

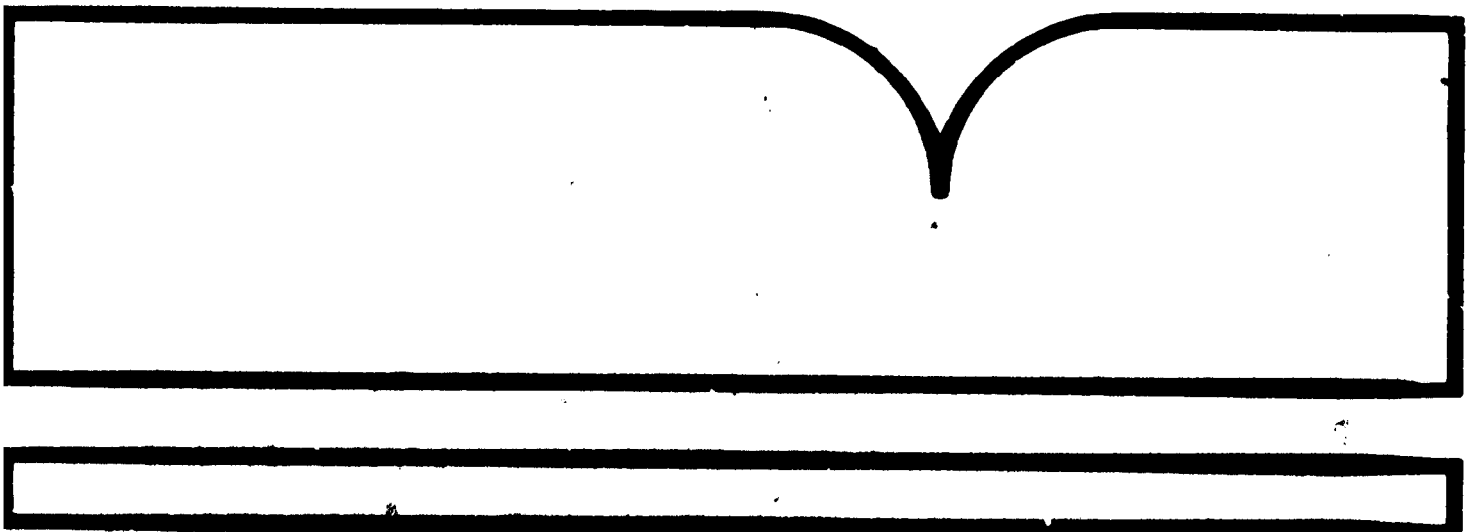
**System Operations Studies for Automated
Guideway Transit Systems: Discrete Event
Simulation Model Programmer's Manual**

**(U.S.) Transportation Systems Center
Cambridge, MA**

Prepared for

**Urban Mass Transportation Administration
Washington, DC**

Jul 82



1. Report No. UMTA-MA-06-0048-82-1		2. Government Accession No.		3. Recipient's Catalog No. PB83 122770	
4. Title and Subtitle System Operations Studies for Automated Guideway Transit Systems: Discrete Event Simulation Model Programmer's Manual.				5. Report Date July 1982	
				6. Performing Organization Code DTS-723	
7. Author(s) J.F. Duke, and R. Blanchard				8. Performing Organization Report No. DOT-TSC-UMTA-82-13	
9. Performing Organization Name and Address General Motors Corporation* GM Transportation Systems Division GM Technical Center Warren, Michigan 48090				10. Work Unit No. (TRAIS) .. MA-06-0048	
				11. Contract or Grant No. DOT-TSC-1220	
12. Sponsoring Agency Name and Address U.S. Department of Transportation Urban Mass Transportation Administration 400 Seventh Street, S.W. Washington, D. C. 20590				13. Type of Report and Period Covered Final Report July 1978	
				14. Sponsoring Agency Code URT-10	
15. Supplementary Notes *Under contract to: U.S. Department of Transportation Transportation Systems Center Kendall Square Cambridge, Massachusetts 02142					
16. Abstract <p>In order to examine specific automated guideway transit (AGT) developments and concepts, UMTA undertook a program of studies and technology investigations called Automated Guideway Transit Technology (AGTT) Program. The objectives of one segment of the AGTT program, the System Operations Studies, were to develop models for the analysis of system operations, to evaluate performance and cost, and to establish guidelines for the design and operation of AGT systems.</p> <p>This document provides a detailed description of the design, programming logic and implementation of the Discrete Event Simulation Model (DESM). The DESM provides the capability to model the operation of a mass transit system operating over a network composed of guideway links and station within a given time period. A wide range of transit classes can be modeled using the DESM. User controls and options are available within the simulator to allow modeling the effects of various operating strategies and service policy options on overall system performance in terms of providing transportation service on an individual patron basis. The Programmer's Manual for the DESM describes the program's purpose, functions, organization, variables, and processing algorithms from a maintenance viewpoint. A global variable dictionary, subprogram logic tables, and subprogram descriptions are included to aid maintenance and modification of this model. A description of debug tools built into the model to aid maintenance is also included.</p>					
17. Key Words Discrete Event Simulation Model Computer Programs Data Collection AGT Systems System Operations Studies AGTT Program Guideway Programmer's Manual Guidelines			18. Distribution Statement Available to the Public through the National Technical Information Service, Springfield, Virginia 22161.		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 730	22. Price

PREFACE

In order to examine specific Automated Guideway Transit (AGT) developments and concepts--and to build a better knowledge base for future decision-making--the Urban Mass Transportation Administration (UMTA) undertook a new program of studies and technology investigations called the UMTA Automated Guideway Transit Technology (AGTT) program. The objectives of one segment of the AGTT program, the System Operations Studies (SOS), were to develop models for the analysis of system operations, to evaluate performance and cost, and to establish guidelines for the design and operation of AGT systems. A team headed by GM Transportation Systems Division (GMTSD) was awarded a contract by the Transportation Systems Center to pursue these objectives. The Technical Monitor for the project at TSC was Arthur Priver, who was assisted by Li Shin Yuan and Thomas Dooley.

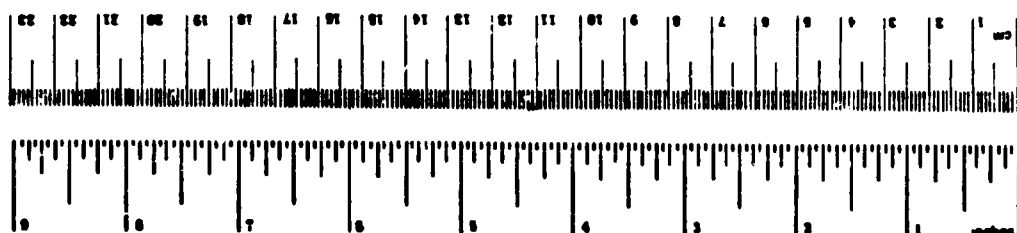
The Discrete Event Simulation Model (DESM) provides the capability to model the operation of a mass transit system operating over a network composed of guideway links and stations within a given time domain. A wide range of transit classes can be modelled using the DESM. User controls and options are available within the simulator to allow modelling the effects of various operating strategies and service policy options on overall system performance in terms of providing transportation service on an individual patron basis.

The Programmer's Manual defines the program's purpose, functions, organization, variables, and processing algorithms from a software maintenance viewpoint. It contains the information for a programmer to maintain and modify the program as required.

METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures		Approximate Conversions from Metric Measures		
Symbol	When You Know	Multiply by	To Find	Symbol
LENGTH				
in	inches	2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
AREA				
sq ft	square feet	0.9	square centimeters	cm ²
sq in	square inches	6.5	square centimeters	cm ²
sq yd	square yards	0.8	square meters	m ²
sq mi	square miles	2.6	square kilometers	km ²
ac	acres	0.4	hectares	ha
MASS (weight)				
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
short ton	short tons	0.9	tonnes	t
VOLUME				
imp gal	imperial gallons	4	liters	l
US gal	US gallons	3.8	liters	l
qt	quarts	0.95	liters	l
pt	pints	0.47	liters	l
cup	cups	0.24	liters	l
fl oz	fluid ounces	0.03	liters	l
gal	gallons	3.8	liters	l
cu ft	cubic feet	0.03	cubic meters	m ³
cu yd	cubic yards	0.76	cubic meters	m ³
TEMPERATURE (exact)				
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

Approximate Conversions from Metric Measures	
When You Know	Multiply by
LENGTH	
centimeters	0.04
inches	0.4
feet	3.3
meters	1.1
kilometers	0.6
AREA	
square centimeters	0.16
square meters	1.2
square kilometers	0.4
hectares (10,000 m ²)	2.5
MASS (weight)	
grams	0.035
kilograms	2.2
tonnes (1000 kg)	1.1
VOLUME	
milliliters	0.03
liters	2.1
hectoliters	1.06
hectoliters	0.26
cubic meters	36
cubic meters	1.3
TEMPERATURE (exact)	
°C	Celsius temperature
°F	Fahrenheit temperature



* 1 in = 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Mon. Publ. 758, Units of Length and Mass, Price \$2.25. SO Catalog No. C13.18-208.

CONTENTS

Section		Page
1.	INTRODUCTION	1-1
1.1	Overview	1-1
1.2	Computer Requirements	1-5
1.2.1	Core Memory	1-5
1.2.2	Peripheral Equipment	1-6
1.2.3	System Control Program	1-8
1.3	Code Segment Description	1-10
2.	PROGRAM DESCRIPTION	2-1
2.1	Input Processor	2-16
2.1.1	Architecture	2-18
2.2	Model Processor	2-22
2.2.1	Architecture	2-25
2.3	Output Processor	2-44
2.3.1	Architecture	2-45
3.	GLOBAL VARIABLE DICTIONARY	3-1
4.	DEBUG TOOLS	4-1
4.1	Subroutine Trace Facility	4-1
4.2	Standardized Error Handling	4-1
4.3	System Traceback	4-6
5.	SUBPROGRAM LOGIC TABLES	5-1
6.	SUBPROGRAM DESCRIPTIONS	6-1
6.1	Input Processors	6-2
6.1.1	DAYTIM - Obtain Date and Time	6-2
6.1.2	DEBUG - Auxiliary Output Macro	6-4
6.1.3	DTIMEL - Read System Clock	6-6
6.1.4	EACOMN - Structured Data COMMONs Ordering	6-8
6.1.5	EAFLAG - Auxiliary Output Request Processing	6-9
6.1.6	EIADDR - Establish COMMON Addressability	6-11
6.1.7	EIALTP - Alternate Path Table Generation	6-13
6.1.8	EIARPT - Active Fleet Size Management Report	6-16
6.1.9	EIBWRT - Binary Input/Output	6-19
6.1.10	EICHCK - Input Parameter Checking	6-21
6.1.11	EIDRPT - Trip Demand Summary Report	6-29

Section		Page
6.1.12	EIDRSP - Demand Responsive Service Planning	6-32
6.1.13	EIEMTY - Empty Vehicle Allocation	6-39
6.1.14	EIERROR - Error Message Processing	6-41
6.1.15	EIFAIL - Failure/Recovery Processing	6-43
6.1.16	EIGDIP4 - Generalized Data Input Variables	6-47
6.1.17	EIINIT - Input Processor Initialization	6-50
6.1.18	EIMNAM - Parameter List Decode	6-52
6.1.19	EIMNTP - Transportation Algorithm	6-55
6.1.19a	EIMORG - Timeout/Group Demand Responsive Service Planning	6-68a
6.1.20	EIMPTH - Least Cost Path Determination	6-69
6.1.21	EINDPP - Network Data Preprocessing	6-73
6.1.22	EINERR - Error Messages	6-77
6.1.23	EINFMT - Network Data Formatting	6-79
6.1.24	EINPUT - Input Processor Control	6-82
6.1.25	EINRPT - Network Summary Report	6-85
6.1.26	EINSLT - Generate Successor Link Table	6-88
6.1.27	EINTWK - Network Processing Control	6-90
6.1.28	EIPARM - PARM Field Addressability	6-92
6.1.29	EIPSAV - COMMON Area Addressability	6-94
6.1.30	EIRNG - Random Number Generator	6-96
6.1.31	EIRSEL - Sample From Probability Distribution	6-97
6.1.32	EISCFG - Station Configurator	6-99
6.1.33	EISCHD - Scheduled Service Planning	6-104
6.1.34	EISERV - Service Planning Control	6-111
6.1.35	EISRPT - System Characteristic Report Generator	6-113
6.1.36	EITNIT - Trip Demand Initialization	6-118
6.1.37	EITRIP - Trip Demand Generation	6-122
6.1.37a	EIUDGN - Deterministic Demand Generation	6-125a
6.1.38	XGDIPF4 - Read Full Word GDIP Data	6-126
6.1.39	XGDIPH4 - Read Half Word GDIP Data	6-129
6.1.40	XGDIPX4 - Read Single Byte GDIP Data	6-132
6.1.41	XNDBOR - Generalized Data Input Processing	6-135
6.1.42	XPSEUDO - I/O Intercept Routine	6-137
6.1.43	XTRACBK - Interrupt Handler	6-140
6.1.44	XTRCBKP - Save Area Trace Formatting	6-143
6.2	Model Processor	6-145
6.2.1	DAYTIM - Obtain Date and Time	6-146
6.2.2	DBUG - Auxiliary Output Macro	6-147
6.2.3	DQUE - Remove First Transaction from a Queue	6-148
6.2.4	DQUEM - Remove a Specified Transaction from a Queue	6-150
6.2.5	DTIMEL - Read System Clock	6-153
6.2.6	EAFSM - Active Fleet Size Management	6-153
6.2.7	EAASYN - Asynchronous Event Processing	6-155
6.2.8	EACKR - Checkprint/Restart Processing	6-159
6.2.9	EACOMN - Structured Data Commons Ordering	6-161
6.2.10	EADADD - Initialize System Status Area	6-162
6.2.11	EAERR - Error Message Processing	6-164
6.2.12	EAFINS - System Termination	6-166

Section		Page
6.2.13	EAFLAG - Auxiliary Output Request Processing	6-170
6.2.14	EAFRPT - Termination Report	6-171
6.2.14a	EAFTRN - Push Couple Vehicle Entrainment	6-172a
6.2.15	EAGCTL - Guideway Link Modeling Control	6-173
6.2.16	EAINDX - Update Index File	6-175
6.2.17	EAINIT - System Initialization	6-177
6.2.17a	EAIVEH - Initialize New Vehicles	6-180a
6.2.18	EANDR - Demand Responsive Service Initialization	6-181
6.2.19	EANFEL - Future Events List Initialization	6-183
6.2.20	EANMDL - Modeling Subsystem Initialization	6-185
6.2.20a	EANMRG - Timeout/Group Demand Responsive Service Initialization	6-186a
6.2.21	EANRPT - Initial Conditions Report	6-187
6.2.22	EANSVA - Establish Status Area Addressability	6-189
6.2.23	EANSCD - Scheduled Service Initialization	6-191
6.2.24	EANTIX - Save PARM Field Information	6-197
6.2.24a	EANTRN - Entrain and Launch Scheduled Service AFSM Vehicles	6-198a
6.2.25	EANTSA - Initialize System Status Area Addresses	6-199
6.2.26	EANXTN - Initialize Transaction Data	6-201
6.2.27	EAPCMP - Periodic Computation Processing	6-202
6.2.28	EAPFEL - Place a Transaction on the Future Events List	
6.2.29	EAPLNK - Dequeue Vehicles from Upstream Guideway Link Queues	6-209
6.2.30	EAPRMT - Dequeue Control	6-212
6.2.31	EAPSTN - Dequeue Vehicles from Upstream Station Queues	6-213
6.2.32	EARFEL - Obtain Next Most Imminent Transaction	6-215
6.2.33	EARRPT - Restart Conditions Report	6-217
6.2.34	EASAMP - Periodic Sampling	6-219
6.2.35	EASCTL - Station Link Modeling Control	6-221
6.2.36	EASRPT - Intermediate Sampling Report	6-223
6.2.36a	EASTOR - Assign Stored Vehicles in Scheduled Service AFSM	6-224a
6.2.37	EATORG - Trip Origination	6-225
6.2.38	EAZNIT - Statistics Initialization	6-228
6.2.39	EGALT - Alternate Path Cost Computation	6-230
6.2.40	EGASTN - Alternate Station Assignment	6-234
6.2.41	EGCNXT - Next Station Determination on Circuitous Empty Route	6-236
6.2.42	EGDSTP - Demand Stop Service Processing	6-238
6.2.43	EGDTRN - Guideway Link Vehicle Detrainment	6-241
6.2.44	EGEMTY - Guideway Link Empty Vehicle Processing	6-248
6.2.45	EGETRN - Dynamic Vehicle Entrainment	6-251
6.2.46	EGFAIL - Guideway Link Failure/Recovery	6-256
6.2.47	EGFNTR - Fixed Headway Travel Segment Entry	6-259
6.2.48	EGFTVL - Fixed Headway Travel Segment Traversal	6-264
6.2.49	EGGNXT - Guideway -- Next Entity	6-271
6.2.50	EGHTRN - Train Headway Traversal Processing	6-276
6.2.51	EGLEAV - Guideway Link Exit	6-283
6.2.52	EGLMDL - Guideway Link Model Control Routine	6-285
6.2.53	EGLNTR - Guideway Link Entry Processing	6-289

Section		Page
6.2.54	EGLWTQ - Guideway Link Queuing	6-294
6.2.55	EGNEXT - Next Guideway Entity Determination Control	6-296
6.2.56	EGPATH - Real-Time Path Selection	6-298
6.2.57	EGPRMY - Primary Path Cost Computation	6-300
6.2.58	EGQMRG - Quasi-Synchronous Guideway Control	6-302
6.2.59	EGQNTR - Advance Positioning Processing	6-310
6.2.60	EGRESV - Guideway Link Empty and Reserved Vehicle Processing	6-316
6.2.61	EGSCHD - Schedule Vehicle Follower	6-319
6.2.62	EGTCHK - Trip Compatibility Check	6-325
6.2.63	EGTCTL - Trip Compatibility Check Control	6-328
6.2.64	EGTEST - Guideway Link Entry Testing	6-330
6.2.65	EGTRNC - Train Compatibility Check at Network Diverges	6-332
6.2.66	EGUNTR - User Headway Model, Link Travel Segment Entry	6-334
6.2.67	EGUTVL - User Headway Model Link Travel Segment Traversal	6-335
6.2.68	EGVALS - Vehicle ETA Calculation and Station Arrival List Recording	6-336
6.2.69	EGVLOG - Vehicle Logging	6-339
6.2.70	EGVNTR - Variable Headway Travel Segment Entry	6-341
6.2.71	EGVTVL - Variable Headway Travel Segment Traversal	6-346
6.2.72	EMGDIP4 - Generalized Data Input Variables	6-353
6.2.73	EMODEL - Model Processor Main Control	6-355
6.2.73a	ESAREQ - Assign Request to Vehicles	6-356a
6.2.74	ESASAV - Initialize System Status Area Adresses	6-357
6.2.75	ESBDL - Board List Generation	6-358
6.2.76	ESCHED - Schedule a Transaction on Future Events List	6-361
6.2.77	ESDBL - Deboard List Generation	6-363
6.2.78	ESDIVF - Diverge Functions	6-365
6.2.79	ESDIVO - Diverge Function Ordering	6-366
6.2.80	ESEVA - Empty Vehicle Assignment	6-367
6.2.80a	ESEVB - Empty Vehicle Bumping	6-369a
6.2.81	ESFAIL - Station Failure/Recovery	6-370
6.2.82	ESLDLY - Launch Delay Computation	6-372
6.2.83	ESLEAV - Station Link Exit	6-374
6.2.84	ESLMDL - Station Link Model	6-376
6.2.85	ESLWTQ - Station Link Queuing	6-379
6.2.86	ESMDLA - Station Link Event Completion Processing	6-381
6.2.87	ESMDLB - Station Link Event Initiation Processing	6-385
6.2.88	ESMDLN - Station Link Next Event Determination	6-391
6.2.89	ESMDLY - Merge Scheduling	6-393
6.2.90	ESNEXT - Station Model Next Entity Determination	6-398
6.2.91	ESNSTN - Next Station Determination	6-402

Section		Page
6.2.92	ESNTRN - Static Vehicle Entrainment	6-404
6.2.93	ESPATH - Station Model Path Selection	6-406
6.2.94	ESSDLY - Schedule Delay Determination	6-409
6.2.95	ESTABQ - Perform Trip/Vehicle Assignment at Trip Arrival	6-412
6.2.96	ESTCHK - Station Model Multi-Party Trip Compatibility Check	6-417
6.2.97	ESTEST - Station Link Entry Testing	6-420
6.2.98	ESTLOG - Completed Trip Log Processing	6-423
6.2.99	ESVALS - Vehicle Arrival List Recording	6-425
6.2.99a	ESVREQ - Process Ready Vehicle Request	6-428a
6.2.100	ESVRES - Perform Trip Reservation	6-429
6.2.101	EUDIVF - User Diverge Function Interface Routine	6-431
6.2.102	EUEVA - User Empty Vehicle Assignment Interface Routine	6-432
6.2.102a	EVEVB - User Empty Vehicle Bumping Interface Routine	6-432a
6.2.103	EUPCMP - User Periodic Computation Event Interface Routine	6-433
6.2.104	EZINT - Correct Time Integrals and Record System Totals	6-434
6.2.105	EZHDR - Sampling Header Record Processing	6-436
6.2.106	EZZERO - Initialize Periodic Sampling Statistics	6-438
6.2.107	FREE - Return a Transaction to an Available List	6-439
6.2.108	GET - Obtain an Available Transaction	6-441
6.2.109	MULTICK - Test Transaction for Current Enqueue	6-443
6.2.110	NQUE - Enqueue a Transaction	6-444
6.2.111	PROMPT - Schedule a Transaction Dequeue Event	6-446
6.2.112	QCHECK - Check Transaction Availability in a Queue	6-448
6.2.113	XGDI PF4 - Read Full Word GDIP Data	6-450
6.2.114	XGDIP H4 - Read Half-Word GDIP Data	6-451
6.2.115	XGDIP X4 - Read Single Word GDIP Data	6-452
6.2.116	XNDBOR - Generalized Data Input Processing	6-453
6.2.117	XPSEUDO - I/O Interrupt Routine	6-454
6.2.118	XTRACBK - Interrupt Handler	6-455
6.2.119	XTRCBKP - Save Area Trace Formatting	6-456
6.3	Output Processor	6-457
6.3.0	AADATE - Provide Date in Character Format	6-457a
6.3.1	CKFOLLOW - Validate Follower Record	6-458
6.3.2	DAYTIM - Obtain Date and Time	6-459
6.3.3	DEBUG - Auxiliary Output Macro	6-460
6.3.3a	DERROR - Error Termination Route	6-460a
6.3.4	DTIMEL - Read System Clock	6-461
6.3.5	EABIN - Bin Area Reallocation	6-462
6.3.6	EBNCHK - Check Bin Allocation	6-465
6.3.7	EDBIN - Bin Allocation	6-467
6.3.8	EDUMBIN - Formatted Dump of Bin Area	6-469
6.3.9	EGRAPH - Time Series Plotting	6-471
6.3.10	EHEADER - Read a Header Record	6-474

Section		Page
6.3.11	EHIST - Format and Display Histogram	6-475
6.3.12	ELIST - Format and Display Time Series List or Statistical Summary	6-478
6.3.13	EMNMX - Compute Minimum and Maximum Values of Sampled Items	6-481
6.3.14	EODATA - Output Processor Data Initialization	6-483
6.3.15	EOERR - Error Message Processing	6-485
6.3.16	EOFLAG - Auxiliary Output Request Processing	6-487
6.3.17	EOINDX - Update Index File	6-489
6.3.18	EONTIX - Save PARM Field Information	6-491
6.3.19	EOPRPT - Performance Summary Report Processing	6-493
6.3.20	EOPSUM - Performance Summary Computation Processing	6-494
6.3.21	EOSRPT - System Summary Report Processing	6-496
6.3.22	EOSSUM - System Summary Statistics Computation Processing	6-497
6.3.23	EOUTPT - Output Processor Control	6-499
6.3.24	EOZKIT - Output Processor Initialization	6-502
6.3.25	EREAD02 - Process Type 2 System Statistics	6-504
6.3.26	EREAD03 - Process Type 3 Station Statistics	6-506
6.3.27	EREAD04 - Process Type 4 Station Link Statistics	6-508
6.3.28	EREAD05 - Process Type 5 Guideway Statistics	6-510
6.3.29	EREAD06 - Process Type 6 Route Statistics	6-512
6.3.30	EREQTLU - Record/Request Correlation	6-514
6.3.31	EREQU - Store an Output Processor Request	6-516
6.3.32	ESETUP - Initialize Data Tables	6-519
6.3.33	ESHIFT - Shift Data in Bin Area	6-520
6.3.34	ESKIPFO - Skip a Follower Record	6-521
6.3.35	ESTORE - Store Sampled Data Item in Bin	6-522
6.3.35a	ETACUM - Accumulate the Raw Statistics	6-523a
6.3.35b	ETCAPT - Capture the Raw Data	6-523b
6.3.35c	ETCOMP - Compute the Measure Requested	6-523d
6.3.35d	ETMEAS - Compute Raw Data for Measure	6-523f
6.3.35e	ETMERG - Convert Statistics to UPTS Format	6-523h
6.3.35f	ETMNR - Convert a Number to EBCDIC	6-523j
6.3.35g	ETRPTS - Generate the Station-to-Station Measures Report	6-523l
6.3.35h	ETSSPM - Station-to-Station Measures Control	6-523n
6.3.35i	ETSTAT - Compute Statistics	6-523p
6.3.35j	ETSTID - Convert Station ID to EBCDIC	6-523r
6.3.35k	ETTOTL - Total the Raw Statistics	6-523s
6.3.36	EZHIST - Histogram Output Control	6-524
6.3.37	EZLIST - List Output Control	6-526
6.3.38	EZPLOT - Plot Output Control	6-528
6.3.39	EZREAD - Data Acquisition	6-530
6.3.40	XPSEUDO - I/O Intercept Routine	6-533
6.3.41	XTRACBK - Interrupt Handler	6-534
6.3.42	XTRCBKP - Save Area Trace Formatting	6-535/6-536
Appendix A	- Report of New Technology	A-1/A-2

ILLUSTRATIONS

Figure		Page
1-1	DESM Organization	1-3
2-1	Input Processor Control Flow	2-2
2-2	Model Processor Control Flow	2-7
2-3	Output Processor Control Flow	2-13
2-4	Input Processor Data Interface	2-17
2-5	Input Processor Functional Flow	2-19
2-6	Model Processor Data Interface	2-23
2-7	Model Processor Functional Flow	2-26
2-8	Architecture/Modeling Control Relationship	2-28
2-9	DESM Queue Structures	2-31
2-10	Queue List Addition	2-32
2-11	Queue List Removal	2-33
2-12	DESM Entity Modeling Architecture	2-34
2-13	DESM Link Structure	2-36
2-14	Station Link Canonical Definition	2-38
2-15	DESM Station Configuration	2-39
2-16	Clock Table Organization	2-41
2-17	Multiple Thread List Organization	2-43
2-18	Output Processor Data Interface	2-46
2-19	Output Processor Functional Flow	2-47

Figure		Page
2-20	Output Processor Bin Referencing	2-50
2-21	Request Filing/Bin Storage Relationship	2-53
2-22	Data Matching Process	2-55
2-23	Data Matching Results	2-56
4-1	Sample Traceback Printout	4-8
6-1	Transaction Dequeing	6-152
6-2	Future Events List Definition	6-168
6-3	System Status Area Organization	6-179
6-4	DESM Headway Regulation	6-288
6-5	Vehicle Retardation Process	6-306
6-6	Desired Advance Computation	6-307
6-7	Actual Advance Computation	6-308
6-8	Deboard/Board Algorithm	6-360
6-9	Algorithm/Total Number Passengers Boarded	6-360a
6-10	Reservation Table Time Base Updating	6-395
6-11	Merge Reservation Process	6-397

TABLES

Table		Page
1-1	DESM Core Requirements	1-7
1-2	Input Processor Code Segments	1-11
1-3	Model Processor Code Segments	1-16
1-4	Output Processor Code Segments	1-22
3-1	Global Variables	3-4
4-1	Input Processor Program Maintenance Flags	4-2
4-2	Model Processor Program Maintenance Flags	4-3
4-3	Output Processor Program Maintenance Flags	4-5
5-1	Input Processor Subroutine Logic Table	5-2
5-2	Model Processor Subroutine Logic Table	5-7
5-3	Output Processor Subroutine Logic Table	5-18
6-1	Automatic Station Link Selection	6-400
6-2	Scheduled Service Route Spacing Algorithms	6-411

SECTION 1. INTRODUCTION

This document provides a detailed description of the design, programming logic and implementation of the Discrete Event Simulation Model (DESM).

1.1 OVERVIEW

The DESM provides the ability to simulate the detailed operation of a transit system operating on a network of automated guideways and stations. Vehicles traverse the automated network according to preplanned schedules or in response to patron requests for service, depending upon the selected mode of service. Simulation functions associated with patrons include: arrival at a station, assignment of a vehicle to service the trip request, waiting for the assigned vehicle, boarding, deboarding, and transferring. The travel portion of the patron activity is modeled in conjunction with vehicle travel. Vehicles move along the guideway network and through stations according to a user-selected system management strategy, which consists of individually selected policies for: type of service, berth assignment, entrainment, empty vehicle allocation, path selection, dispatch, longitudinal control, position regulation, and merge control. Vehicle movement in the model is affected by operational interactions caused by the simultaneity of vehicle movements on the guideways and in stations, and the occurrence of asynchronous events, reflecting unexpected or preplanned stimuli that affect system operation. Other system characteristics, such as vehicle capacity, nominal speed, and headway are major factors considered in the simulation of system performance.

The DESM employs a standalone discrete event modeling approach to perform required transit system simulation. In this approach, transactions are scheduled on a time ordered Future Events List (FEL) to reflect time delays associated with a set of actions or interactions which affect or are associated with system operation.

Transactions represent entities (e.g., vehicles, trips), simulation control or modeling requests, and asynchronous or exogenous simulation processes.

When a transaction reaches the top of the list, the required delay time associated with a particular event has been completed, the simulation clock is advanced to the event completion time and any required event processing related to completion is performed. In this manner, the simulation clock is advanced in discrete intervals of time to the occurrence of the most imminent event to be performed. This is

in contrast to a continuous or delta time simulation approach in which the clock is advanced incrementally by a fixed number of clock units with the status of all transactions being continually updated at each clock advance.

Once event processing for a transaction is completed, the transaction is scheduled for its next event completion. If conditions within the system preclude the scheduling of an event, the DESM employs a detailed transaction queuing and dequeuing mechanism for handling event preemption and resumption. This mechanism provides the ability to recognize the interaction effect produced by the simultaneity of event occurrences.

The major processing functions implemented in the DESM are organized into three standalone program components: 1) an Input Processor (IP), 2) a Model Processor (MP), and 3) an Output Processor (OP). All three processors within the DESM interact with the user and communicate with one another through a centralized AGT data base as shown in Figure 1-1. Each processor is initialized from predefined files within the data base, augmented by user run time inputs.

The Input Processor is the primary interface between the data base and user run time data and the simulation. Its major functions are to check the input data for consistency and reasonableness and to transform the initialization data from a user-oriented format to a model-oriented format to provide for efficient operation of the Model Processor.

The Input Processor performs translation of input definition data and selected control options input by the user to structured data files usable by the Model Processor. The IP is basically a sequential processor, having a fixed order of tasks that can be performed. However, the user, through input of control and option selection parameters, defines which of the tasks and processing options are to be executed. The structured data produced by the IP is organized into Network, System, Demand, and Time Dependent Data files. The Network file contains binary images of connectivity and configuration data for describing the system to be simulated to the MP. The System file contains binary images of system characteristics and option data processed or computed by the IP. These files are used by the MP to initialize the global common data area (status region) which provides the initial conditions and options required for a given simulation experiment. The Demand file contains a generated sequence of trip arrivals which represent patron demands for transportation service to the MP. The structured time-dependent data file contains card image data in a generalized input format. This data is processed by the MP to introduce asynchronous events or to modify simulation conditions during a modeling experiment.

The DESM Model Processor performs the event driven simulation for modeling the detailed operation of the automated transit system. Events

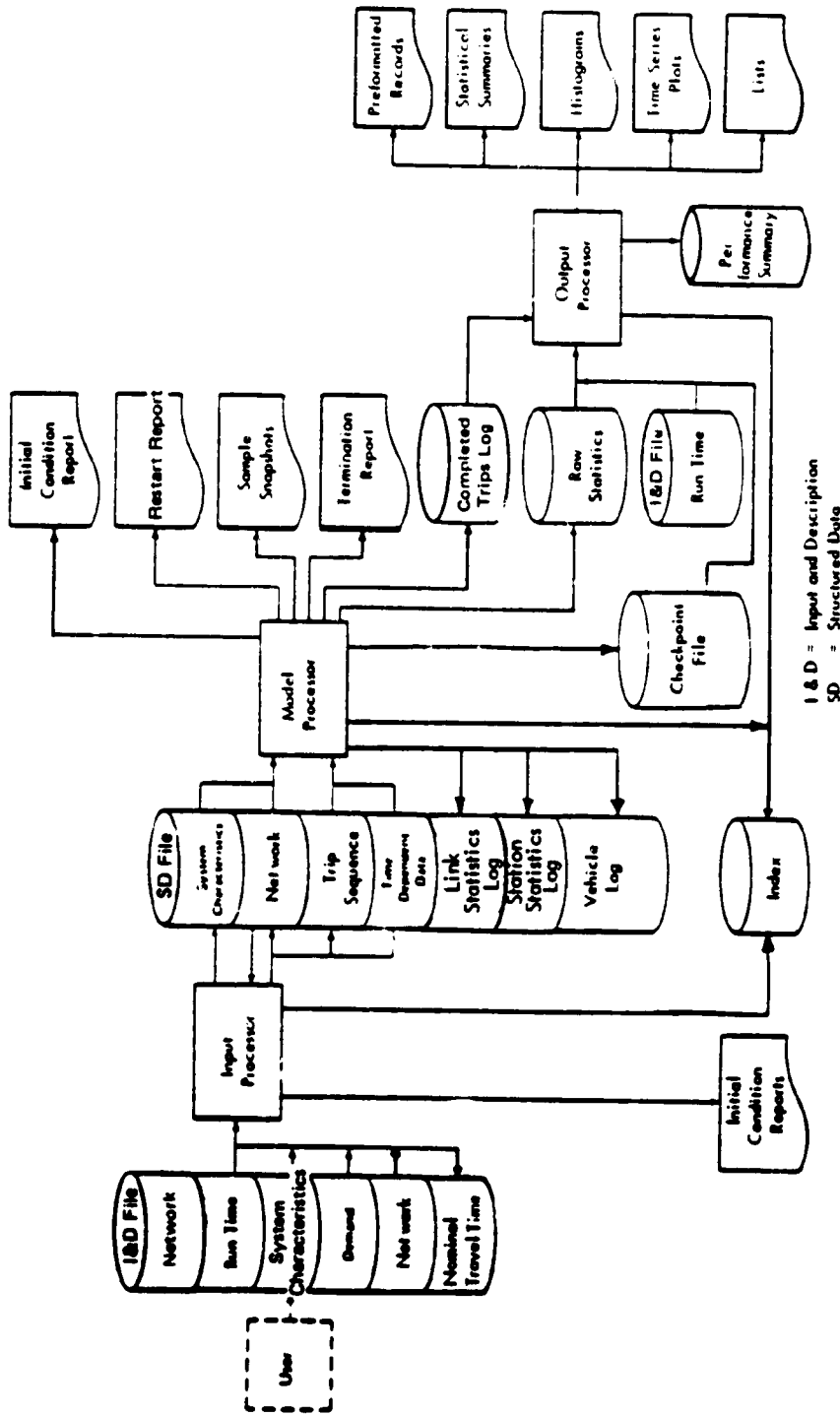


FIGURE 1-1. DESM ORGANIZATION

are scheduled within the simulation for occurrence or completion at some future time, in response to transaction processing requirements. Transactions are appropriately processed when the event time for which they were scheduled is completed and the next event for the transaction becomes the next most imminent task to be performed in the simulation system. Transactions are rescheduled when processing for the current event of the transaction is completed and the next required event and its completion time has been determined.

Upon entry, the MP performs initialization of the simulation experiment which may involve system restart or initial system definition. In the case of system restart, a previously recorded checkpoint of the system status area (structured data and global model common data) is read from the data base to reinitialize and resume the simulation from a specific point in time. If a restart is not to be performed, the structured data created by IP is read from the data base to establish all global common data defining the simulation initial conditions.

Once initialization has been completed, the basic control loop for accomplishing the recognition, scheduling and processing of transactions is started. During this process, the next transaction to be acted upon is determined by performing a sequential scan of successive entries in the clock table, beginning with the currently active interval, until a non-empty interval or the end of the clock table is reached. If a non-empty interval pointer is found, the first transaction chained within the interval is removed and returned as the currently active transaction requiring event processing.

The transaction can be one of the following types: asynchronous event, (data read, checkpoint, failure insertion, or active fleet size management), statistics sampling, trip arrival, dequeuing, periodic computation, guideway link modeling or station link modeling.

The DESM output processor provides the means by which sampling data, written to the raw statistics file during a simulation experiment, can be retrieved and formatted for transit system analysis. The output processor permits access to and manipulation of the raw statistics in a convenient and unrestrictive manner. This is achieved by providing a user interface which does not require apriori knowledge of how data is formatted, acquired from the input source, or arranged internal to the processor itself.

The processing performed by the output processor is directed by service request commands input by the user. These commands invoke the four basic processor provided by the OP as follows:

1. Data storage allocation
2. Command request processing

3. Data acquisition and manipulation

4. Data display.

Data request commands provide the means by which desired statistics and presentation format are specified for retrieval. These requests are accumulated until a read command which causes actual accumulation and formatting of data is encountered.

The OP allows the accumulation of up to 400 user requests before data acquisition is required. The exact number of data request commands which are accumulated prior to initiating data acquisition and processing is entirely dependent upon user requirements since data acquisition is performed only in response to a user command request. The OP also contains an automatic request generation facility which allows the user to obtain via a single request a specific item of data over a range of modeling entities.

The basic control loop of the OP involves the reading and filing of user data requests until a data acquisition (READ) command is encountered. This causes the reading of the raw statistics file to begin and data manipulation, summarization, and display to be performed. Once the desired data display and output have been accomplished, the control loop is once again started to process further sets of user command requests. The control loop is executed until all user data requests have been satisfied.

1.2 COMPUTER REQUIREMENTS

The DESM requires an IBM System 370, Model 155, 158, 165, or 168 CPU, or a compatible equivalent, for program maintenance and execution.

1.2.1 Core Memory

The core storage size required to execute the DESM is a function of maximum problem size, given by System Generation (compile time) definitions for network size, configuration and system capacity. These SYSGEN sizes represent an upper bound on the size of a particular simulation experiment that can be modeled by the DESM. This requirement applies specifically to the MP, which represents the largest of the three components of the DESM. This requirement is exclusive of System Control Program Core requirements, which are CPU and installation dependent. In order to support the problem size requirements placed on the DESM, as outlined in the Functional Specification, a minimum of 4.5 million bytes of core storage is required.

Size requirements can be varied as necessary to support smaller or larger maximum problem sizes or core storage availability, by redefining

compile maximum values and recompiling and link editing the IP, MP, and OP components of the DESM. Only the I/O portion of the OP need be recompiled to reflect new maximum size definitions. As a guide to establishing core storage requirements for a particular SYSGEN of the DESM, core requirements for key configuration and capacity related elements within the MP are shown in Table 1-1. The exact core region size required for execution of the IP, MP, and OP components of the DESM, after recompilation and link editing, is provided as a standard output of the link edit process.

1.2.2 Peripheral Equipment

The DESM is designed for either batch mode or terminal supported background mode operations. In a batch environment, standard system utilities and JCL procedures are used for data base updating and model execution. Background operation requires the use of an online terminal and associated terminal support software to allow online data base editing and job submission. This capability is provided via TSO, if available, in a System 370 operating environment. The mode of operation selected for the DESM places certain constraints on the peripheral equipment required for maintenance and execution of the model. The specific equipment requirements are described below.

1.2.2.1 Data Base Storage

The procedures provided for execution of the DESM require the use of direct access storage for online data base access and control. However, with user modification of the standard procedures, other forms of data storage can be used to eliminate online storage requirements or provide a supplement to the online data base.

Direct Access Storage - The storage requirements for various functional areas of the DESM are given below, in cylinders of IBM 3330 disk storage (approximately 248,000 bytes):

1. Program Development Libraries (Source, Object, and load) -- 20 cylinders
2. Input from Data Base (per configuration) -- 10 cylinders
3. Trip Arrival Sequence (One hour of 30,000 trips) -- 2.5 cylinders
4. Checkpoint Data (each checkpoint assuming 25 stations, 70 guideway links, 20 station links, .15 routes) -- 1.2 cylinders/checkpoint record

TABLE 1-1. DESM CORE REQUIREMENTS

<u>Element</u>	<u>Core Required</u>
Basic DESM MP	480k Bytes
Vehicles	167 Bytes per vehicle
Guideway Links	153 Bytes per link
Stations	282 Bytes per station + 18 (number of stations squared)
Station Links	(83 x maximum number of stations + 24) Bytes per station link
Trips	69 Bytes per concurrent trip
Transactions	14 Bytes per transaction
Routes	(160 + 12 x maximum number of links) Bytes per route + 10 x maximum number of entries in route list
Network	4 x (maximum number of stations x maximum number of links x maximum number of simultaneous path tables) + (maximum number of merges + 2) x maximum number of intervals in merge scheduling table + 16 x (maximum number of merges) Bytes

5. Raw Statistics (assuming 25 stations, 70 guideway links, 20 station links, 15 routes, one hour simulation, one minute sampling interval) -- .27 cylinders/sample (5.1 tracks), 16.3 cylinders/hour of simulated time
6. Auxiliary Output
 - a. Completed trips log -- 5 cylinders
 - b. Vehicle Station Arrival Log -- 1 cylinder
 - c. Link Statistics Log -- 1 cylinder
 - d. Station Statistics Log -- 1 cylinder.

Magnetic Tape - The DESM has no explicit requirement for magnetic tape storage, but it may be a preferable medium over direct access storage for handling of simulation output data. The selection of tape over disk should be based upon the amount of disk space available, frequency of access required, computer center operational procedures, and desired mode of DESM operation. Files resident on magnetic tape are not readily modified and cannot be displayed via background terminal editing and display procedures. For planning estimates, a 2400 foot reel of tape recorded at 1600 bytes/inch has a capacity equivalent to 188 cylinders of 3330 disk space.

1.2.2.2 Unit Record Equipment

The DESM requires a card reader for batch job submission and a high-speed printer for output.

1.2.2.3 Display Terminal

Background data preparation and job submission via standard TSO procedures requires a 3270 display terminal or equivalent for DESM operation.

1.2.3 System Control Program

Maintenance and operation of the DESM requires specific operating system and system support software features as described below.

1.2.3.1 Operating System

The DESM executable load module is structured to avoid requirements for the loading and overlaying of individual program segments. Each execution of the DESM assumes the availability of "unlimited" core storage as provided in a virtual storage and virtual machine operating environment. Specifically, the following operating system or a compatible equivalent is required:

OS/VS2 (SVS or MVS option).

Additionally, the use of the DESM in a background environment requires operating system support of a file/editing, updating, and job submission capability which provides for online terminal operations. In an OS/VS environment, this support is provided via the Time Sharing Option (TSO).

1.2.3.2 Compilers/Linkage Editor

Modification and maintenance of the DESM requires the availability of programming language support as used in initial model development. The DESM is implemented in structured FORTRAN with Assembler Language for accomplishing Operating System interface and data management functions. The structured FORTRAN implementation is provided by PARAFOR which is based upon use of the PL/I preprocessor for extending the syntax available in FORTRAN. The use of the preprocessor within PARAFOR, provides the capability for including code segments from several source libraries to form executable source modules and permits the use of user coded macros for accomplishing application related functions. These capabilities are used extensively in the top down implementation of the DESM to enhance source code readability and leave unobscured the logic flow through the system.

Structured FORTRAN and Assembler language coding within the DESM requires the following system compilers:

1. FORTRAN IV (H level)
2. PL/I Optimizer
3. Assembler (H).

Executable load module creation, requires the use of a Linkage Editor which supports an overlay option. The overlay option is used in the IP and MP components of the DESM for structuring global common area data as described in Section 2. The program segments themselves are not overlaid since the DESM is designed to operate in a virtual storage environment.

1.2.3.3 Support Software

Structured FORTRAN and data base maintenance require the following utility software:

1. PARAFOR
2. OS/VS2 System Utilities.

PARAFOR capabilities and operation are described in "PARAFOR User's Guide", IBM Federal Systems Division, October 15, 1976.

1.3 CODE SEGMENT DESCRIPTION

The code segments listed in Tables 1-2, 1-3, and 1-4 identify each program module, included source member, macro, common data area and entry point name which comprise the IP, MP, and OP components of the DESM, respectively. The descriptive information in the tables identify the level of documentation provided for each listed code segment. The detailed design logic for each routine is provided in the Process Design Language (PDL) presented in Appendix A. The component descriptions, as listed in the tables, are provided in Section 6.

TABLE 1-2. (1 of 5) INPUT PROCESSOR CODE SEGMENTS

DESM - INPUT PROCESSOR --- CODE SEGMENTS & ENTRY POINTS (*)						
NAME	LIB	T Y P E	P D L	C D .	P R E A M	DESCRIPTION
ATYPE	ASM	M	M	M	M	MACRO TO DEFINE CONSTANTS BY TYPE
CALLS	ASM	M	M	M	M	STANDARDIZED ASSEMBLY LANGUAGE ROUTINE LINKAGE
COMN	ASM	M	M	M	M	GDIP COMMON AREA CSECT GENERATION MACRO
DAYTIM	FORT	R	X	X	X	GET CURRENT DATE AND TIME YY/MM/DD/HH/MM
DEBUG	PLI	M	M	X	M	WRITE INTERMEDIATE OUTPUT
DO	ASM	M	M	M	M	STANDARDIZED REGISTER SAVE MACRO
DTIMEL	ASM	R	X	X	X	SOURCE MEMBER FOR SUBROUTINE TIMES
EACOMN	FORT	R	X	X	X	INPUT COMMON AREA ORDERING
EAFLAG	FORT	R	X	X	X	INTERMEDIATE OUTPUT FLAG PROCESSING
ECAMSG	FORT	C	C	C	C	MESSAGE DATA
ECICFG	FORT	C	C	C	C	STATION CONFIGURATION INPUT
ECICOMN	ASM	I	C	I	I	GDIP COMMONS FOR IP AND MP
ECIFEL	FORT	C	C	C	C	FUTURE EVENTS LIST TIMING INPUT DATA
ECIGL	FORT	C	C	C	C	GUIDEWAY LINK DATA
ECIMAX	FORT	C	C	C	C	RUN TIME MAXIMA
ECINET	FORT	C	C	C	C	NETWORK STRUCTURED DATA
ECIPOL	FORT	C	C	C	C	POLICY DATA
ECISL	FORT	C	C	C	C	STATION LINK DATA
ECISTN	FORT	C	C	C	C	STATION DATA
ECISYS	FORT	C	C	C	C	SYSTEM DATA
ECIVEH	FORT	C	C	C	C	VEHICLE DATA
ECNDMD	FORT	C	C	C	C	IP TRIP DEMAND DATA
ECNNET	FORT	C	C	C	C	IP NETWORK DATA
ECNPOL	FORT	C	C	C	C	IP POLICY DATA
ECNSTR	FORT	C	C	C	C	IP NETWORK STRUCTURED DATA
ECNSYS	FORT	C	C	C	C	IP SIMULATION SYSTEM DATA
ECNTRN	FORT	C	C	C	C	IP TRANSPORTATION ALGORITHM DATA
ECOMMAX	ASM	I	I	I	I	ARRAY SIZES FOR ASSEMBLY
EHEADERS	FORT	I	I	I	I	ASYNCHRONOUS INPUT TYPES DEFINITION
EIADDR	ASM	R	X	X	X	SYSTEM CHARACTERISTICS ADDR AND LENGTH SAVE
EIALTP	FORT	R	X	X	X	ALTERNATE PATH TABLE GENERATION
EIALTP1	FORT	I	I	I	I	WRITE ALTERNATE PATH SUMMARY REPORT
EIARPT	FORT	R	X	X	X	WRITE ACTIVE FLEET MANAGEMENT REPORT
EIARPT1	FORT	I	I	I	I	FORMAT FOR ACTIVE FLEET MANAGEMENT REPORT
EIBWRT	FORT	R	X	X	X	STRUCTURED DATA FILE INPUT/OUTPUT
EICHCK	FORT	R	X	X	X	DATA INITIALIZATION, CHECKING & TIME CONVERSION
EICHCK1	FORT	I	X	I	I	INCLUDE COMMONS AND LOCAL VARIABLE DEFINITIONS
EICHCK2	FORT	I	X	I	I	PROCESS GUIDEWAY PARAMETERS
EICHCK3	FORT	I	X	I	I	PROCESS STATION PARAMETERS
EICHCK3A	FORT	I	X	I	I	SCAN EVENT LIST FOR DELIMITERS
EICHCK3B	FORT	I	X	I	I	SCAN UPSTREAM LINK LIST FOR DELIMITERS
EICHCK3C	FORT	I	X	I	I	SCAN DOWNSTREAM LINK LIST FOR DELIMITERS
EICHCK4	FORT	I	X	I	I	CHECK SERVICE POLICY AND TRIP/VEHICLE DATA
EIDRPT	FORT	R	X	X	X	WRITE TRIP GENERATION SUMMARY REPORT
EIDRPT1	FORT	I	I	I	I	INCLUDE COMMONS AND LOCAL VARIABLE DEFINITIONS
EIDRPT2	FORT	I	I	I	I	FORMAT FOR DEMAND REPORT
EIDRPT3	FORT	I	I	I	I	WRITE TRIP SIZE DISTRIBUTIONS
EIDRPT4	FORT	I	I	I	I	WRITE ORIG AND DEST PROBABILITY DISTRIBUTIONS
EIDRPT5	FORT	I	I	I	I	WRITE INPUT DEMAND MATRIX & TRIP GEN SUMMARY
EIDRSP	FORT	R	X	X	X	DEMAND RESPONSIVE SERVICE PLANNING
EIDRSP1	FORT	I	X	I	I	INCLUDE COMMONS AND LOCAL VARIABLE DEFINITIONS
EIDRSP2	FORT	I	X	I	I	ESTIMATE NUMBER OF OCCUPIED VEHICLES REQUIRED
EIDRSP3	FORT	I	X	I	I	ESTABLISH INPUT FOR EIENTY
EIDRSP3A	FORT	I	I	I	I	DEBUG STATEMENTS
EIDRSP4	FORT	I	X	I	I	COMPUTE NUMBER OF EMPTIES AND TOTAL FLEET SIZE
EIDRSP5	FORT	I	X	I	I	COMPUTE EMPTY VEHICLE REDISTRIBUTION PATTERNS
EIDRSP5A	FORT	I	I	I	I	DEBUG STATEMENTS
EIDRSP6	FORT	I	X	I	I	DISTRIBUTE FLEET ON GUIDEWAY

TABLE 1-2. (2 of 5) INPUT PROCESSOR CODE SEGMENTS

NAME	LIB	T Y P E	P D L	C D .	P R E A M	DESCRIPTION
EIDRSP6A	FORT	I	X	I	I	DISTRIBUTE FLEET IN STORAGE
EIDRSP7	FORT	I	X	I	I	VERIFY USER EMPTY VEHICLE DISTRIBUTION DATA
EIDRSP7A	FORT	I	X	I	I	VERIFY EMPTY VEHICLE NEED DATA
EIDRSP8	FORT	I	X	I	I	PROCESS CIRCUITOUS EMPTY ROUTES
EIEMTY	FORT	R	X	X	X	EMPTY VEHICLE REDISTRIBUTION
EIEMTY1	FORT	I	I	I	I	SET ERROR CONDITION CODES
EIERROR	FORT	R	X	X	X	SOURCE MEMBER NAME OF SUBROUTINE ERROR
EIFAIL	FORT	R	X	X	X	NETWORK FAILURE/REPAIR INSERTION
EIFAIL1	FORT	I	I	I	I	INCLUDE COMMONS AND LOCAL VARIABLE DEFINITIONS
EIFAIL2	FORT	I	X	I	I	GET LINK ID FOR LINK/VEHICLE REQUESTS
EIFAIL3	FORT	I	X	I	I	GET STATION ID/LINK FOR STATION REQUESTS
EIFAIL3A	FORT	I	X	I	I	PROCESS STATION REQUEST
EIFAIL4	FORT	I	X	I	I	PROCESS REQUESTS FOR LINK/VEH FAILURE/RECOVERY
EIFAIL4A	FORT	I	X	I	I	LINK FAILURE
EIFAIL4B	FORT	I	X	I	I	LINK RECOVERY
EIFAIL4C	FORT	I	X	I	I	VEHICLE DEGRADATION
EIFAIL4D	FORT	I	X	I	I	VEHICLE DEGRADATION RECOVERY
EIFAIL5	FORT	I	I	I	I	FAILURE/RECOVERY SUMMARY REPORT
EIFAIL5A	FORT	I	I	I	I	WRITE CURRENT MINIMUM PATH TABLE
EIGDIP4	ASM	R	X	X	X	SOURCE MEMBER NAME OF SUBROUTINE GDIPSECT
EIINIT	FORT	R	X	X	X	INPUT PROCESSOR INITIALIZATION
EILIMITS	FORT	R	I	X	I	DATA CHECK LIMITS
EIMNAM	FORT	R	X	X	X	PARM FIELD DECODE
EIMNTP	FORT	R	Y	X	Y	TRANSPORTATION ALGORITHM
EIMORG	FORT	R	X	X	X	TIMEOUT/GROUP DEMAND RESPONSIVE SERVICE PLAN
EIMPTH	FORT	R	X	X	X	MINIMUM PATH GENERATION
EIMPTH1	FORT	I	I	I	I	REDUCE NETWORK
EINADD	FORT	R	E	X	E	SAVE NETWORK DATA ADDR & LGTH (EP IN EIBWRT)
EINDPP	FORT	I	X	X	X	PREPROCESS NETWORK DEFINITION DATA
EINDPP1	FORT	I	X	I	I	CHECK ORIGIN NODES
EINDPP2	FORT	I	X	I	I	CHECK DESTINATION NODES
EINDPP3	FORT	I	X	I	I	ESTABLISH INPUT FOR MINIMUM PATH ROUTINE
EINDPP3A	FORT	I	X	I	I	GUIDEWAY CHARACTERISTICS OF ONLINE STATIONS
EINERR	FORT	R	X	X	X	ERROR MESSAGE GENERATION
EINERR1	FORT	I	I	I	I	ERROR MESSAGES 0-99
EINERR2	FORT	I	I	I	I	ERROR MESSAGES 100'S
EINERR3	FORT	I	I	I	I	ERROR MESSAGES 200'S
EINERR4	FORT	I	I	I	I	ERROR MESSAGES 300'S
EINERR5	FORT	I	I	I	I	ERROR MESSAGES 400'S
EINERR6	FORT	I	I	I	I	ERROR MESSAGES 500'S
EINERR7	FORT	I	I	I	I	ERROR MESSAGES 600'S
EINERR8	FORT	I	I	I	I	ERROR MESSAGES 700'S
EINFMT	FORT	R	X	X	X	NETWORK DATA FORMATTING
EINFMT1	FORT	I	X	I	I	INCLUDE COMMONS AND LOCAL VARIABLE DEFINITIONS
EINFMT2	FORT	I	X	I	I	STATION TO STATION TRAVEL TIME TABLE
EINFMT3	FORT	I	X	I	I	SUCCESSOR STATION TABLE
EINFMT4	FORT	I	X	I	I	STATION PREDECESSOR & SUCCESSOR GUIDEWAY LINKS
EINPUT	FORT	R	X	X	X	DESM INPUT PROCESSOR
EINPUT1	FORT	I	I	I	I	READ SYSTEM CHARACTERISTICS DATA
EINPUT2	FORT	I	I	I	I	READ RUNTIME DATA
EINPUT3	FORT	I	I	I	I	READ TRIP GENERATION INITIAL CONDITIONS
EINPUT4	FORT	I	I	I	I	WRITE ACTIVE FLEET MANAGEMENT DATA
EINRD	FORT	R	E	X	E	READ NETWORK STRUCTURED DATA (EP IN EIBWRT)
EINRPT	FORT	R	X	X	X	WRITE NETWORK INITIAL CONDITIONS REPORT
EINRPT1	FORT	I	I	I	I	INCLUDE COMMONS AND LOCAL VARIABLE DEFINITIONS
EINRPT2	FORT	I	I	I	I	NETWORK CONNECTIVITY SUMMARY
EINRPT2A	FORT	I	I	I	I	LINK WITH STATION
EINRPT2B	FORT	I	I	I	I	LINK WITHOUT STATION
EINRPT3	FORT	I	I	I	I	GUIDEWAY LINK CHARACTERISTICS
EINRPT4	FORT	I	I	I	I	STATION TO STATION SUMMARY

TABLE 1-2. (3 of 5) INPUT PROCESSOR CODE SEGMENTS

NAME	LIB	T Y P E	P D L	C D .	P R E A M	DESCRIPTION
EINRPT5	FORT	I	I	I	I	SUCCESSOR LINK TABLE
EINRPT6	FORT	I	I	I	I	FORMAT STATEMENTS FOR REPORT
EINSLT	FORT	R	X	X	X	GENERATE SUCCESSOR LINK TABLE
EINSLT1	FORT	I	I	I	I	INCLUDE COMMONS AND LOCAL VARIABLE DEFINITIONS
EINTWK	FORT	R	X	X	X	NETWORK PROCESSING CONTROL
EINTWK1	FORT	I	I	I	I	READ NETWORK DEFINITION DATA
EINTWK2	FORT	I	I	I	I	WRITE FAILURE/RECOVERY DATA
EINWRT	FORT	E	E	E	E	WRITE NETWORK STRUCTURED DATA (EP IN EIBWRT)
EIPARM	ASM	R	X	X	X	PARAMETER LIST SAVE
EIPLST	ASM	E	E	E	E	PARAMETER LIST ADDR AND LGTH (EP IN EIPARM)
EIPSAV	ASM	C	X	X	X	ADDR & LGTH OF IP & STRUCTURED DATA COMMONS
EIRNG	FORT	R	X	X	X	RANDOM NUMBER GENERATOR
EIRSEL	FORT	R	X	X	X	CUMULATIVE PROBABILITY DISTRIBUTION SAMPLING
EISAPD	FORT	E	E	E	E	SYSTEM CHAR ADDR + LGTH SAVE (EP IN EIBWRT)
EISCFG	FORT	R	X	X	X	STATION CONFIGURATOR
EISCFG1	FORT	I	I	I	I	LINK TYPE SCAN
EISCFG2	FORT	I	I	I	I	ERROR CHECKING
EISCFG3	FORT	I	I	I	I	STRUCTURED TABLE BUILD
EISCFG4	FORT	I	I	I	I	UPSTREAM POINTER TABLE BUILD
EISCFG4A	FORT	I	I	I	I	DOCK UPSTREAM LINKS LIST BUILD
EISCFG4B	FORT	I	I	I	I	OUTPUT QUEUE UPSTREAM LINKS LIST BUILD
EISCFG4C	FORT	I	I	I	I	OUTPUT RAMP UPSTREAM LINKS LIST BUILD
EISCFG4D	FORT	I	I	I	I	STORAGE UPSTREAM LINKS LIST BUILD
EISCFG4E	FORT	I	I	I	I	DOCK TO STORAGE UPSTREAM LINKS LIST BUILD
EISCFG5	FORT	I	I	I	I	DOWNSTREAM POINTER TABLE BUILD
EISCFG5A	FORT	I	I	I	I	INPUT RAMP DOWNSTREAM LINKS LIST BUILD
EISCFG5B	FORT	I	I	I	I	INPUT QUEUE DOWNSTREAM LINKS LIST BUILD
EISCFG5C	FORT	I	I	I	I	DOCK DOWNSTREAM LINKS LIST BUILD
EISCFG5D	FORT	I	I	I	I	STORAGE TO INPUT DOWNSTREAM LINKS LIST BUILD
EISCHD	FORT	R	X	X	X	SCHEDULED SERVICE PLANNING
EISCHD1	FORT	I	I	I	I	INCLUDE COMMONS AND LOCAL VARIABLE DEFINITIONS
EISCHD2	FORT	I	X	I	I	DEFINE VEHICLE ROUTES FOR USER
EISCHD3	FORT	I	X	I	I	PROCESS USER ROUTE LIST
EISCHD4	FORT	I	X	I	I	DETERMINE NOMINAL TRAVEL TIME
EISCHD5	FORT	I	X	I	I	DEFINE LEVEL OF SERVICE FOR USER
EISCHD6	FORT	I	X	I	I	DETERMINE VEHICLES/ROUTE AND HEADWAY
EISCHD7	FORT	I	X	I	I	SCHEDULE INITIAL VEHICLE LAUNCHES
EISCHD8	FORT	I	X	I	I	CHECK THAT TRAINS FIT ON STATION LINKS
EISCHD9	FORT	I	X	I	I	CALCULATE CLOSEST BARN AND STATION STOPS
EISERV	FORT	R	X	X	X	SERVICE PLANNING CONTROL
EISERV1	FORT	I	I	I	I	INCLUDE COMMONS AND LOCAL VARIABLE DEFINITIONS
EISERV2	FORT	I	X	I	I	COMPUTE STATION DWELL AND NOMINAL TRAVEL TIME
EISRPT	FORT	R	X	X	X	WRITE INITIAL CONDITIONS REPORT
EISRPT1	FORT	I	I	I	I	WRITE SYSTEM CHARACTERISTICS SUMMARY
EISRPT2	FORT	I	I	I	I	WRITE STATION CHARACTERISTICS SUMMARY
EISRPT3	FORT	I	I	I	I	WRITE TRANSIT SERVICE CHARACTERISTICS SUMMARY
EISRPT4	FORT	I	I	I	I	WRITE SIMULATION CONTROL PARAMETERS
EISRPT5	FORT	I	I	I	I	FORMATS FOR REPORTS
EISWRT	FORT	E	E	E	E	WRITE SYS CHAR STRUCTURED DATA (EP IN EIBWRT)
EITNIT	FORT	R	X	X	X	TRIP DEMAND INITIALIZATION
EITNIT1	FORT	I	I	I	I	INCLUDE COMMONS AND LOCAL VARIABLE DEFINITIONS
EITNIT2	FORT	I	X	I	I	PROCESS GROUP SIZE DISTRIBUTION DATA
EITRIP	FORT	R	X	X	X	TRIP DEMAND GENERATION
EITRIP1	FORT	I	I	I	I	INCLUDE COMMONS AND LOCAL VARIABLE DEFINITIONS
EITRIP2	FORT	I	X	I	I	COMPUTE ORIG & DEST PROBABILITY DISTRIBUTIONS
EITRIP3	FORT	I	X	I	I	GENERATE TRIP SEQUENCE
EIUDGN	FORT	E	Y	E	X	DETERMINISTIC (UNIFORM) DEMAND GENERATION
EIWNAM	FORT	E	E	E	E	WRITE FILE MEMBER NAMES (EP IN EIWNAM)
ENDEFS	ASM	M	M	M	M	GDIP COMMON AREA PROCESSOR MACRO
ENTER	ASM	M	M	M	M	STANDARD ASSEMBLY LANGUAGE ENTRY MACRO

TABLE 1-2. (4 of 5) INPUT PROCESSOR CODE SEGMENTS

NAME	LIB	T Y P E	P D L	C D .	P R E A M	DESCRIPTION
ERROR	FORT	E	E	E	E	ERROR MESSAGE WRITE (SOURCE=EIERROR)
ESYSMAX	FORT	I	I	I	I	ARRAY SIZES FOR COMPILATION
GDIPSECT	ASM	R	X	E	X	READ GDIP DATA (SOURCE=EIGDIP4)
GDIPF4	FORT	R	Y	E	X	READ FULLWORD GDIP DATA (SOURCE=XGDIPF4)
GDIPH4	FORT	R	Y	E	X	READ HALFWORD GDIP DATA (SOURCE=XGDIPH4)
GDIPX4	FORT	R	Y	E	X	READ BYTE SIZE GDIP DATA (SOURCE=XGDIPX4)
GDIP4	ASM	E	E	E	E	ENTRY POINT OF GDIPSECT
LBL	ASM	M	M	M	M	MACRO TO GENERATE UNIQUE LABEL
LEAVE	ASM	M	M	M	M	STANDARD ASSEMBLY LANGUAGE EXIT MACRO
LODCOM	ASM	E	E	E	E	SYSTEM CHAR ADDR & LGTH LOAD (EP IN EIPSAV)
NDBOR	FORT	R	X	E	X	READ GDIP FORMAT SPEC'D BY USER(SOURCE=XNDBOR)
NODIMENS	ASM	M	M	M	M	ESTAB. NO. DIMENSIONS FOR BUILDING GDIP TABLE
PSEUDO	ASM	E	E	E	E	XPSEUDO MAIN ENTRY
SPIEL	ASM	E	E	E	E	INIT. FOR PROGRAM INTERRUPT(ENTRY IN TRACBK)
SPIELP	ASM	E	E	E	E	PRINT 6 LINES FOR PROGRAM INTERRUPT(EP-TRCBKP)
SPIELQ	ASM	E	E	E	E	PRINT TERMINATION MESSAGE (EP-TRCBKP)
SUDOGO	ASM	E	E	E	E	INITIALIZE PSEUDO I/O (EP-XPSEUDO)
TIMES	ASM	R	X	E	X	READ SYSTEM CLOCK (SOURCE=DTIMEL)
TRACBK	ASM	R	Y	E	X	PERFORM TRACEBACK (SOURCE=XTRACBK)
TRCBKI	FORT	E	Y	E	E	PRINT HEADING LINE (ENTRY IN TRCBKP)
TRCBKP	FORT	R	Y	E	E	PRINT CALLING ROUTINE INFO (SOURCE=XTRCBKP)
TRCBKR	FORT	E	E	E	E	PRINT 3 LINES FOR GEN. REG. (ENTRY IN TRCBKP)
TRCBKV	FORT	E	E	E	E	PRINT 2 LINES FOR ARGUMENT (ENTRY IN TRCBKP)
UNDO	ASM	M	M	M	M	STANDARD REGISTER RESTORE MACRO
XGDIPF4	FORT	R	Y	X	X	SOURCE MEMBER NAME OF SUBRTN GDIPF4
XGDIPH4	FORT	R	Y	X	X	SOURCE MEMBER NAME OF SUBRTN GDIPH4
XGDIPX4	FORT	R	Y	X	X	SOURCE MEMBER NAME OF SUBRTN GDIPX4
XNDBOR	FORT	R	Y	X	X	SOURCE MEMBER NAME OF SUBROUTINE NDBOR.
XPSEUDO	ASM	R	Y	X	X	PROVIDE PSEUDO I/O
XTRACBK	ASM	R	Y	X	X	SOURCE MEMBER NAME OF SUBROUTINE TRACBK
XTRCBKP	FORT	R	Y	X	X	SOURCE MEMBER NAME OF SUBROUTINE TRCBKP

NOTATIONS:

* = DOES NOT INCLUDE PREAMBLE CODE SEGMENTS(XXXXXX0)

TYPE:

- C = COMMON AREA DEFINITION (WHICH IS INCLUDED)
- E = ENTRY POINT (IN ROUTINE WHOSE NAME IS GIVEN IN DESCRIPTION)
- I = INCLUDED MEMBER
- M = MACRO (PLI OR ASM)
- R = ROUTINE

PDL:

- C = PDL NOT GIVEN: COMMON AREA DEFINITION
- E = PDL NOT GIVEN: ENTRY POINT
- I = PDL NOT GIVEN: INCLUDED MEMBER TREATED IN THE PDL FOR HIGHER LEVEL SEGMENT
- M = PDL NOT GIVEN: MACRO
- X = PDL GIVEN
- Y = PDL NOT GIVEN: EXISTING CODE

COMPONENT DESCRIPTION:

- C = COMPONENT DESCRIPTION NOT GIVEN: COMMON AREA DEFINITION
- E = COMPONENT DESCRIPTION NOT GIVEN: ENTRY POINT
- I = COMPONENT DESCRIPTION NOT GIVEN: INCLUDED MEMBER DESCRIBED IN HIGHER LEVEL COMPONENT DESCRIPTION
- M = COMPONENT DESCRIPTION NOT GIVEN: MACRO
- X = COMPONENT DESCRIPTION GIVEN

TABLE 1-2. (5 of 5) INPUT PROCESSOR CODE SEGMENTS

PREAMBLE:

C = PREAMBLE NOT GIVEN: COMMON AREA DEFINITION
E = PREAMBLE NOT GIVEN: ENTRY POINT
I = PREAMBLE NOT GIVEN: INCLUDED MEMBER IN HIGHER LEVEL SEGMENT
M = PREAMBLE NOT GIVEN: MACRO
X = PREAMBLE GIVEN
Y = PREAMBLE NOT GIVEN: EXISTING CODE

TABLE 1-3. (1 of 6) MODEL PROCESSOR CODE SEGMENTS

DESM - MODEL PROCESSOR --- CODE SEGMENTS & ENTRY POINTS (*)						
NAME	LTB	T Y P E	P D L	C D .	P R E A M	DESCRIPTION
ATYPE	ASM	M	M	M	M	MACRO TO DEFINE CONSTANTS BY TYPE
CALLS	ASM	M	M	M	M	STANDARDIZED ASSEMBLY LANGUAGE ROUTINE LINKGE
COMN	ASM	M	M	M	M	GDIP COMMON AREA CSECT GENERATION MACRO
DAYTIM	FORT	R	X	X	X	OBTAIN CURRENT DATE AND TIME YY/MM/DD/HH/MM
DBUG	PLI	M	M	M	M	WRITE INTERMEDIATE OUTPUT
DO	ASM	M	M	M	M	STANDARDIZED REGISTER SAVE MACRO
DQUE	PLI	M	M	X	M	DEQUEUE XTN FIFO
DQUEM	PLI	M	M	X	M	DEQUEUE XTN FROM ANYWHERE IN CHAIN
DTIMEL	ASM	R	X	X	X	SOURCE MEMBER FOR SUBROUTINE TIMES
EAAFISM	FORT	R	X	X	X	ACTIVE FLEET SIZE MANAGEMENT
EAAFISM1	FORT	I	I	I	I	UPDATE SYSTEM LEVEL VEH STATISTICS
EAAFISM2	FORT	I	I	I	I	INITIALIZE A VEH XACTION FOR D.R. SERVICE
EAAFISM3	FORT	I	I	I	I	MARK PRO-RATION OF TRAINS TO DEBOARD IMMEDIATE
EAA SYN	FORT	R	X	X	X	ASYNCHRONOUS EVENT PROCESSING
EAA SYN1	FORT	I	I	I	I	RECOMPUTE ETA TABLE BASED ON NEW MIN PATH IPT
EACKPT	FORT	E	E	E	E	WRITE CHECKPOINT RECORD(EP-EACKR)
EACKR	FORT	R	X	X	X	CHECKPOINT & RESTART PROCESSING
EACKR1	FORT	I	I	I	I	REPOSITION INPUT FILES FOR SYS RESTART
EACOMN	FORT	R	X	X	X	INPUT COMMON AREA ORDERING
EADADD	FORT	R	X	X	X	INITIALIZE SYS STATUS AREA (STRUC DATA)
EAERR	FORT	R	X	X	X	SOURCE MEMBER NAME OF SUBROUTINE ERROR
EAFINS	FORT	R	X	X	X	SYSTEM TERMINATION PROCESSING
EAFLAG	FORT	R	X	X	X	INTERMEDIATE OUTPUT FLAG PROCESSING
EAFRPT	FORT	R	X	X	X	SYSTEM TERMINATION REPORT GENERATION
EAFTRN	FORT	R	X	X	X	PUSH COUPLING ENTRAINMENT
EAGCTL	FORT	I	X	X	X	GUIDEWAY PROCESSING ARCHITECTURE CONTROL (INCLUDED SEGMENT IN EMODEL)
EAINDX	FORT	R	X	X	X	DECODE PARM FIELD INFO & WRITE INDEX FILE (E.P. EAWTIX)
EAINIT	FORT	R	X	X	X	INITIALIZE SIMULATION
EAINIT1	FORT	I	I	I	I	INITIALIZE SYSTEM LEVEL EVENTS
EAI VEH	FORT	R	X	X	X	INITIALIZE NEW VEHICLES
EANDR	FORT	R	X	X	X	DEMAND RESPONSIVE SERVICE INIT
EANDTA	FORT	E	E	E	E	READ INPUT DATA INTO INPUT COMMONS(EP-EADADD)
EANFEL	FORT	R	X	X	X	INITIALIZE FUTURE EVENT LIST
EANMDL	FORT	R	X	X	X	GUIDEWAY & STATION MODEL INITIALIZATION
EANMDL1	FORT	I	I	I	I	INIT GENERIC STATION LINK TYPES
EANMDL1A	FORT	I	I	I	I	DEFINE PSEUDO STORAGE LINK FOR INITIALIZATION
EANMDL2	FORT	I	I	I	I	INITIALIZE VEHICLE TRANSACTION DATA
EANMDL3	FORT	I	I	I	I	COMPUTE ETA TABLE BASED ON MINIMUM PATH
EANMRG	FORT	R	X	X	X	TIMEOUT/GROUP DEMAND RESPONSIVE SERVICE INIT
EANRPT	FORT	R	X	X	X	INITIAL CONDITIONS REPORT GENERATION
EANRPT1	FORT	I	I	I	I	STATION CONNECTIVITY REPORT GENERATION
EANSV	ASM	R	X	X	X	INIT CKPT & SYSTEM DATA READ PROCESSES
EANSCD	FORT	R	X	X	X	SCHEDULED SERVICE INITIALIZATION
EANSCD1	FORT	I	I	I	I	COMPUTE CONSTANT STATION TIME STORE/LAUNCH
EANSCD2	FORT	I	I	I	I	SCHEDULE VEHICLE FOR STATION DEPARTURE
EANTIX	ASM	R	X	X	X	SAVE PARM FIELD ADDRESS & INVOKE MAIN CONTROL (E.P. EANTIX)
EANTRN	FORT	R	X	X	X	ENTRAIN AND LAUNCH VEHICLES IN SCHEDULED SERVICE ACTIVE FLEET SIZE MANAGEMENT
EANTSA	FORT	R	X	X	X	INITIALIZE SYSTEM STATUS AREA ADDRESSES
EANXTN	FORT	R	X	X	X	INITIALIZE XTN HEADER DATA & AVAILABLE LISTS
EAPCMP	FORT	R	X	X	X	PERIODIC COMPUTATION EVENTS
EAPCMP1	FORT	I	I	I	I	HEURISTIC MERGE CONTROL COMPUTATION
EAPCMP2	FORT	I	I	I	I	FAILURE DETECTION EVENT PROCESSING
EAPCMP2A	FORT	I	I	I	I	PERFORM VEHICLE ENTRAINMENT
EAPCMP2B	FORT	I	I	I	I	MARK FAILURE INACTIVE

TABLE 1-3. (2 of 6) MODEL PROCESSOR CODE SEGMENTS

NAME	LIB	T Y P E	P D L	C D .	P R E A M	DESCRIPTION
EAPCMP3	FORT	I	I	I	I	FAILURE RESTART EVENT PROCESSING
EAPCMP3A	FORT	I	I	I	I	RESTART HELD VEHICLES
EAPCMP4	FORT	I	I	I	I	FAILURE REPLACEMENT VEHICLE INITIALIZATION
EAPCMP5	FORT	I	I	I	I	TIMEOUT/GROUP DR MAX PASSENGER WAIT EVENT
EAPCMP6	FORT	I	I	I	I	TIMEOUT/GROUP DR INVENTORY MANAGEMENT
EAPCMP6A	FORT	I	I	I	I	PERFORM INVENTORY REQUEST
EAPFEL	FORT	R	X	X	X	PUT TRANSACTION ON FUTURE EVENT LIST
EAPFEL1	FORT	I	I	I	I	FIND TIME ORDER POSITION WITHIN CHAIN
EAPLNK	FORT	R	X	X	X	GUIDEWAY LINK DEQUEUEING CONTROL
EAPLNK1	FORT	I	I	I	I	DEVELOP LIST OF POSSIBLE VEH FOR DEQUEUEING
EAPLNK2	FORT	I	I	I	I	PRIORITIZE POSSIBLE VEH FOR DEQUEUEING
EAPLNK2A	FORT	I	I	I	I	DETERMINE LOCAL (STA EXIT) PRIORITIES
EAPLNK3	FORT	I	I	I	I	DEQUEUE VEH FROM GUIDEWAY LINK QUEUES
EAPRMT	FORT	I	X	X	X	SYSTEM VEH DEQUEUEING CONTROL (INCL. IN EMODEL)
EAPSTN	FORT	R	X	X	X	STATION DEQUEUEING CONTROL
EAPSTN1	FORT	I	I	I	I	DEVELOP LIST OF POSSIBLES FOR DEQUEUEING
EAPSTN2	FORT	I	I	I	I	PRIORITIZE POSSIBLE VEH FOR DEQUEUEING
EAPSTN3	FORT	I	I	I	I	DEQUEUE VEH FROM STATION QUEUES
EARESI	FORT	E	E	E	E	READ CHECKPOINT RECORDS & RESET FILES (EP-EACKR)
EARFEL	FORT	I	X	X	X	REMOVE MOST IMMINENT XTN FROM FEL (INC EMODEL)
EARFEL1	FORT	I	I	I	I	RELOAD CLOCK TABLE FROM MULTIPLE THREAD CHAIN
EARRPT	FORT	R	X	X	X	RESTART REPORT GENERATION
EASAMP	FORT	R	X	X	X	SAMPLE EVENT PROCESSING
EASCTL	FORT	I	X	X	X	STATION PROCESSING ARCHITECTURE CONTROL (INCLUDED IN EMODEL)
EASRPT	FORT	R	X	X	X	INTERMEDIATE SAMPLING REPORT GENERATION
EASTOR	FORT	R	X	X	X	ASSIGN STORED VEHICLES IN SCHED SERVICE AFSM
EATORG	FORT	R	X	X	X	ASYNCHRONOUS TRIP ARRIVAL PROCESSING
EATORG1	FORT	I	I	I	I	INITIALIZE A TRIP TRANSACTION
EAUPTX	ASM	E	E	E	E	RETRIEVE PARM FIELD ADDRESS FOR INDEX UPDATE
EAWTIX	FORT	E	E	E	E	WRITE MEMBER NAME FOR OUTPUT FILES TO INDEX (E.P. EAINDX)
EAZNIT	FORT	R	X	X	X	SYSTEM INITIALIZE STATISTICAL VARIABLES
ECAMSG	FORT	C	X	C	C	MESSAGE DATA
ECICOMN	ASM	I	I	I	I	GDIP COMMONS FOR IP AND MP
ECIFEL	FORT	C	X	C	C	FUTURE EVENTS LIST TIMING INPUT DATA
ECIGL	FORT	C	X	C	C	GUIDEWAY LINK DATA
ECIMAX	FORT	C	X	C	C	RUN TIME MAXIMA
ECINET	FORT	C	X	C	C	NETWORK STRUCTURED DATA
ECIPOL	FORT	C	X	C	C	POLICY DATA
ECISL	FORT	C	X	C	C	STATION LINK DATA
ECISTN	FORT	C	X	C	C	STATION DATA
ECISYS	FORT	C	X	C	C	SYSTEM DATA
ECIVEH	FORT	C	X	C	C	VEHICLE DATA
ECMCOM	FORT	I	I	I	I	LIST OF MP DATA COMMON INCLUDES
ECHFEL	FORT	C	X	C	C	FEL TIMING DATA MAINTAINED BY MP
ECHMGL	FORT	C	X	C	C	GUIDEWAY LINK DATA
ECHPOL	FORT	C	X	C	C	POLICY DATA USED BY MODEL
ECHSL	FORT	C	X	C	C	STATION LINK DATA MAINTAINED BY MODEL PROC.
ECHST	FORT	I	I	I	I	LIST OF MP STATISTICS COMMON INCLUDES
ECHSTN	FORT	C	X	C	C	STATION DATA
ECHSYS	FORT	C	X	C	C	SYSTEM DATA MAINTAINED BY MODEL PROC.
ECHTRP	FORT	C	X	C	C	TRIP DATA
ECHVEH	FORT	C	X	C	C	VEHICLE DATA
ECHXTN	FORT	C	X	C	C	TRANSACTION HEADER DATA MAINTAINED BY MP
ECOMMAX	ASM	I	I	I	I	ARRAY SIZES FOR ASSEMBLY
EGALT	FORT	R	X	X	X	ALTERNATE PATH COST COMPUTATION
EGASTN	FORT	R	X	X	X	ALTERNATE STATION ASSIGNMENT
EGCNXT	FORT	R	X	X	X	NXT STA DETERMINATION, CIRCUITOUS EMPTY ROUTING
EGDSTP	FORT	R	X	X	X	DEMAND STOP SERVICE PROCESSING

TABLE 1-3. (3 of 6) MODEL PROCESSOR CODE SEGMENTS

NAME	LIB	T Y P E	P D L	C D .	P R E A M	DESCRIPTION
EGDSTP1	FORT	I	X	I	I	DEMAND STOP, NEXT STATION DETERMINATION
EGDTRN	FORT	R	X	X	X	GUIDEWAY LINK VEHICLE DETRAINMENT
EGEMTY	FORT	R	X	X	X	GUIDEWAY LINK EMPTY VEHICLE PROCESSING
EGETRN	FORT	R	X	X	X	GUIDEWAY LINK DYNAMIC VEHICLE ENTRAINMENT
EGFAIL	FORT	R	X	X	X	GUIDEWAY LINK FAILURE/RECOVERY PROCESSING
EGFNTR	FORT	R	X	X	X	FIXED HEADWAY TRAVEL SEGMENT ENTRY
EGFTVL	FORT	R	X	X	X	FIXED HEADWAY TRAVEL SEGMENT TRAVERSAL
EGFTVL1	FORT	I	X	I	I	FXD HEADWAY TRAVERSAL, MERGE PRIORITY DETERMIN
EGGNXT	FORT	R	X	X	X	GUIDEWAY---NEXT ENTITY DETERMINATION
EGHTRN	FORT	R	X	X	X	TRAIN LINK HEADWAY TRAVERSAL PROCESSING
EGLEAV	FORT	R	X	X	X	GUIDEWAY LINK LEAVE PROCESSING
EGLEAV1	FORT	I	I	I	I	COLLECT GUIDEWAY & VEHICLE TRAVEL STATISTICS
EGLEAV2	FORT	I	I	I	I	UPDATE GUIDEWAY LINK STATUS FOR LEAVING VEH
EGLEAV3	FORT	I	I	I	I	CHECK IF LEAVING VEHICLE CLEARS TOW PATH
EGLMDL	FORT	R	X	X	X	GUIDEWAY LINK MODEL CONTROL
EGLNTR	FORT	R	X	X	X	GUIDEWAY LINK ENTRY PROCESSING
EGLNTR1	FORT	I	X	I	I	GUIDEWAY LINK ENTRY, STATISTICAL PROCESSING
EGLWTQ	FORT	R	X	X	X	ENTER A VEHICLE IN A GUIDEWAY LINK QUEUE
EGNEXT	FORT	R	X	X	X	NEXT ENTITY DETERMINATION CONTROL
EGNEXT1	FORT	I	I	I	I	LINK EXIT FAILURE PROCESSING
EGNEXT2	FORT	I	I	I	I	DETERMINE NEXT STATION - SCHEDULED SERVICE
EGPATH	FORT	R	X	X	X	REAL-TIME PATH SELECTION
EGPRMY	FORT	R	X	X	X	PRIMARY PATH COST COMPUTATION
EGQMRG	FORT	R	X	X	X	QUASI-SYNCHRONOUS LONGITUDINAL CONTROL
EGQMRG1	FORT	I	I	I	I	DETERMINE COMPETING VEHICLES AT MERGE ENTRY
EGQMRG2	FORT	I	I	I	I	PERFORM LOCAL MERGE (STATION EXIT) PRIORITIES
EGQMRG3	FORT	I	I	I	I	COMPUTE VEHICLE RETARDATION MANEUVER
EGQMRG4	FORT	I	I	I	I	COMPUTE VEHICLE ADVANCE MANEUVER
EGQNTR	FORT	R	X	X	X	GUIDEWAY LINK ADVANCE POSITIONING PROCESSING
EGRESV	FORT	R	X	X	X	GUIDEWAY LINK EMPTY & RESERVED VEH PROCESSING
EGRESV1	FORT	I	X	I	I	TRIP SERVICE CRITERIA CHECK AND DIVERSION CTL
EGSCHD	FORT	R	X	X	X	SCHEDULE VEHICLE FOLLOWER
EGTCHK	FORT	R	X	X	X	TRIP COMPATIBILITY CHECK
EGTCHK1	FORT	I	X	I	I	STA TRIP DESTINATION COMPARISON WITH BYPASSED STATIONS
EGTCHK2	FORT	I	X	I	I	ULTIMATE DESTINATION COMPARISON WITH BYPASSED STATIONS OF STATION TRIP ROUTE
EGTCTL	FORT	R	X	X	X	TRIP COMPATIBILITY CHECK CONTROL
EGTEST	FORT	R	X	X	X	GUIDEWAY LINK ENTRY TESTING
EGTEST1	FORT	I	I	I	I	LINK ENTRY FAILURE PROCESSING
EGTRNC	FORT	R	X	X	X	TRAIN COMPATIBILITY CHECK AT NETWORK DIVERGE
EGUNTR	FORT	R	X	X	U	USER INTERFACE ROUTINE FOR LINK ENTRY PROCESS
EGUTVL	FORT	R	X	X	U	USER INTERFACE ROUTINE FOR LINK TRAVERSAL
EGVALS	FORT	R	X	X	X	VEH ETA AND STATION ARRIVAL LIST RECORDING
EGVALS1	FORT	I	X	I	I	VEHICLE ETA INSERTION IN ARRIVAL LIST
EGVLOG	FORT	R	X	X	X	LOGGING OF VEH ARRIVALS AT STATION ENTRY
EGVHTR	FORT	R	X	X	X	VARIABLE HEADWAY TRAVEL SEGMENT ENTRY
EGVTVL	FORT	R	X	X	X	VARIABLE HEADWAY TRAVEL SEGMENT TRAVERSAL
EGVTVL1	FORT	I	X	I	I	VAR HEADWAY TRAVERSAL, MERGE PRIORITY DETER.
EHEADERS	FORT	I	X	I	I	ASYNCHRONOUS INPUT TYPES DEFINITION
EMGDIP4	ASH	R	X	X	X	SOURCE MEMBER NAME OF SUBROUTINE GDIPSECT
EMODEL	FORT	R	X	X	X	DESM MODEL PROCESSOR CONTROL
ENDEFS	ASH	M	M	M	M	GDIP COMMON AREA PROCESSOR MACRO
ENTER	ASH	M	M	M	M	STANDARD ASSEMBLY LANGUAGE ENTRY MACRO
ERROR	FORT	M	E	E	E	ERROR MESSAGE WRITE (SOURCE=EAERR)
ESAREQ	FORT	R	X	X	X	TIMEOUT/GROUP DR ASSIGN REQUEST TO VEHICLE
ESASAV	ASH	R	X	X	X	DEFINE SYSTEM AREA ORGANIZATION
ESBDL	FORT	R	X	X	X	BOARD LIST GENERATION
ESBDL1	FORT	I	I	I	I	PROCESS RESERVED WAITING TRIPS
ESBDL2	FORT	I	I	I	I	PROCESS WAITING TRIPS IN BOARDING QUEUE

TABLE 1-3. (4 of 6) MODEL PROCESSOR CODE SEGMENTS

NAME	LIB	T Y P E	P D L	C D .	P R E A M	DESCRIPTION
ESBDL3	FORT	I	I	I	I	SCHEDULED SERVICE TRIP SELECTION
ESBDL4	FORT	I	I	I	I	TIMEOUT/GROUP DR TRIP SELECTION
ESDBL	FORT	R	X	X	X	DEBOARD LIST DETERMINATION & GENERATION
ESDIVF	FORT	R	X	X	X	STA MODEL DIVERGE FUNCTION -USER INTERFACE
ESDIVO	FORT	R	X	X	X	ORDER STA LINKS BY OCC - USER INTERFACE
ESEVA	FORT	R	X	X	X	EMPTY VEHICLE MANAGEMENT CONTROL
ESEVA1	FORT	I	I	I	I	ANTICIPATED NOT CONSIDERING CURRENT DIS
ESEVA2	FORT	I	I	I	I	ANTICIPATED NEED CONSIDERING CURRENT DIS
ESEVA3	FORT	I	I	I	I	ASSIGN VEHICLE TO CIRCUITOUS ROUTE
ESEVA4	FORT	I	I	I	I	ROUTE EMPTY TO NEXT BEST STATION
ESEVB	FORT	R	X	X	X	EMPTY VEHICLE BUMPING
ESEVB1	FORT	I	I	I	I	ANTICIPATED NOT CONSIDERING CURRENT DIS
ESEVB2	FORT	I	I	I	I	ANTICIPATED NEED CONSIDERING CURRENT DIS
ESEVB3	FORT	I	I	I	I	CIRCUITOUS ROUTE ASSIGNED
ESEVB4	FORT	I	I	I	I	STATION WITH MAX OUTSTANDING REQUESTS
ESFAIL	FORT	R	X	X	X	STATION FAILURE/RECOVERY PROCESSING
ESLDLY	FORT	R	X	X	X	LAUNCH DELAY COMPUTATION
ESLEAV	FORT	R	X	X	X	PROCESS A VEHICLE LEAVING A STATION LINK
ESLEAV1	FORT	I	I	I	I	UPDATE STN LINK STATISTICS TO REFLECT LEAVING VEHICLE
ESLEAV2	FORT	I	I	I	I	DETERMINE PLATOON AVAILABILITY
ESLEAV3	FORT	I	I	I	I	DETERMINE IF TOW PATH CLEAR
ESLMDL	FORT	R	X	X	X	STATION MODEL CONTROL
ESLWTQ	FORT	R	X	X	X	ENTER A VEHICLE IN A STATION QUEUE
ESMDLA	FORT	R	X	X	X	PERFORM AFTER STATION EVENT PROCESSING
ESMDLA1	FORT	I	I	I	I	PROCESS SPLIT TRIP COMPLETION
ESMDLA2	FORT	I	I	I	I	UPDATE SYSTEM STATISTICS
ESMDLA3	FORT	I	I	I	I	UPDATE VEHICLE STATUS
ESMDLB	FORT	R	X	X	X	DO INTIAL PROCESSING FOR A STN MODEL EVENT
ESMDLB1	FORT	I	I	I	I	PERFORM DEBOARD PROCESSING
ESMDLB2	FORT	I	I	I	I	RANK STATIONS BY NOMINAL TRAVEL TIME
ESMDLN	FORT	R	X	X	X	DETERMINE NEXT STATION MODEL EVENT
ESMDLN1	FORT	I	I	I	I	DETERMINE IF NEXT EVENT CAN BE PERFORMED
ESMDLN2	FORT	I	I	I	I	UPDATE SYSTEM STATISTICS
ESMDLY	FORT	R	X	X	X	DO MERGE DELAY COMPUTATION - DETERMINISTIC DISPATCH
ESMDLY1	FORT	I	I	I	I	RESET MERGE RESERVATION TBL PTR & ROTATE LIST
ESMDLY2	FORT	I	I	I	I	FIND MERGES ON VEH PATHS & ARRIVAL TIMES
ESMDLY3	FORT	I	I	I	I	ATTEMPT TO SCHED MERGES ON POTENTIAL VEH PATHS
ESNEXT	FORT	R	X	X	X	DETERMINE NEXT STATION ENTITY FOR VEHICLE
ESNEXT1	FORT	I	I	I	I	DETERMINE LOCAL (STA EXIT) MERGE PRIORITIES
ESNEXT2	FORT	I	I	I	I	DETRAIN VEH HAVING DIFFERENT STN DESTINATIONS
ESNEXT3	FORT	I	I	I	I	DEVELOP LIST OF POSSIBLE NEXT LINKS
ESNEXT3A	FORT	I	I	I	I	ORDER POSSIBILITIES BASED ON VEHICLE STATUS
ESNSTN	FORT	R	X	X	X	DETERMINE VEHICLES NEXT STATION
ESNSTN1	FORT	I	I	I	I	SCHEDULED SERVICE NEXT STATION SELECTION
ESNTRN	FORT	R	X	X	X	PERFORM STATIC VEHICLE ENTRAINMENT
ESPATH	FORT	R	X	X	X	PATH DETERMINATION CONTROL
ESPATH1	FORT	I	I	I	I	EVALUATE POTENTIAL VEHICLE PATHS
ESPATH1A	FORT	I	I	I	I	PRIMARY PATH EVALUATION
ESPATH1B	FORT	I	I	I	I	ALTERNATE PATH EVALUATION
ESSDLY	FORT	R	X	X	X	COMPUTE SCHEDULED SERVICE SCHEDULE DELAY
ESTABQ	FORT	R	X	X	X	PERFORM TRIP/VEH ASSIGNMENT AT TRIP ARRIVAL
ESTABQ1	FORT	I	I	I	I	AVAILABLE ARRIVING NON EMPTY
ESTABQ2	FORT	I	I	I	I	UPSTREAM IN STATION
ESTABQ3	FORT	I	I	I	I	AVAILABLE STORAGE VEHICLE
ESTABQ4	FORT	I	I	I	I	CIRCUITOUS EMPTY
ESTABQ5	FORT	I	I	I	I	SCHEDULED SERVICE CURRENTLY BOARDING VEHICLE
ESTABQ6	FORT	I	I	I	I	AVAILABLE ARRIVING EMPTY REGARDLESS OF STATUS
ESTABQ7	FORT	I	I	I	I	TIMEOUT/GROUP DEMAND RESPONSIVE SERVICE

TABLE 1-3. (5 of 6) MODEL PROCESSOR CODE SEGMENTS

NAME	LIB	T Y P E	P D L	C D .	P R E A M	DESCRIPTION
ESTCHK	FORT	R	X	X	X	D.R. MULTI-PARTY TRIP COMPATIBILITY CHECKING
ESTCHK1	FORT	I	I	I	I	COMPATIBILITY WITH ON BOARD TRIPS
ESTCHK2	FORT	I	I	I	I	COMPATIBILITY WITH RESERVED TRIPS
ESTCHK3	FORT	I	I	I	I	COMPATIBILITY/ BOARDING VEHICLE ON BOARD TRIPS
ESTCHK4	FORT	I	I	I	I	COMPATIBILITY/ BOARDING VEHICLE BOARDING TRIPS
ESTEST	FORT	R	X	X	X	STN MODEL NEXT ENTITY AVAILABILITY TESTING
ESTEST1	FORT	I	I	I	I	PLATOON COMPATIBILITY CHECKING TO DOCK AREA
ESTEST2	FORT	I	I	I	I	PROMPT STORED VEHICLE
ESTEST3	FORT	I	I	I	I	TIMEOUT/GROUP DR STATION OVERFULL PROTECTION
ESTLOG	FORT	R	X	X	X	PERFORM COMPLETED TRIPS LOGGING
ESVALS	FORT	R	X	X	X	RECORD VEH ARRIVAL IN A DOWNSTREAM ARRIVAL LST
ESVALS1	FORT	I	I	I	I	VEHICLE ID IN ARRIVAL LIST TIME ORDERED
ESVREQ	FORT	R	X	X	X	TIMEOUT/GROUP DR PROCESS READY VEHICLE REQUEST
ESVRES	FORT	R	X	X	X	RESERVE SPACE ON A VEHICLE FOR A TRIP
ESYSMAX	FORT	I	I	I	I	ARRAY SIZES FOR COMPILATION
EUDIVF	FORT	R	X	X	U	USER DIVERGE FUNCTION INTERFACE ROUTINE
EUEVA	FORT	R	X	X	U	USER EMPTY VEH ASSINMENT INTERFACE ROUTINE
EUEVB	FORT	R	X	X	U	USER EMPTY VEHICLE BUMPING ALGORITHM
EUPCMP	FORT	R	X	X	U	USER PERIODIC COMPUTE EVENT INTERFACE ROUTINE
EZHDR	FORT	R	X	X	X	WRITE A SAMPLING HEADER RECORD
EZINT	FORT	R	X	X	X	COMPUTE TIME INTEGRALS
EZINT1	FORT	I	I	I	I	COMPUTE SYS LEVEL STATISITCS FOR SAMPLING FILE
EZZERO	FORT	R	X	X	X	RESET SAMPLING STATISITCS
EZZERO1	FORT	I	I	I	I	RESET SYSTEM LEVEL STATISITCS
FREE	PLI	M	M	X	M	RETURN XTN TO AVAILABLE CHAIN
GDIPSECT	ASM	R	X	X	X	READ GDIP DATA (SOURCE=EMGDIP4)
GDIPF4	FORT	R	Y	Y	X	READ FULLWORD GDIP DATA (SOURCE=XGDIPF4)
GDIPH4	FORT	R	Y	Y	X	READ HALFWORD GDIP DATA (SOURCE=XGDIPH4)
GDIPX4	FORT	R	Y	Y	X	READ BYTE SIZE GDIP DATA (SOURCE=XGDIPX4)
GDIP4	ASM	E	M	E	M	ENTRY POINT OF GDIPSECT
GET	PLI	M	M	X	M	GET XTN FROM AVAILABLE CHAIN
LBL	ASM	M	M	M	M	MACRO TO GENERATE UNIQUE LABEL
LEAVE	ASM	M	M	M	M	STANDARD ASSEMBLY LANGUAGE EXIT MACRO
LODCOM	ASM	M	M	M	M	SYSCHAR ADDR+LGTH LOAD (EP-ESASAV)
MULTICK	PLI	M	M	X	M	CHECK IF XTN IS ALREADY IN A CHAIN
NDBOR	FORT	R	X	X	X	READ GDIP FORMAT SPEC'D BY USER (SRCE=XNDBOR)
NODIMENS	ASM	M	M	M	M	ESTAB. NO. DIMENSIONS FOR BUILDING GDIP TABLE
NQUE	PLI	M	M	X	M	ENQUEUE XTN FIFO
PROMPT	PLI	M	M	X	M	SCHEDULE PROMPT EVENT FOR GW OR STN LINK
PSEUDO	ASM	M	M	E	M	XPSEUDO-MAIN ENTRY
SCHED	PLI	M	M	E	M	SCHEDULE TRIP OR VEHICLE ON FEL
SPIEL	ASM	E	E	E	E	INIT. FOR PROGRAM INTERRUPT (ENTRY IN TRCBK)
SPIELP	ASM	E	E	E	E	PRINT 6 LINES FOR PROGRAM INTERRUPT(EP-TRCBKP)
SPIELQ	ASM	E	E	E	E	PRINT TERMINATION MESSAGE (EP-TRCBKP)
SUDOGO	ASM	E	E	E	E	INITIALIZE PSEUDO-I/O(EP-XPSEUDO)
TIMES	ASM	R	R	X	X	READ SYSTEM CLOCK (SOURCE=DTIMEL)
TRACBK	ASM	R	R	Y	X	PERFORM TRACEBACK (SOURCE=XTRACBK)
TRCBKI	FORT	R	E	Y	E	PRINT HEADING LINE (ENTRY IN TRCBKP)
TRCBKP	FORT	R	E	Y	E	PRINT CALLING ROUTINE INFO (SOURCE=XTRCBKP)
TRCBKR	FORT	R	E	E	E	PRINT 3 LINES FOR GEN. REG. (ENTRY IN TRCBKP)
TRCBKV	FORT	R	E	E	E	PRINT 2 LINES FOR ARGUMENT (ENTRY IN TRCBKP)
UNDO	ASM	M	M	H	M	STANDARD REGISTER RESTORE MACRO
XGDIPF4	FORT	R	Y	Y	X	SOURCE MEMBER NAME OF SUBRTN GDIPF4
XGDIPH4	FORT	R	Y	Y	X	SOURCE MEMBER NAME OF SUBRTN GDIPH4
XGDIPX4	FORT	R	Y	Y	X	SOURCE MEMBER NAME OF SUBRTN GDIPX4
XNDBOR	FORT	R	Y	Y	X	SOURCE MEMBER NAME OF SUBROUTINE NDBOR
XPSEUDO	ASM	R	R	Y	X	PROVIDE PSEUDO I/O
XTRACBK	ASM	R	R	Y	X	SOURCE MEMBER NAME OF SUBROUTINE TRACBK
XTRCBKP	FORT	R	Y	Y	X	SOURCE MEMBER NAME OF SUBROUTINE TRCBKP
ZCLNK	FORT	C	X	C	C	GUIDEWAY LINK STATISTICS

TABLE 1-3. (6 of 6) MODEL PROCESSOR CODE SEGMENTS

NAME	LIB	T Y P E	P D L	C D .	P R E A M	DESCRIPTION
ZCRTE	FORT	C	X	C	C	ROUTE STATISTICS
ZCSL	FORT	C	X	C	C	STATION LINK STATISTICS
ZCSTN	FORT	C	X	C	C	STATION STATISTICS
ZCSYSG	FORT	C	X	C	C	SYSTEM STATISTICS - GUIDEWAYS
ZCSYSR	FORT	C	X	C	C	SYSTEM STATISTICS - ROUTES
ZCSYSS	FORT	C	X	C	C	SYSTEM STATISTICS - STATIONS
ZCSYST	FORT	C	X	C	C	SYSTEM STATISTICS
ZCTRP	FORT	C	X	C	C	TRIP STATISTICS
ZCVEH	FORT	C	X	C	C	VEHICLE STATISTICS

NOTATIONS:

* = DOES NOT INCLUDE PREAMBLE CODE SEGMENTS (XXXXXXXX0)

TYPE:

I = INCLUDED MEMBER
 C = COMMON AREA DEFINITION INCLUDE IN HIGER LEVEL SEGMENT
 E = ENTRY POINT
 M = MACRO (PLI OR ASM)
 R = ROUTINE

PDL:

I = PDL NOT GIVEN: INCLUDED MEMBER IN HIGHER LEVEL SEGMENT
 E = PDL NOT GIVEN: ENTRY POINT
 M = PDL NOT GIVEN: MACRO DEFINITION
 X = PDL GIVEN
 Y = PDL NOT GIVEN: EXISTING CODE

COMPONENT DESCRIPTION:

C = COMPONENT DESCRIPTION NOT GIVEN: COMMON AREA DEFINITION
 E = COMPONENT DESCRIPTION NOT GIVEN: ENTRY POINT
 I = COMPONENT DESCRIPTION NOT GIVEN: INCLUDED MEMBER IN
 HIGHER LEVEL SEGMENT
 M = COMPONENT DESCRIPTION NOT GIVEN: MACRO DEFINITION
 X = COMPONENT DESCRIPTION GIVEN
 Y = COMPONENT DESCRIPTION NOT GIVEN: EXISTING CODE

PREAMBLE:

X = PREAMBLE GIVEN
 I = PREAMBLE NOT GIVEN: INCLUDED MEMBER IN HIGHER LEVEL SEGMENT
 C = PREAMBLE NOT GIVEN: COMMON AREA DEFINITION
 E = PREAMBLE NOT GIVEN: ENTRY POINT
 M = PREAMBLE NOT GIVEN: MACRO DEFINTION
 Y = PREAMBLE NOT GIVEN: EXISTING CODE
 U = COMPONENT DESCRIPTION NOT GIVEN: USER-INTERFACE

TABLE 1-4. (1 of 4) OUTPUT PROCESSOR CODE SEGMENTS

DESM - OUTPUT PROCESSOR --- CODE SEGMENTS & ENTRY POINTS (*)						
NAME	LIB	T Y P E	P D L	C D .	P R E A M	DESCRIPTION
AADATE	FORT	R	X	X	X	PROVIDE DATE IN CHARACTER FORMAT
ABIN	FORT	E	X	X	X	EABIN MAIN ENTRY
ATYPE	ASM	M	M	M	M	MACRO TO DEFINE CONSTANTS BY TYPE
BASIC	FORT	C	C	C	C	BIN ASSIGNMENT MAPPING (LOCATION ARRAY)
BNCHK	FORT	E	E	E	E	EBNCHK MAIN ENTRY
CALLS	ASM	M	M	M	M	STANDARDIZED ASSEMBLY LANGUAGE ROUTINE LINKGE
CKFOLLOW	PLI	M	M	X	M	CHECK FOLLOWER RECORD MACRO
DAYTIM	FORT	R	X	X	X	OBTAIN DATE & TIME YY/MM/DD/HH/MM/SS
DBIN	FORT	E	X	X	X	EDBIN MAIN ENTRY
DEBUG	PLI	M	M	X	M	WRITE INTERMEDIATE OUTPUT MACRO
DERROR	FORT	R	X	X	X	ERROR TERMINATION ROUTINE FOR STN-STN MEASURES
DO	ASM	M	M	M	M	STANDARDIZED REGISTER SAVE MACRO
DTIMEL	ASM	R	X	X	X	SOURCE MEMBER FOR SUBROUTINE TIMES
DUMBIN	FORT	E	X	X	X	EDUMBIN MAIN ENTRY
EABIN	FORT	R	X	X	X	REALLOCATE BIN STORAGE AREA
EBNCHK	FORT	R	X	X	X	PERFORM BIN AREA EXPANSION
ECAMSG	FORT	C	C	C	C	ERROR MESSAGE COMMON
EDBIN	FORT	R	X	X	X	INITIALIZE BIN STORAGE AREA
EDUMBIN	FORT	R	X	X	X	DUMP CONTENTS OF BIN AREA FOR DEBUG
EGRAPH	FORT	R	X	X	X	PRODUCE TIME SERIES PLOT
EHEADER	FORT	R	X	X	X	READ NEXT HEADER RECORD
EHIST	FORT	R	X	X	X	PRODUCE A HISTOGRAM
ELIST	FORT	R	X	X	X	PRODUCE A TIME SERIES LIST
EMNMX	FORT	R	X	X	X	COMPUTE MINIMUM AND MAXIMUM VALUES IN BIN
ENTER	ASM	M	M	M	M	STANDARD ASSEMBLY LANGUAGE ENTRY MACRO
EODATA	FORT	B	B	X	B	BLOCK DATA INITIALIZATION OF OUTPUT PROCESSOR
EODCLS	FORT	I	I	I	I	DEFINE DATA AREA FOR ROUTINES (INCLUDED SEG)
EODEFS	FORT	I	I	I	I	DEFINE OP TABLE SIZES (INCLUDED IN EOUTPT)
EOERR	FORT	R	X	X	X	SOURCE MEMBER FOR SUBROUTINE ERROR
EOFLAG	FORT	R	X	X	X	INITIALIZE AUXILIARY OUTPUT INDICATORS
EOINDX	FORT	R	X	X	X	DECODE PARM FIELD INFO & WRITE INDEX FILE
EONTIX	ASM	R	X	X	X	SAVE PARM FIELD ADDRESS & INVOKE MAIN CONTROL
EOPRPT	FORT	R	X	X	X	PERFORMANCE SUMMARY REPORT ROUTINE
EOPRPT1	FORT	I	I	I	I	PERFORMANCE SUMMARY REPORT, PART 1 FORMAT
EOPRPT2	FORT	I	I	I	I	PERFORMANCE SUMMARY REPORT, PART 2 FORMAT
EOPRPT3	FORT	I	I	I	I	PERFORMANCE SUMMARY REPORT, PART 3 FORMAT
EOPSUM	FORT	R	X	X	X	PERFORMANCE SUMMARY DATA, RESOURCE
FOPSUM1	FORT	I	I	I	I	PERFORMANCE SUMMARY DATA, PERFORMANCE
FOPSUM2	FORT	I	I	I	I	PERFORMANCE SUMMARY DATA, LEVEL OF SERVICE
EOSRPT	FORT	R	X	X	X	WRITE SYS SUMMARY STATISTICS, STD REPORT 2
EOSRPT1	FORT	I	I	I	I	STANDARD REPORT 2, PART 1 FORMAT
EOSRPT2	FORT	I	I	I	I	STANDARD REPORT 2, PART 2 FORMAT
EOSRPT3	FORT	I	I	I	I	STANDARD REPORT 2, PART 3 FORMAT
EOSSUM	FORT	R	X	X	X	SYSTEM SUMMARY STATISTICS, SYSTEM-WIDE
EOSSUM1	FORT	I	I	I	I	SYSTEM SUMMARY STATISTICS, STN & GUIDEWAY
EOSSUM2	FORT	I	I	I	I	SYSTEM SUMMARY STATISTICS, ROUTE
EOSTAT	FORT	C	C	C	C	STANDARD REPORT 2 MEASURES DATA
EOUPTX	ASM	E	X	X	X	RETRIEVE PARM FLD ADDR (EP-EONTIX)
EOUTPT	FORT	R	X	X	X	OUTPUT PROCESSOR CONTROL
EOUTPT1	FORT	I	I	I	I	PROCESS USER SAMPLING REQUEST
EOUTPT2	FORT	I	I	I	I	PERFORMANCE SUMMARY REQUEST
EOUTPT3	FORT	I	I	I	I	STATION-TO-STATION MEASURE REQUESTS
EOWTIX	FORT	E	E	E	E	WRITE MEMBER NAMES FOR OUTPUT FILES TO INDEX (EP-EOINDX)
EOZKIT	FORT	R	X	X	X	OUTPUT PROCESSOR INITIALIZATION
EREAD02	FORT	R	X	X	X	READ SYSTEM STATISTICS
EREAD021	FORT	I	I	I	I	COMPUTE PERFORMANCE SUMMARY DATA
EREAD022	FORT	I	I	I	I	COMPUTE DERIVED STATISTICS

TABLE 1-4. (2 of 4) OUTPUT PROCESSOR CODE SEGMENTS

NAME	LIB	T Y P E	P D L	C D .	P R E A M	DESCRIPTION
EREAD03	FORT	R	X	X	X	READ/PROCESS STATION STATISTICS
EREAD031	FORT	I	I	I	I	ACCUMULATE STANDARD REPORT 2 DATA
EREAD04	FORT	R	X	X	X	READ/PROCESS STATION LINK STATISTICS
EREAD041	FORT	I	I	I	I	ACCUMULATE STANDARD REPORT 2 DATA
EREAD042	FORT	I	I	I	I	COMPUTE DERIVED STATISTICS
EREAD05	FORT	R	X	X	X	READ/PROCESS GUIDEWAY LINK STATISTICS
EREAD051	FORT	I	I	I	I	ACCUMULATE STANDARD REPORT 2 DATA
EREAD052	FORT	I	I	I	I	COMPUTE DERIVED STATISTICS
EREAD06	FORT	R	X	X	X	READ/PROCESS ROUTE STATISTICS
EREAD061	FORT	I	I	I	I	ACCUMULATE STANDARD REPORT 2 DATA
EREAD062	FORT	I	I	I	I	COMPUTE DERIVED STATISTICS
EREQTLU	FORT	R	X	X	X	RECORD/REQUEST CORRELATION
EREQU	FORT	R	X	X	X	STORE A DATA REQUEST
ERROR	FORT	E	E	E	E	EOERR MAIN ENTRY
ESETUP	FORT	R	X	X	X	REINITIALIZE OOUTPUT PROCESSOR DATA TABLES
ESHIFT	FORT	R	X	X	X	SHIFT ITEMS IN A BIN
ESKIPFO	FORT	R	X	X	X	SKIP A FOLLOWER RECORD
ESTORE	FORT	R	X	X	X	STORE DATA IN BIN
ESYSMAX	FORT	I	I	I	I	COMPILE TIME MAXIMA
ETACUM	FORT	R	X	X	X	ACCUMULATE STATION-TO-STATION RAW STATISTICS
ETCAPT	FORT	R	X	X	X	CAPTURE STATION-TO-STATION RAW DATA
ETCOMP	FORT	R	X	X	X	COMPUTE REQUESTED STATION-TO-STATION MEASURES
ETMEAS	FORT	R	X	X	X	FUNCTION TO COMPUTE RAW DATA FOR THE MEASURE
ETMERG	FORT	R	X	X	X	CONVERT RAW STATISTICS TO UTPS FORMAT
ETNMBR	FORT	R	X	X	X	FUNCTION TO CONVERT A NUMBER TO EBCDIC
ETRPTS	FORT	R	X	X	X	WRITE STATION-TO-STATION REPORT
ETSSPM	FORT	R	X	X	X	STATION-TO-STATION PERFORMANCE MEASURE CONTROL
ETSTAT	FORT	R	X	X	X	COMPUTE STATISTICS
ETTSTID	FORT	R	X	X	X	FUNCTION TO CONVERT STATION ID TO EBCDIC
ETTOTL	FORT	R	X	X	X	COMPUTE RAW DATA TOTALS
EZHIST	FORT	R	X	X	X	HISTOGRAM CONTROL
EZLIST	FORT	R	X	X	X	TIME SERIES LIST CONTROL
EZPLOT	FORT	R	X	X	X	PLOT OUTPUT CONTROL
EZPEAD	FORT	R	X	X	X	READ SAMPLE FILE HEADER
GRAPH	FORT	E	E	E	E	EGRAPH MAIN ENTRY (FIRST USE OF SUBPROGRAM)
HEADER	FORT	E	E	E	E	EHEADER MAIN ENTRY
HIST	FORT	E	E	E	E	EHIST MAIN ENTRY
HISTO	FORT	C	C	C	C	HISTOGRAM DATA
IENDS	FORT	C	C	C	C	REQUEST STORAGE COUNTER
LBL	ASM	M	M	M	M	MACRO TO GENERATE UNIQUE LABEL
LEAVE	ASM	M	M	M	M	STANDARD ASSEMBLY LANGUAGE EXIT MACRO
LIST	FORT	E	E	E	E	ELIST MAIN ENTRY
MATCH	FORT	C	C	C	C	RECORD TYPE REQUEST CHAINING TABLE
MNMX	FORT	E	E	E	E	EMNMX MAIN ENTRY
NAMES	FORT	C	C	C	C	REUQUEST MNEOMIC TRANSLATION TABLE
NEWGRF	FORT	E	E	E	E	ENTRY POINT TO EGRAPH - REUSE SUBPROGRAM
NEWSCL	FORT	E	E	E	E	ENTRY POINT TO EGRAPH - REUSE SUBPROGRAM WITH NO CHANGE IN ALPHAMERICS
OUTPT	FORT	C	C	C	C	OUTPUT CONTROL DATA
OUTPT1	FORT	C	C	C	C	PLOT LIMITS FOR TIME SERIES DISPLAY
OUTPUT	FORT	C	C	C	C	OUTPUT CONTROL DATA
OUTSUM	FORT	C	C	C	C	STANDARD REPORTS DATA ACCUMULATION ARRAYS
READ02	FORT	E	E	E	E	EREAD02 MAIN ENTRY
READ03	FORT	E	E	E	E	EREAD03 MAIN ENTRY
READ04	FORT	E	E	E	E	EREAD04 MAIN ENTRY
READ05	FORT	E	E	E	E	EREAD05 MAIN ENTRY
READ06	FORT	E	E	E	E	EREAD06 MAIN ENTRY
READER	FORT	C	C	C	C	DATA ACQUISITION PARAMETERS
REQTLU	FORT	E	E	E	E	EREQTLU MAIN ENTRY
REQUES	FORT	C	C	C	C	REQUEST TABLE

TABLE 1-4. (3 of 4) OUTPUT PROCESSOR CODE SEGMENTS

NAME	LIB	T Y P E	P D L	C D .	P R E A M	DESCRIPTION
SAMSCCL	FORT	E	E	E	E	ENTRY POINT TO EGRAPH - REUSE SUBPROGRAM WITH NO CHANGE IN SCALE OR ALPHAMERIC
SETUP	FORT	E	E	E	E	ESETUP MAIN ENTRY
SHIFT	FORT	E	E	E	E	ESHIFT MAIN ENTRY
SKIPFO	FORT	E	E	E	E	ESKIPFO MAIN ENTRY
SPIEL	ASM	E	E	E	E	INTERFACE TO INTERRUPT HANDLER (EP-TRCBK)
SPIELP	FORT	E	E	E	E	PRINT 6 LINES FOR PROGRAM INTERRUPT(EP-TRCBK)
SPIELQ	FORT	E	E	E	E	PRINT TERMINATION MESSAGE (EP-TRCBK)
STOFLO	FORT	E	E	E	E	ENTRY POINT ESTORE
STORE	FORT	E	E	E	E	ESTORE MAIN ENTRY
SUB	FORT	C	C	C	C	GENERAL SYSTEM PARAMETERS
SYSCM1	FORT	C	C	C	C	SIMULATION RELATED PARAMETERS CONTD
SYSCOM	FORT	C	C	C	C	SIMULATION RELATED PARAMETERS
TABLES	FORT	C	C	C	C	REQUEST CORRELATION TABLES
TERMINL	FORT	C	C	C	C	INPUT ECHO FLAG
TIMES	ASM	E	E	E	E	OBTAIN DATE & TIME FROM SYSTEM (EP-DTIMEL)
TRACBK	ASM	E	E	E	E	XTRACBK MAIN ENTRY
TRCBKI	FORT	E	E	E	E	PRINTS HEADING LINE (EP-TRCBK)
TRCBKP	FORT	E	E	E	E	XTRCBKP MAIN ENTRY
TRCBKR	FORT	E	E	E	E	PRINTS THREE LINES FOR GEN.REG. (EP-TRCBK)
TRCBKV	FORT	E	E	E	E	PRINTS TWO LINES FOR AN ARGUMENT (EP-TRCBK)
UNDO	ASM	M	M	M	M	STANDARD REGISTER RESTORE MACRO
XTRACBK	ASM	R	Y	X	X	PERFORM TRACEBACK
XTRCBKP	FORT	R	Y	X	X	PRINTS LINE DESCRIBING CALLING ROUTINE
ZHIST	FORT	E	E	E	E	EZHIST MAIN ENTRY
ZLIST	FORT	E	E	E	E	EZLIST MAIN ENTRY
ZPLOT	FORT	E	E	E	E	EZPLOT MAIN ENTRY
ZREAD	FORT	E	E	E	E	EZREAD MAIN ENTRY
ZREQU	FORT	E	E	E	E	EREQU MAIN ENTRY
	FORT	C	C	C	C	BLANK COMMON DATA STORAGE (BIN AREA REGION)

NOTATIONS:

* = DOES NOT INCLUDE PREAMBLE CODE SEGMENTS(XXXXXXO)

TYPE:

- R = ROUTINE
- I = INCLUDED MEMBER
- M = MACRO (PLI OR ASM)
- C = COMMON AREA DEFINITION (WHICH IS INCLUDED)
- B = BLOCK DATA SUBPROGRAM
- E = ENTRY POINT (IN ROUTINE WHOSE NAME IS GIVEN IN DESCRIPTION)
- F = FUNCTION SUBPROGRAM

PDL:

- B = PDL NOT GIVEN: BLOCK DATA INITIALIZATION
- C = PDL NOT GIVEN: COMMON AREA DEFINITION
- E = PDL NOT GIVEN: ENTRY POINT
- I = PDL NOT GIVEN: INCLUDED MEMBER TREATED IN THE PDL FOR HIGHER LEVEL SEGMENT
- M = PDL NOT GIVEN: MACRO
- X = PDL GIVEN
- Y = PDL NOT GIVEN: EXISTING CODE

TABLE 4-4. (4 of 4) OUTPUT PROCESSOR CODE SEGMENTS

COMPONENT DESCRIPTION:

C = COMPONENT DESCRIPTION NOT GIVEN: COMMON AREA DEFINITION
E = COMPONENT DESCRIPTION NOT GIVEN: ENTRY POINT
I = COMPONENT DESCRIPTION NOT GIVEN: INCLUDED MEMBER DESCRIBED
 IN HIGHER LEVEL COMPONENT DESCRIPTION
M = COMPONENT DESCRIPTION NOT GIVEN: MACRO
X = COMPONENT DESCRIPTION GIVEN

PREAMBLE:

B = PREAMBLE NOT GIVEN: BLOCK DATA INITIALIZATION
C = PREAMBLE NOT GIVEN: COMMON AREA DEFINITION
E = PREAMBLE NOT GIVEN: ENTRY POINT
I = PREAMBLE NOT GIVEN: INCLUDED MEMBER IN HIGHER LEVEL SEGMENT
M = PREAMBLE NOT GIVEN: MACRO
X = PREAMBLE GIVEN

SECTION 2. PROGRAM DESCRIPTION

Figures 2-1, 2-2 and 2-3 contain tree diagrams of the control flow through the program modules which comprise the IP, MP and OP respectively.

The diagrams in Appendix B illustrate the DESM high-level design through the use of Hierarchy plus Input-Process-Output (HIPO) diagrams. The Visual Table of Contents illustrates program organization and contains the names and identification numbers of the detail Input-Process-Output diagrams that define the processing to be performed. These diagrams are intended for use in conjunction with the Process Design Language (PDL) descriptions contained in Appendix A, which provide descriptions of the program design in greater detail. Where the Visual Table of Contents and Input-Process-Output diagrams reference a segment name and identification number, that segment is further expanded in both an Input-Process-Output diagram (having that identification number) and a PDL segment (having that segment name). If an Input-Process-Output diagram references a function by segment name only, then the design of that segment can be found in the PDL segment having that segment name.

The executable load modules for the IP and MP components of the DESM are structured with defined overlay regions for establishing a common data communication interface. The overlay regions within each load module contain identical segments in which all global common area data defining the Network and System Characteristics reside. The common area data within these regions is structured in contiguous core locations beginning and ending with symbolic names which are converted to physical memory addresses during the linkage editing process. The structuring of common areas in this manner permits the Network and System Characteristics to be transmitted via the AGT data base without regard to specific variable assignments within the common area definitions.

The structuring of the overlay regions is accomplished by the following input to the linkage editor:

```
OVERLAY   REGION (REGION)
INSERT    BEGCOM
OVERLAY   REGION
INSERT    PROCESSOR COMMON 1
          :           :           :           M
```

DESM INPUT PROCESSOR:

EACOMN
 EIPSAV
 EIPARM
 EINPUT

```

  ---SPIEL (E.P. IN TRACBK)
  ---DAYTIM
    |
    |---TIMES (E.P. IN DTIMEL)
  ---NDBOR
    |
    |---GDIP4
      |
      |---GDIPF4
        |
        |---PSEUDO
      ---GDIPH4
        |
        |---PSEUDO
      ---GDIPX4
        |
        |---PSEUDO
      ---EINERR
        |
        |---EINERR1
        |---EINERR2
        |---EINERR3
        |---EINERR4
        |---EINERR5
        |---EINERR6
        |---EINERR7
        |---EINERR8
        |---ERROR
          |
          |---TRACBK
            |
            |---TRCBKP
              |
              |---EIINIT
                |
                |---LGDCOM (E.P. IN EIPSAV)
                |---EIADDR
                  |
                  |---EINADD (E.P. IN EIBWRT)
                  |---EISADD (E.P. IN EIBWRT)
                ---EIPLST
                  |
                  |---EIMNAM
                    |
                    |---DAYTIM
  ---EINPUT1
  ---EINPUT2
    |
    |---EAFLAG
  
```

```

ORDER INPUT COMMONS
PROVIDE COMMON STORAGE ADDRESSES
SAVE PARAMETER LIST
DESM INPUT PROCESSOR
INIT. PGM INTERRUPT HANDLING
GET CURRENT DATE AND TIME
READ DATE AND TIME FROM SYS CLOCK
READ GDIP FORMAT DATA
READ GDIP DATA
READ FULLWORD GDIP DATA
PSEUDO DATA READ
READ HALFWORD GDIP DATA
PSEUDO DATA READ
READ SINGLE BYTE GDIP DATA
PSEUDO DATA READ
GEN ERROR MESSAGE
PARAMETER ERROR MSG 0-99
CHARACTERISTIC ERROR MSG 100'S
NET CONFIGURATION ERROR MSG 200'S
DEMAND GEN ERROR MSG 300'S
SERVICE GEN ERROR MSG 400'S
FAIL/RECOVERY ERROR MSG 500'S
ACTIVE FLEET ERROR MSG 600'S
STATION CONFIGURE ERROR MSG 700'S
WRITE ERROR MESSAGE
SYSTEM TRACEBACK FACILITY
PRINT TRACEBACK INFORMATION
INPUT PROCESSOR INITIALIZATION
LOAD ADDR AND LGT OF STRUC DATA
PASS STRUC DATA ADDR + LGT TO SUB
SAVE NETWORK ADDR + LGT
SAVE SYS CHAR ADDR + LGT
PASS PARM LIST ADDR + LGT TO SUB
SEPARATE PARM LIST INTO SUBFIELDS
GET CURRENT DATE AND TIME
TIME FROM SYSTEM CLOCK
READ SYSTEM CHARACTERISTICS DATA
READ RUNTIME DATA
INTERMEDIATE OUTPUT REQUEST
  
```

FIGURE 2-1. (1 of 5) INPUT PROCESSOR CONTROL FLOW

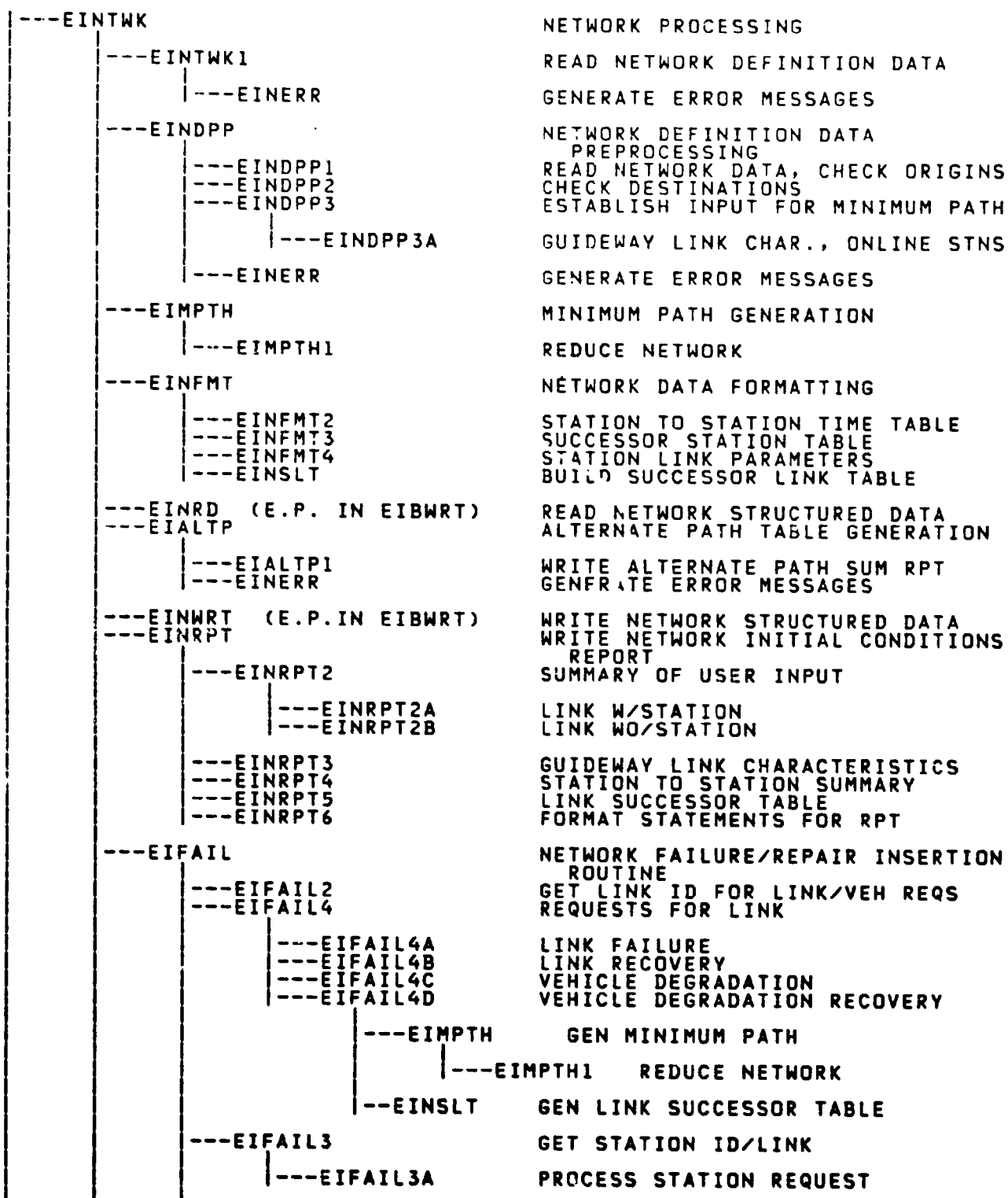


FIGURE 2-1. (2 of 5) INPUT PROCESSOR CONTROL FLOW

---	EIFAIL5	FAILURE/RECOVERY SUM RPT
---	EIFAIL5A	WRITE CURRENT MINIMUM PATH TABLE
---	EINERR	GEN ERROR MESSAGE
---	EINTWK2	WRITE FAILURE/REPAIR DATA
---	EINPUT3	READ TRIP GENERATION INITIAL CONDITIONS
---	EITNIT	TRIP DEMAND INITIALIZATION
---	EITNIT2	BUILD CUMULATIVE PROB. DSTS
---	EINERR	GENERATE ERROR MESSAGES
---	EIUDGN	UNIFORM DEMAND GENERATION
---	EITRIP	TRIP FILE GENERATION
---	EITRIP2	VERIFY INPUT
---	EITRIP3	GENERATE TRIPS
---	EINERR	GENERATE ERROR MESSAGES
---	EIRNG	GENERATE RANDOM NUMBER
---	EIRSEL	SAMPLE PROBABILITY DIST.
---	EIRNG	GENERATE RANDOM NUMBER
---	EIDRPT	WRITE TRIP GENERATION SUMMARY REPORT
---	EIDRPT2	FORMAT FOR DEMAND RPT
---	EIDRPT3	WRITE TRIP SIZE DISTRIBUTIONS
---	EIDRPT4	WRITE ORIG AND DEST PROBABILITIES
---	EIDRPT5	WRITE INPUT DEMAND MATRIX AND TRIPS GENERATED SUMMARY
---	EISCFG	STATION LINK CONFIGURATOR
---	EISCFG1	DETERMINE LINK TYPES
---	EISCFG2	VALIDATE STATION ENTRY EXIT
---	EISCFG3	BUILD STRUCTURED DATA TABLE
---	EISCFG4	BUILD UPSTREAM LINK TABLE
---	EISCFG4A	BUILD DOCK UPSTREAM LINK
---	EISCFG4B	BUILD OUTPUT Q UPSTREAM LINK
---	EISCFG4C	BUILD OUTPUT RAMP UPSTREAM LINK
---	EISCFG4D	BUILD STORAGE UPSTREAM LINK
---	EISCFG4E	BUILD DOCK TO STORAGE LINK
---	EISCFG5	BUILD DOWNSTREAM LINK TABLE
---	EISCFG5A	BUILD INPUT RAMP DOWNSTREAM LINK
---	EISCFG5B	BUILD INPUT Q DOWNSTREAM LINK
---	EISCFG5C	BUILD DOCK DOWNSTREAM LINK
---	EISCFG5D	BUILD STORAGE TO INPUT LINK
---	EINERR	GENERATE ERROR MESSAGE
---	EICHCK	DATA CHECK + INIT
---	EICHCK2	INIT GUIDEWAY PARAMETERS
---	EICHCK3	INIT STATION PARAMETERS
---	EICHCK3A	SCAN EVENT LIST FOR DELIMITERS
---	EICHCK3B	SCAN UPSTREAM LINK LIST FOR DELIMITERS

FIGURE 2-1. (3 of 5) INPUT PROCESSOR CONTROL FLOW

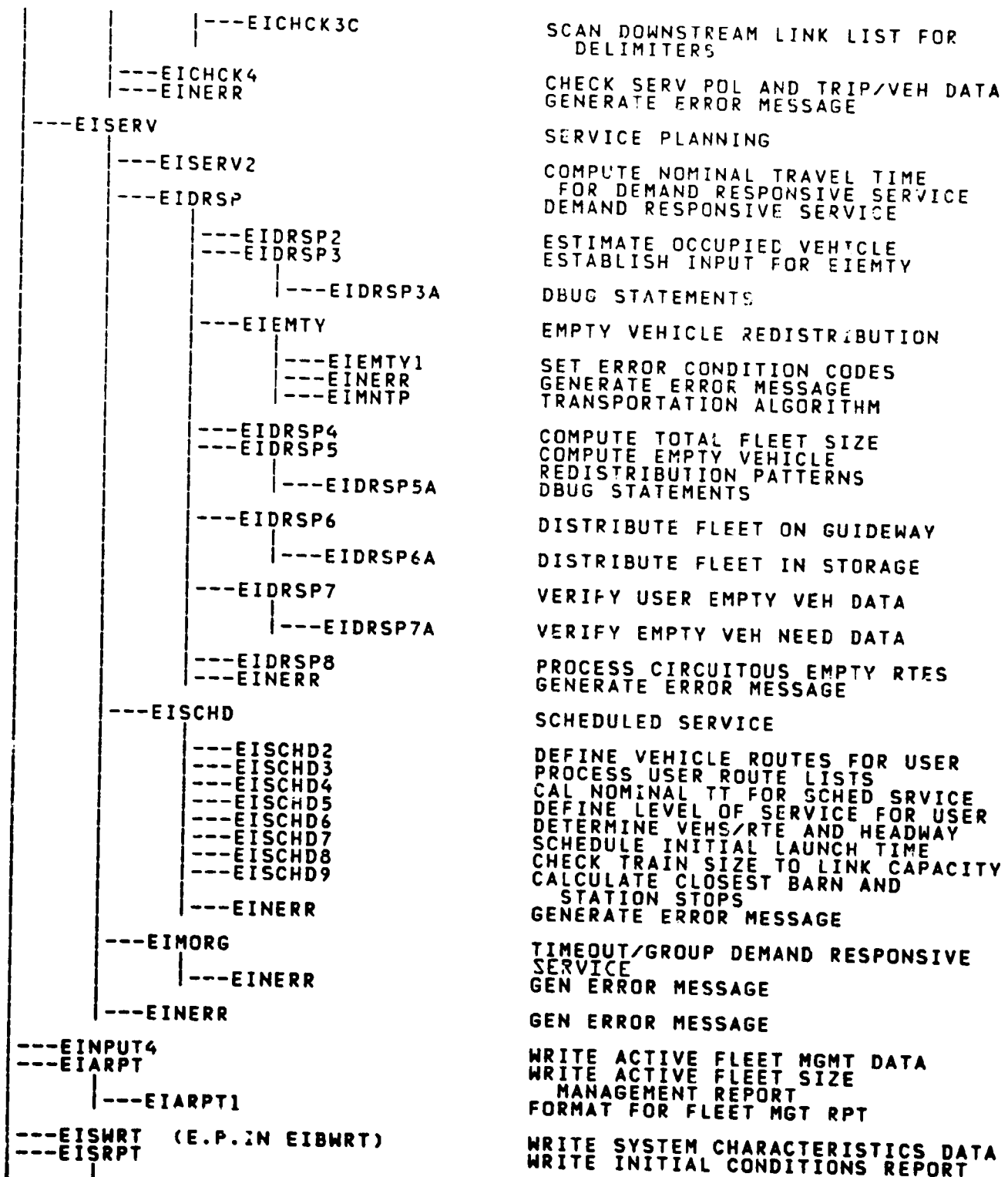


FIGURE 2-1. (4 of 5) INPUT PROCESSOR CONTROL FLOW

---EISRPT1 ---EISRPT2 ---EISRPT3 ---EISRPT4 ---EISRPT5	WRITE SYS CHAR SUMMARY WRITE STA CHAR SUMMARY WRITE TRANSIT SERVICE CHAR SUM WRITE SIM CONTROL PARAMETERS FORMAT FOR REPORTS
---EIWNAM (E.P. IN EIMNAM) ---EINERR	WRITE MEMBER NAME GENERATE ERROR MESSAGE
END	

FIGURE 2-1. (5 of 5) INPUT PROCESSOR CONTROL FLOW

```

DESM MODEL PROCESSOR:
EACOMN (ORDER INPUT COMMONS)
ESASAV (INITIALIZE ADDRESS OF STATUS)
EANTIX (SAVE PARM FIELD ADDRESS)
EMODEL (DISCRETE EVENT SIMULATOR CONTROL)
  ---EAINIT (SYSTEM LEVEL INITIALIZATION)
    ---EAREST (SYSTEM RESTART E.P. EACKR)
      ---EARRPT (SYSTEM RESTART REPORT)
        ---DAYTIM (ACQUIRE DATE AND TIME)
        ---EZHDR (WRITE A SAMPLING FILE HEADER)
          ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
          ---ERROR (ISSUE A FORMATTED ERROR MESSAGE)
        ---SPIEL (INTERRUPT HANDLER INITIALIZATION E.P. TRACBK)
        ---EANTSA (SYSTEM STATUS AREA INITIALIZATION)
          ---LODCOM (INITIALIZE IP DATA AREA ADDRESS E.P. ESASAV)
          ---EANSV (RETRIEVE CORE ADDRESS OF IP DATA AREAS)
            ---EADADD (READ NETWORK & SYSTEM CHARACTERISTICS)
            ---EACKR (INITIALIZE ADDRESSES OF STAUS AREAS
              FOR CHECKPOINT/RESTART)
          ---EAUPTX (RETRIEVE PARM FIELD ADDRESS E.P. EANTIX)
            ---EAINDX (WRITE INITIAL INDEX FILE ENTRIES)
              ---DAYTIM (ACQUIRE DATA & TIME)
                ---TIMES (E.P. DTIMEL)
            ---EANDTA (LOAD SYSTEM STATUS AREA DATA E.P. EADADD)
            ---EANFEL (INITIALIZE FUTURE EVENTS LIST)
            ---EANXTN (INITIALIZE TRANSACTION DATA)
            ---EANMDL (MODEL INITIALIZATION CONTROL)
              ---EANDR (PERFORM DEMAND RESPONSIVE INITIALIZATION)
                ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
                ---EANSVD (PERFORM SCHEDULED SERVICE INITIALIZATION)
                  ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
                  ---EANMRG (PERFORM TIMEOUT/GROUP DEMAND RESPONSIVE
                    INITIALIZATION)
                  ---EANRPT (DISPLAY INITIAL CONDITIONS REPORT)
                    ---DAYTIM (ACQUIRE DATE AND TIME)
                    ---ERROR (ISSUE A FORMATTED ERROR MESSAGE)
                ---EASNIT (INITIALIZE SIMULATION STATISTICS)
                  ---EZHDR (WRITE A SAMPLING FILE HEADER)
                  ---EZZERO (ZERO PERIODIC STATISTICS & INITIALIZE TIME
                    INTEGRALS)
                ---EAFLAG (INITIALIZE AUXILLIARY OUTPUT INDICATORS)
                ---NDBOR (PERFORM GENERALIZED DATA INPUT PROCESSING)
                ---EAPFEL (PLACE A TRANSACTION ON THE FUTURE EVENTS LIST)
                ---ERROR (ISSUE A FORMATTED ERROR MESSAGE)
            ---EAGCTL (GUIDEWAY LINK MODEL CONTROL)

```

FIGURE 2-2. (1 of 6) MODEL PROCESSOR CONTROL FLOW

```

---EGNEXT (NEXT GUIDEWAY ENTITY DETERMINATION CONTROL)
  |
  ---EGGNXT (GUIDEWAY---NEXT ENTITY)
    |
    ---EGPATH (REAL-TIME PATH SELECTION)
      |
      ---EGPRMY (PRIMARY PATH COST COMPUTATION)
      ---EGALT (ALTERNATE PATH COST COMPUTATION)
        |
        ---EGPRMY (PRIMARY PATH COST COMP.)
  ---EGRESV (GUIDEWAY LINK EMPTY &
             RESERVED VEHICLE PROCESSING)
    |
    ---EGRESV1 (TRIP SERVICING CRITERIA CHECK AND
               DIVERSION CONTROL)
    ---EGEMTY (GUIDEWAY LINK EMPTY VEHICLE PROCESSING)
      |
      ---EGCNXT (NEXT STATION - CIRCUITOUS EMPTY)
      ---EGVALS (VEHICLE ETA AND STATION ARRIVAL
                LIST RECORDING)
        |
        ---EGVALS1 (VEHICLE INSERTION IN LIST)
        ---EGGNXT (GUIDEWAY---NEXT ENTITY)
      ---EGTCTL (TRIP COMPATIBILITY CHECK CONTROL)
        |
        ---EGTCHK (TRIP COMPATIBILITY CHECK)
          |
          ---EGTCHK1 (STATION TRIP DESTINATION
                     COMPARISON WITH BYPASSED STNS.)
          ---EGTCHK2 (ULTIMATE DESTINATION COM-
                     PARISON WITH BYPASSED STATIONS
                     OF STATION TRIP ROUTE)
        ---EGDTRN (GUIDEWAY LINK VEHICLE DETRAINMENT)
          |
          ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
    ---EGTRNC (TRAIN COMPATABILITY CHECK AT NETWORK DIVERGE)
      |
      ---EGGNXT (GUIDEWAY NEXT ENTITY)
      ---EGDTRN (GUIDEWAY LINK VEHICLE DETRAINMENT)
    ---EGDSTP (DEMAND STOP SERVICE PROCESSING)
      |
      ---EGDSTP1 (DEMAND STOP, NEXT STATION DETER.)
      ---EGGNXT (GUIDEWAY---NEXT ENTITY)
    ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
  ---EGTEST (GUIDEWAY LINK ENTRY TESTING)
    |
    ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
    ---ERROR (ISSUE A FORMATTED ERROR MESSAGE)
  ---ESLMDL (STATION MODEL CONTROL)
  ---ESTEST (STATION ENTRY TESTING)
  ---EGLMDL (GUIDEWAY LINK MODEL CONTROL)
    |
    ---EGHTRN (TRAIN LINK HEADWAY TRAVERSAL PROCESSING)
      |
      ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
    ---EGLNTR (GUIDEWAY LINK ENTRY PROCESSING)
      |
      ---EGLNTR1 (GUIDEWAY LINK ENTRY, STATISTICAL
                 PROCESSING)
      ---EGQNTR (GUIDEWAY LINK ADVANCE POSITIONING
                 PROCESSING)
        |
        ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
        ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
    ---EGETRN (GUIDEWAY LINK DYNAMIC VEHICLE ENTRAINMENT)

```

FIGURE 2-2. (2 of 6) MODEL PROCESSOR CONTROL FLOW

```

---EGFNTR (FIXED HEADWAY TRAVEL SEGMENT ENTRY)
    ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
---EGVNTR (VARIABLE HEADWAY TRAVEL SEGMENT ENTRY)
    ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
---EGUNTR (USER HEADWAY---TRAVEL SEGMENT ENTRY)
    ---ERROR (ISSUE A FORMATTED ERROR MESSAGE)
---EGFTVL (FIXED HEADWAY TRAVEL SEGMENT TRAVERSAL)
    ---EGLWTQ (ENTITY QUEUING PROCESSING)
    ---EGSCHD (SCHEDULE VEHICLE FOLLOWER)
        ---EAFTRN (PERFORM PUSH COUPLE ENTRAINMENT)
        ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
        ---ERROR (ISSUE A FORMATTED ERROR MESSAGE)
---EGQMRG (MERGE SCHEDULING -QUASI-SYNCHRONOUS
    CONTROL)
        ---EGQMRG1 (PERFORM VEHICLE RETARDATION)
        ---EGQMRG2 (PERFORM VEHICLE ADVANCE)
        ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
---EGFTVL1 (FIXED HEADWAY TRAVERSAL, MERGE PRIOR-
    ITY PROCESSING)
    ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
---EGVTVL (VARIABLE HEADWAY TRAVEL SEGMENT TRAVERSAL)
    ---EGLWTQ (ENTITY QUEUING PROCESSING)
    ---EGSCHD (SCHEDULE VEHICLE FOLLOWER)
    ---EGVTVL1 (VARIABLE HEADWAY TRAVERSAL, MERGE
    PRIORITY PROCESSING)
    ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
---EGUTVL (USER DEFINED---TRAVEL SEGMENT TRAVERSAL)
    ---ERROR (ISSUE A FORMATTED ERROR MESSAGE)
---EGLEAV (ENTITY EXIT PROCESSING)
    ---EGVLOG (WRITE VEHICLE LOG FILE ENTRY)
    ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
---EGLWTQ (ENTITY QUEUING PROCESSING)
---EGASTN (ALTERNATE STATION ASSIGNMENT)
---EASCTL (DESM STATION MODEL CONTROL)
    ---ESLMDL (DESM STATION MODEL)
        ---ESMDLA (STATION MODEL AFTER EVENT PROCESSING)
            ---ESNSTN (DETERMINE NEXT STATION FOR VEHICLE)
                ---ESEVA (EMPTY VEHICLE REDISTRIBUTION)
                    ---EUEVA (USER EMPTY VEHICLE ALGOR-
                        ITM)
                        ---ERROR (ISSUE A FORMATTED
                            ERROR MESSAGE)
                            ---ERROR (ISSUE A FORMATTED ERROR MSG)
                            ---ERROR (ISSUE A FORMATTED ERROR MESSAGE)
            ---ESVALS (INSERT A VEHICLE IN A DOWNSTREAM
                ARRIVAL LIST)
            ---ESLEAV (PROCESS VEHICLE EXIT FROM STATION LINK)
            ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
        ---ESMDLB (STATION MODEL BEFORE EVENT PROCESSING)

```

FIGURE 2-2. (3 of 6) MODEL PROCESSOR CONTROL FLOW

```

---ESTABQ (ATTEMPT TRIP/VEHICLE RESERVATION)
---ESBDL (CREATE A VEHICLE BOARDING LIST FROM
        WAITING TRIPS IN STATION)
        |
        |---ESTCHK (TRIP COMPATIBILITY CHECK)
        |---ERROR (ISSUE A FORMATTED ERROR MESSAGE)
---ESDBL (CREATE A VEHICLE DEBOARD LIST FROM TRIPS
        ONBOARD VEHICLE)
        |
        |---ESTLOG (RECORD TRIP COMPLETION IN LOG)
---ESPATH (DETERMINE PATH(S) FOR A VEHICLE)
---ESNTRN (PERFORM STATIC ENTRAINMENT)
---ESSDLY (DETERMINE SCHEDULE DELAY)
---ESLDLY (DETERMINE LAUNCH TIME FOR VEH)
        |
        |---ESMDLY (DETERMINE MERGE DELAY)
---EANTRN (ENTRAIN AND LAUNCH SCHEDULED SERVICE
        ACTIVE FLEET SIZE MANAGEMENT VEHICLES)
        |
        |---EAPFEL (PLACE A TRANSACTION ON THE FEL)
---EASTOR (ASSIGN STORED VEHICLES DURING SCHEDULED
        SERVICE ACTIVE FLEET SIZE MANAGEMENT)
        |
        |---EANTRN (ENTRAIN AND LAUNCH SCHEDULED SER-
        VICE ACTIVE FLEET SIZE MANAGEMENT
        VEHICLES)
        |---ESLEAV (PROCESS VEHICLE EXIT FROM
        STATION LINK)
        |---EAPFEL (PLACE A TRANSACTION ON THE FEL)
---ESAREQ (ASSIGN READY REQUEST TO VEHICLE)
---ESLEAV (PROCESS VEHICLE EXIT FROM STATION LINK)
---EAPFEL (PLACE A TRANSACTION ON THE FEL)
---ESMDLN (DETERMINE NEXT STATION EVENT FOR A VEHICLE &
        ESTABLISH AVAILABILITY)
        |
        |---EG2MRG (PERFORM QUASI-SYNCHRONOUS CONTROL)
        |---EAPFEL (PLACE A TRANSACTION ON THE FEL)
---ESLEAV (PROCESS A VEHICLE LEAVING A STATION LINK)
        |
        |---EAPFEL (PLACE A TRANSACTION ON THE FEL)
---ESLWTQ (INSERT A VEHICLE IN A STATION LINK QUEUE)
        |
        |---EAFTRN (PERFORM PUSH COUPLE ENTRAINMENT)
        |---EAPFEL (PLACE A TRANSACTION ON THE FEL)
        |---ERROR (ISSUE A FORMATTED ERROR MESSAGE)
---ESNEXT (DETERMINE NEXT ENTITY FOR VEHICLE IN A STATION)
        |
        |---ESDIVF (DIVERGE FUNCTION PROCEESING)
        |
        |---ESDIVO (ORDER STATION LINKS BY OCCUPANCY)
        |---EUDIVF (USER-SPECIFIED DIVERGE FUNCTION)
        |
        |---ERROR (ISSUE A FORMATTED ERROR MESSAGE)
        |---EAPFEL (PLACE A TRANSACTION ON THE FEL)
---ESTEST (STATION ENTRY TESTING)
        |
        |---ESVREQ (PROCESS READY VEHICLE REQUEST)
        |---EAPFEL (PLACE A TRANSACTION ON THE FEL)
        |---ESEVB (EMPTY VEHICLE BUMPING)
        |
        |---EUEVB (USER EMPTY VEHICLE BUMPING)
        |
        |---ERROR (ISSUE A FORMATTED ERROR MESSAGE)
        |---ERROR (ISSUE A FORMATTED ERROR MESSAGE)
---EGLMDL (GUIDEWAY LINK MODEL CONTROL)
---EGTEST (GUIDEWAY LINK ENTRY TESTING)

```

FIGURE 2-2. (4 of 6) MODEL PROCESSOR CONTROL FLOW


```

---EATORG (PROCESS A TRIP ARRIVAL)
  ---ESTABQ (ATTEMPT TRIP/VEHICLE RESERVATION)
    ---ESTCHK (D.R. MULTIPARTY TRIP COMPATIBILITY CHECK)
    ---ESVRES (RESERVE SPACE FOR TRIP ON SELECTED VEH)
      ---ESVREQ (PROCESS READY VEHICLE REQUEST)
        ---ESAREQ (ASSIGN READY REQUEST TO VEHICLE)
          ---EAPFEL (PLACE A TRANSACTION ON FEL)
          ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
        ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
      ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
    ---ERROR (ISSUE A FORMATTED ERROR MESSAGE)

---EASAMP (PERFORM PERIODIC SAMPLING)
  ---EZHDR (WRITE A SAMPLING HEADER RECORD)
  ---EZINT (COMPUTE END POINT TIME INTEGRALS)
  ---EZZERO (ZERO OR RESET PERIODIC STATISTICS)
  ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
  ---EASRPT (DISPLAY PERIODIC SAMPLING REPORT)
    ---DAYTIM (ACQUIRE DATE & TIME)

---EAASYN (PROCESS AN ASYNCHRONOUS DATA EVENT)
  ---NDBOR
    ---EMGDIP4 (DEFINE INPUT COMMON AREAS ORGANIZATION)
      ---XGDIPX4 (READ BYTE SIZE GDIP DATA)
        ---XPSEUDO (PSEUDO I/O INTERCEPT ROUTINE)
      ---XGDIPF4 (READ FULL WORD GDIP DATA)
        ---XPSEUDO (PSEUDO I/O INTERCEPT ROUTINE)
      ---XGDIPH4 (READ HALF WORD GDIP DATA)
        ---XPSEUDO (PSEUDO I/O INTERCEPT ROUTINE)
      ---ERROR (ISSUE A FORMATTED ERROR MESSAGE)
    ---EGFAIL (PROCESS A GUIDEWAY FAILURE)
      ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
      ---ERROR (ISSUE A FORMATTED ERROR MESSAGE)
    ---ESFAIL (PROCESS A STATION FAILURE)
      ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
    ---EAFLAG (PROCESS A DEBUG FLAG REQUEST)
    ---EACKPT (PERFORM DEMAND CHECKPOINT E.P. EACKR)
      ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
      ---ERROR (ISSUE A FORMATTED ERROR MESSAGE)
    ---EAAFISM (ACTIVE FLEET SIZE MANAGEMENT)
      ---EAIVFH (INITIALIZE NEW VEHICLES)
      ---EANTRN (ENTRAIN AND LAUNCH SCHEDULED SERVICE ACTIVE
        FLEET SIZE MANAGEMENT VEHICLES)
      ---EASTOR (ASSIGN STORE VEHICLES DURING SCHEDULED SERVICE
        ACTIVE FLEET SIZE MANAGEMENT)
      ---EAPFEL (PLACE A TRANSACTION ON THE FEL)
      ---ESLEAV (UPDATE STATION STATISTICS)
      ---ERROR (ISSUE A FORMATTED ERROR MESSAGE)

```

FIGURE 2-2. (5 of 6) MODEL PROCESSOR CONTROL FLOW

```

|---ERROR (ISSUE A FORMATTED ERROR MESSAGE)
---EACKPT (PERFORM PERIODIC CHECKPOINT E.P. EACKR)
---EAPRMT (PROCESS A DEQUEUE TRANSACTION)
    ---EAPLNK (DEQUEUE A VEHICLE ON A GUIDEWAY LINK)
        ---EGNEXT (NEXT ENTITY FOR A VEHICLE -GUIDEWAY)
        ---EGTEST (TEST NEXT ENTITY FOR GUIDEWAY VEHICLE)
        ---EGLEAV (LEAVE A GUIDEWAY LINK)
        ---EGLMDL (GUIDEWAY LINK MODEL)
        ---ESNEXT (DETERMINE NEXT ENTITY FOR VEHICLE IN A STN)
        ---ESTEST (STATION LINK ENTRY TESTING)
        ---ESLMDL (STATION MODEL CONTROL) - STN)
        ---ESLEAV (LEAVE A STATION LINK)
        ---EAPFEL (PLACE A TRANSACTION ON THE FUTURE EVENTS LST)
    ---EAPSTN (DEQUEUE A VEHICLE FROM A STATION QUEUE)
        ---EGTEST (TEST NEXT ENTITY FOR GUIDEWAY VEHICLE)
        ---EGLMDL (GUIDEWAY LINK MODEL)
        ---ESNEXT (DETERMINE NEXT ENTITY FOR VEHICLE - STN)
        ---ESTEST (STATION LINK ENTRY TESTING)
        ---ESLMDL (STATION MODEL CONTROL)
        ---ESLEAV (LEAVE A STATION LINK)
---EAPCMP (PERFORM PERIODIC COMPUTATIONS OR FAILURE MANAGEMENT)
    ---EAIVEH (INITIALIZE NEW VEHICLES)
        ---ERROR (ISSUE A FORMATTED ERROR MESSAGE)
        ---EANTRN (ENTRAIN AND LAUNCH SCHEDULED SERVICE ACTIVE FLEET
            SIZE MANAGEMENT VEHICLES)
        ---ESVREQ (PROCESS READY VEHICLE REQUESTS)
        ---ERROR (ISSUE A FORMATTED ERROR MESSAGE)
        ---EUPCMP (USER DEFINED PERIODIC COMPUTATION EVENT)
            ---ERROR (ISSUE A FORMATTED ERROR MESSAGE)
        ---EAPFEL (PLACE A TRANSACTION ON THE FUTURE EVENTS LIST)
            ---ERROR (ISSUE A FORMATTED ERROR MESSAGE)
---EARFEL (REMOVE NEXT TRANSACTION FROM THE FUTURE EVENTS LIST)
---EAFINS (SYSTEM TERMINATION)
    ---EAFRPT (SYSTEM TERMINATION REPORT)
        ---DAYTIM (ACQUIRE DATE AND TIME)
        ---EAWTIX (E.P. EAINDX - WRITE INDEX FILE OUTPUT ENTRIES)
---ERROR (ISSUE A FORMATTED ERROR MESSAGE)
    ---TRACBK (TERMINATE PROCESSING WITH AN EXECUTION TRACE)
        ---TRCBKP (PRINT PROGRAM TRACEBACK INFORMATION)
    ---EAFINS (SYSTEM TERMINATION)
END

```

FIGURE 2-2. (6 of 6) MODEL PROCESSOR CONTROL FLOW

DESM OUTPUT PROCESSOR:

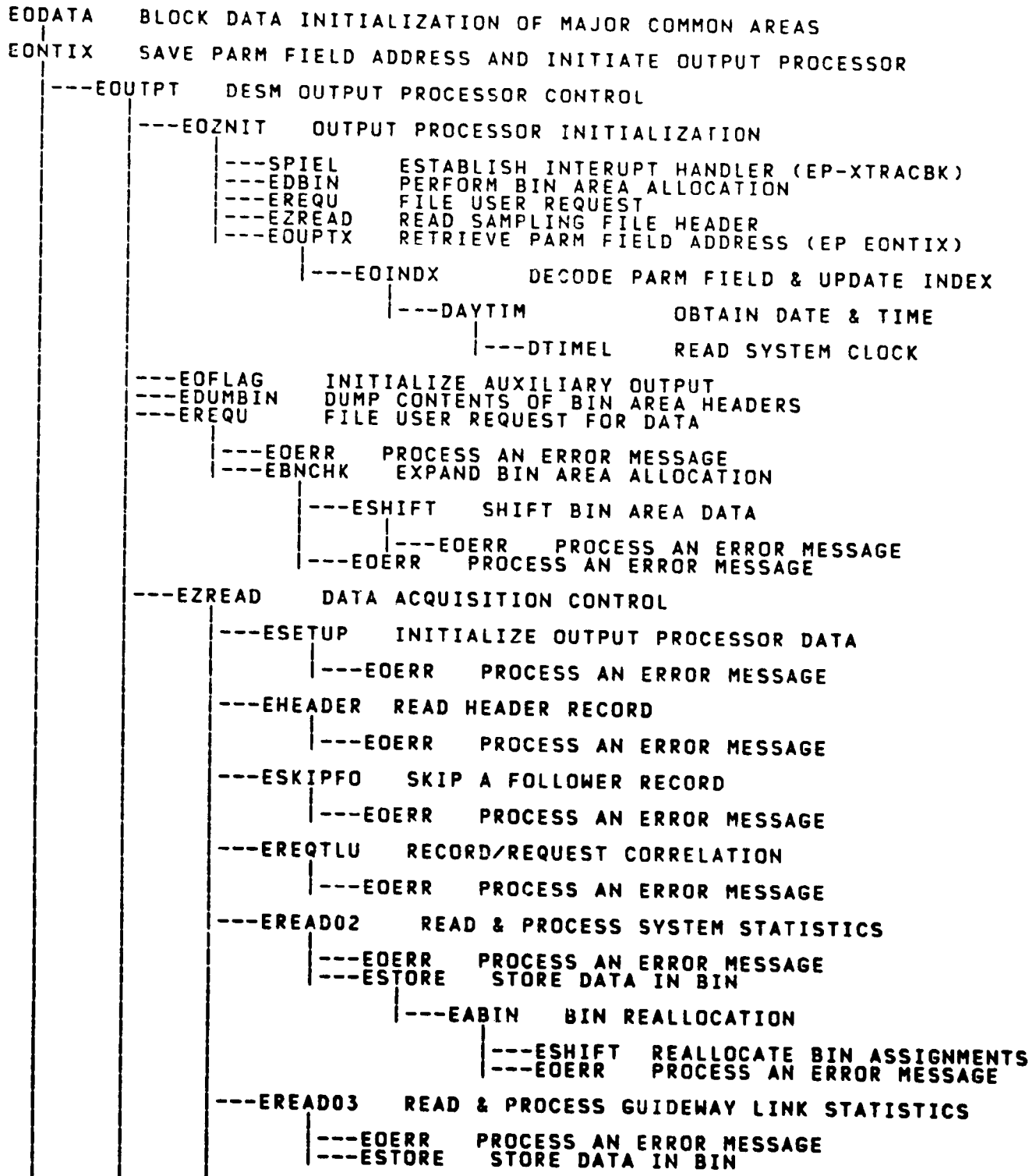


FIGURE 2-3. (1 of 3) OUTPUT PROCESSOR CONTROL FLOW

```

---EREAD04  READ & PROCESS STATION STATISTICS
      |
      |---EOERR  PROCESS AN ERROR MESSAGE
      |---ESTORE  STORE DATA IN BIN
---EREAD05  READ & PROCESS STATION LINK STATISTICS
      |
      |---EOERR  PROCESS AN ERROR MESSAGE
      |---ESTORE  STORE DATA IN BIN
---EREAD06  READ & PROCESS SCHEDULED ROUTE STATISTICS
      |
      |---EOERR  PROCESS AN ERROR MESSAGE
      |---ESTORE  STORE DATA IN BIN
---EOERR    PROCESS AN ERROR MESSAGE
---EZLIST   TIME SERIES LIST CONTROL
      |
      |---ELIST   LIST ITEMS IN A BIN OR PRODUCE SUMMARY
---EZHIST   HISTOGRAM OUTPUT CONTROL
      |
      |---EMNMX   COMPUTE MINIMUM AND MAXIMUM VALUES IN BIN
      |---EHIST   OUTPUT HISTOGRAM OF DATA
      |---EBNCHK  EXPAND BIN AREA ALLOCATION
---EZPLOT   TIME SERIES PLOT CONTROL
      |
      |---EGRAPH  OUTPUT TIME SERIES PLOT OF SAMPLED DATA
---EOPSUM   COMPUTE & WRITE PERFORMANCE SUMMARY DATA
      |
      |---EOPRPT  WRITE PERFORMANCE SUMMARY REPORT, STANDARD
      |          REPORT 1
---EOSSUM   COMPUTE SYSTEM SUMMARY STATISTICS
      |
      |---EOSRPT  WRITE SYSTEM SUMMARY STATISTICS REPORT,
      |          STANDARD REPORT 2
---ETSSPM   STATION-TO-STATION PERFORMANCE MEASURES CONTROL
      |
      |---ETCOMP  COMPUTE REQUESTED STATION-TO-STATION MEASURE
      |
      |---ETCAPT  CAPTURE STATION-TO-STATION RAW DATA
      |
      |---ETACUM  ACCUMULATE STATION-TO-STATION
      |          RAW STATISTICS
      |---ETMEAS  FUNCTION TO COMPUTE RAW DATA
      |          FOR THE MEASURE
      |
      |---DERROR  ERROR TERMINATION ROUTINE
      |
      |---ETTOTL  COMPUTE RAW DATA TOTALS
      |---ETSTAT  COMPUTE STATISTICS
      |---ETRPTS  WRITE STATION-TO-STATION REPORT
      |
      |---AADATE  PROVIDE DATE IN CHARACTER FORMAT
      |
      |---DAYTIM  OBTAIN DATE AND TIME
      |
      |---ETSTID  FUNCTION TO CONVERT STATION ID TO
      |          EBCDIC
      |
      |---ETNMBR  FUNCTION TO CONVERT A
      |          NUMBER TO EBCDIC
      |
      |

```

FIGURE 2-3. (2 of 3) OUTPUT PROCESSOR CONTROL FLOW

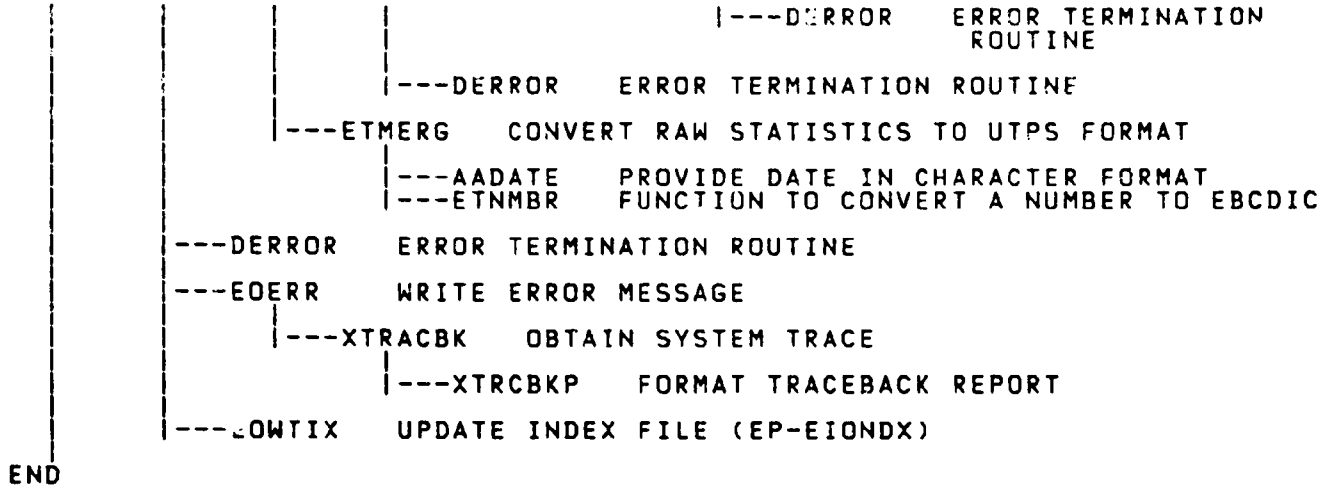


FIGURE 2-3. (3 of 3) OUTPUT PROCESSOR CONTROL FLOW

```

OVERLAY      REGION 2
INSERT      IPNET
OVERLAY      REGION 2A
INSERT      NETWORK DEFINITION COMMON 1
.           .           .           .           .
.           .           .           .           M
OVERLAY      REGION 3
INSERT      IPSYS
OVERLAY      REGION 3A
INSERT      SYSTEM CHARACTERISTICS COMMON 1
.           .           .           .           .
.           .           .           .           M
OVERLAY      REGION 4
INSERT      ENDCOM

```

The symbolic names assigned to each overlay region are given as BEGCOM, IPNET, and IPSYS, respectively. The symbolic name ENDCOM is used in conjunction with BEGCOM for defining the entire overlay region.

Each symbolic name corresponds to a dummy control section for which address constants are established within the program code of each component. These address constants are retrieved during IP and MP initialization to map FORTRAN arrays to each area of the overlay region, such that data transfer can be accomplished via a sequential binary read or write statement.

2-1 INPUT PROCESSOR

The Input Processor is invoked via a cataloged procedure which provides the information necessary to access user-specified input and output data sets within the AGT data base. The primary data interfaces to the IP are shown in Figure 2-4.

Upon entry, preliminary initialization is performed, which includes: determining the location and size of the output data areas that are the interface to the model processor, saving the list of data set member names specified by the user, and defining parameter values and program indicators to establish default conditions and a baseline for verifying and validating specified inputs.

The selected data set member from the system characteristics Input and Description file is then read through a standardized input processing procedure data into the defined common areas. User defined data base override parameter names and values, if present, are then read from a data set member or from the SYSIN stream to alter particular variables in the defined common areas for the current simulation exercise. This override function does not alter the Input and Description file contained in the AGT data base. The user-defined run description information for the index data set is also read and written to the Run Index file.

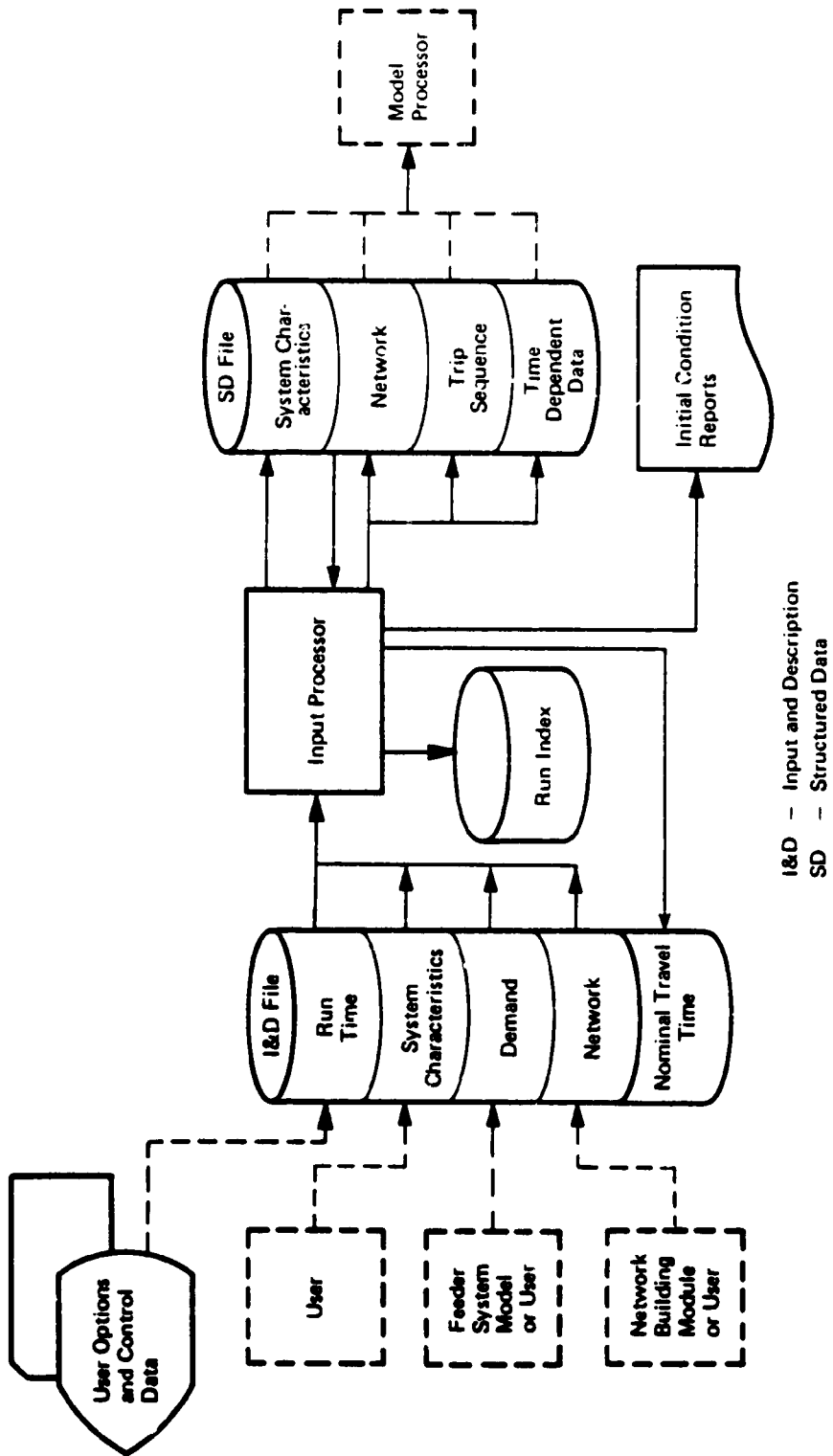


FIGURE 2-4. INPUT PROCESSOR DATA INTERFACE

Simulator control and option selections are read which set the various program indicators and control variables that direct the DESM for the type of transit system, demand profile, and service policy to be simulated. User run-time data, if defined in the input, is read and used as a basis for recomputation of specific aspects of the simulation, such as, the network configuration (for link failure/repair), and service planning (for active fleet size, management) at times specified in the users input data.

2.1.1 Architecture

The IP is basically a sequential processor, having a fixed order of tasks that can be performed. However, the user, through input of control and option selection parameters, defines which of the tasks and processing options are to be executed. The functional flow through the IP and the relationship to the data base is shown in Figure 2-5.

The program begins by determining and saving the address and size of the data areas that will be written as Structured Data files. Next, the data set member names specified by the user are extracted from the parameter list established by the cataloged procedure. Then, a predefined set of parameters is initialized to establish model default conditions and to define a baseline for checking user-entered data. Next, initial condition data is read from a specified Input and Description file dataset which will contain the system characteristics of a generic transit system. Reading of a data file containing database overrides, run control parameters, and option selections is then performed. This permits the user to modify portions of the transit system characteristics for a series of related simulation exercises (e.g., sensitivity analysis).

The IP contains three basic groups of process functions: guideway network configuration processing, trip demand generation, and transit service planning. The user-entered run control parameters define which group(s) are executed in a given run. For example, in a given series of runs, it may be desired to vary the level of service, holding the guideway network and trip demand definitions constant. In this case, the IP executes only the service planning group which uses the characteristics of the network and demand pattern. However, generation of network minimum paths and a trip file is bypassed.

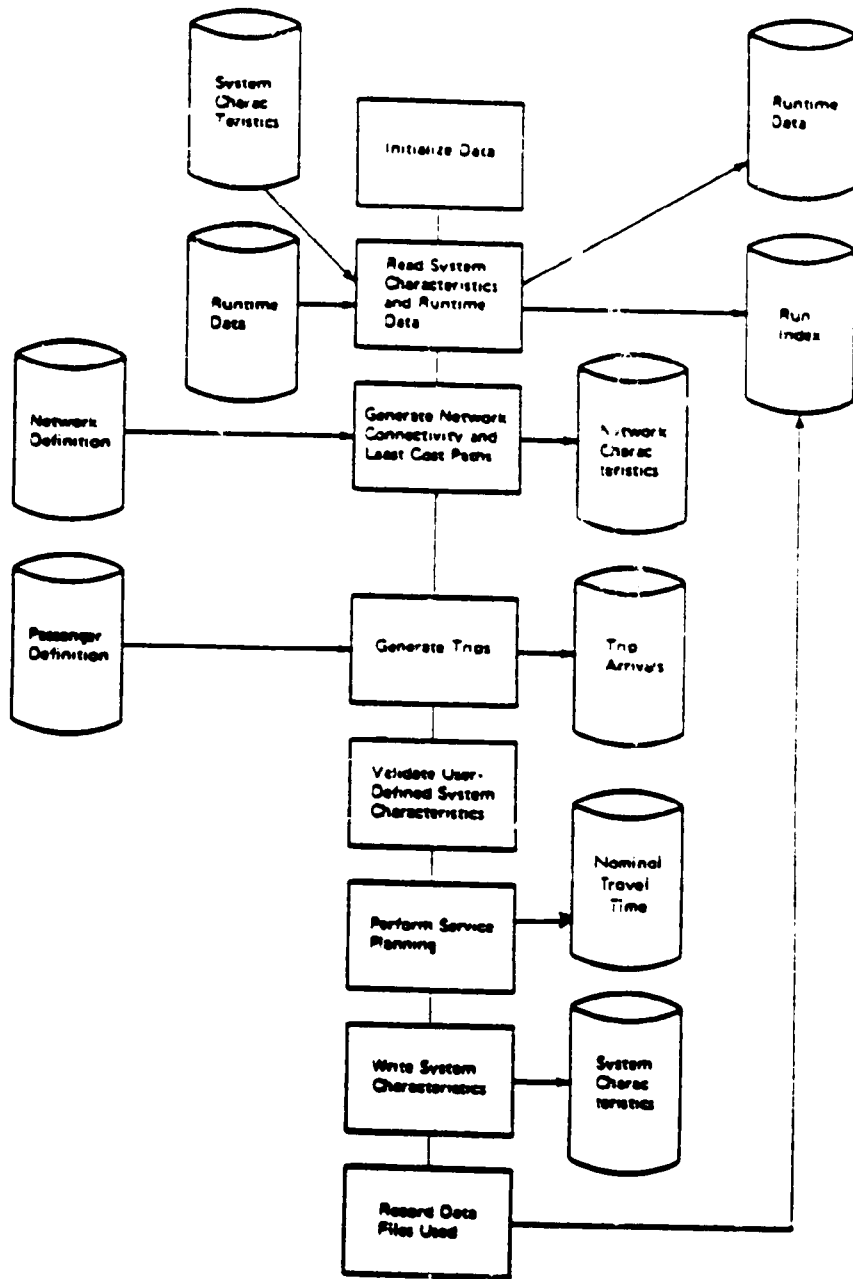


FIGURE 2-5. INPUT PROCESSOR FUNCTIONAL FLOW

Each group of process functions produces an initial conditions report, summarizing the results of the computations performed. User input data will be checked for consistency and reasonableness and appropriate error messages will be printed.

The IP consists a major processing loop based on the applicable time associated with the input data. All three groups of functions are included in the major loop. This permits the generation of a trip file from a series of demand matrices representing a time of day demand profile. It also permits the level of service to be established using recomputation of minimum cost paths when a guideway link failure is planned to occur during an exercise.

All initial conditions, whether read from the database or generated by the IP are written to Structured Data file datasets for access by the Model Processor which is run as a separate job step.

A summary of the major processing performed by the IP is presented in the following sections.

2.1.1.1 Network Processing

If processing of a new guideway network configuration is required, network definition data is read from the user specified input and description file as created by the user. After checking the input data to verify that the network is complete and has properly defined station sites, merges, and diverges, the least cost path between each pair of stations is determined. The network connectivity tables required by the Model Processor are then built using the least cost path definitions.

If a previously processed network is to be used for the current experiment the structured network data is read from the user-specified file for use as needed to support alternate path processing, guideway failure processing, and transit service planning. If one or more alternate paths have been specified in the system characteristics or run time data sets, the program establishes connectivity between the alternate paths and the least cost paths at the common diverge points. The alternate path data is translated from user format to the Model Processor format.

If either a new network has been processed or new alternate paths have been defined, a new network structured data file is written for use by the Model Processor and subsequent Input Processor runs.

If guideway link or vehicle failures are required, the program penalizes the use of the failed link and recalculates the least cost path definitions, if selected by the user. If tow vehicle recovery is selected, links which must be closed to isolate the tow path are also penalized.

The failure definition data is translated to Model Processor parameters and written with the new least cost path table to the structured run time data file. If the failure is within a station, the failure definition data is translated to Model Processor parameters and written to the structured run time data file. Least cost paths are not recomputed. Recovery from failure is handled in a similar manner, except that when the last failure is removed, recomputation of least cost paths is not performed.

2.1.1.2 Trip Demand Generation

Demand generation is performed if a new sequence of trip arrivals is to be generated or if the user requests the Input Processor to estimate the number of vehicles required for a specified level of demand and service policy. The function has the capability to process multiple demand input files, to accept overrides to the demand time interval, and to apply a scale factor to the demand to handle time of day variation in level of demand.

The generation process begins with the reading of the user-supplied demand definition data from the input and description file, which is expressed in terms of total passengers during a specified time interval for each origin/destination pair. This is converted to trips per hour by origin/destination using user-specified trip group size distributions (a trip consists of one or more passengers traveling together by their own choice).

The user can enter up to three group size distributions and assign each origin/destination pair to one of the three distributions. The data is entered in frequency distribution format and converted to cumulative distribution format for later use in generating individual trips. If a new sequence of trips is not required, no further processing is performed.

If the generation of a new sequence of trip arrivals has been requested, probability distributions of origin and destination selection are formed from the trips per hour by origin/destination data and the aggregate trip interarrival time is calculated. A Poisson process is used to select individual trip arrival times and pseudo-random numbers are compared to the probability distributions to assign origin, destination, and group size to the trip. Alternatively, the user may choose to generate trips in a deterministic manner using a uniform distribution. This process guarantees the exact number of trips specified in the demand matrix and a constant interarrival time for all trips for a given O-D pair within each demand interval. The trip parameters are then written to the structured demand file. When the trip arrival time exceeds the time interval specified in the demand input data, trip generation terminates and a report summarizing the generated trips is written.

The entire trip generation process is repeated for each set of demand input data specified by the user.

2.1.1.3 Service Planning

This function includes station configuration, system characteristics checking, and transit service policy processing. If the user has defined the station characteristics in the user-oriented notation, this input data is processed to build the station characteristic and connectivity tables required by the Model Processor. The user may elect to input the station characteristic and connectivity tables directly, in which case no processing is required.

The system characteristics input data is checked for reasonableness and consistency to detect parameter value errors and invalid combinations of functions (e.g., synchronous control requires deterministic dispatch). If an error is encountered, a message is written and if further processing is unrealistic due to the nature of the error, the run terminates.

The service policy can be either demand responsive (single or multiparty or multiparty single stop or timeout/group) or scheduled (with user-defined routes or routes determined by the Input Processor). The user may define the vehicle fleet or request the Input Processor to estimate the number of vehicles required based on the network characteristics and the level of trip demand. For scheduled service the Input Processor calculates the vehicle departure schedules for each station on each route.

After the service planning functions have completed, the structured system characteristics file is written and a report is written summarizing the system characteristics and transit service policy data.

2.2 MODEL PROCESSOR

The Model Processor performs the discrete event modeling of the desired transit system as defined by the user through the IP. The data provided by the IP through various formatted files, specifying the configuration and characteristics for the network of guideways and stations, establishes the two basic modeling entities in the simulation processor. In addition, initial conditions, operational policies, and options which are to direct the flow of transactions, defined as vehicles, trips, and asynchronous stimuli in the simulation system comprise the initialization data set for the Model Processor. The primary data interfaces to the MP are shown in Figure 2-6.

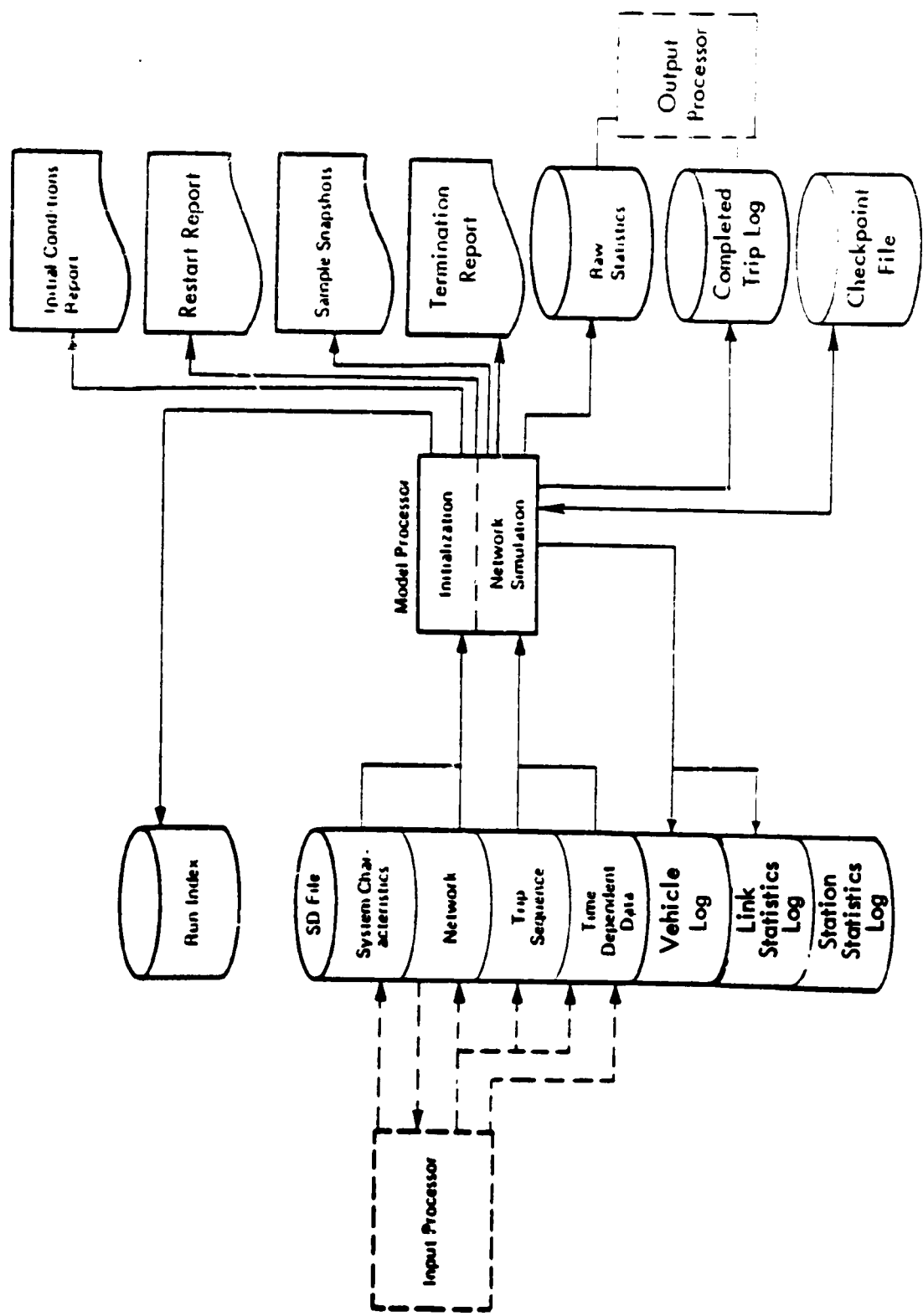


FIGURE 2-6. MODEL PROCESSOR DATA INTERFACE

The Model Processor contains the discrete event simulation architecture which provides the time dependent processing of all functions associated with trip management, station, vehicle, and guideway operations. The interaction of these functions over time can cause queues of patrons in stations and propagation of vehicle congestion on the guideway and in stations. Asynchronous command processing provides for time-dependent inputs such as trip requests, fleet size changes, and introduction of failures and other external stimuli.

The Model Processor, in performing a given simulation experiment, collects, summarizes, and formats statistical data at periodic intervals. This data, which is related to completed events, current operational status and queues, is recorded in a data file for subsequent report generation.

Execution of the MP is initiated by invoking a cataloged job control procedure contained in the procedure library. Upon entry, the MP performs initialization of the simulation experiment. This initialization is based upon reading of structured data files created as the result of input processing and involves the following processing:

1. Establishing system status area addresses. The addresses of all data related to transaction management, links, vehicles, stations, trips, and sampling are established and stored for use in order facilitate the unloading and loading of these areas during system checkpoints or restart.
2. System Restart. This process locates the data (recorded system status) for the desired restart interval from the checkpoint file and initializes the simulator such that the simulation experiment may be resumed from the previously recorded checkpoint.
3. Architecture Initialization. The structured data created by the Input Processor is read to establish all common area definitions containing the simulation initial conditions. The event timing mechanism is also established. Once this is accomplished, the scheduling of system transactions to begin trip processing, sampling, asynchronous data reading, and periodic computations is performed.
4. Modeling subsystem initialization. This process is performed to establish initial conditions for guideway links, stations, sampling statistics. The initial scheduling of vehicles in the network which is also accomplished is based upon the service policy in effect for the simulation run.

Once this is completed, the basic control loop for accomplishing the recognition, scheduling, and processing of transaction events is started. This control process provides for obtaining the next transaction to be processed and invoking the processing procedure required for the transaction as shown in Figure 2-7.

The processing components invoked by the architecture perform the processing tasks as indicated by the active transaction. This processing may cause the reading of asynchronous data input, application of algorithmic computations for updating modeling status, data summarization and recording, or processing within the modeling subsystems (guideway links or stations) of the simulation system. As the result of processing, the transaction which invoked the processing may be rescheduled to occur depending upon the processing task performed. Service request transactions for invoking periodic computations, sampling, system check-pointing, and algorithm execution are always rescheduled to occur at a fixed interval in the simulation. Service request transactions which are used to cause recognition of trip arrivals or model data updates are rescheduled to occur at the time indicated by the next asynchronous trip or data record to be processed. Transactions which are used for restarting queued transactions within the modeling subsystem are not rescheduled, but reclaimed and returned to the available pool of transactions. Reuse of the transaction depends upon operational conditions within the modeling subsystems. Vehicle transactions as described previously are rescheduled or queued depending upon whether their next event can be performed within the modeling subsystem. After transaction processing is completed control is returned to the architecture for execution of the system control mechanism.

The end of simulation occurs in response to recognition of a simulation termination event scheduled as the result of an asynchronous data input request. The occurrence of this event results in performing simulation termination activities and ending the simulation experiment.

2.2.1 Architecture

The MP architecture is designed to provide a separation between system and model dependent functions. Those functions which are system dependent represent the basic control mechanism of the simulation and serve as the fixed structural elements of the system. The interface between the simulation control mechanism and model dependent function is provided via architectural components which perform system level processing functions. Transactions flows are the only linkages that exist between the system architecture and the modeling subsystem. All scheduling and manipulation of transactions is handled by the system architecture through requests made by the modeling subsystem via

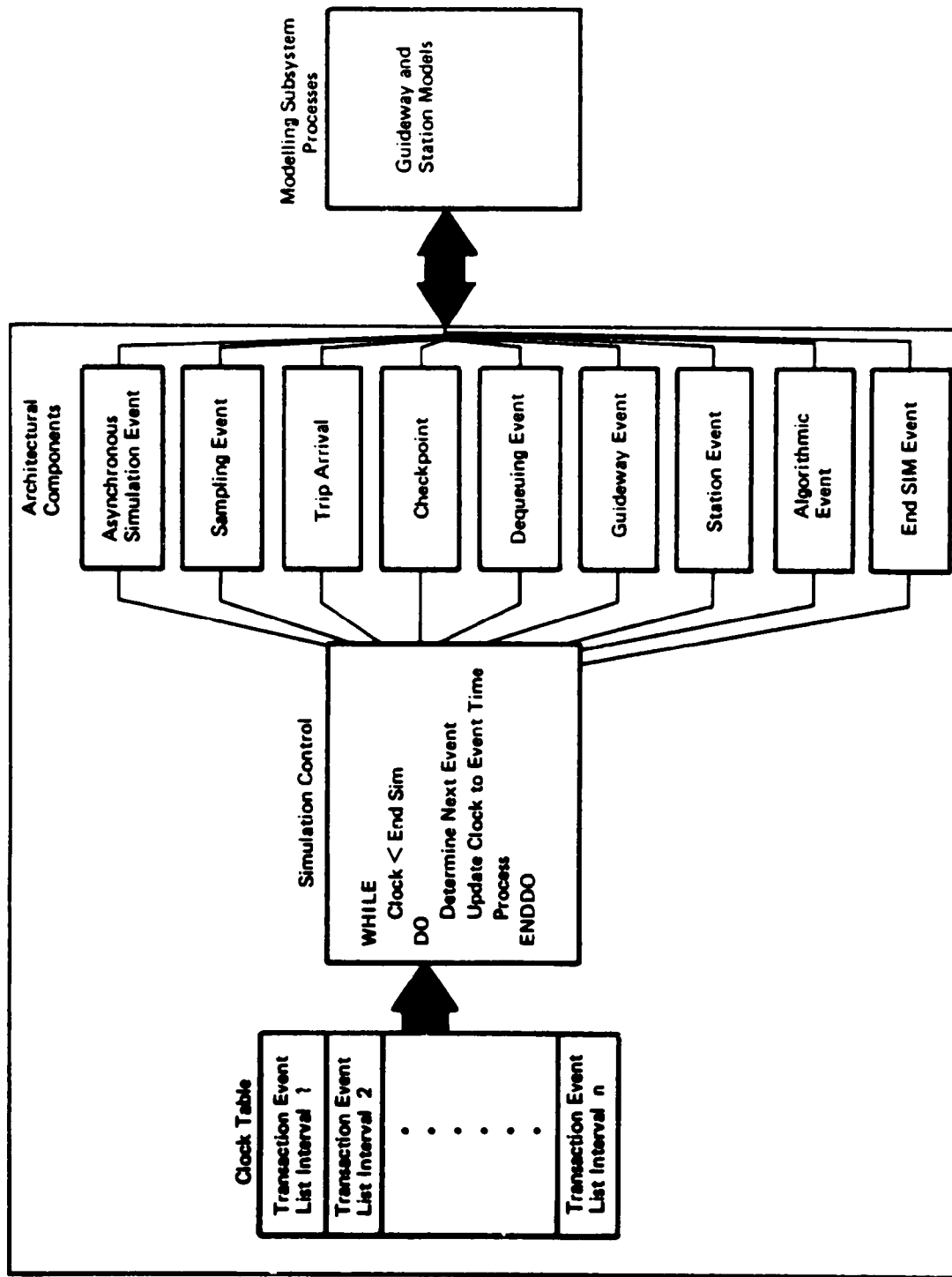


FIGURE 2-7. MODEL PROCESSOR FUNCTIONAL FLOW

standard system macros. The transaction parameters and data are controlled and manipulated by the modeling subsystem. The common transaction attributes recognized and communicated between the architecture and modeling subsystem are the transaction ID, next event function, or branch ID, and the delta time increment for occurrence of the next transaction event. Thus, processing flow within the simulator is maintained with three fixed pieces of information which represent a standard control-modeling interface. This control relationship is shown in Figure 2-8.

The definition of entities within the MP is oriented toward increasing execution efficiency by limiting the amount of event scheduling which must be performed.

Guideway links, stations, vehicles, and trips are designated as simulator elements. Elements are further defined by type as transactions or system entities. Defined as transactions, elements are subject to event scheduling each time processing is required. As system entities, elements are given attribute status and can only be assigned to other simulator elements and therefore do not require any event scheduling. The conceptual view allows greater efficiency in simulator execution since usage demands on the event control mechanism are reduced.

Transactions within the MP are defined as either vehicles, trips, or system service requests. System service requests are used in scheduling events in the future that are not directly related to model processing events. This includes such functions as data input reading trip arrival recognition, sampling and periodic computations. Vehicle transactions are used in the architectural sense to represent requests for simulator control or model processing services. These requests may take the form of a vehicle completing a specific event such as link travel, station arrival, passenger embarkation, etc. Trip transactions represent demand requests for service processed within the modeling subsystem. Regardless of transaction type, control processing and flow through the simulation system is handled in the same manner by the control architecture. The distinctions made between the types of transactions are totally model dependent.

Simulator elements such as links and stations are assignable to system transactions. Any processing performed while an entity is assigned to a transaction is totally dependent upon the organization of the link and station models. These models can contain as many internal processing paths and event points as desired, providing transaction flow back to the control program is handled according to the requirements specified above.

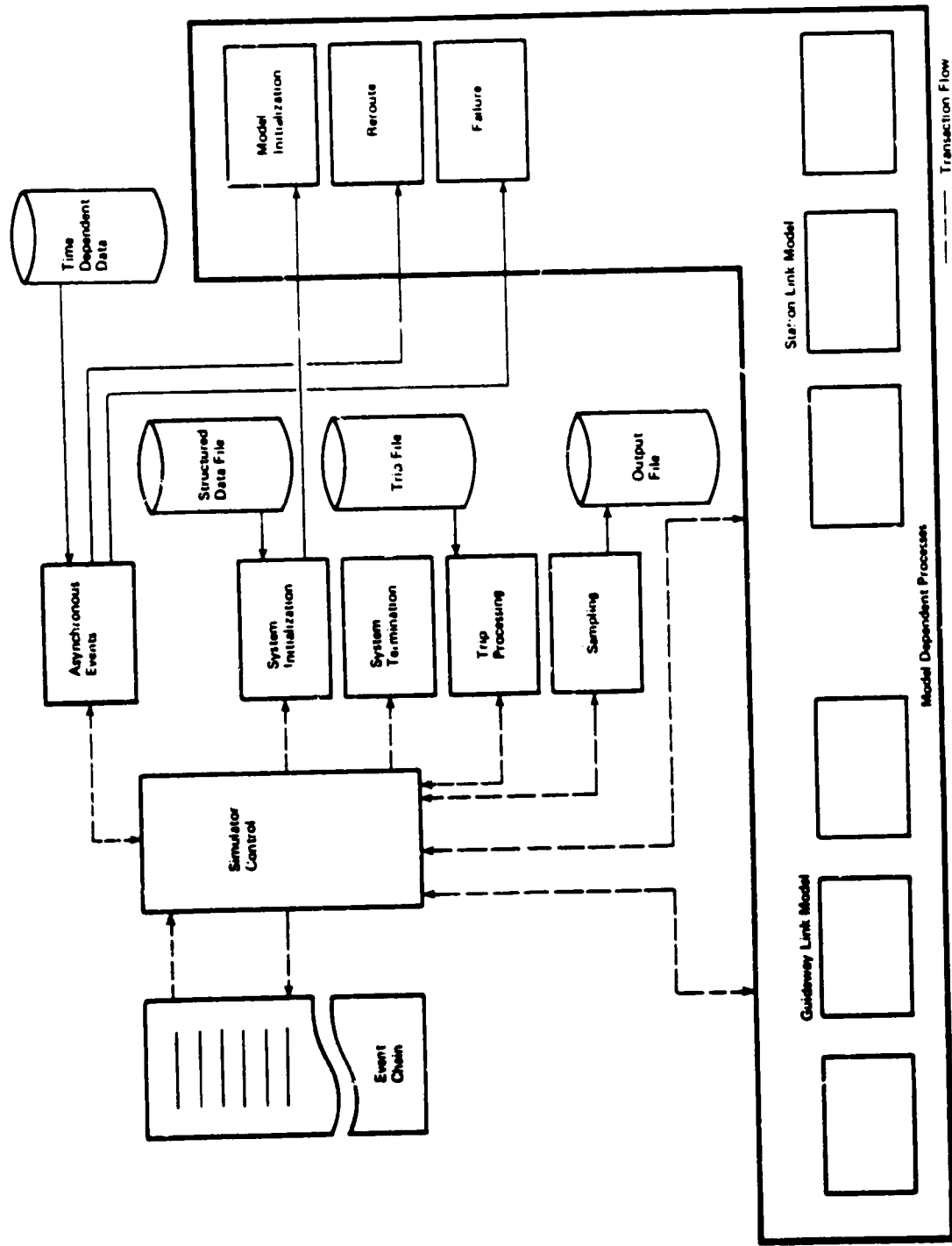


FIGURE 2-8. ARCHITECTURE/MODELING CONTROL RELATIONSHIP

Trips (passenger demand requests) can assume both entity and transaction status within the simulation environment. Typically, trips are regarded as entities which are assignable to either a network station or a vehicle transaction. In this case, trips are considered as attributes of the simulation element to which they are assigned. As trips are served during the course of simulation, their assigned status is changed from one entity to another (station to vehicle). By considering trips in this manner, demand loading has little direct impact on simulator run time since trips do not require event scheduling, thereby reducing simulator overhead required for event manipulation. In limited circumstances, however, trips must be considered as transactions which require individual event processing. These circumstances arise when independent actions must be taken by trips, such as movement (transfer time) within system elements. However, once the special event for which the trip was scheduled is completed, the trip again assumes entity status as described above.

Since vehicle and trip transactions are associated with different station and link entities as a simulation progresses, the system architecture must maintain records to account for all transactions at all times. Accordingly, transactions must always be a member of one of three possible lists:

- o An Available List (AL), or
- o The Future Event List (FEL), or
- o A Queue List (QL) from which transaction restart is required.

At the start of simulation, all of the transactions are allocated to an Available List. As vehicle transactions are initialized in the simulation, they are removed from the AL and remain created throughout the simulation experiment. As trips arrive, trip transactions are removed from the Available List and are initialized to be located at their origin stations. As a trip leaves the system (i.e., reaches its destination), its transaction is returned to the appropriate Available List for future reuse. Similarly, system service request transactions can be reused during the simulation.

Queue lists in the simulator are associated with transaction assignment to physical modeling entities, e.g. vehicles, guideway links, stations, etc. The lists themselves are complex data structures designed for efficient maintenance by the system architecture in performing transaction management functions. Each queue list consists of a header which serves a pointer to the last transaction in the list. Transactions are associated with the queue list via a chain pointer which is used to indicate the following transaction in the list, and

form a circular list of transaction membership as shown in Figure 2-9. The circular structure of a queue list provides an efficient mechanism for adding transactions to the list since at most two pointers within the structure must be modified as shown in Figure 2-10. Transactions can be removed from the list without regard to queue ordering by a standard dequeuing procedure as shown in Figure 2-11.

Within the DESM, a set of Transaction Management Service macros are utilized to perform the following list management functions:

- o Obtain an available transaction by type
- o Free a transaction by insertion on the available list
- o Schedule a transaction on the FEL for event completion
- o FIFO or LIFO enqueue a transaction in a queue list
- o FIFO or LIFO dequeue a transaction from a queue list
- o Search a queue list for a specific transaction
- o Dequeue a specific transaction from a queue list

2.2.1.1 Modeling Entity Control

From an architectural or control program view, links and stations are considered as common entities requiring a basic set of fixed processes or macro events. Both require entry and exit testing and a processing component to provide for transaction movement within the entity being modeled. As such, within the MP, the link and station models contain parallel structural components to handle macro event processing as shown in Figure 2-12. The actual decision logic and macro events within these structural components differs for links and stations.

This structure for links and stations allows all interfacing between the link and station models and transaction control to be contained within the system architecture without requiring direct communication between the models. Thus, changes to any model or modeling component does not effect any other subsystem and modularity and flexibility is enhanced. The only requirement for model substitution is that it contains the basic processing components required by the simulation architecture.

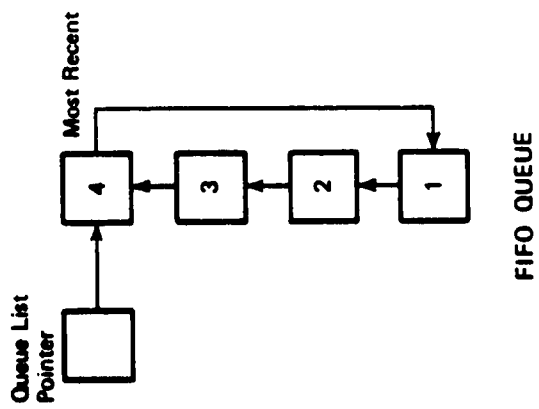
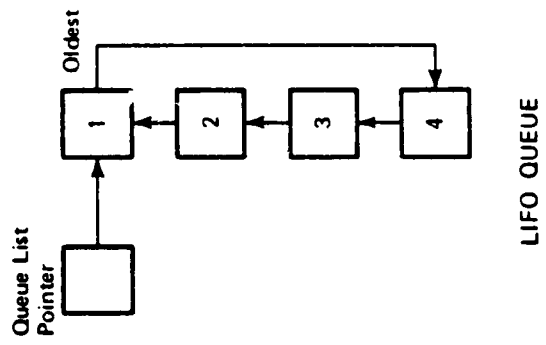
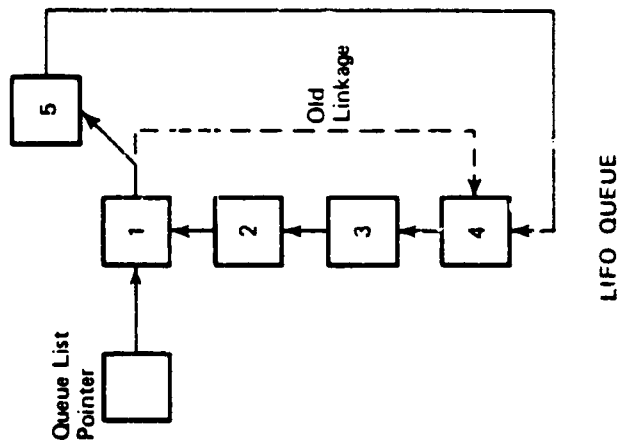
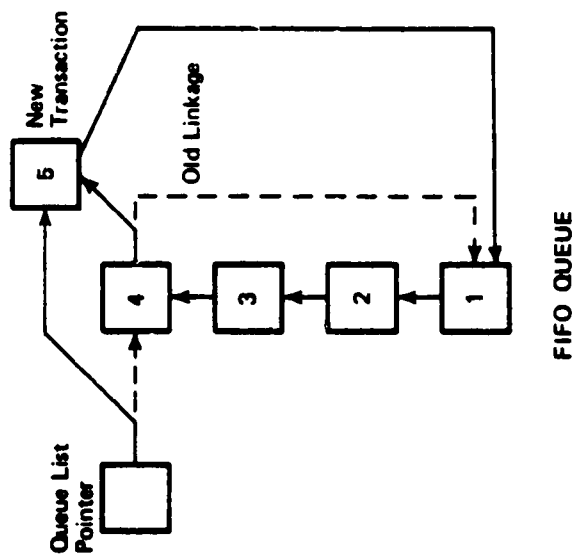


FIGURE 2-9. DESM QUEUE STRUCTURES

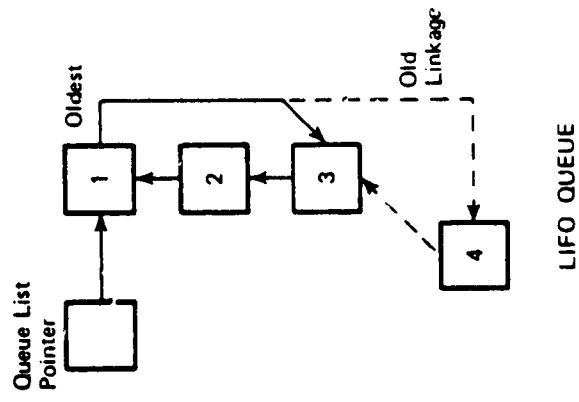


LIFO QUEUE

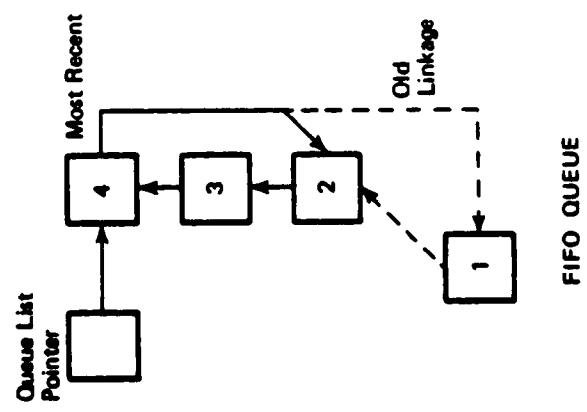


FIFO QUEUE

FIGURE 2-10. QUEUE LIST ADDITION



LIFO QUEUE



FIFO QUEUE

FIGURE 2-11. QUEUE LIST REMOVAL

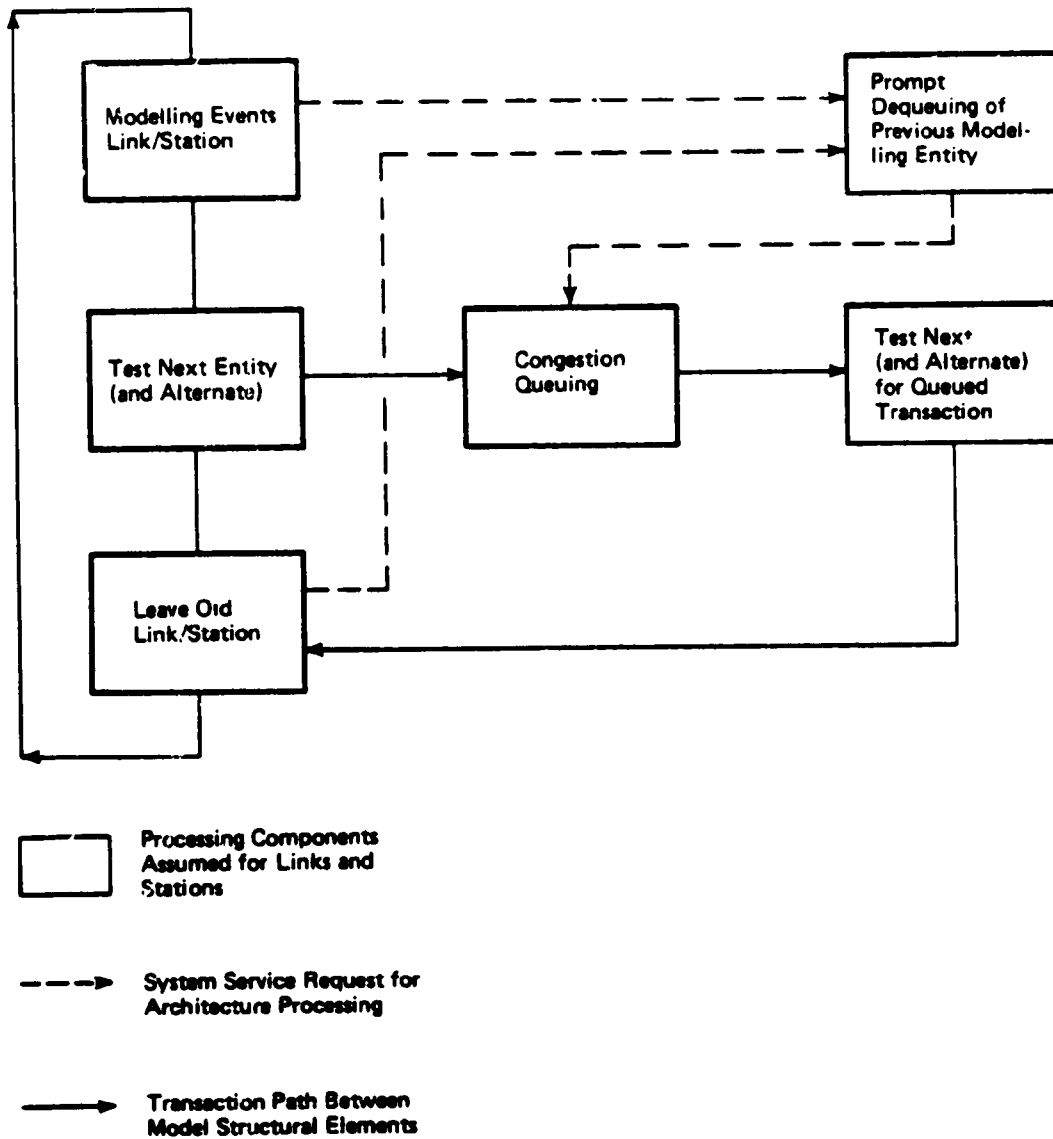


FIGURE 2-12. DESM ENTITY MODELING ARCHITECTURE

2.2.1.1.1 Link Event Processes

Each link in the simulator is modeled as a basic system entity requiring a fixed set of processes to be performed in providing continuous vehicle movement through the simulated network. As mentioned previously, these fixed processes include link entry testing, link travel, and link exit processing. Link entry testing insures that proper vehicle spacing is maintained and downstream congestion is accounted for in moving vehicles between successive links. The traversal of the link is handled by the link travel process. Link exit processing provides the means for controlling queueing and dequeuing of upstream vehicles or communicating downstream congestion conditions to connecting upstream links.

The implementation of these processes in the simulator is facilitated by viewing each link as being composed of a link entry segment necessary for maintaining vehicle spacing, a link travel segment and a link exit queue as shown in Figure 2-13. In this manner, link connectivity is stated in terms of pointers to upstream link exit queues. This simplifies vehicle control and allows alternative link configurations (mainline, merge, diverge) to be described by at most two upstream link exit queues. A mainline or diverge exit link can be described in terms of one upstream queue pointer, whereas a merge is defined in terms of two upstream queue pointers. Defining link connectivity in this manner allows an online station or intersection to be modeled as a combination of diverges and merges. In the case of modeling a station, only one diverge/merge combination is necessary to simulate the station entry/exit process. The merge component of this geometry consists of the mainline entry diverge queue and the station exit queue.

Intersections within the simulator are modeled in a similar manner as a pair of diverge/merge combinations. The modeling of intersection bypass ramps requires the definition of two additional links as does the inclusion of cross over tracks. The same effect can be achieved in the simulator without significant loss of detail by modeling the intersection as a simple pair of diverge/merge combinations. This configuration requires less links to be defined and thereby can reduce simulator run time.

2.2.1.1.2 Station Event Processes

Each station in the network is viewed from an architectural control standpoint, as consisting of the same basic elements as a link in that transactions entering a station require entry, exit, and internal event processing. Although these components are similar as viewed

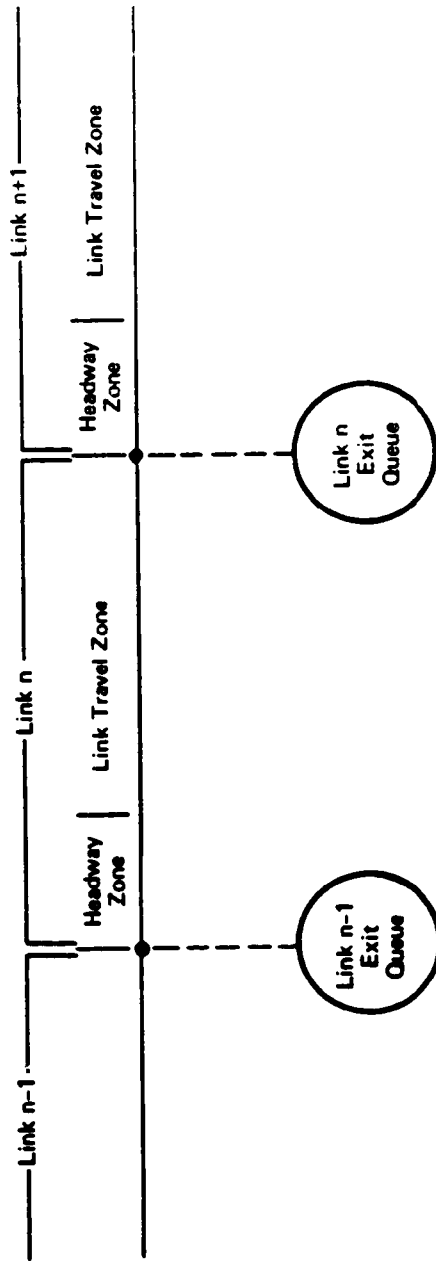


FIGURE 2-13. DESM LINK STRUCTURE

from the control program, the processing for stations is more involved. For example, entry testing involves alternate station assignment in the case of blocked entry and internal station processing encompasses a sequence of events which can be dynamically configured under control of the modeling subsystem.

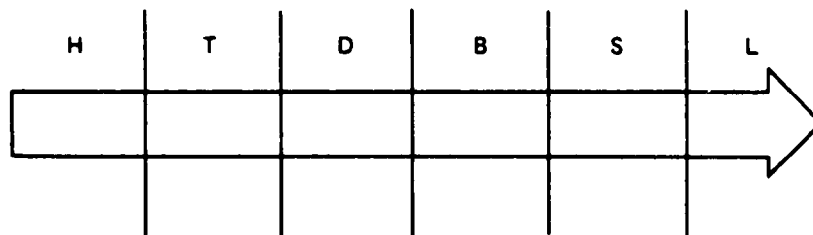
Stations within the DESM are configured as a set of canonical links upon which various micro event processes within stations are performed as shown in Figure 2-14. The specification for each link of the micro events to be performed are made by designating which of the following basic processes are required on each station link.

1. Headway zone travel
2. Link traversal
3. Passenger deboarding
4. Passenger boarding
5. Empty vehicle storage
6. Launch time determination and processing.

The selection of processes on any one station link are constrained by the configuration of the basic station structure supported in the DESM as shown in Figure 2-15. For example,

1. The station input ramp can be modeled by events (1) and (2)
2. A docking lane can be modeled by a combination of events (1) (2), (3), (4), and (6)
3. Output lanes can be modeled by events (1), (2), and (6)
4. Station storage areas by event (5).

For a given simulation experiment, not all station links need be present to support internal station processing. The minimal configuration for station modeling requires at least one link containing a docking and launch area to be configured. Such a configuration can be used to model a simple point station. The processing of the micro events on the station links and management within the station is totally model dependent and separate from the system architecture. Once a vehicle enters a station, it is subject to the internal processing of the station model. If upon leaving the station processing area, a vehicle cannot exit the station, it is queued in the station exit queue in a manner analogous to the procedure employed in exiting a guideway



- H - travel the headway zone;
- T - travel the main body of the link;
- D - undergo the deboarding of passengers;
- B - undergo the boarding of passengers;
- S - store the vehicle on this link;
- L - undergo the delay waiting for launch.

FIGURE 2-14. STATION LINK CANONICAL DEFINITION

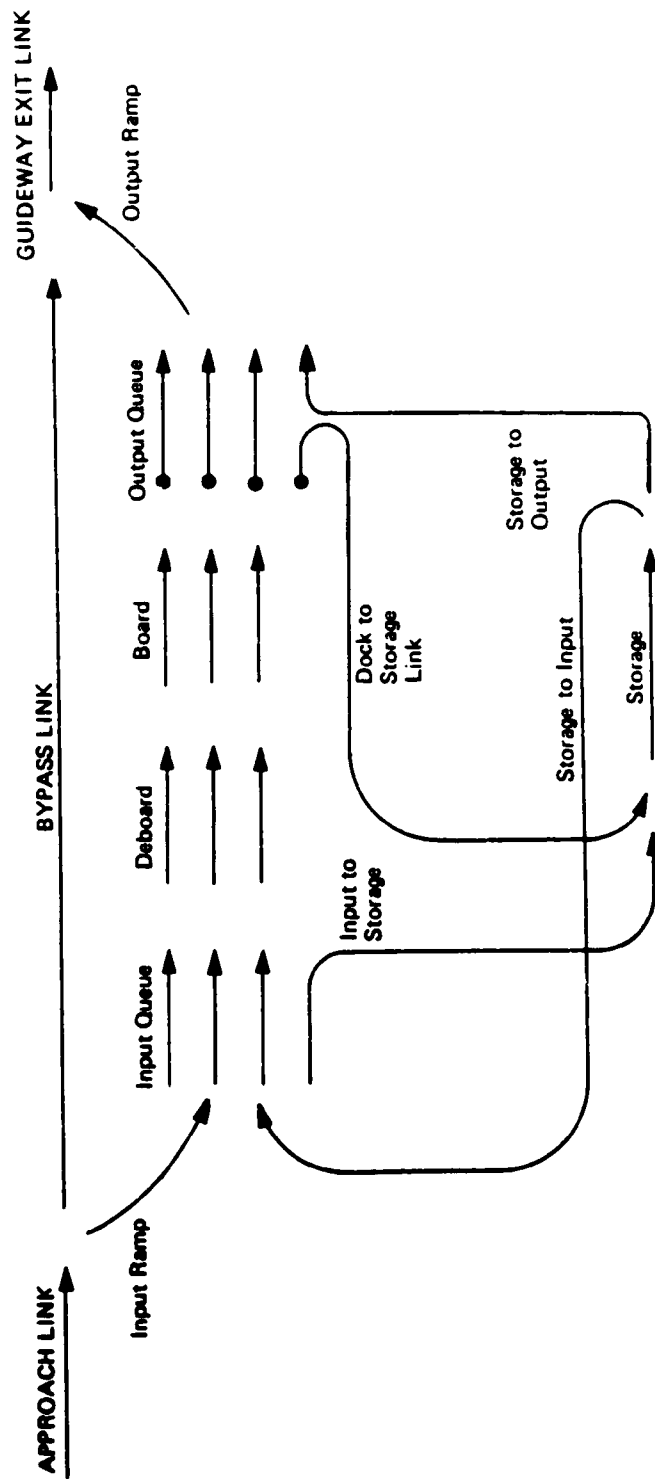


FIGURE 2-15. DESM STATION CONFIGURATION

link. This exit queue forms one component of a merge with the mainline guideway.

2.2.1.1.3 Transaction Dequeuing

As previously mentioned, vehicle transactions in the simulation are subject to queuing within the modeling subsystem depending upon whether the next entity or processing event for which they are to be scheduled is available or can be performed. The MP provides the means by which queued transactions can be restarted (scheduled for their next event) when conditions are such that the event upon which they are waiting can now be performed. This is accomplished by the scheduling of a system transaction which causes the prompting (interrogation and dequeuing) of upstream guideway or station modeling entities. The scheduling of prompt transactions results from completion of specific event processes within the simulation system. Specifically, the scheduling of a prompt transaction occurs each time an entity macro or micro exit or entry event is processed. Additionally, prompts are scheduled in response to asynchronous events such as failure recoveries to provide the means for transaction restart.

2.2.1.2 Future Events List

The Future Events List (FEL) is a time ordered list of pointers used to chain transaction IDs for scheduling of events for occurrence in future simulated time. Time is quantized into discrete, finite units called 'clock units', with each unit representing some period of simulated time, e.g., one millisecond. The FEL is divided into two components, a clock table for scheduling near time transactions, and an extension (multiple thread list) for scheduling distant time transactions. Each pointer in the clock table begins the list of transaction IDs which require processing during a simulation interval. The point in real time at which the simulator is currently operating is given by clock time, which provides the number of clock units which have passed since the start of the simulation experiment.

Every transaction that represents an action to be performed at some future time is placed into the FEL at the proper time point. To record when events are to occur, each transaction has a time word that defines the time at which it is to be processed. Scheduling of transactions on the FEL is performed by both the architecture control and modeling subsystems. Each transaction has as part of its definition a chain word which is used for inserting it into the FEL. Transactions are inserted into the FEL by determining the time interval (pointer) within which the event being scheduled is to occur. The transaction is then chained in time order into the list of transactions which are to become active in the specified simulation interval. The organization of the clock table portion of FEL is shown in Figure 2-16.

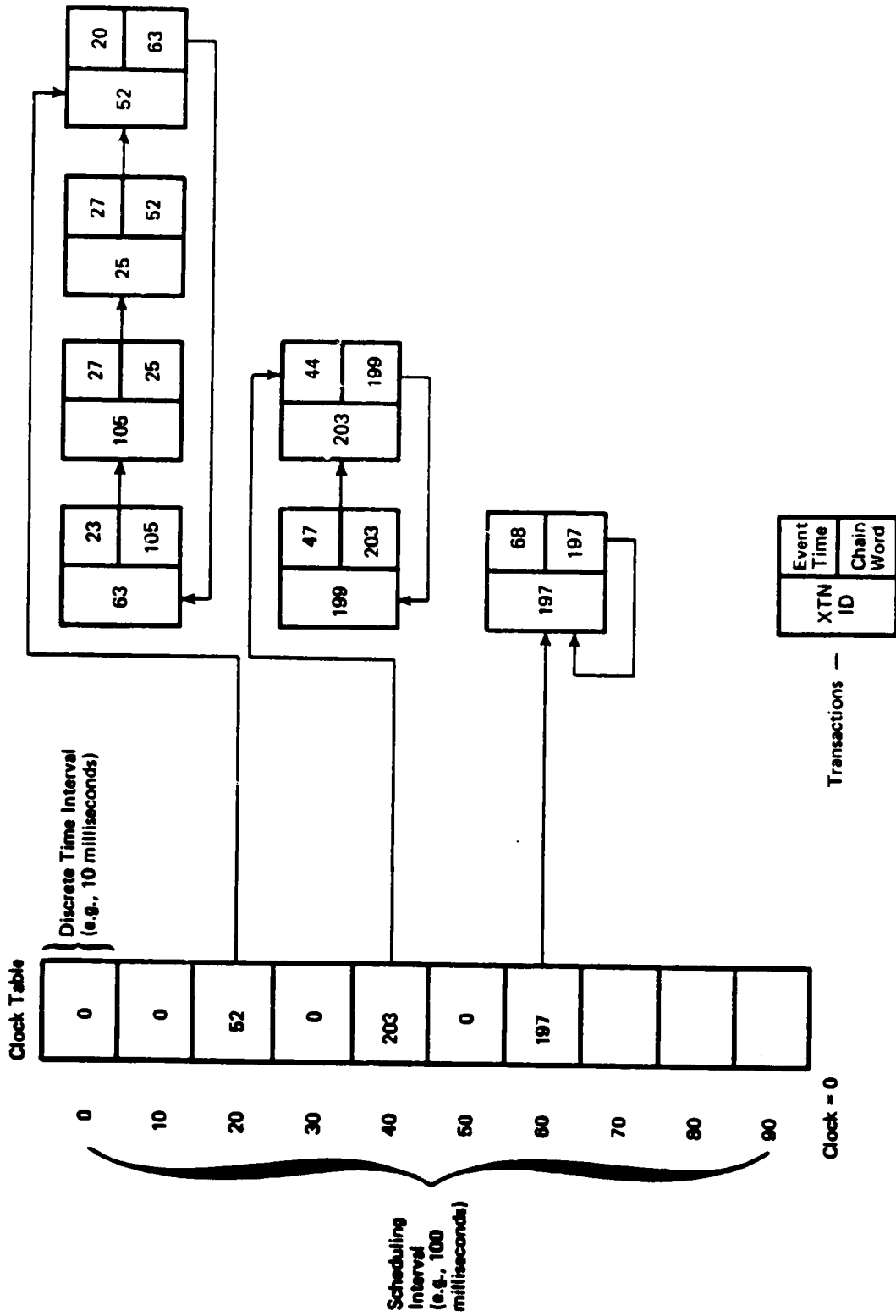


FIGURE 2-16. CLOCK TABLE ORGANIZATION OF THE FEL

Since the clock table portion of the FEL is of finite length, only a finite number of time intervals can be represented. Transactions which must be scheduled for a time interval greater than the time period represented by the clock table are scheduled on the FEL extension or multiple thread future events list. Entries or quantized intervals in the multiple thread list represent an interval of time corresponding to an entire clock table interval. Multiple thread list pointers differ from clock table pointers in that they are created dynamically as required during the simulation experiment by chaining available transactions which serve as the FEL pointer for chaining transactions which are scheduled during that simulation time interval. Transactions placed on the multiple thread list are chained from the multiple thread transaction without regard to discrete simulation intervals as maintained in the clock table. The organization of the multiple thread list is shown in Figure 2-17.

Once the simulation interval encompassed by the clock table has passed (all transactions processed and clock updated to last transaction time), the clock table is updated from the next available multiple thread list.

2.2.1.3 Event Recognition and Control

The basic control loop in the MP is to determine the next event to be performed, update the simulation clock, and perform the event. Since every event is represented by a transaction, the transaction is the basis for determining the next process to be performed. The control loop in the simulator consists of the following as shown by PDL segment EMODEL:

1. Obtain the next most imminent transaction. The next event to be performed is indicated within the transaction which is first on the FEL.
2. Remove the transaction from the FEL.
3. Update the simulation clock to the time of the transaction. Whenever the simulation clock is updated, it is updated to the time of the next most imminent event.
4. Perform the indicated event. The type of event to be performed is indicated by another item of information associated with a transaction. This item is used to determine which architectural processing component is required and a control transfer is performed. The architectural component invokes the required processing functions as listed below.

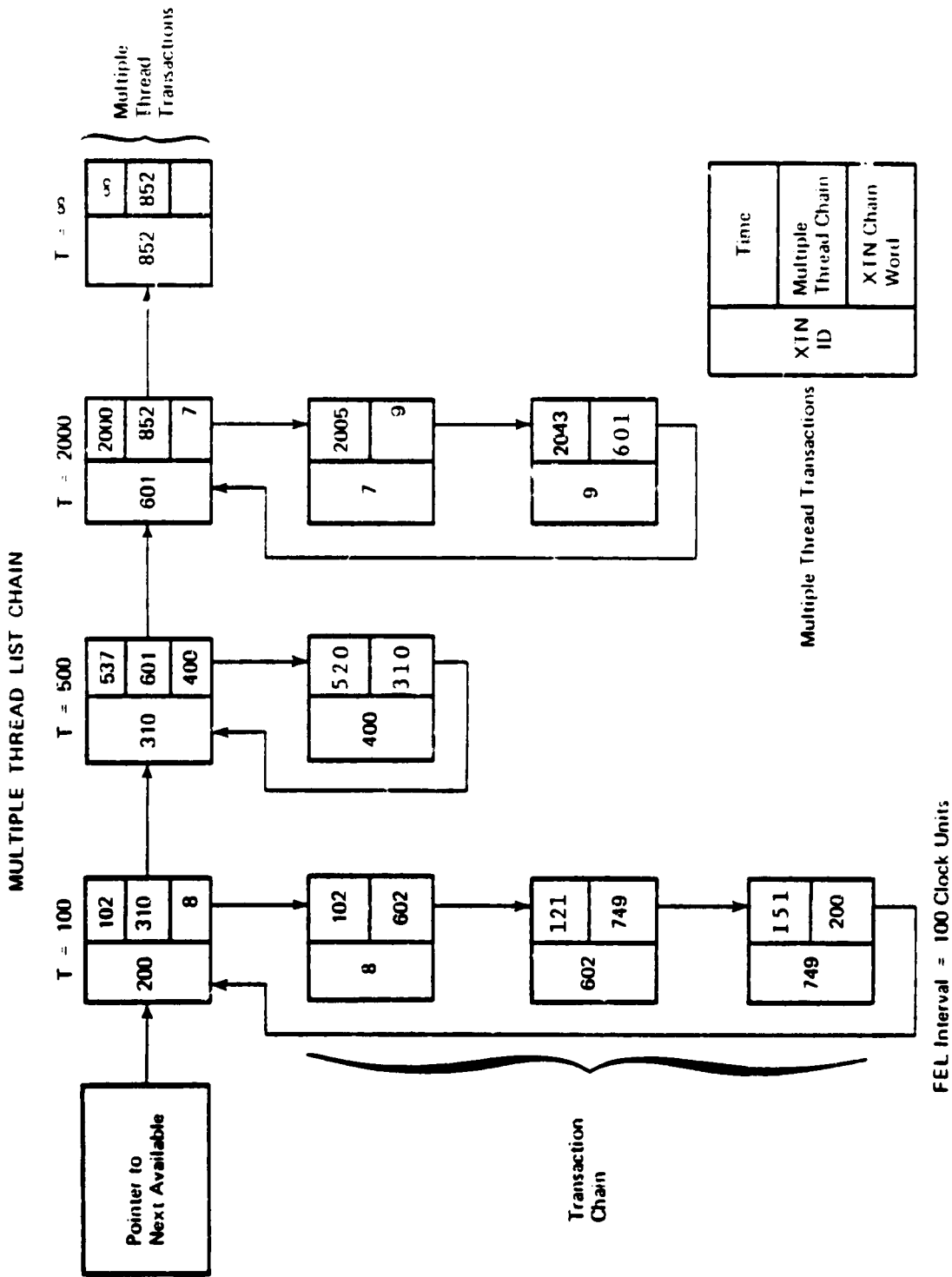


FIGURE 2-17. MULTIPLE THREAD LIST ORGANIZATION OF THE FEL.

- a. Asynchronous Simulation Event - This event reads asynchronous command data and performs processing to execute the command. The transaction which triggered the asynchronous event is rescheduled to occur at the time indicated by the next asynchronous data header card.
- b. Sampling Event - This event causes statistics to be accumulated and recorded. The transaction which triggered the sampling event is rescheduled to occur at the time of the next recording interval.
- c. Trip Arrival Event - An arriving trip entering the simulation is processed. The system transaction is rescheduled to occur at the time of the next trip arrival.
- d. Periodic System Checkpoint - This event causes the recording of current simulation status. The system transaction is rescheduled to the next interval at which checkpoint is required.
- e. Transaction Dequeuing Event - The queue associated with a system modeling entity is interrogated for the existence of waiting transactions. If required, waiting transactions are dequeued and scheduled for their next event. The system transaction which triggered the event is released and returned to the available transaction list.
- f. Guideway Link Event - The transaction is passed to the guideway modeling subsystem for processing.
- g. Station Link Event - The transaction is passed to the station modeling subsystem for processing.
- h. Periodic Computation Event - This event causes a simulation related algorithmic computation to be performed. The system transaction may be rescheduled for the next periodic event time.
- i. Simulation Termination Event - System termination activities are performed.

2.3 OUTPUT PROCESSOR

The Output Processor provides the services necessary to retrieve the statistical output from the Model Processor, perform summarization functions as requested by the user, and prepare printed reports in a requested format suitable for analysis. The summaries include time series

listings, plots, statistical summaries, histograms, and predefined composite reports. Selected performance measures are also evaluated and written to a Performance Summary File for later comparison with the results of other simulation experiments.

The primary data interfaces to the Output Processor are shown in Figure 2-18.

2.3.1 Architecture

The OP is designed to provide a flexible input/output procedure which limits dependencies on specific data formats or the volume of data resident in the raw statistics file. This is achieved by viewing the functions required to achieve data retrieval, storage, manipulation, and formatting as modular components invoked as specific services as required in response to user input commands or performance of other service functions. The functional flow through the OP is shown in Figure 2-19. Additionally, the sampling data read from the input source does not depend on a fixed record format, but is processed as required via identifying information which precedes each stored data record in the file.

Execution of the OP is initiated by invoking a cataloged job control procedure contained in the procedure library. Upon initial entry, the OP performs initialization processing to establish initial conditions for the processing of a raw statistics file. This involves the initial formatting and allocation of the bin storage area which is used for data (sample value) storage during the acquisition process. Default parameters relating to problem size definition are acquired from the raw statistics file to define the following characteristics of the simulation experiment from which the raw statistics were derived.

1. Number of Guideway Links
2. Number of Station
3. Number of Station Links
4. Number of Routes and Route Groups
5. Simulation Clock Granularity
6. Sampling Interval
7. Vehicle Capacity
8. Station Link Type Classifications.

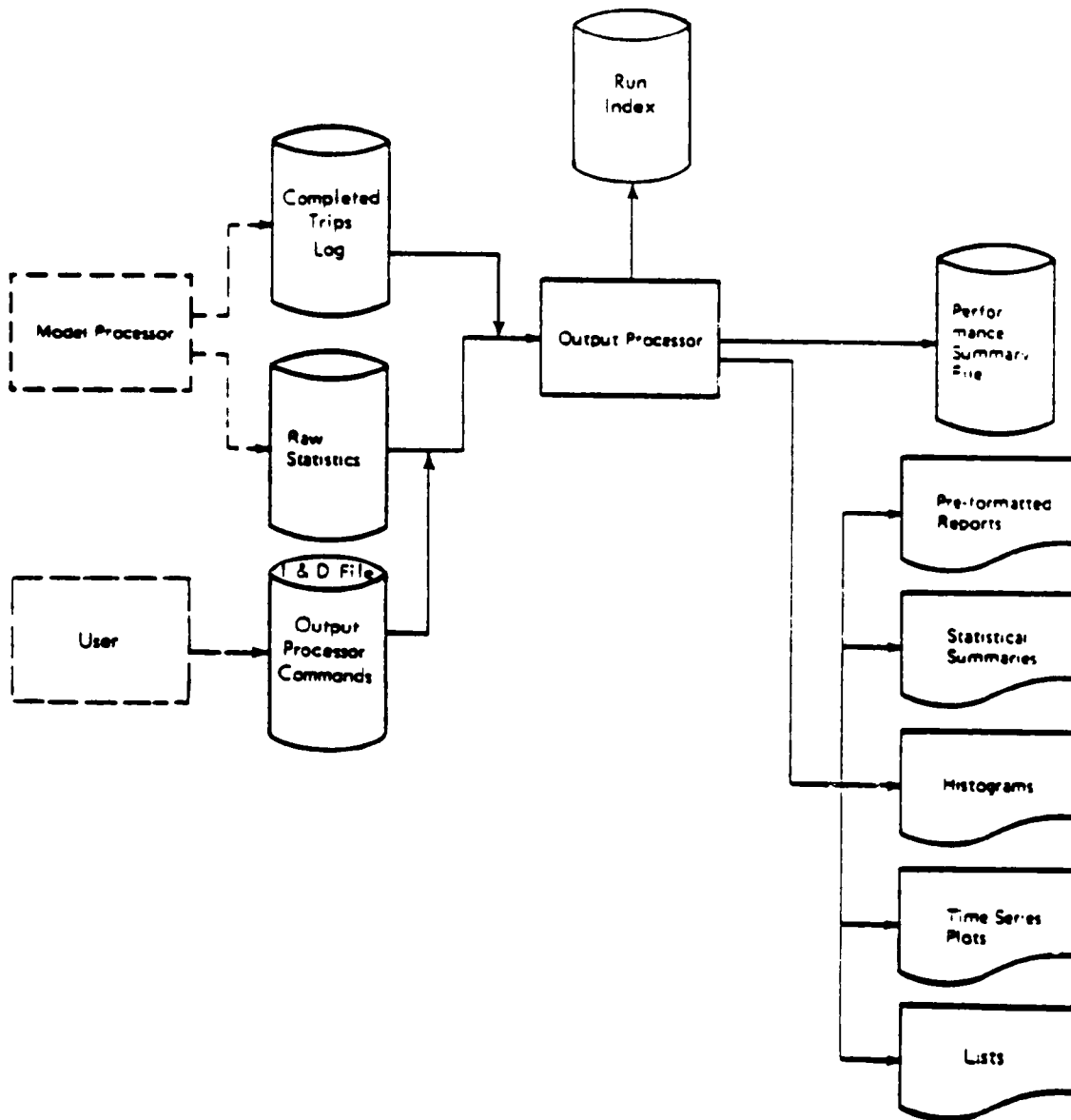


FIGURE 2-18. OUTPUT PROCESSOR DATA INTERFACE

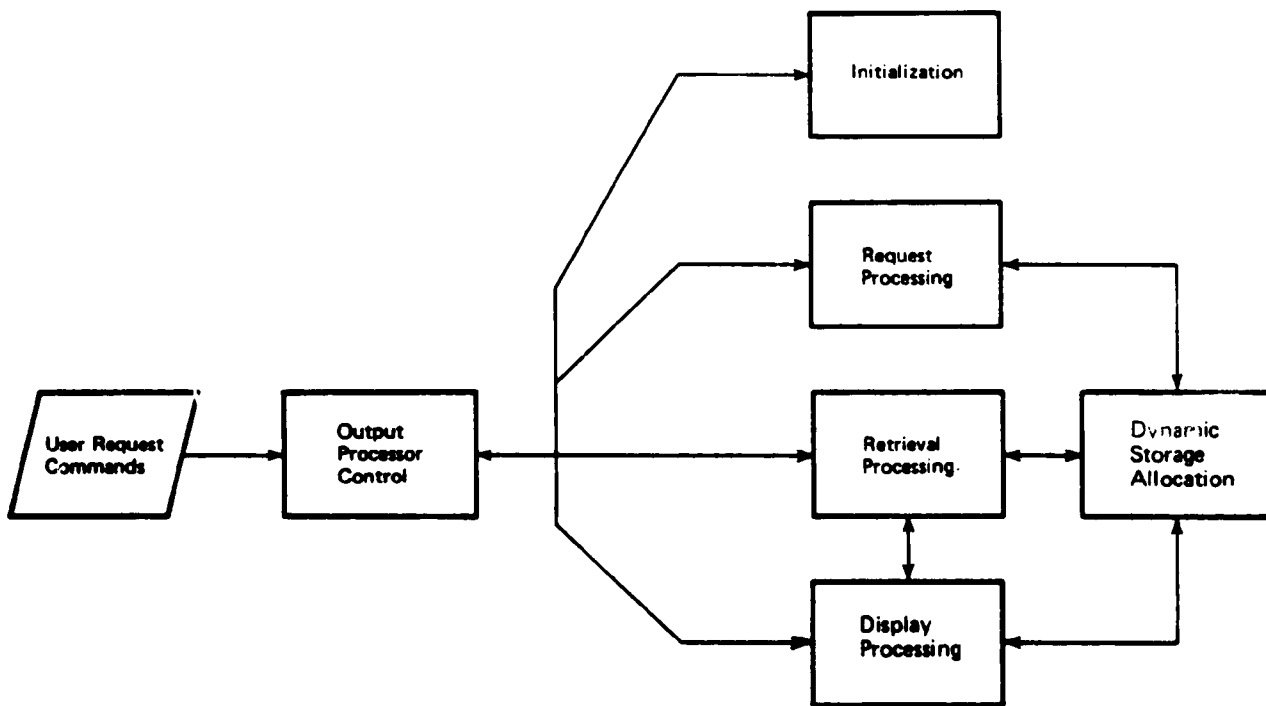


FIGURE 2-19. OUTPUT PROCESSOR FUNCTIONAL FLOW

These characteristics are used to control data acquisition and format control in processing the raw statistics file and in the computation of system wide performance measures and derived statistical measures.

Once initialization has been performed, the basic control loop of the OP is initiated. Each OP request is processed until an acquisition (READ) command is encountered. These requests are entered by the user to control auxiliary (trace) output printing, acquisition of individual statistics, performance summary generation, and pre-formatted report preparation.

As each command request is processed, a region within the data storage area is assigned to contain the data required for satisfying the request, and decoding and filing of the command request is performed. During the decoding process, those requests for individual statistics which require data to be collected over a range of entities, cause the automatic generation and filing of individual requests for each entity in the specified range.

The actual acquisition of data, from the raw statistics file, is initiated upon encountering an acquisition (READ) command during the processing of user data requests. The acquisition process begins with positioning of the raw statistics file to the beginning of the request accumulation interval and correlating the data associated with filed requests to record types and formats contained in the raw statistics file. The raw statistics file is composed of a sequence of time tagged header records identifying the type of data which follows and a group of follower records which contain the recorded sampled data.

Actual data acquisition from required record types is performed by I/O processing based on individual record type for the major data category indicated in the record group header. This processing iterates upon each of the follower records in turn and then upon each of the requests in the request table associated with the particular record type.

For each iteration, the required data item, located in the follower record, is retrieved, and stored in the appropriate storage area. If during the store process, an allocated storage area for a particular request becomes full, it is automatically reallocated to contain more space. Thus, the file reading process does not require a "second guess" of how much of each type of data actually resides in the raw statistics file. Once storage of a data item has been performed, the storage space pointers contained in the request table entry are updated to reflect storage utilization.

Data display processing is performed upon completion of data acquisition to produce the desired output format requested by the user. The display process involves cycling through the filed requests and

manipulating, formatting, and outputting the associated data. As part of the formatting process, sixteen character descriptive titles for the requested statistics are retrieved for labeling purposes. In the case of station link data requests, a type mnemonic is assigned for output labeling, based upon the numeric configuration type designation for the requested station link.

2.3.1.1 Data Storage Allocation

The OP contains a unique data storage mechanism which does not require a priori knowledge of the amount of data to be processed. All data in storage is viewed as consisting of vectors of successive values for one kind of data, e.g., a time series of queue occupancy levels, a list of average vehicles in service, etc. In the OP, these vectors are designated as "bins" which are consecutively numbered and referenced by number. Bin allocations are initially made during initialization processing according to descriptive information contained in the first record of the raw statistics file. This information is provided by the MP during its initialization phase and reflects the configuration (entity requirements) used in the simulation run.

Figure 2-20 illustrates the manner in which bins are referenced within the OP. The bin number, i , is used to index the bin location pointer. The number in the location pointer (i) provides the position in the bin storage area, at which the bin is located. By convention, $j = \text{location pointer } (i)$ always indicates the third word that has been allocated to the bin such that data retrieval by bin can be accomplished. Each bin is initially allocated as five words consisting of four header words plus one unused data word. The initial number of bins allocated is given by the number of entities (links, stations, vehicles) used in the simulation experiment as reflected in the raw statistics file.

In general, a bin consists of several distinct areas as shown in Figure 2-20.

1. The system header — entries $j-2$ and $j-1$. This area is used exclusively by the bin storage allocation and maintenance services. It specifies:
 - a. The total number of words in this bin area, including those in the system header
 - b. The identification number of this bin.

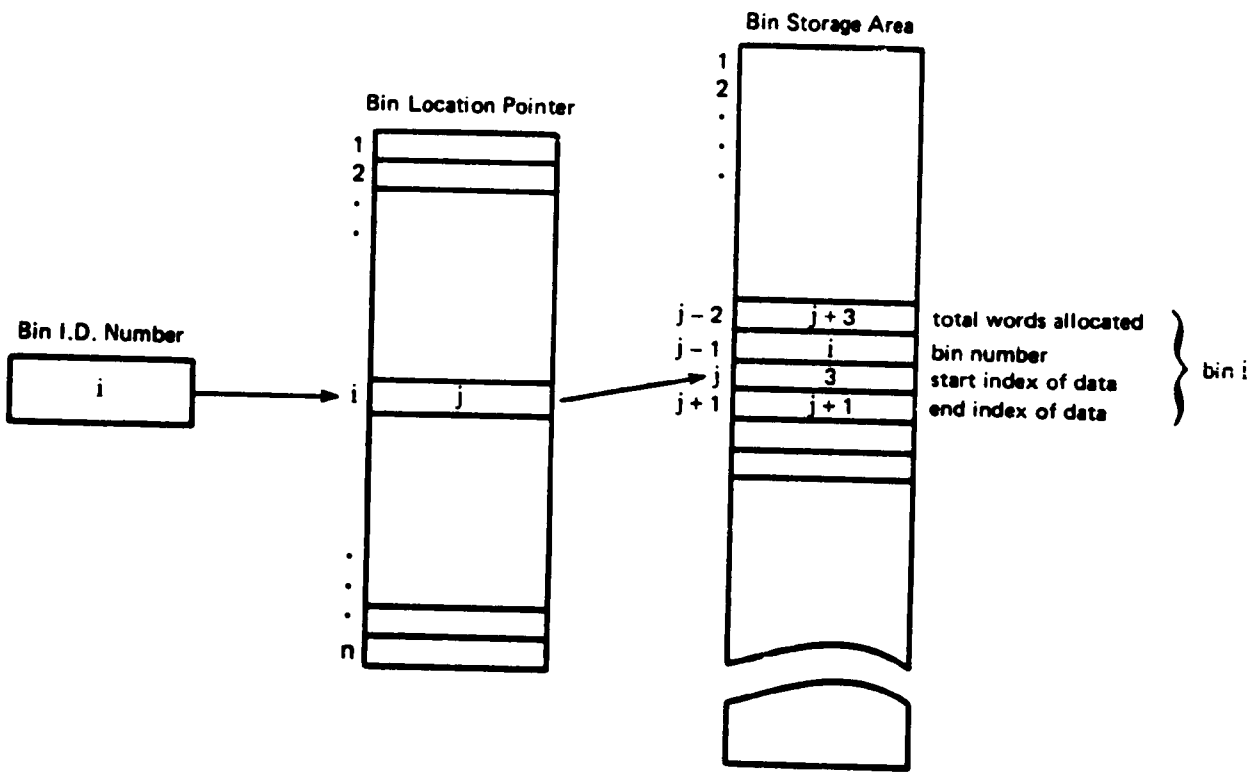


FIGURE 2-20. OUTPUT PROCESSOR BIN REFERENCING

2. The data header - entries j through j+2 used during the data retrieval process:
 - a. The starting index of the data in the bin
 - b. The ending index of the data in the bin
 - c. A data identity area used to identify the data in the bin.

The unused portion of the bin allocation area is set up as a pseudo-bin, with bin number set to zero to indicate it is unused.

2.3.1.2 Command Request Storage

Each data request entered by the user causes the contents of the request to be filed in a request table used in data retrieval processing. As part of this filing process, a bin assignment and reallocation is made for internal storage of the sampling data to be retrieved during request servicing. The amount of space reallocated to a particular bin depends upon the display mode specified in the request. The amount of space allocated at this point serves only as an initial estimate of storage required. If further space is required during the data retrieval process it is obtained dynamically.

In addition, to the above, request filing results in the category definition for the data item selected. This definition is stored with the request for identifying the required sampling records which must be processed to service the data request. In the raw statistics file, data is stored on a time sample basis according to a category hierarchy. Data is classified as to which portion of the model it pertains (major category): system, link, station, station link, or route.

System-level data requires no subscript or index—each data element is a single number. Link, station, or route data elements require an entity index since each element consists of multiple values—one per link, station, or route. In addition station link data requires two indices to identify the particular station and link for which sample values are required.

Further, data is classified as to whether it is status data or historical data (subcategories). Status data reflects the status of a modeled area at the instant at which sampling took place (e.g., the number of vehicles on link five). On the other hand, historical data reflects what events transpired over the interval preceding the sampling event (beginning after the previous sample and ending at the time of the current sample). Examples of historical data are the number of vehicles leaving a link exit queue during the sampling interval, and the average number of vehicles on a link over the sampling interval.

The organization of data in the raw statistics file consists of groups of unformatted logical records. The first record of a group is a header record. The header contains the following information.

1. A code number that indicates the type of group (major category)
2. A count of the number of logical records (sampling data follows) in the group, excluding the header. If the group consists of the header only, this count is zero
3. The value of the simulation clock at the time the record group was written. (This value is non-decreasing along the file.)

The remaining logical records of the group, if any, have a format unique to that type of group, which is indicated by the header. This header-follower organization has several distinct advantages:

1. Widely varying kinds of data may be interspersed in the file
2. Within a given simulator clock value, the order of information is unimportant
3. Groups of records may be placed in the file or omitted from it, at the sole discretion of the program which is writing the file (i.e., the simulator). The sampled data items recorded for a given simulation experiment is dependent upon the transit system being modeled.
4. Information not needed on a given part of the tape can be quickly skipped, simply by skipping the number of followers specified by the header.

The relationship between the request table and data storage bins resulting from the request filing process is shown in Figure 2-21.

2.3.1.3 Establishing Request/Record Correlation

Once the data acquisition process is started in response to a read command, a matching process which identifies which data records can be used to satisfy filed requests is performed. The matching process is started during data acquisition for the first record groups contained within the desired accumulation interval. Subsequent occurrences of record groups need not be matched, once the matching process has been performed for that type.

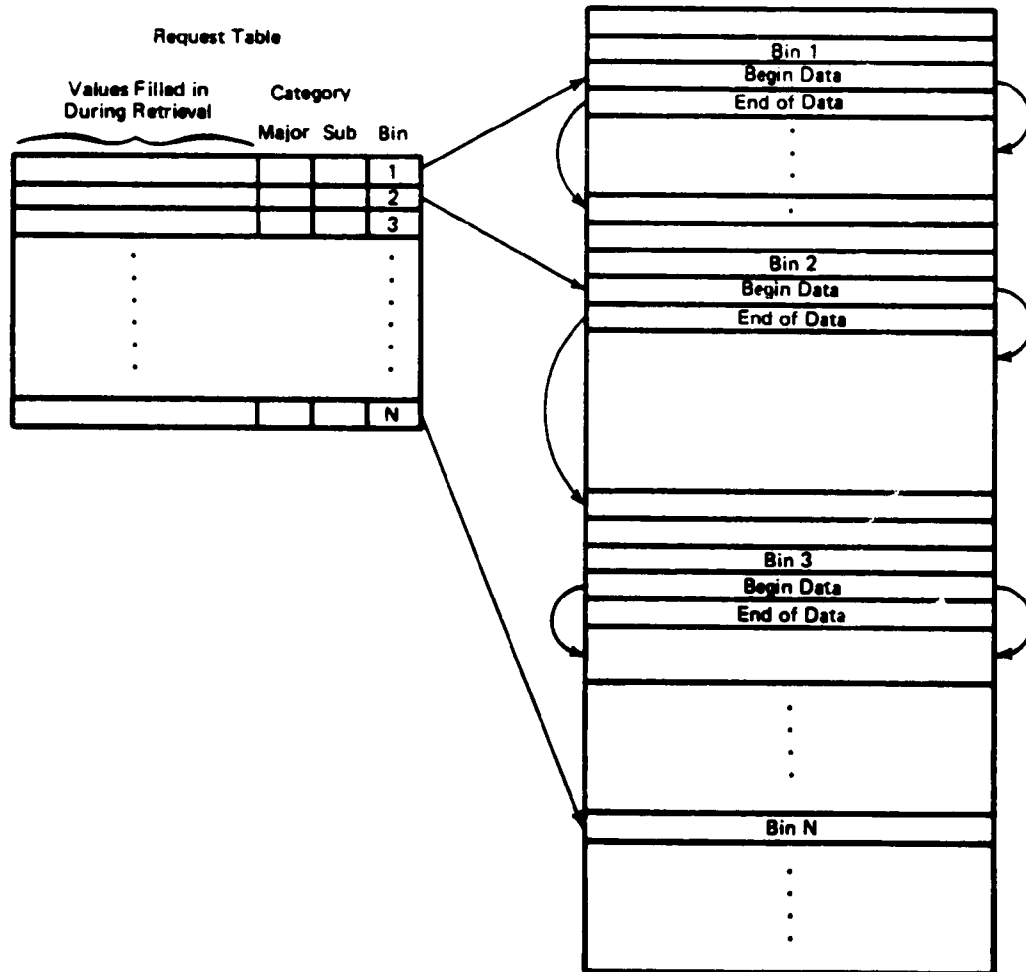


FIGURE 2-21. REQUEST FILING/BIN STORAGE RELATIONSHIP

Prior to reading the header for the first record group in the requested interval, a match table, which provides for an indicator for each category (record group), M, is initialized as follows:

- o 0 - Record groups of type M are to be ignored (if found on the tape, they will be skipped)
- o 1 - Groups of type M contain data necessary for proper operation of the OP and are always to be read. (Example are conversion tables, system dimensions, etc.)
- o -1 - Groups of type M might be required, depending upon their existence in the file and upon a request for data contained within them.

As reading begins, if the major category indicator for the record group is -1, the match process is performed as shown in the example given by Figures 2-22 and 2-23.

As shown in Figure 2-22, record type M has the main category ' α ' and a list of subcategories given by column j of the subcategory mnemonic table. This list includes subcategories ' β ' and ' γ '. In looking for matches, the request table entries containing the major and subcategory mnemonic designators are compared with ' α ' and the subcategory list for all i. Lines s and t of the request table matched both ' α ' and either ' β ' or ' γ '. Consequently:

1. The subcategory indicator of the request table was set to indicate the index of the matching subcategory. That is, since line s matched the first subcategory (' γ '), the indicator was set to l. Likewise, the indicator for request (t) was set to k.
2. A list of requests that can be satisfied by record type M (namely, requests s and t) was created by forming a chain:
 - a. The entry in the major category indicator table which was -1 prior to the matching process, was set to "t", one of the request table lines satisfied by record type M.
 - b. Item next request for entry (t) in the request table gives a second line (s) satisfied by record type M.
 - c. Item next request for entry (s)=0 indicates that no other lines in the table are satisfied by record type M.

As a result of these actions the major category indicator indicates those lines of the request table that are satisfied by records of type M, via the chain. Also, the subcategory entry of the request table indicates

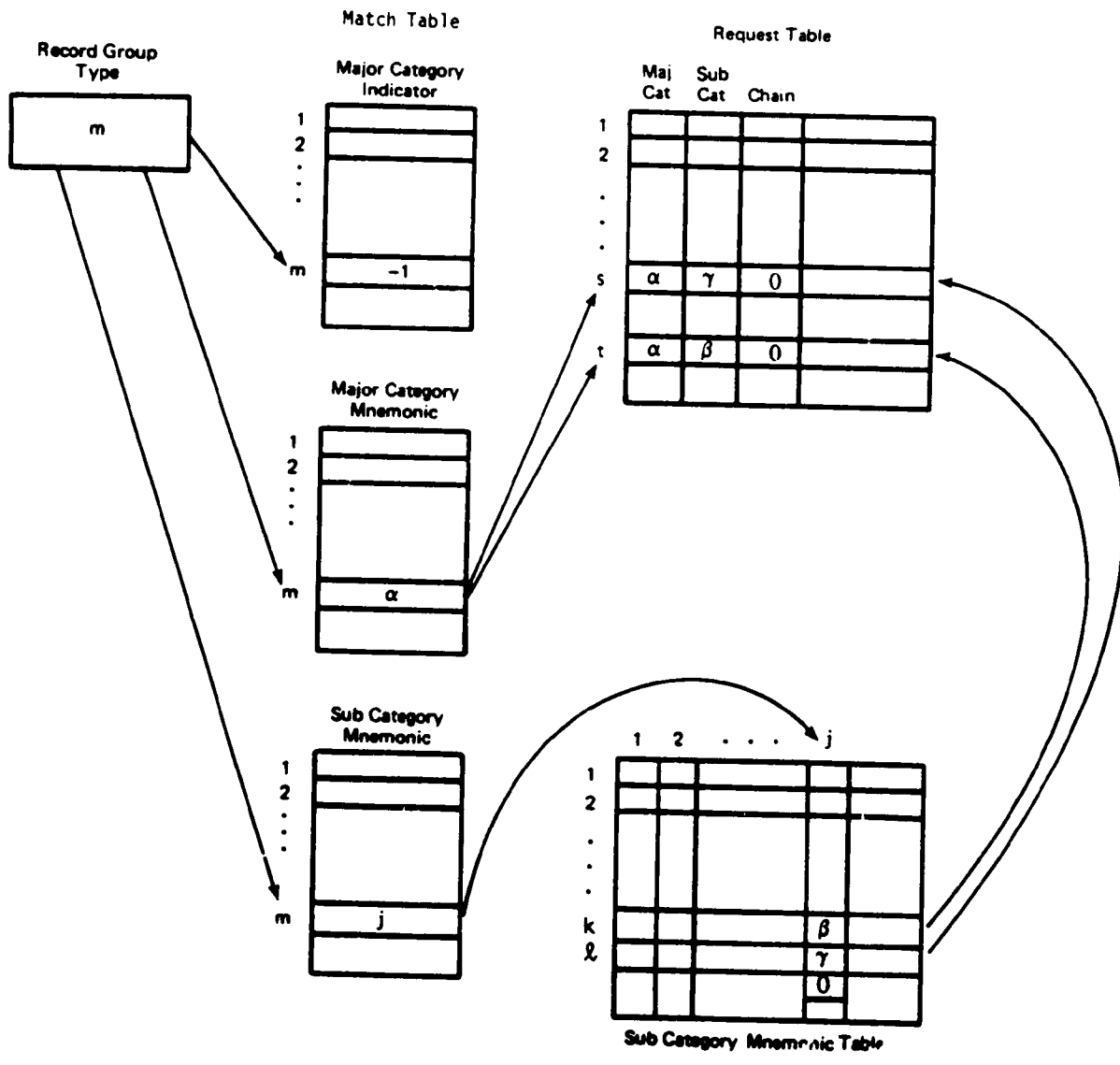


FIGURE 2-22. DATA MATCHING PROCESS

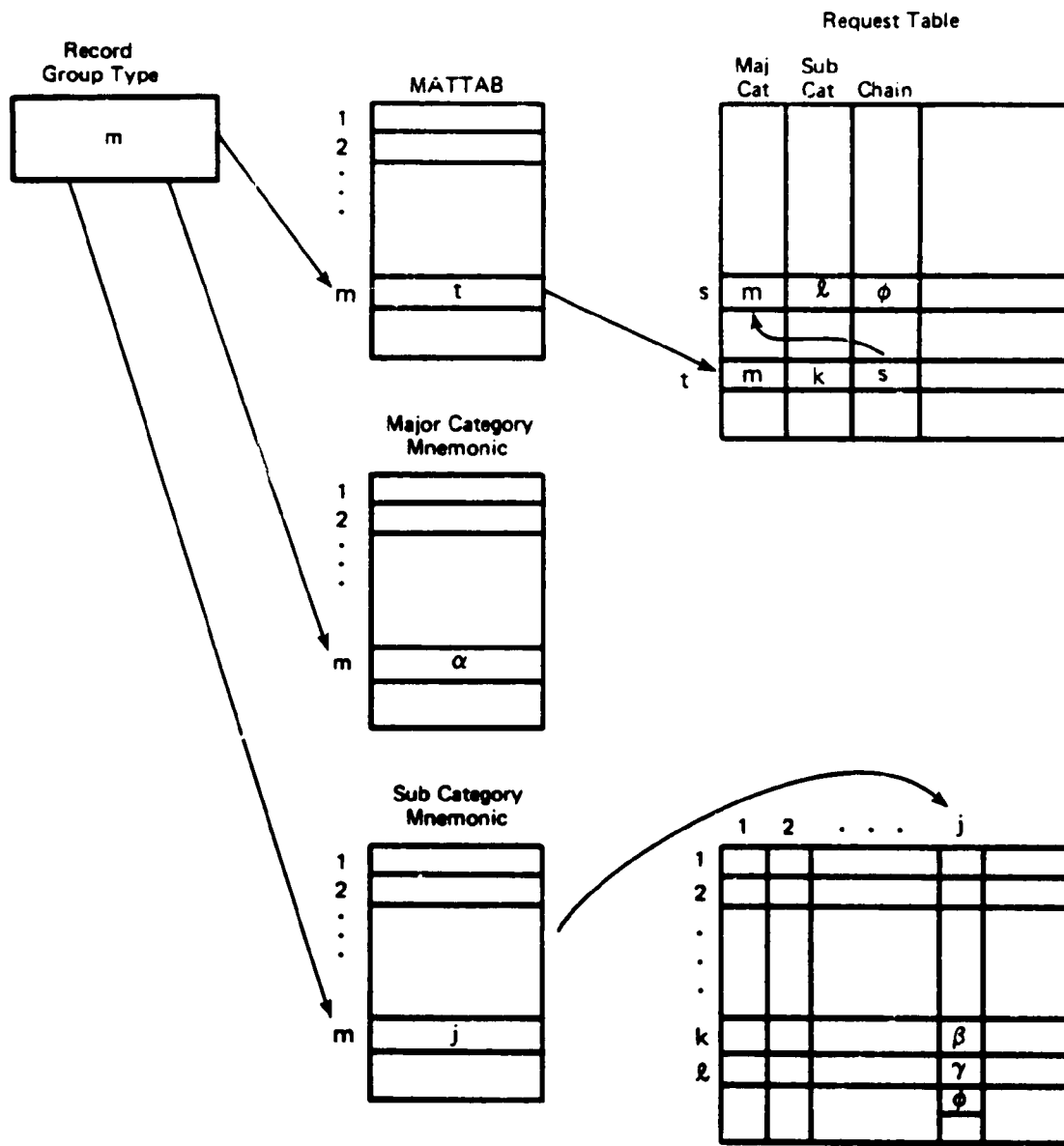


FIGURE 2-23. DATA MATCHING RESULTS

the exact subcategory of data that was requested, by number. Consequently, the main and subcategory mnemonic tables need not be referenced hereafter for records of this type.

The subcategory indicator now contained in the request table entries, serves as the position designator of the required item within the sampled data record.

SECTION 3. GLOBAL VARIABLE DICTIONARY

Table 3-1 defines variables that are global within the processors. Those COMMON areas beginning with ECI are the structured system and network characteristics data established by the input processor based on user input and used by the model processor. COMMON areas beginning with ECN are used only by the input processor. COMMON areas beginning with ECM are used only by the model processor. COMMON ECAMSG passes initial conditions from each of the processors to the error message handling routine. COMMON areas beginning with ZC are the sampled data generated by the model processor and used by the output processor. The following data areas are included in the table.

- o ECAMSG - error message data
- o ECICFG - station configurator input data (2)
- o ECIFEL - future events timing control data (1)
- o ECIGL - guideway link data (1)
- o ECIMAX - run time problem sizes (1)
- o ECINET - network structured data (1)
- o ECIPOL - policy data (1)
- o ECISL - station link data (1)
- o ECISTN - station data (1)
- o ECISYS - simulation system data (1)
- o ECIVEH - vehicle data (1)
- o ECMFEL - MP future events timing control data
- o ECMGL - MP guideway link data
- o ECMPOL - MP policy data
- o ECMSL - MP station link data
- o ECMSTN - MP station data
- o ECMSYS - MP simulation system data
- o ECMTRP - MP trip data
- o ECMVEH - MP vehicle data
- o ECMXTN - MP transaction header data

- o ECNDMD - IP demand generation data (2)
- o ECNNET - IP network data (2)
- o ECNPOL - IP policy data (2)
- o ECNSTR - IP network structured data (2)
- o ECNSYS - IP simulation system data (2)
- o ECNTRN - IP transportation algorithm data
- o EILIMITS - IP limits for reasonableness checks
- o ESYSMAX/ECOMMAX - compile/assembly maxima
- o (blank) - OP data storage (bin area region)
- o BASIC - OP bin assignment mapping
- o EOSTAT - OP standard report 2 measures
- o HISTO - OP histogram data
- o IENDS - OP request storage counter
- o MATCH - OP record/request chaining table
- o NAMES - OP request translation data
- o OUTPT - OP output display label data
- o OUTPT1 - OP plot limits for time series display
- o OUTPUT - OP output control data
- o OUTSUM - OP standard reports data accumulation arrays
- o READER - OP data acquisition parameters
- o REQUES - OP request table
- o SUB - OP bin allocation data
- o SYSCOM - OP simulation related parameters
- o SYSCM1 - OP simulation related parameters (cont)
- o TABLES - OP request correlation table
- o TERMNL - OP input echo flag

- o ZCLNK - guideway link statistics
- o ZCRTE - route statistics
- o ZCSL - station link statistics
- o ZCSTN - station statistics
- o ZCSYSG - system-wide guideway link statistics
- o ZCSYSR - system-wide route statistics
- o ZCSYSS - system-wide station statistics
- o ZCSYST - system-wide performance statistics
- o ZCTRP - trip statistics
- o ZCVEH - vehicle statistics

- (1) Variables also defined in assembly member ECICOMN to be made known to GDIP routine in input and mode processors.
- (2) Variables also defined in assembly member EIGDIP4 to be made known to GDIP routine in input processor only.

The format of the definitions is as follows:

<u>VARIABLE</u>	<u>DIM</u>	<u>TYPE</u>	<u>DESCRIPTION</u>
A	B	C	D

where:

A is the name of the variable

B is the dimension of the variable; dash (-) implies it is a scalar

C is the type of the variable

L1 - logical, 1 byte

I2 - integer, 2 bytes

I4 - integer, 4 bytes

R4 - real, 4 bytes

R8 - real, 8 bytes

D is the definition of the variable including the values it can assume if a fixed set exists.

TABLE 3-1 (1 of 53) GLOBAL VARIABLES

NAME: ECAMSG			CATEGORY: ERROR MESSAGE DATA
VARIABLE	DIM	TYPE	DESCRIPTION
KMSG	3	I2	NUMBER OF MESSAGES ISSUED DURING AN EXPERIMENT, BY CLASS: 1 - INFORMATION 2 - WARNING 3 - SEVERE
NMSG	-	I2	TOTAL NUMBER OF MESSAGES OF ANY CLASS ISSUED DURING AN EXPERIMENT
MSGC	KMMSG	I2	MESSAGE NUMBERS ISSUED DURING EXPERIMENT.
MSGCN	KMMSG	I2	NUMBER OF REMAINING MESSAGES OF THIS TYPE ALLOWED PRIOR TO TERMINATION.
TERM	-	L1	INDICATOR TO SIGNAL TERMINATION DUE TO EXCESSIVE MESSAGES.
MFLAG	-	L1	ERROR PROCESSING IN PROGRESS INDICATOR TO HALT RECURSIVE ERROR PROCESSING.
MSGID	-	L1	ID OF PROCESSOR BEING EXECUTED (1 = DESM).

TABLE 3-1 (2 of 53) GLOBAL VARIABLES

NAME: ECICFG			CATEGORY: STATION CONFIGURATOR INPUT DATA																														
VARIABLE	DIM	TYPE	DESCRIPTION																														
SLCFIG	13,KMSL	R4	<p>SLCFIG CONTAINS 13 DESCRIPTORS FOR EACH LINK (L) IN THE STATION. COLUMN 1 IS SLCFIG(1,L) AND COLUMN 13 IS SLCFIG(13,L) IN THE FOLLOWING DEFINITIONS:</p> <p>COL 1 - STATION LINK TYPE 1 = IR = INPUT RAMP 2 = IQ = INPUT QUEUE 3 = D = DOCK 4 = OQ = OUTPUT QUEUE 5 = OR = OUTPUT RAMP 6 = S = STORAGE 7 = IS = INPUT TO STORAGE 8 = SI = STORAGE TO INPUT 9 = DS = DOCK TO STORAGE 10 = SO = STORAGE TO OUTPUT</p> <p>COL 2 - STATION LINK TRAVEL TIME, INCLUDING HEADWAY ZONE TRAVEL TIME (OPTIONAL; REQUIRED IF COL 3 AND SLVEL ARE NOT USED).</p> <p>COL 3 - STATION LINK LENGTH (OPTIONAL; REQUIRED WITH SLVEL IF COL 2 IS NOT USED).</p> <p>COL 4 - DEFAULT STATION LINK CAPACITY (MAY BE OVERRIDDEN BY SPECIFYING SLCAP(L,STATION))</p> <p>COL 5 - FIRST EVENT ON LINK. EVENTS MUST BE IN THE ORDER HEADWAY, TRAVEL, DEBOARD, BOARD, STORE, LAUNCH WHEN SPECIFIED. THE FOLLOWING NUMBERS REPRESENT EVENTS: 1 = HEADWAY 2 = TRAVEL 3 = DEBOARD 4 = BOARD 5 = STORE 6 = LAUNCH</p> <p>COL 6 - SECOND EVENT ON LINK COL 7 - THIRD EVENT ON LINK COL 8 - FOURTH EVENT ON LINK COL 9 - FIFTH EVENT ON LINK COL 10 - DIVERGE FUNCTION REQUEST (NONE IMPLEMENTED)</p> <p>COL 11 - UPSTREAM LINK ORDERING OPTION USED WHEN UPSTREAM VEHICLES ARE DEQUEUED IN PRIORITY ORDER (SLPF = 1) ORDER OF UPSTREAM LINKS</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th rowspan="2">LINK</th> <th rowspan="2">OPT</th> <th colspan="2">UPSTREAM LINKS</th> </tr> <tr> <th>FIRST</th> <th>SECOND</th> </tr> </thead> <tbody> <tr> <td>IQ</td> <td>1</td> <td>IR</td> <td>SI</td> </tr> <tr> <td>IQ</td> <td>2</td> <td>SI</td> <td>IR</td> </tr> <tr> <td>D(NO IQ)</td> <td>1</td> <td>IR</td> <td>SI</td> </tr> <tr> <td>D(NO IQ)</td> <td>2</td> <td>SI</td> <td>IR</td> </tr> <tr> <td>OR</td> <td>1</td> <td>OQ OR D</td> <td>SO</td> </tr> <tr> <td>OR</td> <td>2</td> <td>SO</td> <td>OQ OR D</td> </tr> </tbody> </table> <p>COL 12 - HEADWAY TIME PER TRAIN USED TO COMPUTE TIME TO TRAVEL THE HEADWAY ZONE. TOTAL HEADWAY ZONE TRAVEL TIME = (COL 13) * (TRAIN LENGTH) + (COL 12)</p>	LINK	OPT	UPSTREAM LINKS		FIRST	SECOND	IQ	1	IR	SI	IQ	2	SI	IR	D(NO IQ)	1	IR	SI	D(NO IQ)	2	SI	IR	OR	1	OQ OR D	SO	OR	2	SO	OQ OR D
LINK	OPT	UPSTREAM LINKS																															
		FIRST	SECOND																														
IQ	1	IR	SI																														
IQ	2	SI	IR																														
D(NO IQ)	1	IR	SI																														
D(NO IQ)	2	SI	IR																														
OR	1	OQ OR D	SO																														
OR	2	SO	OQ OR D																														

TABLE 3-1 (3 of 53) GLOBAL VARIABLES

NAME: ECICFG (CONT)			CATEGORY: STATION CONFIGURATOR INPUT DATA
VARIABLE	DIM	TYPE	DESCRIPTION
SLVEL	-	R4	COL 13 - HEADWAY TIME PER VEHICLE USED TO COMPUTE TIME TO TRAVEL THE HEADWAY ZONE. TOTAL HEADWAY ZONE TRAVEL TIME = (COL 13) * (TRAIN LENGTH) + (COL 12) AVERAGE STATION LINK VELOCITY (REQUIRED WITH COL 3 OF SLCFIG IF COL 2 OF SLCFIG IS NOT SPECIFIED. REQUIRED FOR ONLINE STATIONS).

NAME: ECIFEL			CATEGORY: FUTURE EVENTS TIMING CONTROL DATA
VARIABLE	DIM	TYPE	DESCRIPTION
Csize	-	I4	NUMBER OF CLOCK UNITS PER MINUTE.
CLOOP	-	I4	MAXIMUM NUMBER OF EVENTS IN ONE CLOCK UNIT.
CLSMAL	-	I4	SPACING BETWEEN CLOCK TABLE ENTRIES IN CLOCK UNITS * 10.
CLSIZE	-	I4	NUMBER OF INTERVALS COVERED BY CLOCK TABLE.

NAME: ECIGL			CATEGORY: GUIDEWAY LINK DATA
VARIABLE	DIM	TYPE	DESCRIPTION
GLVSD	-	R4	STANDARD DEVIATION OF VELOCITY.
GLRTIM	-	R4	REACTION TIME FOR ACCELERATING TO LINE SPEED FROM STOP.
GLMDLY	KMDLY1, KMDLY2	R4	MERGE DELAY TABLE FOR HEURISTIC MERGE POLICY. ELEMENT I, J CONTAINS THE VEHICLE DELAY ON A LINK WHEN THE TRAFFIC DENSITY ON THE LINK IS I / KMDLY1 AND THE TRAFFIC DENSITY ON THE COMPETING LINK IS J / KMDLY2.
GLOCC	KML	I2	LINK OCCUPANCY.

TABLE 3-1 (4 of 53) GLOBAL VARIABLES

NAME: ECIMAX			CATEGORY: RUN TIME PROBLEM SIZES
VARIABLE	DIM	TYPE	DESCRIPTION
KNSL	-	I2	STATION LINKS.
KNV	-	I2	NUMBER OF VEHICLES ACTIVE IN SIMULATION RUN.
KNFLT	-	I2	MAXIMUM FLEET SIZE.
KNT	-	I2	CONCURRENT TRIPS.
KNR	-	I2	ROUTES FOR SCHEDULED SERVICE.
KNR1	-	I2	ENTRIES IN ROUTE POINTER (PVRPTR)(KNR + 1).
KNCRS	-	I2	ENTRIES IN CIRCUITOUS EMPTY ROUTE TABLE (PECRTE).
KNANT	-	I2	ENTRIES IN EMPTY VEHICLE ANTICIPATED NEED LISTS (PANSTN, PANEEED, PANCD).
KNRT	-	I2	ENTRIES IN SCHEDULED ROUTE LIST (PVRLST).
KNEVP	-	I2	ENTRIES IN ORDERED EMPTY VEHICLE PRIORITY LIST OF WHERE TO SEND EMPTIES (PVEPR).
KNSVP	-	I2	ENTRIES IN ORDERED LIST OF WHERE TO SEARCH FOR EMPTIES (PVSPR).
KNLSLT	-	I2	MINIMUM PATH TABLES (PLSLT) IN SIMULATION CONCURRENTLY.
KNSLE	-	I2	ENTRIES IN STATION LINK EVENT LIST (SLEVL).
KNSLD	-	I2	ENTRIES IN STATION LINK DOWNSTREAM STATION LINK LIST (SLDSL).
KNSLU	-	I2	ENTRIES IN STATION LINK UPSTREAM STATION LINK LIST (SLUSL).
KNFAIL	-	I2	NUMBER OF FAILURE OR RECOVERY CARDS PROCESSED
KNTOW	-	I2	NUMBER OF LINKS IN FAILURE RESPONSE TOW PATH

TABLE 3-1 (5 of 53) GLOBAL VARIABLES

NAME: ECINET			CATEGORY: NETWORK STRUCTURED DATA
VARIABLE	DIM	TYPE	DESCRIPTION
GLVEL	KML	R4	LINE SPEED.
G.LHDWY	KML	R4	TIME SEPARATION BETWEEN VEHICLES AT LINE SPEED.
GLTTIM	KML	R4	LINK NOMINAL TRAVEL TIME.
GLLEN	KML	I4	LINK LENGTH.
PLSLT	KML, KMS, KMSLST	I4	MINIMUM PATH TABLE: $PLSLT(I,J) = A * 1000 + B$, WHERE: B = NEXT GUIDEWAY LINK TO ENTER ON MINIMUM PATH TO STATION J WHEN CURRENTLY ON GUIDEWAY LINK I (B = 0 ==> STATION J IS AT END OF LINK I) A = ENTRY NUMBER OF START OF LIST OF ALTERNATE ROUTE(S) (PALRTE) TO STATION J FROM GUIDEWAY LINK I (A = 0 ==> NO ALTERNATE ROUTES EXIST)
KNL	-	I2	NUMBER OF GUIDEWAY LINKS.
KNS	-	I2	NUMBER OF STATIONS.
KNS1	-	I2	NUMBER OF ENTRIES IN EMPTY VEHICLE NEED POINTER (PANPTR) (KNS + 1).
KNM	-	I2	NUMBER OF MERGES.
GLDVGN	KML	I2	DIVERGE ID AT END OF LINK.
GLMRGN	KML	I2	MERGE ID AT BEGINNING OF LINK.
GLENTY	KML,2	I2	UPSTREAM LINK POINTERS.
GMENTY	KMM	I2	POINTS AT LINKS WHICH BEGIN WITH MERGES.
GLSTN	KML	I2	STATION ID AT END OF LINK: 0==>NO STATION AT END OF LINK N==>STATION N AT END OF LINK
PLSLTN	-	I2	NUMBER OF CURRENTLY ACTIVE MINIMUM PATH TABLE (PLSLT).
PALRTE	KMALT	I2	ALTERNATE PATH ROUTE LIST. CONCATENATED LIST OF LISTS OF ALTERNATE PATHS TO DESTINATION STATIONS FROM COMMON DIVERGE POINTS.
PSSTN	KMS, KMS	I2	SUCCESSOR STATION TABLE: ELEMENT I,J IS NEXT STATION ON MINIMUM PATH FROM I TO J.
SILINK	KMS	I2	STATION PREDECESSOR GUIDEWAY LINK.
SELINK	KMS	I2	STATION SUCCESSOR GUIDEWAY LINK.
SNXDSS	KMS	I2	ID OF CLOSEST DOWNSTREAM STATION TO THIS STATION.
GLCAP	KML	I2	LINK CAPACITY.
GLBLK	-	I2	BLOCK LENGTH FOR FIXED HEADWAY REGULATION (METERS).
GLMRGC	KML	I2	ID OF LINK COMPETING FOR MERGE WITH GIVEN LINK.

TABLE 3-1 (6 of 53) GLOBAL VARIABLES

NAME: ECIPOI		CATEGORY: POLICY DATA	
VARIABLE	DIM	TYPE	DESCRIPTION
PSTWT	-	R4	WEIGHTING FACTOR FOR NOMINAL TRAVEL TIME.
PSUWT	-	R4	WEIGHTING FACTOR FOR LINK UTILIZATION.
PMDWT	-	R4	WEIGHTING FACTOR FOR MERGE DELAY.
PANCD	KMANT	R4	CUMULATIVE DISTRIBUTION FUNCTIONS OF ANTICIPATED NEED FOR EMPTY VEHICLE MANAGEMENT OPTION 3.
PMRGWW	-	R4	MERGE RESERVATION TABLE WINDOW WIDTH.
PMRGTT	-	R4	TOTAL AMOUNT OF TIME SPANNED BY MERGE RESERVATION TABLE.
PNXSLV	KMRT	I4	CONTAINS THE TIME AT WHICH THE NEXT VEHICLE ON THIS ROUTE SHOULD LEAVE FROM THIS STATION.
PLSTLV	KMRT	I4	CONTAINS THE TIME AT WHICH THE VEHICLE AHEAD OF THE VEHICLE NOW BEING SCHEDULED ON THIS ROUTE ACTUALLY DID LEAVE THIS STATION.
PNVDIS	KMRT	I4	NUMBER OF VEHICLES TO DISPATCH ON THIS ROUTE FROM THIS STATION (EQUIVALENCED WITH PLSTLV).
PRTEHW	KMR	I4	DESIRED HEADWAY BETWEEN VEHICLES ON THE SAME ROUTE FOR SCHEDULED SERVICE.
PWLKTS	-	I4	DEFAULT SYSTEM-WIDE TRANSFER WALK TIME PRIOR TO BOARDING QUEUE RE-ENTRY
PWALKT	KMS, KMS	I4	TRANSFER WALK TIME PRIOR TO BOARDING QUEUE ENTRY (BY DEBOARD/REBOARD STATION PAIR).
PALTET	-	I4	ALTERNATE STATION EGRESS TIME (APPLIED TO PASSENGERS DEBOARDING AT OTHER THAN THEIR DESIRED DESTINATION).
PHIST1	-	I4	FIRST THRESHOLD FOR EXCESS TRAVEL TIME HISTOGRAM.
PHIST2	-	I4	SECOND THRESHOLD FOR EXCESS TRAVEL TIME HISTOGRAM.
POLLC	-	I2	LONGITUDINAL CONTROL POLICY IN EFFECT: 1===>SYNCHRONOUS 2===>QUASI-SYNCHRONOUS 3===>ASYNCHRONOUS
POLDIS	-	I2	DISPATCH POLICY IN EFFECT: 1===>DETERMINISTIC 2===>QUASI-DETERMINISTIC 3===>NON-DETERMINISTIC
POLVPR	-	I2	VEHICLE POSITION REGULATION SCHEME IN EFFECT: 1===>FIXED BLOCK 2===>VARIABLE BLOCK 3===>USER SPECIFIED MODEL
POLSER	-	I2	THE SERVICE POLICY TO BE USED FOR THIS RUN: 1===>DEMAND RESPONSIVE SINGLE PARTY 2===>DEMAND RESPONSIVE MULTIPARTY 3===>SCHEDULED 4===>TIMEOUT/GROUP DEMAND RESPONSIVE
PVSPAC	-	I2	ALGORITHM SELECTION FOR SPACING BETWEEN VEHICLES ON THE SAME SCHEDULED ROUTE: 1===>FIXED SCHEDULE 2===>MIDPOINT SCHEDULE 3===>FIXED INTERVAL 4===>MIDPOINT INTERVAL 5===>IMMEDIATE
PVRLST	KMRT	I2	SCHEDULED ROUTE LIST - A CONCATENATION OF ALL THE SCHEDULED ROUTES; FIRST ENTRY OF EACH ROUTE POINTED TO BY PVRPTR.
PVRPTR	KMR1	I2	POINTER TO STARTING ENTRY FOR EACH ROUTE IN PVRLST: I < KMR1 ===> ENTRY NUMBER OF PVRLST OF FIRST STATION STOP ON ROUTE I I = KMR1 ===> (LENGTH OF PVRLST)+1 (= KNRT + 1)

TABLE 3-1 (7 of 53) GLOBAL VARIABLES

NAME: ECIPO (CONT)			CATEGORY: POLICY DATA
VARIABLE	DIM	TYPE	DESCRIPTION
PBERTH	-	I2	BERTH ASSIGNMENT POLICY: 1==>TO MOST DOWNSTREAM AVAILABLE BERTH AS SOON AS POSSIBLE 2==>FORM PLATOONS IF BERTH AREA NOT CLEAR, ASSIGN PLATOON WHEN BERTH AREA CLEARS 3==>RIPPLE BERTH ADVANCEMENT FOR TIMEOUT/ GROUP DEMAND RESPONSIVE
PNVRTE	KMR	I2	NUMBER OF VEHICLES ON EACH ROUTE IN SCHEDULED SERVICE.
PRTLEN	KMR	I2	TRAIN LENGTH ON EACH ROUTE IN SCHEDULED SERVICE.
PECRTE	KMCRS	I2	CONCATENATION OF LISTS OF STATIONS THAT FORM ROUTES ON WHICH TO CIRCULATE EMPTY VEHICLES.
PECPTR	KMCR	I2	POINTER TO STARTING ENTRY IN PECRTE: PECPTR(I) = ENTRY NUMBER OF PECRTE OF FIRST STATION STOP ON EMPTY VEHICLE CIRCULATION ROUTE I
PSMETH	-	I2	PATH SELECTION METHOD: 1==>APRICRI (IN THE STATION) 2==>REAL-TIME (AT DIVERGES) (USING ALTERNATE PATH DEFINITION PALRTE)
PSTYPE	-	I2	PATH SELECTION TYPE: 1==>TABLE LOOK-UP 2==>ALGORITHMIC (USING PSALGM)
PSALGM	-	I2	PATH SELECTION ALGORITHM TO BE USED: 1==>LINK NOMINAL TRAVEL TIME 2==>LINK LENGTH 3==>LINK UTILIZATION 4==>WEIGHTED COMBINATION OF 1 & 3
PVEPR	KMEVP	I2	ORDERED LIST OF WHERE TO PUT EMPTY VEHICLE: PVEPR(1)=FIRST PLACE TO TRY PVEPR(2)=SECOND PLACE TO TRY PVEPR(KMEVP)=LAST PLACE TO TRY 1==>LOCAL STORAGE 2==>REGIONAL STORAGE 3==>DISTRIBUTE ACCORDING TO ANTICIPATED NEED WITHOUT CONSIDERING THE CURRENT AVAILABILITY OF EMPTIES 4==>DISTRIBUTE ACCORDING TO ANTICIPATED NEED WHILE CONSIDERING THE CURRENT AVAILABILITY OF EMPTIES 5==>CIRCULATE ON THE GUIDEWAY - ON A PREDETERMINED ROUTE 6==>SEND TO STATION WITH MOST REQUESTS 7==>USER EMPTY VEHICLE ALGORITHM
PVSPR	KMSVP	I2	ORDERED LIST OF WHERE TO LOOK FOR EMPTY: PVSPR(1)=FIRST PLACE TO LOOK PVSPR(2)=SECOND PLACE TO LOOK PVSPR(KMSVP)=LAST PLACE TO LOOK 1==>A NON-CIRCUITOUS VEHICLE (OCCUPIED/ EMPTY) ABOUT TO ARRIVE/BYPASS THE STATION (I.E., ON THE STATION'S ARRIVAL LIST) 2==>LOOK IN STATION AT LINKS SPECIFIED IN PSLIST 3==>LOCAL STORAGE 4==>REGIONAL STORAGE 5==>AN EMPTY CIRCULATING ON THE GUIDEWAY (I.E., ON THE STATION'S ARRIVAL LIST)

TABLE 3-1 (8 of 53) GLOBAL VARIABLES

NAME: ECIPOL (CONT)			CATEGORY: POLICY DATA
VARIABLE	DIM	TYPE	DESCRIPTION
			6==>EARLIEST AVAILABLE VEHICLE CONSIDERING ALL SOURCES
			7==>ANY EXPECTED ARRIVAL (1 AND 5 COMBINED)
			8==>AN EMPTY FROM CLOSEST STATION STORAGE AREA.
PSLIST	KMSL	I2	LIST OF STATION LINKS WHERE EMPTY VEHICLE IS TO BE LOOKED FOR. USER ENTERS LINK TYPE, IP
PANPTR	KMSI	I2	TRANSLATES TO STATION LINK ID. POINTER TO START OF LIST OF STATIONS TO RECEIVE EMPTIES FROM THIS STATION BASED ON ANTICIPATED NEED (POINTER TO PANSTN, PANEED, PANCD).
PANEED	KMANT	I2	ANTICIPATED NEED (# VEHICLES) FOR EMPTY VEHICLE MANAGEMENT OPTIONS 3 AND 4.
PANSTN	KMANT	I2	LIST OF RECEIVING STATIONS CORRESPONDING TO PANEED AND PANCD.
PTSTN	KMS, KMS	I2	TRANSFER STATION TABLE: PTSTN(I,J) = THE INTERMEDIATE DESTINATION (IN TERMS OF A STATION NUMBER) OF A TRIP CURRENTLY AT STATION I AND WHOSE FINAL DESTINATION IS STATION J
PWSTN	KMS, KMS	I2	STATION TO WALK TO BEFORE REBOARDING A VEHICLE WHEN TRANSFERRING: PWSTN(I,J) = THE STATION TO WALK TO WHEN DEBOARDING AT STATION I AND THE FINAL DESTINATION IS STATION J (IF PWSTN(I,J) = 0, REBOARD WITHOUT AN INTERMEDIATE WALK)
PMXTRL	-	I2	MAXIMUM TRAIN LENGTH.
PTSPLT	-	I2	TRIP SPLIT SIZE; ANY TRIP OF SIZE N WILL BE SPLIT INTO: K TRIPS OF SIZE (PTSPLT) AND 1 TRIP OF SIZE L
PEVALM	-	I2	THE LENGTH OF A TIME INTERVAL SUCH THAT WHEN THE ETA OF A VEHICLE IN THE ARRIVAL LIST OF A STATION IS GREATER THAN THE CURRENT CLOCK PLUS THIS INTERVAL, THE VEHICLE WILL NOT BE CONSIDERED.
POLMRG	-	I2	MERGE POLICY INDICATOR: 1==>FIFO (OF QUEUE) 2==>MANUEVERS BASED ON DELAY TABLE GLMDLY 3==>PRIORITY (ASSUMES THE PRIORITY LINK IS THE FIRST SPECIFIED OF A PAIR OF LINKS IN GLENTY) 4==>FIRST ARRIVAL AT MERGE POINT
PRASGN	KMS, KMS	I2	STATION ROUTE ASSIGNMENT TABLE: ELEMENT I,J IDENTIFIES ROUTE TO USE FOR TRAVEL FROM I TO J. IF ELEMENT I,J IS GREATER THAN KNR, A GROUP OF ROUTES CAN SERVE THE TRIP (POINTS TO PRGPTR).
PRGPTR	KMR	I2	ROUTE GROUP POINTER. ENTRY N - KNR POINTS TO FIRST ROUTE IN LIST OF ROUTES (PRGLST) DEFINING A GROUP OF ROUTES THAT CAN SERVE A TRIP FROM I TO J. PRASGN(I,J) = N > KNR INVOKES USE OF PRGPTR.
PRGLST	KHGT	I2	ROUTE GROUP LIST. CONCATENATED LIST OF LISTS OF ROUTES COMPRISING GROUPS. ZEROS SEPARATE THE LISTS. FIRST ROUTE IN EACH LIST POINTED TO BY PRGPTR.
PNOMTH	KMS, KMS	I2	ELEMENT I,J CONTAINS THE NOMINAL TRAVEL TIME FROM STATION I TO STATION J.

TABLE 3-1 (9 of 53) GLOBAL VARIABLES

NAME: ECIPOL (CONT)			CATEGORY: POLICY DATA
VARIABLE	DIM	TYPE	DESCRIPTION
PARMAX	-	I2	MAXIMUM VEHICLE MANEUVER AT A MERGE (SLOTS).
PARTIM	-	I2	MAXIMUM VEHICLE MANEUVER AT A MERGE (C.U.).
PADVNC	-	I2	VEHICLE ADVANCE MANEUVER INDICATOR: 0===>NO ADVANCE PERMITTED N===>ADVANCE OF UP TO N SLOTS PERMITTED
PNVMRG	KMM	I2	NUMBER OF MERGE RESERVATIONS ALLOWED PER WINDOW FOR EACH MERGE.
PMRGL	2,2	I2	LOCAL MERGE PRIORITY TABLE FOR MERGE OF STATION EXIT RAMP WITH GUIDEWAY: ELEMENT(1,1)===>PRIORITY OF EMPTY VEHICLE ON GUIDEWAY ELEMENT(2,1)===>PRIORITY OF OCCUPIED VEHICLE ON GUIDEWAY ELEMENT(1,2)===>PRIORITY OF EMPTY VEHICLE IN STATION ELEMENT(2,2)===>PRIORITY OF OCCUPIED VEHICLE IN STATION
PNTRLM	-	I2	MAXIMUM WAIT TIME FOR ENTRAINMENT IN STATION.
SPLTOD	KMS, KMS	I2	TIMEOUT/GROUP DEMAND RESPONSIVE PLATFORM TYPE USED BY ORIGIN DESTINATION TRIPS. 1===>SOUTHBOUND A PLATFORM 2===>SOUTHBOUND B PLATFORM 3===>NORTHBOUND A PLATFORM 4===>NORTHBOUND B PLATFORM
PMXTIM	-	I2	TIMEOUT/GROUP DEMAND RESPONSIVE MAXIMUM PASSENGER WAIT TIME PRIOR TO VEHICLE REQUEST.
PMNGRP	-	I2	TIMEOUT/GROUP DEMAND RESPONSIVE MINIMUM PASSENGER GROUP SIZE PRIOR TO VEHICLE REQUEST.
PNTVEH	KMR	I2	NUMBER OF TRANSITION VEHICLES FOR SCHEDULED SERVICE ACTIVE FLEET SIZE MANAGEMENT.
PRBARN	KMR	I2	MAINTENANCE BARN ASSIGNED TO EACH ROUTE (CALCULATED BY INPUT PROCESSOR).
PRSTOP	KMR	I2	LAST STATION STOP ON ROUTE BEFORE VEHICLE SENT TO BARN FOR SCHEDULED SERVICE ACTIVE FLEET SIZE CHANGE (CALCULATED BY INPUT PROCESSOR).
PRNTRY	KMR	I2	POINTER TO FIRST STATION STOP (IN PVRLST) ON ROUTE AFTER RELAUNCH FROM BARN FOR SCHEDULED SERVICE ACTIVE FLEET SIZE MANAGEMENT (CALCULATED BY INPUT PROCESSOR).
PSADSP	-	I2	TIMEOUT/GROUP DEMAND RESPONSIVE STATION OVER-FULL PROTECTION ADEQUATE SPACE FOR VEHICLE STATION ENTRY.
PXFER	-	L1	TRANSFER POLICY: F===>NO TRANSFERS T===>TRANSFERS PERMITTED
PVDVRT	-	L1	VEHICLE DIVERSION FROM GUIDEWAY TO BOARD TRIPS WHEN STATION IS NOT THE VEHICLE'S DESTINATION: F===>NO DIVERSION T===>DIVERSION PERMITTED
PVRES	-	L1	RESERVATIONS REQUIRED INDICATOR (FOR DEMAND RESPONSIVE SERVICE): F===>NO RESERVATIONS T===>RESERVATIONS REQUIRED
PENTS	-	L1	STATIC (IN STATION) ENTRAINMENT INDICATOR: F===>NO STATIC ENTRAINMENT T===>STATIC ENTRAINMENT PERMITTED
PENTD	-	L1	DYNAMIC (ON GUIDEWAY) ENTRAINMENT/DETRAINMENT: F===>NO DYNAMIC ENTRAINMENT T===>DYNAMIC ENTRAINMENT PERMITTED
POLDMS	-	L1	DEMAND STOP INDICATOR (SCHEDULED SERVICE): F===>STOP AT EACH SCHEDULED STOP T===>STOP ONLY IF DEMAND EXISTS

TABLE 3-1 (10 of 53) GLOBAL VARIABLES

NAME: ECIPOL (CONT)		CATEGORY: POLICY DATA	
VARIABLE	DIM	TYPE	DESCRIPTION
PVBMP	-	L1	VEHICLE BUMPING INDICATOR (DEMAND RESPONSIVE SERVICE) F==>DO NOT BUMP EMPTY VEHICLE T==>BUMP EMPTY VEHICLE FROM STORAGE AS NEXT VEHICLE ENTERS STATION
PSDIRT	-	L1	SINGLE STOP INDICATOR FOR DEMAND RESPONSIVE MULTI-PARTY SERVICE F==>USUAL DEMAND RESPONSIVE MULTI-PARTY T==>ONLY SINGLE STOP DEMAND RESPONSIVE TOURS ARE PLANNED
HCBDLO	-	L1	DEBOARD/BOARD TIME LIMIT CHECK OVERRIDE F==>DO NOT OVERRIDE T==>OVERRIDE VALIDITY CHECK
HCPASS	-	L1	HANDICAPPED PASSENGER PROCESSING INDICATOR F==>DO NOT ACCEPT HANDICAPPED TRIPS T==>PROCESS HANDICAPPED TRIPS

TABLE 3-1 (11 of 53) GLOBAL VARIABLES

NAME: ECISL			CATEGORY: STATION LINK DATA
VARIABLE	DIM	TYPE	DESCRIPTION
SLPENT	KMSL, KMS	R4	PENALTY FACTOR TO BE APPLIED TO LINK TRAVERSAL TIME FOR DEGRADED STATION LINK.
SLTTIM	KMSL	R4	TRAVEL TIME ON STATION LINK, INCLUDING HEADWAY ZONE TRAVEL TIME.
SLHTA	KMSL	R4	HEADWAY ZONE TRAVEL TIME PER VEHICLE: TRAVEL TIME = SLHTA * (TRAIN LENGTH) + SLHTB
SLHTB	KMSL	R4	HEADWAY ZONE TRAVEL TIME PER TRAIN: TRAVEL TIME = SLHTA * (TRAIN LENGTH) + SLHTB
SLBFAC	-	R4	STATION LINK BYPASS FACTOR FOR ON-LINE STATIONS (RATIO OF STATION LINK VELOCITY TO STATION BYPASS SPEED).
SLCAP	KMSL, KMS	I2	STATION LINK CAPACITY (NUMBER OF VEHICLES).
SLOCC	KMSL, KMS	I2	STATION LINK OCCUPANCY (NUMBER OF VEHICLES).
SLTYPE	KMSL	I2	STATION LINK TYPE TO GROUP LINKS FOR REPORTING PURPOSES: 1===>INPUT RAMP 2===>INPUT QUEUE 3===>DOCK 4===>OUTPUT QUEUE 5===>OUTPUT RAMP 6===>STORAGE 7===>INPUT TO STORAGE 8===>STORAGE TO INPUT 9===>DOCK TO STORAGE 10===>STORAGE TO OUTPUT
SLEVP	KMSL	I2	POINTER TO STARTING ENTRY IN SLEVL FOR EACH STATION LINK.
SLEVL	KMSLE	I2	CONCATENATED LIST OF LISTS OF EVENTS FOR EACH STATION LINK. EVENTS ON A LINK MUST BE IN INCREASING NUMERICAL ORDER. 1===>TRAVEL HEADWAY ZONE 2===>TRAVEL STATION LINK 3===>DEBOARD 4===>BOARD 5===>STORE 6===>LAUNCH
SLUSP	KMSL	I2	POINTER TO STARTING ENTRY IN SLUSL FOR EACH STATION LINK.
SLUSL	KMSLU	I2	CONCATENATED LIST OF LISTS OF UPSTREAM STATION LINKS THAT FEED INTO EACH STATION LINK.
SLDSP	KMSL	I2	POINTER TO STARTING ENTRY IN SLDSL FOR EACH STATION LINK.
SLDSL	KMSLD	I2	CONCATENATED LIST OF LISTS OF DOWNSTREAM STATION LINKS THAT EACH STATION LINK FEEDS INTO.
SLDIVC	KMSL	I2	DIVERGE FUNCTION SELECTION FOR MULTIPLE DOWNSTREAM STATION LINKS (NONE ARE IMPLEMENTED).
SLPF	KMSL	I2	PRIORITY/FIFO INDICATOR FOR DEQUEUE FROM UPSTREAM STATION LINKS. IF PRIORITY, STATION LINKS MUST BE LISTED IN SLUSL IN ORDER OF PRIORITY. 0===>FIFO 1===>PRIORITY
SLDTYP	KMSL, KMS	I2	TIMEOUT/GROUP DEMAND RESPONSIVE STATION LINK DOCK TYPE (INPUT FOR DOCK LINKS ONLY) 1===>SOUTH TURNAROUND 2===>SOUTH SIDE CHANNEL 3===>NORTH TURNAROUND 4===>NORTH SIDE CHANNEL

TABLE 3-1 (12 of 53) GLOBAL VARIABLES

NAME: ECISL (CONT)		CATEGORY: STATION LINK DATA	
VARIABLE	DIM	TYPE	DESCRIPTION
SLPLAT	KMSL, KMS	I2	TIMEOUT/GROUP DEMAND RESPONSIVE DOCK LINK PLATFORM ASSIGNMENT TABLE (INPUT FOR DOCK LINKS ONLY) 1==>SOUTHBOUND A PLATFORM 2==>SOUTHBOUND B PLATFORM 3==>NORTHBOUND A PLATFORM 4==>NORTHBOUND B PLATFORM
SLAVAL	KMSL, KMS	L1	AVAILABILITY INDICATOR FOR EACH STATION LINK: F==>STATION LINK IS NOT AVAILABLE T==>STATION LINK IS AVAILABLE

TABLE 3-1 (13 of 53) GLOBAL VARIABLES

NAME: ECISTN		CATEGORY: STATION DATA	
VARIABLE	DIM	TYPE	DESCRIPTION
STDBA	-	R4	DEBOARD TIME PER PASSENGER.
STDHFF	-	R4	ESTIMATED DWELL ADJUSTMENT FACTOR FOR NOMINAL TRAVEL TIME.
STBA	-	R4	BOARD TIME PER PASSENGER.
SMNDBT	-	R4	MINIMUM DOOR OPEN TIME.
SMXDBT	-	R4	MAXIMUM BOARDING TIME LIMIT ASSUMING NO DEBOARDING PASSENGERS.
SHCBA	-	R4	HANDICAPPED PASSENGER BOARD DOOR TIME.
SHCDBA	-	R4	HANDICAPPED PASSENGER DEBOARD DOOR TIME.
SHCBB	-	R4	HANDICAPPED PASSENGER BOARD SECURE TIME.
SHCDBB	-	R4	HANDICAPPED PASSENGER DEBOARD RELEASE AND MOVE TO DOOR TIME.
STMOV1	-	R4	TIMEOUT/GROUP DEMAND RESPONSIVE RIPPLE BERTH ADVANCEMENT TIME IF MOVING.
STMOV2	-	R4	TIMEOUT/GROUP DEMAND RESPONSIVE RIPPLE BERTH ADVANCEMENT TIME IF STOPPED.
PSRCTO	KMS	I2	NUMBER OF THE STATION THAT ACTS AS THE REGIONAL CENTER TO WHICH THIS STATION SENDS EMPTIES WHEN SENDING THEM TO A REGIONAL CENTER.
PSRCFM	KMS	I2	NUMBER OF THE STATION THAT ACTS AS THE REGIONAL CENTER FROM WHICH THIS STATION GETS EMPTIES WHEN GETTING THEM FROM A REGIONAL CENTER.
PECRTN	KMS	I2	NUMBER OF THE EMPTY VEHICLE CIRCUITOUS ROUTE ONTO WHICH THIS STATION SENDS EMPTIES WHEN SENDING THEM ON A CIRCUITOUS ROUTE (POINTER TO PECPTR).
SBQCAP	KMS	I2	CAPACITY OF BOARDING QUEUE (NUMBER OF PASSENGERS) USED TO TURN AWAY ARRIVALS, IF EXCEEDED. MAY BE VIOLATED WITH TRANSFERS ENTERING BOARDING QUEUE, SINCE THEY CANNOT BE TURNED AWAY.
SLIR	-	I2	NUMBER OF THE STATION LINK ACTING AS THE INPUT RAMP IN ALL STATIONS.
SLOR	-	I2	NUMBER OF THE STATION LINK ACTING AS THE OUTPUT RAMP IN ALL STATIONS.
SLSTOR	-	I2	NUMBER OF THE STATION LINK ACTING AS THE STORAGE LINK IN ALL STATIONS.
SEMTIM	-	I2	STATION LINK TRAVEL TIME FROM LAUNCH TO EXIT RAMP MERGE WITH GUIDEWAY.
SEMSLT	-	I2	NUMBER OF SLOTS FROM LAUNCH TO EXIT RAMP MERGE WITH GUIDEWAY.
SMXDBP	-	I2	MAXIMUM BOARDING PASSENGERS LIMIT ASSUMING NO DEBOARDING PASSENGERS.
SNCAP1	KMS	I2	TOTAL STATION CAPACITY.
SHCPA	-	I2	HANDICAPPED PASSENGER BOARD DOOR TIME ORDINARY PASSENGER EQUIVALENT COUNT.
SHCPB	-	I2	HANDICAPPED PASSENGER BOARD SECURE TIME ORDINARY PASSENGER EQUIVALENT COUNT.
SMNINV	KMS, 4	I2	TIMEOUT/GROUP DEMAND RESPONSIVE MINIMUM INVENTORY GOAL FOR EACH DIRECTIONAL PLATFORM. 1===>SOUTHBOUND A PLATFORM 2===>SOUTHBOUND B PLATFORM 3===>NORTHBOUND A PLATFORM 4===>NORTHBOUND B PLATFORM
STYPE	KMS	L1	TYPE OF STATION: F===>OFF-LINE T===>ON-LINE

TABLE 3-1 (14 of 53) GLOBAL VARIABLES

NAME: ECISTN (CONT)		CATEGORY: STATION DATA	
VARIABLE	DIM	TYPE	DESCRIPTION
SBARN	KMS	L1	STATION MAINTENANCE BARN INDICATOR: F==>STATION NOT A BARN T==>STATION IS A BARN (MUST HAVE CONNECTED STORAGE)
STNGFL	KMS	L1	SWITCHBACK STATION INDICATOR: F==>NOT A SWITCHBACK STATION T==>STATION IS A SWITCHBACK STATION

TABLE 3-1 (15 of 53) GLOBAL VARIABLES

NAME: ECISYS			CATEGORY: SIMULATION SYSTEM DATA			
VARIABLE	DIM	TYPE	DESCRIPTION			
AFMDEG	KMFAIL	R4	TOW VEHICLE SPEED DEGRADATION FACTOR.			
AKSEED	-	I4	SEED FOR RANDOM NUMBER GENERATOR. MUST BE AN ODD INTEGER GREATER THAN OR EQUAL TO 3.			
ASAMPI	-	I4	SAMPLING INTERVAL.			
ACKPTI	-	I4	PERIODIC CHECKPOINT INTERVAL (0 = NO PERIODIC CHECKPOINTING).			
ATREAD	-	I4	TIME TO BEGIN READING TRIP RECORDS.			
AFAIL	10, KMFAIL	I4	FAILURE REQUEST TO MODEL PROCESSOR:			
			INDEX VALUE	GUIDEWAY	STATION	VEHICLE
			1	0	GUIDEWAY	VEHICLE
			2	0	LINK ID	STN ID ENTIRE STN LINK ID
			3	0	ENTIRE LINK	STN LINK#
				1	LINK ENTRY	LINK ENTRY
				2	LINK EXIT	LINK EXIT
			4	1	FAILURE	DEGRADED
				2	RECOVERY	DEGRADED
				3	-	REMOVED
				4	-	RECOVERY
			5	N	DELAY TO DETECTION	DELAY TO DETECTION
			6	1	-	RESTART
				2	-	PUSH COUPLE
				3	-	TOW VEHICLE
			7	N	-	DELAY TO RESTART
			8	N	-	DELAY TO REPLACEMENT
			9	N	MIN PATH TABLE #	MIN PATH TABLE #
			10	N	REFERENCE TO FAILURE	REFERENCE TO FAILURE
AVLOG	-	I2	VEHICLE LOG REQUEST: 0===>NO LOG REQUIRED N===>LOG VEHICLES AS THEY ARRIVE AT DIVERGE TO STATION N			
ASTATU	-	I2	NUMBER OF SAMPLING INTERVALS BETWEEN SNAPSHOT OUTPUTS.			
APCOM1	-	I2	TIME INTERVAL FOR PERIODIC COMPUTATION OF MERGE DELAY FACTORS FOR HEURISTIC MERGE POLICY (POLMRG = 2).			
AFMTOW	KMI, KIFAIL	I2	TOW VEHICLE PATH LINK SEQUENCE.			
AFMRSP	K'R, KMFAIL	I2	OTHER VEHICLE RESPONSE CHOICE (BY ROUTE, USE ROUTE = 1 IF NOT SCHEDULED SERVICE). 1===>CONTINUE REVENUE SERVICE 2===>DEBOARD ONLY MODE OF REVENUE SERVICE 3===>EMPTY CIRCULATION MODE, DEBOARD ALL PASSENGERS AT FIRST STATION STOP 4===>WAIT IN STATION PRIOR TO BOARD EVENT MODE			
ATRPLG	-	L1	TRIP LOG REQUEST: F===>NO TRIP LOG REQUIRED T===>WRITE LOG OF COMPLETED TRIPS			
AFLAG	KNFLAG	L1	AUXILIARY OUTPUT REQUEST FLAGS: AFLAG(I) = F===>NO AUXILIARY OUTPUT AFLAG(I) = T===>WRITE AUXILIARY OUTPUT FOR REQUEST I			

TABLE 3-1 (16 of 53) GLOBAL VARIABLES

NAME: ECISYS (CONT)			CATEGORY: SIMULATION SYSTEM DATA
VARIABLE	DIM	TYPE	DESCRIPTION
ALLOG	-	L1	LINK STATISTICS LOG INDICATOR: F==>NO LOG REQUESTED T==>LOG LINK STATS AT SAMPLING INTERVAL
ASLOG	-	L1	STATION STATISTICS LOG INDICATOR: F==>NO LOG REQUESTED T==>LOG STATION STATS AT SAMPLING INTERVAL

NAME: ECIVEH			CATEGORY: VEHICLE DATA
VARIABLE	DIM	TYPE	DESCRIPTION
VDFACT	KMFAIL	R4	DEGRADATION FACTOR FOR VEHICLE DEGRADATION.
VLEN	-	I2	VEHICLE LENGTH.
VCAP	-	I2	TOTAL VEHICLE CAPACITY.
VSEAT	-	I2	NUMBER OF SEATS ON A VEHICLE.
VHCAP	-	I2	VEHICLE HANDICAPPED PASSENGER CAPACITY.

NAME: ECMFEL			CATEGORY: MP FUTURE EVENTS TIMING CONTROL DATA
VARIABLE	DIM	TYPE	DESCRIPTION
CLSTAT	3, 10	R4	FUTURE EVENTS TIMING STATISTICS (I,J): I = 1, NUMBER OF ENTRIES 2, SUM OF ENTRIES 3, SUM OF SQUARES OF ENTRIES J = 1, CLOCK TABLE INTERVAL 1 2, CLOCK TABLE INTERVAL 2 3, CLOCK TABLE INTERVAL 3 4, CLOCK TABLE INTERVAL 4 5, CLOCK TABLE INTERVAL 5 6, CLOCK TABLE INTERVAL 6 7, CLOCK TABLE INTERVAL 7 8, CLOCK TABLE INTERVAL 8 9, CLOCK TABLE INTERVAL 9 10, GREATER THAN 9 CLOCK TABLE INTERVALS
CLPOS	-	I4	CURRENT POSITION IN CLOCK TABLE.
CLBASE	-	I4	BASE TIME VALUE FOR FIRST ENTRY IN CLOCK TABLE (C.U. * 10).
CLMINI	-	I4	TIME (C.U. * 10) OF CURRENT CLOCK TABLE INTERVAL GIVEN BY CLPOS: (CLMINI = CLBASE + CLSMAL * (CLPOS - 1)).
CLBIG	-	I4	TIME (C.U. * 10) OF CURRENT CLOCK TABLE: (CLBIG = CLSIZE * CLMINI).
CNFEL	-	I4	NUMBER OF ENTRIES IN FUTURE EVENTS LIST (CLOCK TABLE AND MULTIPLE THREAD LIST)
CLOCK	-	I4	SIMULATION CLOCK.
CLTABL	KMCLTA	I2	CLOCK TABLE; LIST HEAD POINTER TO TRANSACTIONS ACTIVE IN CLOCK INTERVAL CLMINI.
CMTHRD	-	I2	MULTIPLE THREAD LIST POINTER.
CLSCAN	-	L1	FLAG FOR USE IN SCANNING CLOCK TABLE WHEN RESCAN OF CLOCK TABLE REQUIRED UPON REACHING END (POS < CLSIZE).

TABLE 3-1 (17 of 53) GLOBAL VARIABLES

NAME: ECMGL			CATEGORY: MP GUIDEWAY LINK DATA
VARIABLE	DIM	TYPE	DESCRIPTION
GTIME	KML	I4	CLOCK TIME AT LAST VEHICLE ENTRY ONTO TRAVEL SEGMENT.
GLEQHD	KML	I4	POINTER TO HEAD OF LINK EXIT QUEUE.
GEXTR1	KML	I4	SPARE INTEGER*4 VARIABLE.
GLVID	KML	I2	ID OF LAST VEHICLE TO ENTER LINK.
GLQOCC	KML	I2	LINK QUEUE OCCUPANCY: VARIABLE HEADWAY: NUMBER OF VEHICLES IN QUEUE FIXED HEADWAY: NUMBER OF FIXED BLOCKS REQUIRED FOR THE QUEUED VEHICLES
GTOTVP	KML	I2	ACTUAL NUMBER OF VEHICLE POSITIONS TO BE OCCUPIED BY THE LAST VEHICLE ENTERING THE LINK.
GLENT	KML	I2	INDICATOR FOR LINK ENTRY BLOCKAGE: 0=== NO BLOCKAGE 1-99=== LINK FAILURE 100+=== VEHICLE FAILURE
GLEXIT	KML	I2	INDICATOR FOR LINK EXIT BLOCKAGE OR PRIORITY PASSAGE: 0=== NO BLOCKAGE 1-9=== LINK FAILURE 10-99=== LINK BLOCKED FOR TOW PATH 100+=== VEHICLE FAILURE
GVEXIT	KML	I2	NEXT VEHICLE TO EXIT LINK.
GMDLYF	KML	I2	MERGE DELAY FACTOR FOR TRAVEL TIME (HEURISTIC MERGE ALGORITHM).
GEXTR2	KML	I2	SPARE INTEGER*2 VARIABLE.
GLHZF	KML	L1	LOGIC VARIABLE FOR MAINTAINING HEADWAY ON LINK ENTRY.
GLMPR	KMLS	L1	PRIORITY FLAG FOR MERGING (SET WHEN MAXIMUM RETARD REACHED IN QUASI-SYNCHRONOUS CONTROL).
GEXTR3	KML	L1	SPARE LOGICAL*1 VARIABLE.

TABLE 3-1 (18 of 53) GLOBAL VARIABLES

NAME: ECMPOL		CATEGORY: MP POLICY DATA	
VARIABLE	DIM	TYPE	DESCRIPTION
PVPRMC	-	R4	PRIMARY PATH COST.
PVALTC	-	R4	ALTERNATE PATH COST.
PEXTRI	-	R4	SPARE REAL*4 VARIABLE.
PMBASE	-	I4	BASE TIME VALUE FOR MERGE RESERVATION TABLE.
PMTIME	KMM, 2	I4	TIME TO REACH MERGES.
KNRX	-	I4	NUMBER OF ROUTE GROUPS.
PMNTRY	-	I2	CURRENT ENTRY NUMBER IN RESERVATION TABLE.
PMLIST	KMM, 2	I2	MERGE ID'S ON VEHICLE PATHS (1 = PRIMARY, 2 = ALTERNATE).
PMDELY	2	I2	MERGE DELAY FOR COMPETING PATHS (1 = PRIMARY, 2 = ALTERNATE).
PETATM	KMS, KMS	I2	ESTIMATED TIME OF ARRIVAL TABLE.
PEXTR3	-	I2	SPARE INTEGER*2 VARIABLE.
ANVRTE	KMR	I2	PREVIOUS NUMBER OF VEHICLES ON ROUTE.
ARTLEN	KMR	I2	PREVIOUS TRAIN CONSIST ON ROUTE.
PCMPAT	-	L1	LOGIC VARIABLE TO INDICATE VEHICLE/TRIP COMPATIBILITY.
SVTERM	-	L1	LOGIC VARIABLE TO INDICATE A VEHICLE MUST BE REMOVED FROM SERVICE AFTER DEBOARDING.
PMRGR	KMM2, KMWMAX	L1	MERGE RESERVATION TABLE.

NAME: ECMSL		CATEGORY: MP STATION LINK DATA	
VARIABLE	DIM	TYPE	DESCRIPTION
SLMENT	KMSL, KMS	I2	POINTER TO THE TAIL OF THE MEMBERSHIP CHAIN OF THIS STATION LINK.
SLPOCC	KMSL, KMS	I2	PSEUDO-OCCUPANCY; MAINTAINED ONLY FOR STATION LINKS WITH DEBOARD/BOARD EVENTS AND EQUAL TO THE CAPACITY MINUS THE NUMBER OF AVAILABLE UPSTREAM BERTHS.
SLEXIT	KMSL, KMS	I2	INDICATES WHETHER EXIT OF STATION LINK IS FAILED OR ACTIVE. 0===>ACTIVE 1-9===>FAILED 10+===>BLOCKED FOR TOW PATH
SLSPD	KMSL, KMS	I2	INDICATES WHETHER PRECEDING VEHICLE IS TRAVELING AT BYPASS SPEED OR STATION LINK SPEED. 0===>STATION LINK SPEED 1===>BYPASS SPEED
SLHZF	KMSL, KMS	L1	HEADWAY ZONE FLAG FOR THIS STATION LINK: F===>NOT OCCUPIED T===>OCCUPIED
SLENT	KMSL, KMS	L1	INDICATES WHETHER ENTRY OF STATION LINK IS FAILED OR ACTIVE: F===>ACTIVE T===>FAILED

TABLE 3-1 (19 of 53) GLOBAL VARIABLES

NAME: ECMSTN			CATEGORY: MP STATION DATA
VARIABLE	DIM	TYPE	DESCRIPTION
SVBRDL	KMS	I4	POINTER TO TAIL OF BOARDING LIST.
SLSI	-	I4	STATION LINK ACTING AS THE STORAGE TO INPUT LINK.
SLSO	-	I4	STATION LINK ACTING AS THE STORAGE TO OUTPUT LINK.
SLIS	-	I4	STATION LINK ACTING AS THE INPUT TO STORAGE LINK.
SLDS	-	I4	STATION LINK ACTING AS THE DOCK TO STORAGE LINK.
SEXTR1	KMS	I4	SPARE INTEGER*4 VARIABLE.
SVREQ	KMS	I2	NUMBER OF TRIPS THAT HAVE NOT BEEN ABLE TO MAKE A RESERVATION (DECREMENTED BY TRIP BOARDING OR EMPTY VEHICLE DIVERSION INTO STATION).
SBQTL	KMS	I2	POINTER TO TAIL OF CHAIN OF TRIPS IN BOARDING QUEUE.
SBQOCC	KMS	I2	ACTUAL NUMBER OF PASSENGERS IN BOARDING QUEUE.
SVATL	KMS	I2	LIST OF ARRIVING VEHICLES.
SDBQTL	KMS	I2	POINTER TO TAIL OF CHAIN OF TRIPS IN DEBOARD QUEUE WHERE TRIPS THAT HAVE BEEN SPLIT INTO MULTIPLE TRIPS WAIT TO BE REUNITED.
STOTP	-	I2	THE TOTAL NUMBER OF PASSENGERS TO BOARD OR DEBOARD.
SEXTR2	KMS	I2	SPARE INTEGER*2 VARIABLE.
SLLAST	-	I2	LAST EVENT IN STATION BEFORE EXIT: 2==>TRAVEL 6==>LAUNCH
SSCREQ	KMS,2	I2	NUMBER OF OUTSTANDING SIDE CHANNEL REQUESTS FOR EACH DIRECTIONAL STATION.
STNUVC	KMS,4	I2	UNASSIGNED VEHICLE COUNT FOR EACH DIRECTIONAL PLATFORM.
STNSOW	KMS,2	I2	STATION-ON-WAY COUNT FOR EACH DIRECTIONAL STATION.
STNDPB	KMS,4	I2	DIRECTIONAL PLATFORM BACKLOG - NUMBER OF VEHICLES REQUIRED TO SERVE DISPATCH REQUESTS.
STNPIR	KMS,4	I2	DIRECTIONAL PLATFORM INVENTORY REQUIREMENT.
STNSIR	KMS,2	I2	DIRECTIONAL STATION INVENTORY REQUIREMENT.
STNSIS	KMS,2	I2	DIRECTIONAL STATION INVENTORY SURPLUS.
STRIDL	KMS	I2	POINTER TO LIST OF TIMEOUT/GROUP DEMAND RESPONSIVE TRANSACTIONS.
STRIDL	KMS	I2	POINTER TO LIST OF TIMEOUT/GROUP DEMAND RESPONSIVE READY TRANSACTIONS.

TABLE 3-1 (20 of 53) GLOBAL VARIABLES

NAME: ECMSYS			CATEGORY: MP SIMULATION SYSTEM DATA
VARIABLE	DIM	TYPE	DESCRIPTION
ARANDN	-	R4	RANDOM NUMBER FROM A UNIFORM DISTRIBUTION BETWEEN 0 AND 1.
NAME	-	I4	NAME OF MOST RECENTLY READ ASYNCHRONOUS INPUT HEADER CARD.
AASYNC	17	I4	CONTAINS THE CONTENTS OF COLUMNS 11 - 78 OF THE MOST RECENTLY READ DATA HEADER CARD IN CHARACTER FORM.
SYXTR1	-	I4	SPARE INTEGER*4 VARIABLE.
AVNXSL	-	I2	NEXT STATION LINK THE VEHICLE BEING PROCESSED IS TO USE.
ATREC	-	I2	NUMBER OF TRIP RECORDS READ.
ASREC	-	I2	NUMBER OF SAMPLING RECORDS WRITTEN.
AVREC	-	I2	NUMBER OF VEHICLE LOG RECORDS WRITTEN.
ATLREC	-	I2	NUMBER OF COMPLETED TRIP LOG RECORDS WRITTEN.
ACARD	-	I2	NUMBER OF ASYNCHRONOUS HEADER CARDS READ.
ASLLST	KMSLDS	I2	LIST OF DOWNSTREAM STATION LINKS PASSED BETWEEN ESDIVF AND ESTEST.
AKNVA	-	I2	ACTUAL NUMBER OF VEHICLES CURRENTLY IN THE SIMULATION.
AFMVID	KMFAIL	I2	VEHICLE ID OF DEGRADED VEHICLE.
SYXTR2	-	I2	SPARE INTEGER*2 VARIABLE.
AENTRG	-	L1	LOGIC VARIABLE FOR INDICATING NEXT LINK (GUIDEWAY OR STATION) AVAILABLE FOR ENTRY.
ADONEG	-	L1	LOGIC VARIABLE FOR INDICATING LINK TRAVERSAL COMPLETED.
ADONES	-	L1	RETURNED BY ESLMDL AND USED TO INDICATE: F==>NOT DONE; THERE ARE MORE EVENTS TO OCCUR TO VEHICLE ON ITS CURRENT STATION LINK T==>DONE WITH ALL EVENTS FOR VEHICLE ON ITS CURRENT STATION LINK
AENTRS	-	L1	USED TO INDICATE: F==>CANNOT ENTER T==>CAN ENTER
AERROR	-	L1	LOGIC VARIABLE TO INDICATE ERROR OCCURRENCE.
AZSCHD	-	L1	LOGIC VARIABLE TO INDICATE THAT A ZERO TIME EVENT SCHEDULE WAS PERFORMED.
AEND	-	L1	LOGIC VARIABLE TO INDICATE THAT THE SIMULATION IS TO END.
APRMT	-	L1	ZERO TIME SCHEDULE FLAG: F==>DO NOT PERFORM SCHEDULE T==>PERFORM SCHEDULE
VALIST	-	L1	LOGIC VARIABLE INDICATING THAT ARRIVAL LIST RECORDING IS REQUIRED.
AMNAME	11, 8	L1	MEMBER NAMES OF FILES USED FOR THIS EXPERIMENT (11 NAMES, EACH CONSISTING OF 8 CHARACTERS)
AREST	-	L1	LOGIC VARIABLE INDICATING WHETHER RESTART WAS PERFORMED.
ASAMPL	-	L1	LOGIC VARIABLE INDICATING WHETHER SAMPLING WAS PERFORMED.
ACHKPL	-	L1	LOGIC VARIABLE INDICATING WHETHER CHECKPOINT WAS PERFORMED.
ATLOGL	-	L1	LOGIC VARIABLE INDICATING WHETHER A COMPLETED TRIPS LOG WAS WRITTEN.
AVLOGL	-	L1	LOGIC VARIABLE INDICATING WHETHER A VEHICLE LOG WAS WRITTEN.
ALLOGL	-	L1	LOGIC VARIABLE INDICATING WHETHER A LINK STATISTICS LOG WAS WRITTEN.
ASLOGL	-	L1	LOGIC VARIABLE INDICATING WHETHER A STATION STATISTICS LOG WAS WRITTEN.
AFMACT	KMFAIL	L1	FAILURE CONDITION ACTIVE FLAG
AFMCLR	KMFAIL	L1	TOW VEHICLE PATH CLEAR FLAG

TABLE 3-1 (21 of 53) GLOBAL VARIABLES

NAME: ECMTRP		CATEGORY: MP TRIP DATA	
VARIABLE	DIM	TYPE	DESCRIPTION
TARRT	KMT	I4	ARRIVAL TIME OF THE TRIP.
TASTNT	KMT	I4	STATION ENTRY TIME (SET NEGATIVE AFTER DISPATCH)
TEXTR1	KMT	I4	SPARE INTEGER*4 VARIABLE.
TORIG	KMT	I2	ORIGIN STATION OF TRIP.
TDEST	KMT	I2	DESTINATION STATION OF TRIP.
TPASS	KMT	I2	NUMBER OF PASSENGERS ON THE TRIP.
TIDEST	KMT	I2	INTERMEDIATE DESTINATION OF THE TRIP.
TMEMCH	KMT	I2	USED TO CHAIN TRIPS THAT ARE MEMBERS OF A BOARDING QUEUE.
TGRPCH	KMT	I2	TRIP GROUP CHAIN WORD; USED TO CHAIN TRIPS TOGETHER THAT WERE SPLIT APART AT THEIR ORIGIN STATION.
TRESCH	KMT	I2	TRIP RESERVATION CHAIN WORD; USED TO CHAIN TRIPS TO THE VEHICLE THEY HAVE RESERVED.
TROUTE	KMT	I2	ROUTE OR ROUTE GROUP TO WHICH THE TRIP IS ASSIGNED.
TADEST	KMT	I2	ORIGINAL DESTINATION OF TRIP WHEN AN ALTERNATE DESTINATION ASSIGNED.
TSTOP	KMT	I2	NUMBER OF INTERMEDIATE PASSENGER STOPS.
THCFL	KMT	I2	HANDICAPPED TRIP INDICATOR: 0==>TRIP NOT HANDICAPPED N==>TRIP IS HANDICAPPED
TTRNID	KMT	I2	GROUP TRANSACTION ID FOR TIMEOUT/GROUP DEMAND RESPONSIVE SERVICE.
TREQ	KMT	L1	SPARE LOGICAL*1 VARIABLE.

TABLE 3-1 (22 of 53) GLOBAL VARIABLES

NAME: ECMVEH			CATEGORY: MP VEHICLE DATA
VARIABLE	DIM	TYPE	DESCRIPTION
VPENT	KMV	R4	DEGRADED SPEED FACTOR TO BE APPLIED TO NORMAL LINK TRAVEL TIME FOR DEGRADED VEHICLE FAILURE. SPARE REAL*4 VARIABLE.
VEXTRI	KMV	R4	MAXIMUM BOARD DOOR OPEN TIME FOR A VEHICLE.
STBDMX	KMV	R4	MINIMUM BOARD DOOR OPEN TIME FOR A VEHICLE.
STBDMN	KMV	R4	OVERLAP BETWEEN VEHICLE BOARD AND DEBOARD OF LATEST DEBOARDING VEHICLE OF TRAIN.
STBDOV	KMV	R4	OVERLAP BETWEEN BOARD OF EARLIEST BOARDING VEHICLE AND DEBOARD OF LATEST DEBOARDING VEHICLE IN A TRAIN.
STBOVX	KMV	R4	DIFFERENCE BETWEEN LINK ENTRY TIME OF CURRENT AND PREVIOUS VEHICLE.
VSEP	KMV	I4	ESTIMATED ARRIVAL TIME OF VEHICLE AT RESERVED STATION.
VETA	KMV	I4	ENTRY TIME OF VEHICLE INTO HEADWAY ZONE.
VENTRY	KMV	I4	LINK EXIT TIME; STATION HEADWAY TIME.
VEXIT	KMV	I4	TRAVEL TIME ALONG THE ALTERNATE PATH.
VALPT	KMV	I4	USED TO CHAIN ALL VEHICLES ON A STATION LINK IN THE ORDER THEY ARRIVED (TO FORM A MEMBERSHIP CHAIN)
VMEMCH	KMV	I2	TRAIN LENGTH OF ENTRAINED GROUP: >0===>LEAD VEHICLE OF TRAIN =0===>SINGLE VEHICLE <0===>INTERMEDIATE ENTRAINED VEHICLES (NUMBER OF VEHICLES THAT CAN BE ADDED)
VTRLEN	KMV	I2	CHAIN WORD FOR MAINTAINING VEHICLE ENTRAINMENT: =0===>NOT ENTRAINED >0===>IS ENTRAINED (LAST VEHICLE IN TRAIN POINTS TO LEAD VEHICLE)
VTRNCH	KMV	I2	VEHICLE RETRY STATUS AND REASON: 0===>NO RETRY REQUIRED 1===>VEHICLE QUEUED BECAUSE OF STATION LINK FAILURE 2===>WAITING FOR PRECEDING VEHICLE 3===>EVENT RETRY INDICATOR 4===>VEHICLE IS IN STORAGE 5===>TIMEOUT/GROUP DEMAND RESPONSIVE VEHICLE AT LOAD BERTH READY FOR ASSIGNMENT 6===>TIMEOUT/GROUP DEMAND RESPONSIVE VEHICLE WAITING FOR RIPPLE BERTH ADVANCEMENT
VQREAS	KMV	I2	VEHICLE ID OF FOLLOWING VEHICLE (FOR LEAD VEHICLE OF TRAIN, VFPTR = ID OF LAST VEHICLE IN TR.
VFPTR	KMV	I2	VEHICLE ID OF LEADING VEHICLE.
VLPTR	KMV	I2	DISTANCE TRAVELED ON CURRENT LINK.
VDIST	KMV	I2	LINK ID OF NEXT GUIDEWAY LINK OR STATION LINK TO BE TRAVERSED.
VNCURR	KMV	I2	CURRENT OCCUPANCY OF THE VEHICLE.
VNPASS	KMV	I2	ARRIVAL LIST CHAIN WORD.
VSALCH	KMV	I2	THE ID OF THE STATION IN WHOSE ARRIVAL LIST THIS VEHICLE IS RECORDED.
VRESTN	KMV	I2	NUMBER OF RESERVED SEATS ON THE VEHICLE.
VRPASS	KMV	I2	POINTER TO TAIL OF CHAIN OF TRIPS RESERVED ON THE VEHICLE. FOR TIMEOUT/GROUP DEMAND RESPONSIVE, TRANSACTION NUMBER OF VEHICLE REQUEST ASSIGNED TO VEHICLE.
VRESTL	KMV	I2	VEHICLE'S TRIP QUEUE; POINTER TO CHAIN OF ONBOARD TRIPS.
VTRIPQ	KMV	I2	POINTER TO TAIL OF CHAIN OF TRIPS THAT ARE ABOUT TO BOARD THE VEHICLE.
VBLTL	KMV	I2	

TABLE 3-1 (23 of 53) GLOBAL VARIABLES

NAME: ECMVEH (CONT)			CATEGORY: MP VEHICLE DATA
VARIABLE	DIM	TYPE	DESCRIPTION
VDBLTL	KMV	I2	POINTER TO TAIL OF CHAIN OF TRIPS THAT ARE ABOUT TO DEBOARD THE VEHICLE AND LEAVE THE STATION.
VDXLTL	KMV	I2	POINTER TO TAIL OF CHAIN OF TRIPS THAT ARE ABOUT TO DEBOARD THE VEHICLE AND TRANSFER.
VNXSTN	KMV	I2	NEXT STOP OF THE VEHICLE, A STATION ID.
VONXST	KMV	I2	THE NUMBER OF THE STATION INTO WHICH THE VEHICLE WAS GOING WHEN IT WAS REDIRECTED TO ITS RESERVED STATION.
VDIVST	KMV	I2	INDICATES IF THE VEHICLE IS TO DIVERT TO STORAGE AT THE NEXT AVAILABLE OPPORTUNITY.
VRROUTE	KMV	I2	FOR DEMAND RESPONSIVE SERVICE: THE NUMBER OF THE CIRCUITOUS EMPTY VEHICLE ROUTE TO WHICH THE VEHICLE HAS BEEN ASSIGNED (POINTER TO PECPTR).
			FOR SCHEDULED SERVICE: THE NUMBER OF THE SCHEDULED ROUTE TO WHICH THE VEHICLE HAS BEEN ASSIGNED (POINTER TO PVRPTR).
VRTPOS	KMV	I2	CURRENT POSITION WITHIN THE SCHEDULED ROUTE LIST FOR THE ROUTE THIS VEHICLE IS ON.
VAPATH	KMV	I2	VEHICLE ENTRY NUMBER IN ALTERNATE PATH TABLE FOR CURRENT LINK.
VPPATH	KMV	I2	VEHICLE PRIMARY PATH POINTER; INSERTED AT LAUNCH TIME AS THE NUMBER OF THE CURRENTLY ACTIVE MINIMUM PATH TABLE.
VAPSAV	KMV	I2	POINTER TO ALTERNATE ROUTE TABLE FOUND WHEN THE VEHICLE WAS IN THE STATION. SET TO 0 IF THE MINIMUM PATH TABLE IS TO BE USED.
VTROCC	KMV	I2	TRAIN OCCUPANCY: VARIABLE HEADWAY: NUMBER OF VEHICLES IN TRAIN FIXED HEADWAY: NUMBER OF FIXED BLOCKS REQUIRED FOR THE TRAIN.
VADV	KMV	I2	ADVANCE REQUIRED AFTER MERGE.
VSLIP	KMV	I2	NUMBER OF TIMES VEHICLE RETARDED.
VPLATN	KMV	I2	VEHICLE PLATOON ASSIGNMENT FOR PLATOON BERTH POLICY.
VNBRD	KMV	I2	NUMBER OF PASSENGERS CURRENTLY BOARDING.
VULTD	KMV	I2	VEHICLE ULTIMATE DESTINATION.
VSTOP	KMV	I2	STATION STOP INDICATOR: 0==>BYPASS STATION 1==>STOP AT STATION
VNSBRD	KMV	I2	NUMBER OF BOARDING PASSENGERS THAT WILL TAKE SEATS.
SPBDMX	KMV	I2	MAXIMUM NUMBER OF PASSENGERS THAT CAN BOARD A VEHICLE DUE TO TIME LIMIT.
VLBRD	KMV	I2	VEHICLE PERFORM BOARD EVENT INDICATOR.
VLBYP	KMV	I2	VEHICLE BYPASS STATION IN EMPTY CIRCULATION MODE FLAG
VLHOLD	KMV	I2	VEHICLE HOLD AT DOCK FAILURE RESPONSE FLAG.
VPUSHV	KMV	I2	VEHICLE ID OPERATING AS PUSH VEHICLE FOR FAILED VEHICLE.
VHOCC	KMV	I2	NUMBER OF HANDICAPPED PASSENGERS ON VEHICLE.
STOTPT	KMV	I2	PSEUDO PASSENGER ON BOARD COUNT FOR COMPUTING BOARD TIME.
VRHCP	KMV	I2	NUMBER OF HANDICAPPED PASSENGERS HOLDING RESERVATIONS ON VEHICLE.
VBERTH	KMV	I2	BERTH NUMBER VEHICLE CURRENTLY ASSIGNED TO IN TIMEOUT/GROUP
VPRSTN	KMV	I2	PREVIOUS STATION STOP IN MORGANTOWN DEMAND.MODE RESPONSIVE SERVICE.

TABLE 3-1 (24 of 53) GLOBAL VARIABLES

NAME: ECMVEH (CONT)			CATEGORY: MP VEHICLE DATA
VARIABLE	DIM	TYPE	DESCRIPTION
VQUED	KMV	L1	LOGIC VARIABLE INDICATING WHETHER VEHICLE IS QUEUED ON A GUIDEWAY OR STATION LINK.
VLAGAN	KMV	L1	INDICATES THAT THE LAUNCH EVENT MUST BE PERFORMED AGAIN BECAUSE IT WAS NOT POSSIBLE TO OBTAIN A RESERVATION THROUGH THE MERGES.
VTERM	KMV	L1	VEHICLE FLUSH (TERMINATION FLAG) FOR ACTIVE FLEET SIZE MANAGEMENT.
VNTRN	KMV	L1	ENTRAINMENT RETRY INDICATOR.
VATERM	KMV	L1	FLAG TO TERMINATE VEHICLE TRAIN FOR ACTIVE FLEET SIZE MANAGEMENT DECREASE ON ROUTE.
VLDBRD	KMV	L1	VEHICLE PERFORM DEBOARD EVENT FLAG (NOT USED).
VMOVUP	KMV	L1	FLAG TO INDICATE VEHICLE PERFORMING TIMEOUT/ MORGANTOWN RIPPLE BERTH ADVANCE MOVEMENT.

TABLE 3-1 (25 of 53) GLOBAL VARIABLES

NAME: ECMXTN			CATEGORY: MP TRANSACTION HEADER DATA
VARIABLE	DIM	TYPE	DESCRIPTION
XTIME	KMX	I4	SYSTEM TRANSACTION TIME: IF XTIME > CLOCK, THEN THIS IS THE TIME AT WHICH THE TRANSACTION/TRIP/VEHICLE IS TO COME OFF THE FUTURE EVENTS LIST. IF XTIME < CLOCK, THEN THIS IS THE TIME AT WHICH THE TRANSACTION/TRIP/VEHICLE CAME OFF THE FUTURE EVENTS LIST AND WAS PUT IN A QUEUE.
VTIME	KMV	I4	VEHICLE TRANSACTION TIME. EQUIVALENCED WITH XTIME(1). SEE XTIME DESCRIPTION.
TTIME	KMT	I4	TRIP TRANSACTION TIME. EQUIVALENCED WITH XTIME(KMV + 1). SEE XTIME DESCRIPTION.
XACTIV	-	I4	ID OF THE CURRENT SYSTEM TRANSACTION BEING PROCESSED.
VACTIV	-	I4	ID OF THE CURRENT VEHICLE TRANSACTION BEING PROCESSED. EQUIVALENCED WITH XACTIV.
TACTIV	-	I4	ID OF THE CURRENT TRIP TRANSACTION BEING PROCESSED. EQUIVALENCED WITH XACTIV.
XSEVNT	KMX	I2	SYSTEM EVENT; WHERE TO GO IN MAIN ROUTINE WHEN EVENT COMES OFF FUTURE EVENTS LIST.
VSEVNT	KMV	I2	VEHICLE EVENT; VEHICLE PROCESS TO BE PERFORMED WHEN EVENT COMES OFF FUTURE EVENTS LIST. EQUIVALENCED WITH XSEVNT(1).
TSEVNT	KMT	I2	TRIP EVENT; TRIP PROCESS TO BE PERFORMED WHEN EVENT COMES OFF FUTURE EVENTS LIST. EQUIVALENCED WITH XSEVNT(KMV + 1).
TCURR	KMT	I2	CURRENT STATION OF TRIP. EQUIVALENCED WITH XSEVNT(KMV + 1).
XMEVNT	KMX	I2	SYSTEM EVENT; MODELING PROCESS TO BE PERFORMED WHEN EVENT COMES OFF FUTURE EVENTS LIST.
VMEVNT	KMV	I2	VEHICLE EVENT; MODELING PROCESS TO BE PERFORMED WHEN EVENT COMES OFF FUTURE EVENTS LIST. EQUIVALENCED WITH XMEVNT(1).
TMEVNT	KMT	I2	TRIP EVENT; MODELING PROCESS TO BE PERFORMED WHEN EVENT COMES OFF FUTURE EVENTS LIST. EQUIVALENCED WITH XMEVNT(KMV + 1).
XFELCH	KMX	I2	CHAIN WORD TO PUT SYSTEM TRANSACTION ON FUTURE EVENTS LIST.
VFELCH	KMV	I2	CHAIN WORD TO PUT VEHICLE TRANSACTION ON FUTURE EVENTS LIST. EQUIVALENCED WITH XFELCH(1).
TFELCH	KMT	I2	CHAIN WORD TO PUT TRIP TRANSACTION ON FUTURE EVENTS LIST. EQUIVALENCED WITH XFELCH(KMV + 1).
XQUECH	KMX	I2	CHAIN WORD TO PUT SYSTEM TRANSACTION INTO A QUEUE WHEN TRANSACTION IS NOT ON FUTURE EVENTS LIST. EQUIVALENCED WITH XFELCH(1).
VQUECH	KMV	I2	CHAIN WORD TO PUT VEHICLE TRANSACTION INTO A QUEUE WHEN TRANSACTION IS NOT ON FUTURE EVENTS LIST. EQUIVALENCED WITH XFELCH(1).
TQUECH	KMT	I2	CHAIN WORD TO PUT TRIP TRANSACTION INTO A QUEUE WHEN TRANSACTION IS NOT ON FUTURE EVENTS LIST. EQUIVALENCED WITH XFELCH(KMV + 1).
XEXTR1	KMX	I2	INTEGER*2 VARIABLE FOR DATA ASSOCIATED WITH A TRANSACTION.
VCURR	KMV	I2	THE GUIDEWAY LINK OR STATION ON WHICH THE VEHICLE IS LOCATED WHEN ITS TRANSACTION COMES OFF THE FUTURE EVENTS LIST. EQUIVALENCED WITH XEXTR1(1).
TRVEHN	KMT	I2	VEHICLE ID ON WHICH THE TRIP HAS A RESERVATION. EQUIVALENCED WITH XEXTR1(KMV + 1).
XEXTR2	KMX	I2	INTEGER*2 VARIABLE FOR DATA ASSOCIATED WITH A TRANSACTION.

TABLE 3-1 (26 of 53) GLOBAL VARIABLES

NAME: ECMXTN (CONT)			CATEGORY: MP TRANSACTION HEADER DATA
VARIABLE	DIM	TYPE	DESCRIPTION
VSL	KMV	I2	THE STATION LINK ON WHICH THE VEHICLE IS LOCATED WHEN ITS TRANSACTION COMES OFF THE FUTURE EVENTS LIST. EQUIVALENCED WITH XEXTR2(1).
XAVAIL	-	I2	POINTER TO AVAILABLE CHAIN OF SYSTEM SERVICE TRANSACTIONS.
VAVAIL	-	I2	POINTER TO AVAILABLE CHAIN OF VEHICLE TRANSACTIONS.
TAVAIL	-	I2	POINTER TO AVAILABLE CHAIN OF TRIP TRANSACTIONS.

TABLE 3-1 (27 of 53) GLOBAL VARIABLES

NAME: ECNDMD		CATEGORY: IP DEMAND GENERATION DATA	
VARIABLE	DIM	TYPE	DESCRIPTION
DDDIST	KMS,	R4	CUMULATIVE PROBABILITY DISTRIBUTIONS OF DESTINATION SELECTION FROM EACH ORIGIN.
DIATIM	-	R4	INTERARRIVAL TIME BETWEEN PREVIOUS AND CURRENT TRIPS.
DLAMDA	-	R4	MEAN TRIP INTERARRIVAL TIME.
DMNTRP	3	R4	MEAN GROUP SIZE OF EACH OF THREE GROUP SIZE DISTRIBUTIONS.
TARRT	-	R4	TRIP ARRIVAL TIME.
DODIST	KMS	R4	CUMULATIVE PROBABILITY DISTRIBUTION OF ORIGIN USAGE.
DTRDST	KMG,	R4	EACH OF THREE COLUMNS IS A PROBABILITY DISTRIBUTION OF TRIP GROUP SIZE. USER ENTERS FREQUENCY DISTRIBUTIONS, IP CONVERTS THEM TO CUMULATIVE DISTRIBUTIONS.
DOUSE	KMS	R4	NUMBER OF TRIPS PER HOUR ORIGINATING AT EACH STATION.
DNTRPS	-	R4	TOTAL NUMBER OF TRIPS PER HOUR OVER ALL STATIONS.
DTDMND	KMS,	R4	ORIGIN/DESTINATION DEMAND MATRIX IN TRIPS PER HOUR.
DSCFAC	-	R4	NET SCALE FACTOR FOR DEMAND SCALING.
DMPROF	3,	R4	DEMAND PROFILE GENERATION CONTROL VARIABLE: DMPROF(1,I)===>SCALE FACTOR FOR ITH DEMAND INTERVAL DMPROF(2,I)===>TIME BASE FOR ITH DEMAND INTERVAL (IF = 0, USE THE TIME BASE STORED WITH THE DEMAND INPUT DATA) DMPROF(3,I)===>= 0, USE THE DEMAND MATRIX CURRENTLY IN MEMORY > 0, READ A NEW DEMAND MATRIX
NOTOUT	-	R4	THEORETICAL MINUS ACTUAL PASSENGER COUNT FOR DEMAND INTERVAL IN UNIFORM DEMAND GENERATION.
DMAX	-	R4	HIGH VALUED CONSTANT.
DENDTM	-	I4	SIMULATED TIME AT WHICH THE CURRENT DEMAND MATRIX BECOMES OBSOLETE.
DPASTL	-	I4	TOTAL NUMBER OF PASSENGERS GENERATED.
DKSEED	-	I4	SEED FOR RANDOM NUMBER GENERATOR USED FOR TRIP GENERATION (INITIALIZED TO AKSEED).
TORIG	-	I2	TRIP ORIGIN STATION.
IDEST	-	I2	TRIP DESTINATION STATION.
TPASS	-	I2	NUMBER OF PASSENGERS ON THE TRIP.
DNSD2	-	I2	NUMBER OF STATION PAIRS USING THE SECOND GROUP SIZE DISTRIBUTION.
DNSD3	-	I2	NUMBER OF STATION PAIRS USING THE THIRD GROUP SIZE DISTRIBUTION.
DPDMND	KMS,	I2	ORIGIN/DESTINATION DEMAND MATRIX IN PASSENGERS PER TIME BASE.
DSTSEL	KMS,	I2	GROUP SIZE DISTRIBUTION SELECTION MATRIX: ELEMENT I,J CONTAINS 1,2 OR 3 TO SELECT ONE OF THREE GROUP SIZE DISTRIBUTIONS FOR USE IN DEFINING THE GROUP SIZE FOR TRIPS FROM STATION I TO STATION J.
DTMBAS	-	I2	LENGTH OF TIME (MINUTES) OVER WHICH PASSENGER DEMAND MATRIX IS VALID.
DTRIP3	3	I2	TOTAL TRIPS PER HOUR FOR EACH OF THREE GROUP SIZE DISTRIBUTIONS.
DTRPTB	KMS,	I2	TABULATION OF GENERATED TRIPS BY ORIGIN AND DESTINATION.
DNDMND	-	I2	NUMBER OF DEMAND PROFILE INTERVALS TO PROCESS.
DIS2OD	KMNOD	I2	LIST OF ORIGIN/DESTINATION PAIRS THAT USE THE SECOND GROUP SIZE DISTRIBUTION.

TABLE 3-1 (28 of 53) GLOBAL VARIABLES

NAME: ECNDMD (CONT)			CATEGORY: IP DEMAND GENERATION DATA
VARIABLE	DIM	TYPE	DESCRIPTION
DIS3OD	KMNOD	I2	LIST OF ORIGIN/DESTINATION PAIRS THAT USE THE THIRD GROUP SIZE DISTRIBUTION.
DNSTA	-	I2	NUMBER OF STATIONS FOR WHICH DEMAND DATA EXISTS.
KNG	-	I2	LARGEST TRIP GROUP SIZE.
DPFPTR	-	I2	POINTER TO THE DEMAND PROFILE INTERVAL CURRENTLY BEING PROCESSED.
TSZARR	3, KMDPRF	I2	TRIP SIZE ARRAY FOR UNIFORM DEMAND GENERATION.
DTRPFL	-	L1	DEMAND GENERATION REQUEST: F===>DO NOT GENERATE TRIPS T===>GENERATE A SEQUENCE OF TRIP ARRIVALS
UNFDMD	-	L1	UNIFORM DEMAND GENERATION INDICATOR: F===>USE POISSON DEMAND GENERATION T===>USE UNIFORM DEMAND GENERATION

NAME: ECNNET			CATEGORY: IP NETWORK DATA
VARIABLE	DIM	TYPE	DESCRIPTION
NCOST	KMN, KMN	I4	COST OF TRAVEL BETWEEN PAIRS OF NETWORK NODES. NCOST(I,J) IS DISTANCE OR TRAVEL TIME FROM NODE I TO NODE J.
NBYLNK	-	I2	ID OF STATION BYPASS LINK.
NDELST	KMN	I2	TABULATION OF NUMBER OF OCCURRENCES OF EACH NODE AS A DESTINATION IN THE INPUT DATA.
NDEST	KML	I2	ID OF NODE AT END OF EACH LINK.
NDIVID	-	I2	DIVERGE IDENTIFICATION COUNTER
NDNODE	-	I2	NUMBER OF UNIQUE DESTINATION NODE IDENTIFICATIONS.
NODE	KML	I2	ID OF NODE AT BEGINNING OF EACH LINK.
NONODE	-	I2	NUMBER OF UNIQUE ORIGIN NODE IDENTIFICATIONS.
NORLST	KMN	I2	TABULATION OF NUMBER OF OCCURRENCES OF EACH NODE AS AN ORIGIN IN THE INPUT DATA.
NTYPE	KML	I2	LIST OF NODE TYPES: NTYPE(I) = 0===>NO STATION ON LINK I > 0===>STATION ON LINK I
NLNPRI	KML	I2	LIST OF NODE PAIRS IDENTIFYING LINKS THAT HAVE PRIORITY AT MERGES. ODD SUBSCRIPTS CONTAIN NODE AT BEGINNING OF LINK, EVEN CONTAIN NODE AT END OF LINK.
NMAPNR	KMN	I2	NODE MAP, ORIGINAL NETWORK TO REDUCED NETWORK
NUPSTN	KMS	I2	ID OF MAJOR NODE UPSTREAM FROM EACH STATION
NEWNET	-	L1	NETWORK PROCESSING REQUEST: F===>USE PREVIOUSLY PROCESSED NETWORK DATA T===>READ AND PROCESS NEW NETWORK DEFINITION DATA
NETDAT	-	L1	INDICATES WHETHER NETWORK STRUCTURED DATA EXISTS IN MEMORY: F===>DATA DOES NOT EXIST IN MEMORY T===>DATA EXISTS IN MEMORY

TABLE 3-1 (29 of 53) GLOBAL VARIABLES

NAME: ECPOL			CATEGORY: IP POLICY DATA
VARIABLE	DIM	TYPE	DESCRIPTION
PAVTRP	-	R4	AVERAGE NUMBER OF PASSENGERS PER TRIP OVER THE ENTIRE NETWORK.
PLDFAC	-	R4	ESTIMATED ACHIEVABLE VEHICLE LOAD FACTOR FOR DEMAND RESPONSIVE MULTIPARTY AND SCHEDULED SERVICE.
SLBVEL	-	R4	VELOCITY THROUGH ONLINE STATION WHEN NOT REQUIRED TO STOP.
PMRGTH	-	R4	FRACTION OF MERGE WINDOW TO BE RESERVED.
PMAXWT	-	I4	MAXIMUM TIME A TRIP SHOULD WAIT FOR A VEHICLE IN SCHEDULED SERVICE.
PSRCET	KMS	I2	NUMBER OF EXCESS TRIPS AT SOURCE STATIONS.
PSNKDT	KMS	I2	NUMBER OF DEFICIT TRIPS AT SINK STATIONS.
PSNKID	KMS	I2	LIST OF STATIONS THAT ARE EXPECTED TO REQUIRE EMPTY VEHICLES.
PSRCID	KMS	I2	LIST OF STATIONS THAT ARE EXPECTED TO BE A SOURCE OF EMPTY VEHICLES.
PLOSBS	-	I2	SOURCE OF LEVEL OF SERVICE DEFINITION: 0==>DEFINED BY USER N==>DEFINED BY IP USING THE NTH DEMAND PROFILE INTERVAL
PNEMPV	-	I2	ESTIMATED NUMBER OF CONCURRENT EMPTY VEHICLES.
PNOCCV	-	I2	ESTIMATED NUMBER OF CONCURRENT OCCUPIED VEHICLES.
PNARR	KMS	I2	NUMBER OF TRIPS DESTINATING AT EACH STATION PER HOUR.
PNLV	KMS	I2	NUMBER OF TRIPS ORIGINATING AT EACH STATION PER HOUR.
PNSNK	-	I2	NUMBER OF STATIONS THAT ARE EXPECTED TO REQUIRE EMPTY VEHICLES.
PNSRC	-	I2	NUMBER OF STATIONS THAT ARE EXPECTED TO BE A SOURCE OF EMPTY VEHICLES.
PNXSTA	-	I2	NEXT STATION ON PATH TO DESTINATION STATION.
PEVOD	KMS, KMS	I2	EMPTY VEHICLE REDISTRIBUTION MATRIX. PEVOD(I,J) CONTAINS THE NUMBER OF EMPTY VEHICLES TO BE SENT FROM STATION I TO STATION J PER HOUR SUCH THAT THE COST OF REDISTRIBUTION IS A MINIMUM.
PRSEQ	KMS	I2	SEQUENCE OF STATIONS ON A CANDIDATE ROUTE.
PRTDEF	-	I2	SOURCE OF ROUTE DEFINITION IN SCHEDULED SERVICE: 0==>ROUTES DEFINED BY USER 1==>ROUTES, DEFINED BY IP
PSURPL	KMS	I2	NET SURPLUS OF EMPTY VEHICLES AT EACH STATION (NEGATIVE INDICATES DEFICIENCY).
PALTRT	KMALT	I2	USER'S DEFINITION OF ALTERNATE PATH NODE SEQUENCES.
PDWELL	-	I2	NOMINAL VEHICLE DWELL TIME IN STATION.
PXFLST	4, KMXFER	I2	TRANSFER STATION LIST: PXFLST(1,I)===>NODE ID OF ORIGIN STATION PXFLST(2,I)===>NODE ID OF DESTINATION STATION PXFLST(3,I)===>NODE ID OF STATION TO DEBOARD AT WHEN TRANSFERRING PXFLST(4,I)===>NODE ID OF STATION TO REBOARD AT WHEN TRANSFERRING (= 0 INDICATES REBOARD WITHOUT AN INTERMEDIATE WALK TO ANOTHER STATION)

TABLE 3-1 (30 of 53) GLOBAL VARIABLES

NAME: ECNPOL (CONT)			CATEGORY: IP POLICY DATA
VARIABLE	DIM	TYPE	DESCRIPTION
PRLIST	KMS	L1	BOOLEAN VECTOR IDENTIFYING WHETHER A STATION APPEARS ON A ROUTE.
PRTABL	KMR, KMS	L1	TABLE OF BOOLEAN VECTORS DEFINING THE UNIQUE ROUTES FOUND.
PSRCST	-	L1	INDICATES WHETHER A SOURCE STATION FOR EMPTY VEHICLES HAS BEEN FOUND.

NAME: ECNSTR			CATEGORY: IP NETWORK STRUCTURED DATA
VARIABLE	DIM	TYPE	DESCRIPTION
NODTIM	KMS, KMS	R4	STATION TO STATION TRAVEL TIME MATRIX. NODTIM(I,J) CONTAINS THE NOMINAL TRAVEL TIME FROM STATION I TO STATION J.
PSPEED	-	R4	NOMINAL VEHICLE SPEED ON GUIDEWAY.
KNN	-	I2	NUMBER OF NODES IN NETWORK.
NLDEST	KML	I2	LIST OF IDENTIFICATIONS OF NODES AT END OF LINKS IN IP NUMBERING SEQUENCE.
NLORIG	KML	I2	LIST OF IDENTIFICATIONS OF NODES AT BEGINNING OF LINKS IN IP NUMBERING SEQUENCE.
NMAPUS	KMNID	I2	MAPS INPUT NODE IDENTIFICATIONS TO A CONTIGUOUS SET OF NODE IDENTIFICATIONS BEGINNING AT 1.
NMAPSU	KMN	I2	MAPS INTERNAL NODE IDENTIFICATIONS TO INPUT NODE IDENTIFICATIONS.
NLLIST	KMN, 2	I2	DEFINES LINK(S) THAT LEAVE EACH NODE.
NDLLST	KMN, 2	I2	DEFINES LINK(S) THAT LEAD TO EACH NODE.
NSLIST	KMN	I2	NODE LIST IN WHICH STATIONS HAVE UNIQUE IDENTIFICATIONS.
NCESEL	-	I2	COST SELECTION FOR MINIMUM PATH DETERMINATION: 0==>USE LINK NOMINAL TRAVEL TIME 1==>USE LINK LENGTH

TABLE 3-1 (31 of 53) GLOBAL VARIABLES

NAME: ECNSYS			CATEGORY: IP SIMULATION SYSTEM DATA			
VARIABLE	DIM	TYPE	DESCRIPTION			
ARANDN	-	R4	RANDOM NUMBER FROM A UNIFORM DISTRIBUTION BETWEEN 0 AND 1.			
ATIME	-	R4	SIMULATED TIME ASSOCIATED WITH DATA CURRENTLY BEING PROCESSED.			
ATIMTG	-	R4	VALUE IN TIME FIELD OF DATA HEADER CARD.			
AFALRE	11	R4	FAILURE/RECOVERY REQUEST:			
			INDEX VALUE	GUIDEWAY	STATION	VEHICLE
			1	0	GUIDEWAY	VEHICLE
				N	-	-
			2	0	START	START
				N	ENTIRE STN	ENTIRE STN
					LINK TYPE	LINK TYPE
			3	N	END	END
					NODE ID	NODE ID
			4	0	ENTIRE LINK	-
				1	LINK ENTRY	LINK ENTRY
				2	LINK EXIT	LINK EXIT
			5	1	FAILURE	FAILURE
				2	RECOVERY	RECOVERY
				3	-	DEGRADED
				4	-	RECOVERY
			6	N	-	DEGRADE FACTOR
						1<DF<10
			7	N	DELAY TO	DELAY TO
					DETECTION	DETECTION
			8	1	-	RECOVERY
				2	-	PUSH COUPLE
				3	-	TOW VEHICLE
			9	N	-	DELAY TO
						RESTART
			10	N	-	DELAY TO
						REPLACEMENT
			11	0	MIN PATH	MIN PATH
					NOT REDONE	NOT REDONE
				N	MIN PATH	MIN PATH
					REDONE	REDONE
AFIDEG	-	R4	TOW VEHICLE SPEED DEGRADATION FACTOR			
AMIN	-	I4	MINIMUM OF A SET OF VALUES.			
ATYPE	-	I4	FIRST 4 CHARACTERS OF DATA TYPE ON DATA HEADER CARD.			
ATEXT	18	I4	72 CHARACTERS OF TEXT FROM A DATA CARD.			
ACOL	-	I2	POINTS TO A COLUMN OF A MATRIX.			
APTR	-	I2	POINTS TO AN ELEMENT OF A LIST.			
APTR1	-	I2	POINTS TO AN ELEMENT OF A LIST (USED WITH APTR FOR COMPRESSING A LIST).			
AROW	-	I2	POINTS TO A ROW OF A MATRIX.			
ABCD	31	I2	CONTENTS OF COLUMNS 11 - 72 OF DATA HEADER CARD IN CHARACTER FORM.			
ADCTR	-	I2	NUMBER OF DEMAND PROFILE INTERVALS THAT HAVE BEEN PROCESSED.			
ADMCTR	-	I2	NUMBER OF SETS OF DEMAND GENERATION INITIAL CONDITIONS THAT HAVE BEEN READ.			
AFITOW	KML	I2	TOW VEHICLE PATH LINK ID SEQUENCE			

TABLE 3-1 (32 of 53) GLOBAL VARIABLES

NAME: ECNSYS (CONT)			CATEGORY: IP SIMULATION SYSTEM DATA
VARIABLE	DIM	TYPE	DESCRIPTION
AFIRSP	KMR	I2	OTHER VEHICLE RESPONSE REQUEST: 1===>CONTINUE REVENUE SERVICE 2===>DEBOARD ONLY MODE OF REVENUE SERVICE 3===>EMPTY CIRCULATION MODE, DEBOARD ALL PASSENGERS AT FIRST STATION STOP 4===>WAIT IN STATION MODE, VEHICLE HELD AT DOCK PRIOR TO BOARD EVENT
ADFCTR	-	I2	NUMBER OF FILES OF DEMAND GENERATION INITIAL CONDITIONS THAT HAVE BEEN READ.
AMNAME	11, 8	L1	FILE MEMBER NAMES USED FOR CURRENT EXPERIMENT (11 NAMES, EACH CONSISTING OF 8 CHARACTERS).
AFIRST	-	L1	FIRST PASS INDICATOR.
AEOD	-	L1	END OF INPUT DATA INDICATOR.
AINDEX	-	L1	INDICATES INDEX FILE INFORMATION HAS BEEN PROCESSED.
AIFAIL	-	L1	INDICATES THAT A FAILURE/RECOVERY REQUEST HAS BEEN READ.
AEOF	-	L1	INDICATES END OF FILE HAS BEEN DETECTED WHILE READING INPUT DATA.
AHDR	-	L1	INDICATES THAT A DATA HEADER CARD HAS BEEN PROCESSED.
AAFMS	-	L1	INDICATES THAT AN ACTIVE FLEET SIZE MANAGEMENT REQUEST HAS BEEN READ.
AERROR	-	L1	INDICATES A SERIOUS ERROR IN THE INPUT DATA HAS BEEN DETECTED.
ARNTIM	-	L1	INDICATES THAT A STRUCTURED RUN TIME FILE WAS CREATED.
ANOMTT	-	L1	INDICATES THAT A REQUEST TO WRITE A NOMINAL TRAVEL TIME FILE WAS READ.
ASYSI	-	L1	INDICATES THAT DATA WAS READ FROM A SYSTEM CHARACTERISTICS INPUT AND DESCRIPTION FILE.
ARNTMI	-	L1	INDICATES THAT DATA WAS READ FROM A RUN TIME INPUT AND DESCRIPTION FILE.
ANETI	-	L1	INDICATES THAT NETWORK DATA WAS READ FROM: F===>NETWORK STRUCTURED FILE T===>NETWORK INPUT AND DESCRIPTION FILE
ANETO	-	L1	INDICATES THAT STRUCTURED NETWORK FILE WAS WRITTEN.

TABLE 3-1 (33 of 53) GLOBAL VARIABLES

NAME: ECNTRN			CATEGORY: IP TRANSPORTATION ALGORITHM DATA
VARIABLE	DIM	TYPE	DESCRIPTION
X	1002	I4	CURRENT SOLUTION MATRIX SPECIFIED BY CODED VECTORS INDEXED BY K.
U	501	I4	INITIALLY THE SUPPLY VECTOR, THEN BECOMES THE ROW DUAL VARIABLE (SIMPLEX MULTIPLIER).
V	501	I4	INITIALLY THE DEMAND VECTOR, THEN BECOMES THE COLUMN DUAL VARIABLE (SIMPLEX MULTIPLIER).
IOPTN	10	I4	OPTION STATUS: IOPTN(5) = 0 NO ACTION = 1 DUMP VECTOR INTERIM STATUS AT EACH ITERATION (SET BY AUXILIARY OUTPUT FLAGS 64 AND 84)
LINKU	501	I4	ROW POINTER (J) TO LOOP ELEMENT X(I,J).
LINKV	501	I4	COLUMN POINTER (I) TO LOOP ELEMENT X(I,J).
TNODE	1002	I4	ELEMENTS X(I,J) DESIGNATED BY K IN CURRENT LOOP.
C	1002	I4	CURRENT COST ELEMENTS ASSOCIATED WITH X IN THE BASIS.
MXPUSH	-	I4	MAGNITUDE OF VARIABLE TO BE BROUGHT INTO BASIS.
IOUT	-	I4	NOT USED.
JOUT	-	I4	POINTER TO CANDIDATES TO BE REMOVED WHEN LOOP FOUND.
NBR	-	I4	NUMBER.
NEGOBJ	-	I4	REDUCTION IN OBJECTIVE FUNCTION.
NONZ	-	I4	NUMBER OF NON ZERO ELEMENTS IN BASIS.
INFIN	-	I4	NUMBER REPRESENTING INFINITY.
IDBASE	-	I4	IDENTIFICATION OF BASE ROW DUAL VARIABLE.
INFINC	-	I4	COST VALUE REPRESENTING INFINITY.
IDEXIT	-	I4	INDICATES BASE ROW DUAL VARIABLE ENCOUNTERED IN LOOP.
IPARA	-	I4	PARAMETRIC PROCESSING REQUEST: 0==>NORMAL PROCESSING 1==>PERFORM PARAMETRIC PROCESSING
M	-	I4	NUMBER OF ROWS (ORIGINS) IN PROBLEM.
N	-	I4	NUMBER OF COLUMNS (DESTINATIONS) IN PROBLEM.
NRCCELL	-	I4	TOTAL NUMBER OF CELLS IN PROBLEM.
PERROR	-	I4	ERROR CONDITION INDICATOR.
KCOST	6000	I4	ENTIRE COST MATRIX ROW BY ROW, INDEXED BY K*.
COLUMN	1002	I2	COLUMN DESIGNATION OF CORRESPONDING X(K) ELEMENT.
ROW	501	I2	NUMBER OF SEQUENTIAL ELEMENTS IN ALL ROWS UP TO EACH ROW IN X(K).
KCOL	6000	I2	COLUMN DESIGNATION OF CORRESPONDING KCOST(K*).
KROW0	-	I2	PROVIDES VALUE ZERO WHEN KROW(0) IS REFERENCED
KROW	501	I2	NUMBER OF SEQUENTIAL COST ELEMENTS IN ALL ROWS UP TO EACH ROW IN KCOST(K*).

TABLE 3-1 (34 of 53) GLOBAL VARIABLES

NAME: EILIMITS			CATEGORY: IP LIMITS FOR REASONABLENESS CHECKS
VARIABLE	DIM	TYPE	DESCRIPTION
LADSPH	-	CHAR	TIMEOUT/GROUP DEMAND RESPONSIVE ADEQUATE SPACE IN CHANNELS PARAMETER HIGH LIMIT.
LCKPTH	-	CHAR	CHECKPOINT INTERVAL HIGH LIMIT.
LCLOPH	-	CHAR	TRANSACTIONS PER CLOCK TABLE ENTRY HIGH LIMIT.
LCLOPL	-	CHAR	TRANSACTIONS PER CLOCK TABLE ENTRY LOW LIMIT.
LCLSMH	-	CHAR	SPACING BETWEEN CLOCK TABLE ENTRIES HIGH LIMIT.
LCLSML	-	CHAR	SPACING BETWEEN CLOCK TABLE ENTRIES LOW LIMIT.
LCLSZH	-	CHAR	NUMBER OF ENTRIES IN CLOCK TABLE HIGH LIMIT.
LCLSZL	-	CHAR	NUMBER OF ENTRIES IN CLOCK TABLE LOW LIMIT.
LCSELH	-	CHAR	LEAST COST PATH COST SELECTION HIGH LIMIT.
LCSIZH	-	CHAR	CLOCK SCALE FACTOR HIGH LIMIT.
LGLENH	-	CHAR	GUIDEWAY LINK LENGTH HIGH LIMIT.
LGLVSH	-	CHAR	GUIDEWAY SPEED STANDARD DEVIATION HIGH LIMIT.
LGVELH	-	CHAR	VEHICLE SPEED ON GUIDEWAY HIGH LIMIT.
LHBATH	-	CHAR	HANDICAPPED PASSENGER BOARD/DEBOARD DOOR TIME HIGH LIMIT.
LHBBTH	-	CHAR	HANDICAPPED PASSENGER BOARD SECURE TIME HIGH LIMIT.
LHDBTH	-	CHAR	HANDICAPPED PASSENGER DEBOARD RELEASE TIME HIGH LIMIT.
LHVCPH	-	CHAR	VEHICLE HANDICAPPED PASSENGER CAPACITY HIGH LIMIT.
LMDLYH	-	CHAR	MERGE DELAY TABLE HIGH LIMIT.
LMDWTH	-	CHAR	MERGE DELAY WEIGHTING FACTOR HIGH LIMIT.
LMNGRH	-	CHAR	TIMEOUT/GROUP DEMAND RESPONSIVE MINIMUM GROUP SIZE TO REQUEST VEHICLE HIGH LIMIT.
LMNGRL	-	CHAR	TIMEOUT/GROUP DEMAND RESPONSIVE MINIMUM GROUP SIZE TO REQUEST VEHICLE LOW LIMIT.
LMRGLH	-	CHAR	LOCAL MERGE PRIORITY TABLE HIGH LIMIT.
LMRGWH	-	CHAR	MERGE RESERVATION TABLE WINDOW WIDTH HIGH LIMIT.
LMXTMH	-	CHAR	TIMEOUT/GROUP DEMAND RESPONSIVE MAXIMUM WAIT TIME TO REQUEST VEHICLE HIGH LIMIT.
LMXTML	-	CHAR	TIMEOUT/GROUP DEMAND RESPONSIVE MAXIMUM WAIT TIME TO REQUEST VEHICLE LOW LIMIT.
LMXWTH	-	CHAR	MAXIMUM WAIT TIME FOR VEHICLE IN SCHEDULED SERVICE HIGH LIMIT.
LMXWTL	-	CHAR	MAXIMUM WAIT TIME FOR VEHICLE IN SCHEDULED SERVICE LOW LIMIT.
LNEEDH	-	CHAR	EMPTY VEHICLE NEED PER STATION HIGH LIMIT.
LPALGH	-	CHAR	PATH SELECTION ALGORITHM HIGH LIMIT.
LPCOMH	-	CHAR	PERIODIC COMPUTATION INTERVAL HIGH LIMIT.
LPMETH	-	CHAR	PATH SELECTION METHOD HIGH LIMIT.
LPTYPH	-	CHAR	PATH SELECTION TYPE HIGH LIMIT.
LPVEPH	-	CHAR	EMPTY VEHICLE DISBURSEMENT OPTIONS HIGH LIMIT.
LPVSPH	-	CHAR	EMPTY VEHICLE SOURCE OPTIONS HIGH LIMIT.
LRTIMH	-	CHAR	TIME TO ACCELERATE TO LINESPEED FROM STOP HIGH LIMIT.
LSAMPH	-	CHAR	SAMPLING INTERVAL HIGH LIMIT.
LSBQCH	-	CHAR	BOARDING QUEUE CAPACITY HIGH LIMIT.
LSDEGH	-	CHAR	STATION LINK DEGRADATION FACTOR HIGH LIMIT.
LSDEGL	-	CHAR	STATION LINK DEGRADATION FACTOR LOW LIMIT.
LSLDVH	-	CHAR	STATION LINK DIVERGE FUNCTION HIGH LIMIT.
LSLHTA	-	CHAR	STATION LINK HEADWAY ZONE TIME PER VEHICLE HIGH LIMIT.
LSLHTB	-	CHAR	STATION LINK HEADWAY ZONE TIME CONSTANT TERM HIGH LIMIT.
LSLPFH	-	CHAR	STATION LINK MERGE POLICY HIGH LIMIT.
LSLTTH	-	CHAR	STATION LINK TRAVEL TIME HIGH LIMIT.
LSMOVH	-	CHAR	TIMEOUT/GROUP DEMAND RESPONSIVE RIPPLE BERTH ADVANCEMENT TIME HIGH LIMIT.

TABLE 3-1 (35 of 53) GLOBAL VARIABLES

NAME: EILIMITS (CONT)			CATEGORY: IP LIMITS FOR REASONABLENESS CHECKS
VARIABLE	DIM	TYPE	DESCRIPTION
LSTATH	-	CHAR	SNAPSHOT INTERVAL HIGH LIMIT.
LSTBAH	-	CHAR	BOARD/DEBOARD TIME PER PASSENGER HIGH LIMIT.
LSTBBH	-	CHAR	DOOR OPEN MINIMUM TIME AND HEADWAY ADJUSTMENT TERM HIGH LIMIT.
LSTWTH	-	CHAR	TRAVEL TIME WEIGHTING FACTOR HIGH LIMIT.
LSUWTH	-	CHAR	UTILIZATION WEIGHTING FACTOR HIGH LIMIT.
LTLENH	-	CHAR	TRAIN LENGTH HIGH LIMIT.
LTMBSH	-	CHAR	DEMAND TIME BASE HIGH LIMIT.
LTMBSL	-	CHAR	DEMAND TIME BASE LOW LIMIT.
LTREDH	-	CHAR	TIME TO BEGIN READING DEMAND INPUT HIGH LIMIT.
LVCAPH	-	CHAR	VEHICLE CAPACITY HIGH LIMIT.
LVDEGH	-	CHAR	VEHICLE DEGRADATION FACTOR HIGH LIMIT.
LVDEGL	-	CHAR	VEHICLE DEGRADATION FACTOR LOW LIMIT.
LVSPCH	-	CHAR	VEHICLE SPACING METHOD HIGH LIMIT.
LWALKH	-	CHAR	TRANSFER WALK TIME HIGH LIMIT.

TABLE 3-1 (36 of 53) GLOBAL VARIABLES

NAME: ESYSMAX			CATEGORY: COMPILE TIME MAXIMUM ARRAY SIZES
VARIABLE	DIM	TYPE	DESCRIPTION
KMALT	-	CHAR	NUMBER OF ENTRIES IN ALTERNATE ROUTE LIST.
KMANT	-	CHAR	NUMBER OF ENTRIES IN EMPTY VEHICLE ANTICIPATED NEED LISTS.
KMCLTA	-	CHAR	NUMBER OF ENTRIES IN CLOCK TABLE.
KMCR	-	CHAR	NUMBER OF EMPTY VEHICLE CIRCULATION ROUTES.
KMCRS	-	CHAR	NUMBER OF ENTRIES IN EMPTY VEHICLE CIRCULATION ROUTE LIST.
KMDLY1	-	CHAR	NUMBER OF ROWS IN MERGE DELAY TABLE.
KMDLY2	-	CHAR	NUMBER OF COLUMNS IN MERGE DELAY TABLE.
KMDFRF	-	CHAR	NUMBER OF INTERVALS IN DEMAND PROFILE.
KMEVP	-	CHAR	NUMBER OF ENTRIES IN PRIORITY LIST OF WHERE TO SEND EMPTY VEHICLES.
KMFAIL	-	CHAR	NUMBER OF FAILURE AND RECOVERY CARDS.
KMFLAG	-	CHAR	NUMBER OF AUXILIARY OUTPUT FLAGS.
KMG	-	CHAR	NUMBER OF PASSENGERS PER TRIP.
KMGT	-	CHAR	NUMBER OF ENTRIES IN ROUTE GROUP LIST.
KMHDR	-	CHAR	NUMBER OF DATA CARD HEADER TYPES.
KML	-	CHAR	NUMBER OF GUIDEWAY LINKS.
KMLS	-	CHAR	KML + KMS.
KMLSLT	-	CHAR	NUMBER OF SIMULTANEOUS LEAST COST PATH TABLES.
KMM	-	CHAR	NUMBER OF MERGES (INCLUDING THOSE AT EXIT OF OFFLINE STATIONS).
KMM2	-	CHAR	KMM + 2.
KMMSGI	-	CHAR	NUMBER OF INFORMATION MESSAGES ISSUED BEFORE TERMINATION.
KMMSGS	-	CHAR	NUMBER OF MESSAGES OF ANY KIND ISSUED BEFORE TERMINATION.
KMMSGW	-	CHAR	NUMBER OF WARNING MESSAGES ISSUED BEFORE TERMINATION.
KMMTYP	-	CHAR	NUMBER OF MESSAGES OF ANY TYPE ISSUED BEFORE TERMINATION.
KMN	-	CHAR	NUMBER OF NETWORK NODES.
KMNID	-	CHAR	MAXIMUM NETWORK NODE IDENTIFICATION LABEL.
KMNOD	-	CHAR	NUMBER OF ENTRIES IN LIST OF O/D PAIRS USING THE SECOND OR THIRD TRIP GROUP SIZE DISTRIBUTIONS.
KMODPM	-	CHAR	(KMS * 50) + (KML * 20) + (KMR * 45).
KMR	-	CHAR	NUMBER OF ROUTES.
KMR1	-	CHAR	KMR + 1.
KMRT	-	CHAR	NUMBER OF ENTRIES IN SCHEDULED ROUTE LIST.
KMRT2	-	CHAR	KMRT * 2.
KMS	-	CHAR	NUMBER OF STATIONS.
KMS1	-	CHAR	KMS + 1.
KMSL	-	CHAR	NUMBER OF STATION LINKS.
KMSLD	-	CHAR	NUMBER OF ENTRIES IN STATION LINK DOWNSTREAM LINK LIST.
KMSLDS	-	CHAR	NUMBER OF ENTRIES IN LIST OF STATION LINKS DOWNSTREAM FROM A STATION LINK DIVERGE.
KMSLE	-	CHAR	NUMBER OF ENTRIES IN STATION LINK EVENT LIST.
KMSLU	-	CHAR	NUMBER OF ENTRIES IN STATION LINK UPSTREAM LINK LIST.
KMSSQ	-	CHAR	NUMBER OF ENTRIES IN STATION-TO-STATION MEASURES WORKSPACE (KMS*KMS*5).
KMSVP	-	CHAR	NUMBER OF ENTRIES IN PRIORITY LIST OF WHERE TO SEARCH FOR EMPTY VEHICLES.
KMT	-	CHAR	NUMBER OF TRIPS.
KMTR	-	CHAR	NUMBER OF TRIPS GENERATED BY UNIFORM DEMAND GENERATION ALGORITHM.
KMTR1	-	CHAR	KMTR + 1.
KMV	-	CHAR	NUMBER OF VEHICLES.
KMV1	-	CHAR	KMV + 1.
KMMHAX	-	CHAR	NUMBER OF INTERVALS IN MERGE RESERVATION TABLE
KMX	-	CHAR	NUMBER OF TRANSACTIONS.
KMXFER	-	CHAR	NUMBER OF O/D PAIRS REQUIRING A TRANSFER.

TABLE 3-1 (37 of 53) GLOBAL VARIABLES

NAME: (BLANK COMMON)			CATEGORY: DATA STORAGE (BIN AREA REGION)
VARIABLE	DIM	TYPE	DESCRIPTION
A	40000	R4	DATA STORAGE AREA DYNAMICALLY ALLOCATED TO BIN STORAGE DATA ACCUMULATION REGIONS

NAME: BASIC			CATEGORY: BIN ASSIGNMENT MAPPING
VARIABLE	DIM	TYPE	DESCRIPTION
LOC	400	I4	DISPLACEMENT IN DATA STORAGE REGION FOR START OF BIN AREA FOR ACCUMULATING DATA FOR A FILED USER REQUEST, LOC(1)=REQUEST 1, ETC.

NAME: EOSTAT			CATEGORY: STANDARD REPORT 2 MEASURES
VARIABLE	DIM	TYPE	DESCRIPTION
SYMEAS	13,3	R4	SYSTEM MEASURES.
STMEAS	KMS,15,3	R4	STATION MEASURES.
GLMEAS	KML,6,3	R4	GUIDEWAY LINK MEASURES.
RTMEAS	KMR,13,3	R4	ROUTE MEASURES.
RGMEAS	KMR,2,3	R4	GUIDEWAY LINK MEASURES.
LRMEAS	KML,KMR,3	R4	LINK-ROUTE MEASURES.

NAME: HISTO			CATEGORY: HISTOGRAM DATA
VARIABLE	DIM	TYPE	DESCRIPTION
MAX	-	I4	NOT USED.
MIN	-	I4	NOT USED.
AMAX	-	R4	MAXIMUM VALUE IN BIN.
AMIN	-	R4	MINIMUM VALUE IN BIN.
NSLOT	-	I4	NUMBER OF CLASS INTERVALS IN HISTOGRAM.

TABLE 3-1 (38 of 53) GLOBAL VARIABLES

NAME: IENDS			CATEGORY: REQUEST STORAGE COUNTER
VARIABLE	DIM	TYPE	DESCRIPTION
IEND	-	I4	NUMBER OF FILED REQUESTS

NAME: MATCH			CATEGORY: RECORD/REQUEST CHAINING TABLE
VARIABLE	DIM	TYPE	DESCRIPTION
MATTAB	120	I2	MATCH TABLE USED IN RECORD/REQUEST CORRELATION TABLE INDEX INDICATES MAJOR CATEGORY TYPE & POINTS TO FIRST REQUEST ENTRY IN ZREQUE FOR WHICH THE RECORD TYPE IS REQUIRED TO SATISFY OUTPUT REQUIREMENTS.
MATTAX	120	I2	INITIAL VALUES FOR RESETTING MATCH TABLE FOR EACH ACCUMULATION INTERVAL. SEE EODATA FOR SPECIFICATION.

NAME: NAMES			CATEGORY: REQUEST TRANSLATION DATA
VARIABLE	DIM	TYPE	DESCRIPTION
TSER	-	I4	TIME SERIES LIST = 'TSER'
STAT	-	I4	STATISTICAL SUMMARY = 'STAT'
PLOT	-	I4	TIME SERIES PLOT = 'PLOT'
SYSTEM	-	I4	SYSTEM WIDE MEASURE = 'SYST'
LINK	-	I4	GUIDEWAY LINK MEASURE = 'LINK'
LIST	-	I4	TIME SERIES LIST = 'LIST'
HIST	-	I4	HISTOGRAM = 'HIST'
STN	-	I4	STATION MEASURE = 'STN'
RTE	-	I4	ROUTE MEASURE = 'RTE'
STNL	-	I4	STATION LINK MEASURE = 'STNL'

NAME: OUTPT			CATEGORY: OUTPUT DISPLAY LABEL DATA
VARIABLE	DIM	TYPE	DESCRIPTION
TITLES	2100	I4	OUTPUT DISPLAY LABELS (4 WORDS/STATISTIC MNEMONIC)
MAJC	400	I2	SEE EODATA FOR DEFINITION. MAJOR CATEGORY INDICATOR BY FILED REQUEST MAJC(1)=REQUEST 1, ETC.
NCAT	120	I2	DISPLACEMENT IN TITLE ARRAY FOR START OF LABEL INFORMATION BY MAJOR CATEGORY

TABLE 3-1 (39 of 53) GLOBAL VARIABLES

NAME: OUTPT1			CATEGORY: PLOT LIMITS FOR TIME SERIES DISPLAY
VARIABLE	DIM	TYPE	DESCRIPTION
PLOTS	400	R4	LOWER LIMIT FOR OUTPUT PLOT BY FILED REQUEST PLOTS(1)=REQUEST 1, ETC.
PLOTE	400	R4	UPPER LIMIT FOR OUTPUT PLOT BY FILED REQUEST PLOTE(1)=REQUEST 1, ETC.

NAME: OUTPUT			CATEGORY: OUTPUT CONTROL DATA
VARIABLE	DIM	TYPE	DESCRIPTION
NBIN	10	I4	POINTER TO BIN LOCATIONS.
PAR	7	R4	CLASS INTERVAL WIDTH FOR HISTOGRAMS.
IPAR	7	I4	STEP WHEN LOOPING THROUGH STATISTICS SAMPLES FOR LIST.

NAME: OUTSUM			CATEGORY: STD REPORTS DATA ACCUMULATION ARRAYS
VARIABLE	DIM	TYPE	DESCRIPTION
Psumm	250	I4	PERFORMANCE SUMMARY MEASURES
Pmax	250	R4	MAX VALUES FOR PERFORMANCE SUMMARY MEASURES
Pmin	250	R4	MIN VALUES FOR PERFORMANCE SUMMARY MEASURES
PRFSUM	100	R4	COMPUTED OVERALL PERFORMANCE SUMMARY MEASURES WRITTEN TO FILE
DPM	KMODPI	R4	STANDARD REPORT 2 DPM MEASURES
DPMLR	KML, KMR, 4	R4	STANDARD REPORT 2 DPM MEASURES LINK BY ROUTE

NAME: READER			CATEGORY: DATA ACQUISITION PARAMETERS
VARIABLE	DIM	TYPE	DESCRIPTION
MSTART	-	I4	BEGIN TIME C.U. FOR DATA ACCUMULATION
MSTOP	-	I4	END TIME C.U. FOR DATA ACCUMULATION
MSOT	-	I4	DATA SET REFERENCE NUMBER FOR RAW STATISTICS
MSOTX	-	I4	DATA SET REFERENCE NUMBER FOR RAW STATISTICS
MNAME	-	R8	I/O PARM FOR RECORD IDENTIFIER
CUSEC	-	R4	C.U./SECOND
MCLOCK	-	I4	I/O PARM CURRENT VALUE OF TIME IN C.U. FOR RECORD BEING PROCESSED
EOF	-	I4	INDICATOR FOR END FILE ENCOUNTERED
MBYTES	-	I2	I/O PARM FOR NUMBER OF BYTES IN RECORD
MFOLL	-	I2	I/O PARM FOR NUMBER OF FOLLOWER RECORDS
MTYPE	-	I2	I/O PARM FOR RECORD TYPE PROCESSED

TABLE 3-1 (40 of 53) GLOBAL VARIABLES

NAME: REQUES			CATEGORY: REQUEST TABLE
VARIABLE	DIM	TYPE	DESCRIPTION
ZREQUE	12,400	I4	REQUEST STORAGE TABLE 1= OUTPUT FORMAT REQUIRED 2= MAJOR CATEGORY MNEMONIC 3= SUB CATEGORY MNEMONIC 4= REQUEST NUMBER 5= ASSIGNED BIN STORAGE AREA 6= SIZE OF ASSIGNED BIN AREA 7= CHAIN POINTER TO NEXT REQUEST AFTER RECORD REQUEST CORRELATION 8= LENGTH OF ASSIGNED BIN AREA 9= SUB CATEGORY INDICATOR AFTER RECORD 10= NUMBER OF WORDS REMAINING IN BIN AREA REQUEST CORRELATION 11= UNUSED 12= UNUSED

NAME: SUB			CATEGORY: BIN ALLOCATION DATA
VARIABLE	DIM	TYPE	DESCRIPTION
JN	-	I4	NUMBER OF BINS SET.
ITOT	-	I4	SIZE OF BIN AREA (40000).
KN	-	I4	NUMBER OF BINS IN USE.
JK	-	I4	NOT USED.
AFLAG	99	L1	AUXILIARY OUTPUT REQUEST FLAGS: F==>OUTPUT NOT ENABLED T==>OUTPUT ENABLED
PERF	-	L1	FLAG INDICATING PERFORMANCE SUMMARY REQUESTED.
PRT1	-	L1	FLAG INDICATING STANDARD REPORT 1 REQUESTED.
PRT2	-	L1	FLAG INDICATING STANDARD REPORT 2 REQUESTED.
AMNAME	4,8	L1	FILE MEMBER NAMES USED FOR CURRENT EXPERIMENT (4 NAMES, EACH CONSISTING OF 8 CHARACTERS).
PEROUT	-	L1	FLAG INDICATING PERFORMANCE SUMMARY OUTPUT FILE WRITTEN.

NAME: SYSCOM			CATEGORY: SIMULATION RELATED PARAMETERS
VARIABLE	DIM	TYPE	DESCRIPTION
NLINES	-	I4	OUTPUT DISPLAY PAGE SIZE LIMIT (88)
KNSL	-	I4	UNUSED INTEGER*4 VARIABLE
KNTL	-	I4	FLEET SIZE DEFINED FOR SIMULATION
KNR	-	I4	REQUEST STORAGE TABLE MAX (400)
KNB	-	I4	ALLOCATED BIN AREA MAX (400)
KNA	-	I4	ALLOCATED DATA STORAGE REGION MAX (40000)
CLOCK	-	I4	TIME IN C.U. FOR CURRENT RECORD
CSAMPL	-	I4	SAMPLING INTERVAL C.U.
CSIZE	-	I4	CLOCK GRANULARITY (C.U./MINUTE) FOR SIMULATION
NUMRG	-	I4	NUMBER OF DEFINED ROUTE GROUPS IN SIMULATION
VSEAT	-	I4	VEHICLE SEAT CAPACITY DEFINED IN SIMULATION
PSTART	-	I4	TIME OF FIRST RECORD IN ACCUMULATION INTERVAL
PEND	-	I4	TIME OF LAST RECORD IN ACCUMULATION INTERVAL
NSAMP	-	I4	NUMBER OF RECORDS IN ACCUMULATION INTERVAL

TABLE 3-1 (41 of 53) GLOBAL VARIABLES

NAME: SYSCM1			CATEGORY: SIMULATION RELATED PARAMETERS CONTD
VARIABLE	DIM	TYPE	DESCRIPTION
PHIST1	-	I4	MINI HISTOGRAM CUTOFF VALUE 1
PHIST2	-	I4	MINI HISTOGRAM CUTOFF VALUE 2
NUMS	-	I4	NUMBER OF DEFINED STATIONS IN SIMULATION
NUML	-	I4	NUMBER OF DEFINED GUIDEWAY LNKS IN SIMULATION
NUMSL	-	I4	NUMBER OF DEFINED STATION LNKS IN SIMULATION
NUMR	-	I4	NUMBER OF DEFINED SCHED RTEs IN SIMULATION
VCAP	-	I2	VEHICLE CAPACITY DEFINED IN SIMULATION
STNNO	400	I2	STATION ID FOR STORED USER REQUEST STNNO(1)= REQUEST 1, ETC.
KTYPE	KMSL	I2	STATION LINK TYPES FOR DEFINED STATION LINKS

NAME: TABLES			CATEGORY: REQUEST CORRELATION TABLES
VARIABLE	DIM	TYPE	DESCRIPTION
MAINTA	120	I4	MAIN CATEGORY MNEMONIC TABLE 1= UNUSED 2= SYSTEM STATISTICS - 'SYST' 3= STATION STATISTICS - 'STN' 4= STATION LINK STATISTICS - 'STNL' 5= GUIDEWAY LINK STATISTICS - 'LINK' 6= ROUTE STATISTICS - 'RTE' 7-120 = UNUSED
MCOTAB	120	I2	COLUMN INDICES INTO SUBCATEGORY TABLE FOR EACH MAJOR CATEGORY 1= UNUSED 2= 1 (SYSTEM STATISTICS) 3= 19 (STATION STATISTICS) 4= 25 (STATION LINK STATISTICS) 5= 27 (GUIDEWAY LINK STATISTICS) 6= 30 (ROUTE STATISTICS) 7-120 = UNUSED
MSUTAB	15,35	I4	SUBCATEGORY MNEMONIC DEFINITIONS BY MAJOR CATEGORY. MNEMONICS STORED COLUMN WISE FOR EACH CATEGORY IN SEQUENTIAL ORDER. A ZERO ENTRY TERMINATES LIST FOR CATEGORY. SUB CATEGORY IDS ARE ASSIGNED BASED UPON ORDER OF MNEMONIC IN TABLE. SEE EODATA FOR MNEMONIC DEFINITIONS.
MSUTYP	15,35	I2	SUBCATEGORY MNEMONIC TYPES CURRENTLY UNUSED

TABLE 3-1 (42 of 53) GLOBAL VARIABLES

NAME: TERMNL			CATEGORY: INPUT ECHO FLAG
VARIABLE	DIM	TYPE	DESCRIPTION
XTERM	-	I4	FLAG INDICATING INPUT ECHO

NAME: ZCLNK			CATEGORY: GUIDEWAY LINK STATISTICS
VARIABLE	DIM	TYPE	DESCRIPTION
ZGTIOL	KML	I4	TIME INTEGRAL OF VEHICLE OCCUPANCY.
ZGTIQL	KML	I4	TIME INTEGRAL OF VEHICLE QUEUE OCCUPANCY.
ZGTSCCL	KML	I4	SUM OF COMPLETED LINK TRAVEL TIMES.
ZGDSCCL	KML	I4	SUM OF COMPLETED LINK DISTANCES.
ZGTIPL	KML	I4	TIME INTEGRAL PASSENGERS ON LINK.
GL41	KML	I4	UNDEFINED.
GL42	KML	I4	UNDEFINED.
GL43	KML	I4	UNDEFINED.
ZGVNOL	KML	I2	CURRENT NUMBER OF VEHICLES OCCUPYING LINK.
ZGVNQL	KML	I2	CURRENT NUMBER OF VEHICLES OCCUPYING QUEUE.
ZGMTQL	KML	I2	MAXIMUM TIME OF OCCUPANCY FOR VEHICLES LEAVING.
ZGMNOL	KML	I2	MAXIMUM NUMBER OF VEHICLES ON THE LINK.
ZGNEL	KML	I2	TOTAL # OF VEHICLES ENTERING THE LINK.
ZGNLL	KML	I2	TOTAL # OF VEHICLES LEAVING THE LINK.
ZGNVEL	KML	I2	TOTAL # OF VEHICLES ENTRAINED ON LINK.
ZGNVDL	KML	I2	TOTAL # OF VEHICLES DETAINED ON LINK.
ZGMNQL	KML	I2	MAXIMUM NUMBER OF VEHICLES QUEUED ON LINK
ZGNLQL	KML	I2	TOTAL # OF VEHICLES LEAVING QUEUE.
ZGMDQL	KML	I2	MAXIMUM TIME DELAY FOR VEHICLES LEAVING QUEUE.
ZGSDQL	KML	I2	SUM OF DELAY FOR VEHICLES LEAVING QUEUE.
ZGSTAL	KML	I2	GUIDEWAY LINK STATUS.
ZGMXOL	KML	I2	MINIMUM NUMBER OF VEHICLES OCCUPYING LINK.
ZGXPL	KML	I2	MAXIMUM NUMBER PASSENGERS ON LINK.
ZGMPL	KML	I2	MINIMUM NUMBER PASSENGERS ON GUIDEWAY LINK.
ZGNPL	KML	I2	CURRENT NUMBER OF PASSENGER ON GUIDEWAY LINK.
GL22	KML	I2	UNDEFINED.
GL23	KML	I2	UNDEFINED.

TABLE 3-1 (43 of 53) GLOBAL VARIABLES

NAME: ZCRTE		CATEGORY: ROUTE STATISTICS	
VARIABLE	DIM	TYPE	DESCRIPTION
ZRTSTI	KMR	I4	TIME INTEGRAL OF TRIPS ON ROUTE.
ZRPSTI	KMR	I4	TIME INTEGRAL OF PASSENGERS ON ROUTE.
ZRVMDV	KMR	I4	MAXIMUM SCHEDULE DEVIATION FOR ROUTE.
ZRVTDV	KMR	I4	SUM SCHEDULE DEVIATION FOR ROUTE.
ZRTIDT	KMR	I4	SUM INTERDISPATCH TIME FOR ROUTE.
ZRXIDT	KMR	I4	MAXIMUM INTERDISPATCH TIME FOR ROUTE.
ZRMIDT	KMR	I4	MINIMUM INTERDISPATCH TIME FOR ROUTE.
ZRTIVF	KMR	I4	TIME INTEGRAL VEHICLES IN SERVICE ON ROUTE.
ZRTISP	KMR	I4	TIME INTEGRAL SEATED PASSENGERS ON ROUTE.
ZRTIPW	KMR	I4	TIME INTEGRAL PASSENGERS WAITING ON ROUTE.
ZRVXDV	KMR	I4	MINIMUM SCHEDULE DEVIATION FOR ROUTE.
ZRTSAS	KMR	I4	SUM ACTUAL TRAVEL TIME COMPLETED TRIPS ON ROUTE.
ZRTSNS	KMR	I4	SUM NOMINAL TRAVEL TIME COMPLETED TRIPS ON ROUTE.
ZRXRTT	KMR	I4	MAXIMUM RATIO NOMINAL TT / ACTUAL TT.
ZRMRTT	KMR	I4	MINIMUM RATIO NOMINAL TT / ACTUAL TT.
ZRGVST	KMR	I4	SUM VEHICLE TIMES ON GUIDEWAY.
ZRSVTS	KMR	I4	SUM TIME IN STATIONS FOR VEHICLES LEAVING.
ZRPSDS	KMR	I4	SUM TIME PASSENGERS DEMAND TO DISPATCH.
ZRPUXS	KMR	I4	MAXIMUM PASSENGER TIME DEMAND TO DISPATCH.
ZRPDMS	KMR	I4	MINIMUM PASSENGER TIME DEMAND TO DISPATCH.
ZRGVSD	KMR	I4	SUM VEHICLE DISTANCE ON GUIDEWAY.
ZRPDST	KMR	I4	SUM PASSENGER DISTANCE TRAVELLED.
ZRSdst	KMR	I4	TOTAL DISTANCE TRAVELLED BY VEHICLES LEAVING STATIONS.
ZRTIVL	KML, KMR	I4	TIME INTEGRAL VEHICLE OCCUPANCY GUIDEWAY LINK N, ROUTE M.
ZRTIPL	KML, KMR	I4	TIME INTEGRAL PASSENGER OCCUPANCY GUIDEWAY LINK N, ROUTE M.
RTE42	KMR	I4	UNDEFINED.
RTE43	KMR	I4	UNDEFINED.
ZRTSER	KMR	I2	TOTAL # OF TRIPS SERVED ON ROUTE.
ZRPSER	KMR	I2	NUMBER OF PASSENGERS SERVED ON ROUTE.
ZRTNO	KMR	I2	CURRENT NUMBER OF TRIPS TRAVELLING ROUTE.
ZRPNO	KMR	I2	CURRENT NUMBER OF PASSENGERS TRAVELLING ROUTE.
ZRVDIS	KMR	I2	NUMBER OF VEHICLES DISPATCHED ON ROUTE.
ZRXFLT	KMR	I2	MAXIMUM FLEET SIZE.
ZRMFLT	KMR	I2	MINIMUM FLEET SIZE.
ZRTSXS	KMR	I2	TOTAL # OF ARRIVING TRANSFER PASSENGERS.
ZRNPAR	KMR	I2	TOTAL # OF ARRIVING PASSENGERS.
ZRNPCS	KMR	I2	TOTAL # OF COMPLETED PASSENGERS.
ZRNPT	KMR	I2	TOTAL NUMBER OF PASSENGERS WAITING.
ZRXPWT	KMR	I2	MAXIMUM NUMBER OF PASSENGERS WAITING.
ZRMPWT	KMR	I2	MINIMUM NUMBER OF PASSENGERS WAITING.
ZRPND	KMR	I2	TOTAL NUMBER OF PASSENGERS DISPATCHED.
ZRNPSV	KMR	I2	CURRENT NUMBER OF SEATED PASSENGERS ON VEHICLES.
ZRMVLF	KMR	I2	MINIMUM VEHICLE LOAD FACTOR.
ZRXVLF	KMR	I2	MAXIMUM VEHICLE LOAD FACTOR.
ZRVNOL	KML, KMR	I2	NUMBER OF VEHICLES ON GUIDEWAY LINK N, ROUTE M.
ZRNPL	KML, KMR	I2	NUMBER OF PASSENGERS ON GUIDEWAY LINK N, ROUTE M.
RTE23	KMR	I2	UNDEFINED.

TABLE 3-1 (44 of 53) GLOBAL VARIABLES

NAME: ZCSL			CATEGORY: STATION LINK STATISTICS
VARIABLE	DIM	TYPE	DESCRIPTION
ZSVAN	KMSL, KMS,2	R4	(I,J,1)=AVERAGE # OF VEHICLES OCCUPYING LINK.
ZSVAT	KMSL, KMS,2	R4	(I,J,2)=AVERAGE # OF VEHICLES QUEUED ON LINK. (I,J,1)=AVERAGE TIME OF OCCUPANCY FOR VEHICLES LEAVING. (I,J,2)=AVERAGE QUEUE TIME OF VEHICLES LEAVING QUEUES.
ZSVTI	KMSL, KMS,2	I4	(I,J,1)=TIME INTEGRAL OF VEHICLE OCCUPANCY. (I,J,2)=TIME INTEGRAL OF VEHICLE QUEUE OCCUPANCY.
ZSVST	KMSL, KMS,2	I4	(I,J,1)=SUM OF TIME FOR VEHICLES OCCUPYING LINK. (I,J,2)=SUM OF TIME FOR VEHICLES OCCUPYING LINK QUEUE.
ZSVMT	KMSL, KMS,2	I4	(I,J,1)=MAXIMUM TIME FOR ANY VEHICLE OCCUPYING LINK. (I,J,2)=MAXIMUM TIME FOR ANY VEHICLE OCCUPYING LINK QUEUE.
ZSVNE	KMSL, KMS,2	I2	(I,J,1)=TOTAL # OF VEHICLES ENTERING LINK. (I,J,2)=TOTAL # OF VEHICLES ENTERING LINK QUEUE.
ZSVNL	KMSL, KMS,2	I2	(I,J,1)=TOTAL # OF VEHICLES LEAVING THE LINK. (I,J,2)=TOTAL # OF VEHICLES LEAVING THE LINK QUEUE.
ZSVNI	KMSL, KMS,2	I2	(I,J,1)=CURRENT NUMBER OF VEHICLES OCCUPYING LINK. (I,J,2)=CURRENT NUMBER OF VEHICLES OCCUPYING LINK QUEUE.
ZSVMN	KMSL, KMS,2	I2	(I,J,1)=MAX # OF VEHICLES OCCUPYING LINK. (I,J,2)=MAX # OF VEHICLES OCCUPYING LINK QUEUE.
ZSMXQ	KMSL, KMS,2	I2	(I,J,1)=MIN # OF VEHICLES OCCUPYING LINK. (I,J,2)=MIN # OF VEHICLES OCCUPYING LINK QUEUE.

TABLE 3-1 (45 of 53) GLOBAL VARIABLES

NAME: ZCSTN		CATEGORY: STATION STATISTICS	
VARIABLE	DIM	TYPE	DESCRIPTION
ZSVANS	KMS	R4	AVERAGE NUMBER OF VEHICLES IN STATION.
ZSVATS	KMS	R4	AVERAGE TIME OF VEHICLES IN STATION.
ZSVTIS	KMS	I4	TIME INTEGRAL OF VEHICLES IN STATION.
ZSTTIS	KMS	I4	TIME INTEGRAL OF TRIPS WAITING IN STATION.
ZSPTIS	KMS	I4	TIME INTEGRAL OF PASSENGERS WAITING IN STATION.
ZSVSTS	KMS	I4	SUM OF TIMES FOR VEHICLES LEAVING STATION.
ZSVMTS	KMS	I4	MAXIMUM TIME IN STATION FOR VEHICLES LEAVING.
ZSVTMC	KMS	I4	SUM OF TIMES FOR MERGE CONFLICT RESOLUTION.
ZSTDXS	KMS	I4	MAX DELAY DEMAND TO DISPATCH FOR ANY PASSENGER (= MAX DELAY TRIP GROUP/ # PASS IN GROUP).
ZSTSDS	KMS	I4	SUM DELAY DEMAND TO DISPATCH FOR ALL TRIPS.
ZSPDXS	KMS	I4	MAX DELAY DEMAND TO DISPATCH FOR ANY TRIP GROUP (= DELAY * # PASSENGERS).
ZSPSDS	KMS	I4	SUM DELAY DEMAND TO DISPATCH ALL PASSENGERS.
ZSTSAS	KMS	I4	SUM ACTUAL TRAVEL TIME FOR COMPLETED TRIPS.
ZSTSNS	KMS	I4	SUM OF NOMINAL TRAVEL TIME FOR COMPLETED TRIPS.
ZSMETS	KMS	I4	MAX EXCESS TRAVEL TIME FOR ANY COMPLETED TRIP.
ZSTCAS	KMS	I4	SUM ACTUAL TRAVEL TIME FOR ALL COALESCED TRIPS.
ZSTCNS	KMS	I4	SUM OF NOMINAL TRAVEL TIME FOR COALESCED TRIPS.
ZSMCES	KMS	I4	MAXIMUM EXCESS TRAVEL TIME FOR A COALESCED TRIP.
ZSTDST	KMS	I4	TOTAL DISTANCE TRAVELLED FOR ALL COMPLETING TRIPS.
ZSVNTS	KMS	I4	MINIMUM TIME IN STATION FOR VEHICLES LEAVING.
ZSPDMS	KMS	I4	MINIMUM DELAY DEMAND TO DISPATCH FOR A PASSENGER.
STN41	KMS	I4	UNDEFINED.
STN42	KMS	I4	UNDEFINED.
STN43	KMS	I4	UNDEFINED.
ZSVNES	KMS	I2	TOTAL # OF VEHICLES ENTERING STATION.
ZSVNLS	KMS	I2	TOTAL # OF VEHICLES LEAVING STATION.
ZSVNIS	KMS	I2	CURRENT NUMBER OF VEHICLES IN STATION.
ZSVALT	KMS	I2	TOTAL # VEHICLES DENIED TIMELY ENTRY.
ZSVNLN	KMS	I2	TOTAL # OF VEHICLES LAUNCHED FROM STATION.
ZSVHNS	KMS	I2	MAXIMUM NUMBER OF VEHICLES IN STATION.
ZSTNIS	KMS	I2	TOTAL # OF TRIPS IN STATION INCLUDING THOSE ON VEHICLES CURRENTLY IN STATION.
ZSTNRS	KMS	I2	TOTAL # OF TRIPS REJECTED (BOARDING QUEUE CAPACITY).
ZSTNDS	KMS	I2	TOTAL # OF TRIPS DISPATCHED FROM STATION.
ZSPNIS	KMS	I2	TOTAL # OF PASSENGERS IN STATION INCLUDING THOSE ON VEHICLES CURRENTLY IN STATION.
ZSPNRS	KMS	I2	TOTAL # OF PASSENGERS REJECTED (BOARDING QUEUE CAPACITY).
ZSPNDS	KMS	I2	TOTAL # OF PASSENGERS DISPATCHED FROM STATION.
ZSVREQ	KMS	I2	TOTAL # OF UNSATISFIED EMPTY REQUESTS.
ZSVEDC	KMS	I2	TOTAL # OF EMPTIES DISPATCHED ON CIRCUITOUS ROUTES.
ZSVELS	KMS	I2	TOTAL # OF EMPTIES DISPATCHED TO LOCAL STORAGE.
ZSVERC	KMS	I2	TOTAL # OF EMPTIES DISPATCHED TO REGIONAL CENTER.
ZSVEAN	KMS	I2	TOTAL # OF EMPTIES DISPATCHED BASED ON ANTICIPATED NEED (NOT CONSIDERING CURRENT DISTRIBUTION).
ZSVENB	KMS	I2	TOTAL # OF EMPTIES DISPATCHED BASED ON ANTICIPATED NEED (CONSIDERING CURRENT DISTRIBUTION).

TABLE 3-1 (46 of 53) GLOBAL VARIABLES

NAME: ZCSTN (CONT)			CATEGORY: STATION STATISTICS
VARIABLE	DIM	TYPE	DESCRIPTION
ZSVEDR	KMS	I2	TOTAL # OF EMPTIES DISPATCHED FOR UNSATISFIED REQUESTS.
ZSVED	KMS	I2	TOTAL NUMBER OF EMPTIES DISPATCHED FROM STATION.
ZSTNES	KMS	I2	TOTAL # OF TRIPS ENTERING STATION INCLUDING TRANSFERS & THOSE REJECTED FOR CAPACITY.
ZSTNLS	KMS	I2	TOTAL # OF TRIPS SERVED IN STATION.
ZSPNLS	KMS	I2	TOTAL # OF PASSENGERS SERVED IN STATION.
ZSPNES	KMS	I2	TOTAL # OF PASSENGERS ENTERING STATION INCLUDING TRANSFERS & THOSE REJECTED FOR CAPACITY.
ZSTNCS	KMS	I2	TOTAL # OF TRIPS COMPLETED AT STATION.
ZSPNCS	KMS	I2	TOTAL # OF PASSENGERS COMPLETED AT STATION.
ZSTCNC	KMS	I2	TOTAL # OF COALESCED TRIPS COMPLETED AT STATION.
ZSTXSX	KMS	I2	TOTAL NUMBER OF ARRIVING TRANSFERS AT STATION.
ZSTMXS	KMS	I2	MAXIMUM NUMBER OF TRANSFERS FOR COMPLETED TRIP.
ZSN TAR	KMS	I2	TOTAL # OF TRIPS ARRIVING & ENTERING BOARDING QUEUE.
ZSNPAR	KMS	I2	TOTAL # OF PASSENGERS ARRIVING & ENTERING BOARDING QUEUE.
ZSN TWT	KMS	I2	CURRENT NUMBER OF TRIPS WAITING AT STATION.
ZSXTWT	KMS	I2	MAX NUMBER OF TRIPS WAITING AT STATION.
ZSNPWT	KMS	I2	CURRENT NUMBER OF PASSENGERS WAITING AT STATION.
ZSXPWT	KMS	I2	MAX NUMBER OF PASSENGER WAITING AT STATION.
ZSTNSS	KMS	I2	TOTAL # OF SPLIT TRIPS CREATED.
ZSTNGS	KMS	I2	TOTAL # OF COMPLETED TRIP SUBGROUPS.
ZSVVNS	VMS	I2	TOTAL # OF ENTRAINED VEHICLES LEAVING STATION.
ZSVRNT	KMS	I2	TOTAL # EMPTIES REQUESTED NON-LOCAL STORAGE.
ZSVRLS	KMS	I2	TOTAL # OF EMPTIES REQUESTED LOCAL STORAGE.
ZSTTXS	KMS	I2	TOTAL NUMBER OF TRANSFERS FOR COMPLETED TRIPS.
ZSTPXS	KMS	I2	TOTAL NUMBER PASSENGER XFERS COMPLETED TRIPS (= TOTAL XFERS FOR TRIPS X # PASSENGERS).
ZSMPWT	KMS	I2	MINIMUM NUMBER PASSENGERS WAITING AT STATION.
ZSNPEG	KMS	I2	TOTAL # OF PASSENGERS ENTERING STATION FROM GUIDEWAY.
ZSNPLG	KMS	I2	TOTAL # OF PASSENGERS EXITING STATION TO GUIDEWAY.
ZSVXNS	KMS	I2	MINIMUM NUMBER OF VEHICLES IN STATION.

TABLE 3-1 (47 of 53) GLOBAL VARIABLES

NAME: ZCSYSG			CATEGORY: OVERALL GUIDEWAY STATISTICS
VARIABLE	DIM	TYPE	DESCRIPTION
ZTQDV	-	R4	MAX AVERAGE QUEUE DELAY/VEH FOR ANY LINK.
ZTQDQV	-	R4	MAX AVERAGE QUEUE DELAY/QUEUED VEH FOR ANY LINK.
GTHDWY	-	R4	SUM OF GUIDEWAY LINK HEADWAY TIMES.
ZTTIOL	-	I4	TIME INTEGRAL OF VEHICLE OCCUPANCY ON GUIDEWAY.
ZTTIQL	-	I4	TIME INTEGRAL OF VEHICLE QUEUE OCCUPANCY ON GUIDEWAY.
ZTTIPL	-	I4	TIME INTEGRAL PASSENGERS ON GUIDEWAY.
ZTTSCL	-	I4	SUM OF COMPLETED LINK TRAVEL TIMES ON GUIDEWAY.
ZTDSCL	-	I4	SUM OF COMPLETED LINK DISTANCES ON GUIDEWAY.
SYSG41	-	I4	UNDEFINED.
SYSG42	-	I4	UNDEFINED.
SYSG43	-	I4	UNDEFINED.
ZTVNOL	-	I2	CURRENT NUMBER OF VEHICLES OCCUPYING GUIDEWAY.
ZTVNQL	-	I2	CURRENT NUMBER OF VEHICLES QUEUED ON GUIDEWAY.
ZTMTQL	-	I2	MAXIMUM TIME OF VEHICLE OCCUPANCY FOR VEHICLES CURRENTLY QUEUED ON ANY GUIDEWAY LINK.
ZTMNOL	-	I2	MAXIMUM NUMBER OF VEHICLES ON THE GUIDEWAY.
ZTNEL	-	I2	TOTAL # OF VEHICLES ENTERING ALL GUIDEWAY LINKS.
ZTNLL	-	I2	TOTAL # OF VEHICLES LEAVING ALL GUIDEWAY LINKS.
ZTNVEL	-	I2	TOTAL # OF VEHICLES ENTRAINED ON GUIDEWAY LINKS.
ZTNVDL	-	I2	TOTAL # OF VEHICLES DETAINED ON GUIDEWAY LINKS.
ZTMNQL	-	I2	MAXIMUM NUMBER OF VEHICLES QUEUED ON ANY LINK.
ZTNLQL	-	I2	TOTAL NUMBER OF VEHICLES LEAVING LINK QUEUES.
ZTMDQL	-	I2	MAXIMUM TIME DELAY FOR VEHICLES LEAVING ANY QUEUE.
ZTSDQL	-	I2	SUM OF DELAY FOR VEHS LEAVING GUIDEWAY QUEUES.
GTCAP	-	I2	TOTAL GUIDEWAY CAPACITY.
ZTMXOL	-	I2	MINIMUM NUMBER VEHICLES OCCUPYING ANY LINK.
ZTGXPL	-	I2	MAXIMUM NUMBER PASSENGERS ON ANY LINK.
ZTGMPL	-	I2	MINIMUM NUMBER PASSENGERS ON ANY LINK.
ZTGNPL	-	I2	TOTAL NUMBER OF PASSENGERS ON GUIDEWAY LINKS.
SYSG22	-	I2	UNDEFINED.
SYSG23	-	I2	UNDEFINED.

TABLE 3-1 (48 of 53) GLOBAL VARIABLES

NAME: ZCSYSR		CATEGORY: OVERALL ROUTE STATISTICS	
VARIABLE	DIM	TYPE	DESCRIPTION
ZTTSTI	-	I4	TIME INTEGRAL OF TRIPS ON ALL ROUTES.
ZTPSTI	-	I4	TIME INTEGRAL OF PASSENGERS ON ALL ROUTES.
ZTVMDV	-	I4	MAXIMUM SCHEDULE DEVIATION FOR ANY ROUTE.
ZTVTDV	-	I4	SUM SCHEDULE DEVIATION FOR ALL ROUTES.
ZTTIDT	-	I4	SUM INTERDISPATCH TIME FOR ALL ROUTES.
ZTXIDT	-	I4	MAXIMUM INTERDISPATCH TIME FOR ANY ROUTE.
ZTMIDT	-	I4	MINIMUM INTERDISPATCH TIME FOR ANY ROUTE.
ZiVXDV	-	I4	MINIMUM SCHEDULE DEVIATION FOR ANY ROUTE.
SYSR41	-	I4	UNDEFINED.
SYSR42	-	I4	UNDEFINED.
SYSR43	-	I4	UNDEFINED.
ZTTSER	-	I2	TOTAL # OF TRIPS SERVED ON ROUTES.
ZTPSER	-	I2	TOTAL # OF PASSENGERS SERVED ON ROUTES.
ZTTNO	-	I2	CURRENT # OF TRIPS TRAVELLING ALL ROUTES.
ZTPNO	-	I2	CURRENT # OF PASSENGERS TRAVELLING ALL ROUTES.
ZTVDIS	-	I2	TOTAL # OF VEHICLES DISPATCHED ON ALL ROUTES.
SYSR21	-	I2	UNDEFINED.
SYSR22	-	I2	UNDEFINED.
SYSR23	-	I2	UNDEFINED.

TABLE 3-1 (49 of 53) GLOBAL VARIABLES

NAME: ZCSYSS			CATEGORY: OVERALL STATION STATISTICS
VARIABLE	DIM	TYPE	DESCRIPTION
ZTVTIS	-	I4	TIME INTEGRAL OF VEHICLES IN STATIONS.
ZTTTIS	-	I4	TIME INTEGRAL OF TRIPS WAITING IN STATIONS.
ZTPTIS	-	I4	TIME INTEGRAL OF PASSENGERS WAITING IN STATIONS.
ZVTIIR	-	I4	TIME INTEGRAL OF VEHICLES ON INPUT RAMPS.
ZVTIIQ	-	I4	TIME INTEGRAL OF VEHICLES ON INPUT QUEUES.
ZVTIDK	-	I4	TIME INTEGRAL OF VEHICLES AT DOCKS.
ZVTIOQ	-	I4	TIME INTEGRAL OF VEHICLES ON OUTPUT QUEUES.
ZVTIOR	-	I4	TIME INTEGRAL OF VEHICLES ON OUTPUT RAMPS.
ZVTIST	-	I4	TIME INTEGRAL OF VEHICLES IN STATION STORAGES.
ZQTIIR	-	I4	TIME INTEGRAL OF VEHICLES QUEUED ON INPUT RAMPS.
ZQTIIQ	-	I4	TIME INTEGRAL OF VEHICLES QUEUED ON INPUT QUEUES.
ZQTIDK	-	I4	TIME INTEGRAL OF VEHICLES QUEUED AT DOCKS.
ZQTIOQ	-	I4	TIME INTEGRAL OF VEHICLES QUEUED ON OUTPUT QUEUES.
ZQTIOR	-	I4	TIME INTEGRAL OF VEHICLES QUEUED ON OUTPUT RAMPS.
ZQTIST	-	I4	TIME INTEGRAL OF VEHICLES QUEUED IN STORAGES.
ZTVSTS	-	I4	SUM OF TIMES IN STATIONS FOR VEHICLES LEAVING.
ZTVMTS	-	I4	MAXIMUM TIME IN A STATION FOR VEHICLES LEAVING.
ZTVTMC	-	I4	SUM OF TIME FOR MERGE CONFLICT RESOLUTIONS.
ZTTDXS	-	I4	MAX DELAY DEMAND TO DISPATCH FOR ANY TRIP GROUP.
ZTTSDS	-	I4	SUM DELAY DEMAND TO DISPATCH FOR TRIPS.
ZTPDXS	-	I4	MAX DELAY DEMAND TO DISPATCH ANY PASSENGER.
ZTPSDS	-	I4	SUM DELAY DEMAND TO DISPATCH FOR PASSENGERS.
ZTTSAS	-	I4	SUM ACTUAL TRAVEL TIME FOR COMPLETED TRIPS.
ZTTSNS	-	I4	SUM OF NOMINAL TRAVEL TIME FOR COMPLETED TRIPS.
ZTMETS	-	I4	MAXIMUM EXCESS TRAVEL TIME FOR COMPLETED TRIPS.
ZTTCAS	-	I4	SUM ACTUAL TRAVEL TIME FOR COALESCED TRIPS.
ZTTCNS	-	I4	SUM OF NOMINAL TRAVEL TIME FOR COALESCED TRIPS.
ZDTIR	-	I4	TOTAL QUEUE DELAY VEHICLES LEAVING INPUT RAMPS.
ZDTIQ	-	I4	TOTAL QUEUE DELAY VEHICLES LEAVING INPUT QUEUES.
ZDTDK	-	I4	TOTAL QUEUE DELAY VEHICLES LEAVING DOCKS.
ZDTOQ	-	I4	TOTAL QUEUE DELAY VEHICLES LEAVING OUTPUT QUEUES.
ZDTOR	-	I4	TOTAL QUEUE DELAY VEHICLES LEAVING OUTPUT RAMPS.
ZDTST	-	I4	TOTAL QUEUE DELAY VEHICLES LEAVING STORAGE AREAS.
ZTMCES	-	I4	MAXIMUM EXCESS TRAVEL TIME FOR COALESCED TRIPS.
ZTPDMS	-	I4	MINIMUM TIME DEMAND TO DISPATCH PASSENGERS.
ZTVNTS	-	I4	MINIMUM TIME IN ANY STATION FOR VEHICLES LEAVING.
ZXDIR	-	I4	MAXIMUM TIME FOR VEHICLES LEAVING IR QUEUE.
ZXDIOQ	-	I4	MAXIMUM TIME FOR VEHICLES LEAVING IQ QUEUE.
ZXDDK	-	I4	MAXIMUM TIME FOR VEHICLES LEAVING DK QUEUE.
ZXDOQ	-	I4	MAXIMUM TIME FOR VEHICLES LEAVING OQ QUEUE.
ZXDOR	-	I4	MAXIMUM TIME FOR VEHICLES LEAVING OR QUEUE.
ZXDST	-	I4	MAXIMUM TIME FOR VEHICLES LEAVING ST QUEUE.
ZTTDST	-	I4	TOTAL DISTANCE TRAVELLED FOR COMPLETED TRIPS.
ZTSDST	-	I4	TOTAL STATION DISTANCE FOR VEHICLES LEAVING.

TABLE 3-1 (50 of 53) GLOBAL VARIABLES

NAME: ZCSYSS (CONT)			CATEGORY: OVERALL STATION STATISTICS
VARIABLE	DIM	TYPE	DESCRIPTION
ZTSTOP	-	I4	TOTAL NUMBER OF INTERMEDIATE PASSENGER STOPS.
SYSS43	-	I4	UNDEFINED.
ZTVNES	-	I2	TOTAL # OF VEHICLES ENTERING STATIONS.
ZTVNLS	-	I2	TOTAL # OF VEHICLES LEAVING STATIONS.
ZTVNIS	-	I2	CURRENT NUMBER OF VEHICLES IN STATIONS.
ZTVALT	-	I2	TOTAL # OF VEHICLES DENIED TIMELY ENTRY.
ZTVNLN	-	I2	TOTAL # OF VEHICLES LAUNCHED FROM STATIONS.
ZTVMNS	-	I2	MAXIMUM NUMBER OF VEHICLES IN ANY STATION.
ZTTNIS	-	I2	CURRENT # OF TRIPS IN STATIONS INCLUDING THOSE ON VEHICLES IN STATION.
ZTTNRS	-	I2	TOTAL # OF TRIPS REJECTED (BOARDING QUEUE CAPACITY).
ZTTNDS	-	I2	TOTAL # OF TRIPS DISPATCHED FROM STATIONS.
ZTPNIS	-	I2	CURRENT # PASSENGERS IN STATIONS INCLUDING THOSE ON VEHICLES IN STATION.
ZTPNRS	-	I2	TOTAL # OF PASSENGERS REJECTED (BOARDING QUEUE CAPACITY).
ZTPNDS	-	I2	TOTAL # OF PASSENGERS DISPATCHED FROM STATIONS EXCLUDING TRANSFERS.
ZTVREQ	-	I2	TOTAL # OF UNSATISFIED EMPTY VEHICLE REQUESTS.
ZTVEDC	-	I2	TOTAL # OF EMPTIES DISPATCHED ON CIRCUITOUS ROUTES.
ZTVELS	-	I2	TOTAL # OF EMPTIES DISPATCHED TO LOCAL STORAGE.
ZTVERC	-	I2	TOTAL # OF EMPTIES DISPATCHED TO REGIONAL CENTERS.
ZTVEAN	-	I2	TOTAL # OF EMPTIES DISPATCHED BASED ON ANTICIPATED NEED (NOT CONSIDERING CURRENT DISTRIBUTION).
ZTVENB	-	I2	TOTAL # OF EMPTIES DISPATCHED BASED ON ANTICIPATED NEED (CONSIDERING CURRENT DISTRIBUTION).
ZTVEDR	-	I2	TOTAL # OF EMPTIES DISPATCHED TO OUTSTANDING REQUESTS.
ZTVED	-	I2	TOTAL NUMBER OF EMPTIES DISPATCHED.
ZTTNES	-	I2	TOTAL # OF TRIPS ENTERING STATIONS INCLUDING TRANSFERS & REJECTIONS FOR CAPACITY.
ZTTNLS	-	I2	TOTAL # OF TRIPS SERVED IN STATIONS.
ZTPNLS	-	I2	TOTAL # OF PASSENGERS SERVED IN STATIONS.
ZTPNES	-	I2	TOTAL # OF PASSENGERS ENTERING STATIONS INCLUDING TRANSFERS & THOSE REJECTED FOR CAPACITY.
ZTTNCS	-	I2	TOTAL # OF TRIPS COMPLETED AT STATIONS.
ZTPNCS	-	I2	TOTAL # OF PASSENGERS COMPLETED AT STATIONS.
ZTTCNC	-	I2	TOTAL # OF COALESCED TRIPS COMPLETED AT STATIONS.
ZTTSXS	-	I2	TOTAL NUMBER OF ARRIVING TRANSFERS AT STATIONS.
ZTTMXS	-	I2	MAXIMUM NUMBER OF TRANSFERS FOR ANY COMPLETED TRIP.
ZTNTAR	-	I2	TOTAL # OF TRIPS ARRIVING AT STATIONS AND ENTERING BOARDING QUEUE INCLUDING TRANSFERS.
ZTNPAR	-	I2	TOTAL # OF PASSENGERS ARRIVING AT STATIONS & ENTERING BOARDING QUEUE INCLUDING TRANSFERS.
ZTNTWT	-	I2	CURRENT NUMBER OF TRIPS WAITING AT STATIONS.
ZTXTWT	-	I2	MAX NUMBER OF TRIPS WAITING IN ALL STATIONS.
ZTNPWT	-	I2	CURRENT NUMBER PASSENGERS WAITING IN STATIONS.
ZTXPWT	-	I2	MAX NUMBER OF PASSENGER WAITING IN ALL STATIONS.
ZTTNSS	-	I2	TOTAL # OF SPLIT TRIPS CREATED (SUBGROUPS).
ZTTNGS	-	I2	TOTAL # OF COMPLETED TRIP SUBGROUPS AT STATIONS.

TABLE 3-1 (51 of 53) GLOBAL VARIABLES

NAME: ZCSYSS (CONT)			CATEGORY: OVERALL STATION STATISTICS
VARIABLE	DIM	TYPE	DESCRIPTION
ZTVNNS	-	I2	TOTAL # OF ENTRAINED VEHICLES LEAVING STATIONS.
ZMOIR	-	I2	MAXIMUM NUMBER OF VEHICLES ON INPUT RAMPS.
ZMOIQ	-	I2	MAXIMUM NUMBER OF VEHICLES ON INPUT QUEUES.
ZMODK	-	I2	MAXIMUM NUMBER OF VEHICLES AT DOCKS.
ZMOOQ	-	I2	MAXIMUM NUMBER OF VEHICLES ON OUTPUT QUEUES.
ZMOOR	-	I2	MAXIMUM NUMBER OF VEHICLES ON OUTPUT RAMPS.
ZMOST	-	I2	MAXIMUM NUMBER OF VEHICLES IN STORAGE AREAS.
ZMQIR	-	I2	MAXIMUM NUMBER OF VEHICLES QUEUED ON INPUT RAMPS.
ZMQIQ	-	I2	MAXIMUM NUMBER OF VEHICLES QUEUED ON INPUT QUEUES.
ZMQDK	-	I2	MAXIMUM NUMBER OF VEHICLES QUEUED AT DOCKS.
ZMQOQ	-	I2	MAXIMUM NUMBER OF VEHICLES QUEUED ON OUTPUT QUEUES.
ZMQOR	-	I2	MAXIMUM NUMBER OF VEHICLES QUEUED ON OUTPUT RAMPS.
ZMQST	-	I2	MAXIMUM NUMBER OF VEHICLES QUEUED IN STORAGE.
ZNQIR	-	I2	TOTAL NUMBER VEHICLES LEAVING INPUT RAMPS.
ZNQIQ	-	I2	TOTAL NUMBER VEHICLES LEAVING INPUT QUEUES.
ZNQDK	-	I2	TOTAL NUMBER VEHICLES LEAVING DOCKS.
ZNQOQ	-	I2	TOTAL NUMBER VEHICLES LEAVING OUTPUT QUEUES.
ZNQOR	-	I2	TOTAL NUMBER VEHICLES LEAVING OUTPUT RAMPS.
ZNQST	-	I2	TOTAL NUMBER VEHICLES LEAVING STORAGE AREAS.
ZNEIR	-	I2	TOTAL NUMBER VEHICLES ENTERING INPUT RAMPS.
ZNEIQ	-	I2	TOTAL NUMBER VEHICLES ENTERING INPUT QUEUES.
ZNEDK	-	I2	TOTAL NUMBER VEHICLES ENTERING DOCKS.
ZNEOQ	-	I2	TOTAL NUMBER VEHICLES ENTERING OUTPUT QUEUES.
ZNEOR	-	I2	TOTAL NUMBER VEHICLES ENTERING OUTPUT RAMPS.
ZNEST	-	I2	TOTAL NUMBER VEHICLES ENTERING STORAGE AREAS.
ZTVRNT	-	I2	TOTAL NUMBER EMPTIES REQUESTED FROM NON-LOCAL STORAGE AREAS.
ZTVRLS	-	I2	TOTAL NUMBER EMPTIES REQUESTED FROM LOCAL STORAGE AREAS.
ZTTTXS	-	I2	TOTAL NUMBER OF TRANSFERS FOR COMPLETED TRIPS.
ZTTPXS	-	I2	TOTAL NUMBER OF PASSENGER TRANSFERS FOR COMPLETED TRIPS (# XFERS X # PASSENGERS).
ZTMPWT	-	I2	MINIMUM NUMBER OF PASSENGERS WAITING IN STATIONS.
ZTMQIR	-	I2	MINIMUM NUMBER VEHICLES QUEUED INPUT RAMPS.
ZTMQIQ	-	I2	MINIMUM NUMBER VEHICLES QUEUED INPUT QUEUES.
ZTMQDK	-	I2	MINIMUM NUMBER VEHICLES QUEUED DOCK AREAS.
ZTMQOQ	-	I2	MINIMUM NUMBER VEHICLES QUEUED OUTPUT QUEUES.
ZTMQOR	-	I2	MINIMUM NUMBER VEHICLES QUEUED OUTPUT RAMPS.
ZTMQST	-	I2	MINIMUM NUMBER VEHICLES QUEUED STORAGE AREAS.
ZTXFER	-	I2	TOTAL # OF COMPLETED PASSENGERS HAVING TRANSFERRED AT LEAST ONCE.
SYSS22	-	I2	UNDEFINED.
SYSS23	-	I2	UNDEFINED.

TABLE 3-1 (52 of 53) GLOBAL VARIABLES

NAME: ZCSYST		CATEGORY: OVERALL SYSTEM STATISTICS	
VARIABLE	DIM	TYPE	DESCRIPTION
ZTTIRV	-	I4	TIME INTEGRAL OF REVENUE SERVICE (OCCUPIED) VEHICLES.
ZTTIDH	-	I4	TIME INTEGRAL OF DEADHEADING (EMPTY) VEHICLES.
ZTTISV	-	I4	TIME INTEGRAL OF VEHICLES IN STORAGE.
ZTTITV	-	I4	TIME INTEGRAL OF TRIPS ON VEHICLES.
ZPTITV	-	I4	TIME INTEGRAL OF PASSENGERS ON VEHICLES.
ZTSEAT	-	I4	TIME INTEGRAL OF SEATED PASSENGERS ON VEHICLES.
ZTSDCS	-	I4	SUM OF TIMES DEMAND TO COMPLETION FOR COMPLETED TRIPS.
ZTXRTT	-	I4	MAXIMUM RATIO NOMINAL TT / ACTUAL TT.
ZTMRTT	-	I4	MINIMUM RATIO NOMINAL TT / ACTUAL TT.
ZDTDST	-	I4	SUM OF PASSENGER DISTANCE TRAVELLED ON GUIDEWAY.
ZDDDST	-	I4	SUM OF VEHICLE DEADHEADING (EMPTY) DISTANCE.
ZDRDST	-	I4	SUM OF REVENUE SERVICE (OCCUPIED) DISTANCE.
ZTVDST	-	I4	TOTAL VEHICLE DISTANCE TRAVELLED (STN & GDWY).
SYST42	-	I4	UNDEFINED.
SYST43	-	I4	UNDEFINED.
ZNTOV	-	I2	CURRENT NUMBER OF TRIPS ON VEHICLES.
ZNPOV	-	I2	CURRENT NUMBER OF PASSENGERS ON VEHICLES.
ZNVRVS	-	I2	CURRENT NUMBER OF VEHICLES IN REVENUE SERVICE (OCCUPIED).
ZNVDEH	-	I2	CURRENT NUMBER OF VEHICLES DEADHEADING(EMPTY).
ZNVSTO	-	I2	CURRENT NUMBER OF VEHICLES IN STORAGE.
ZNTT1	-	I2	# TRIPS EXCESS TRAVEL TT <= PHIST1.
ZNTT2	-	I2	# TRIPS EXCESS TRAVEL TT >PHIST1 & <=PHIST2.
ZNTT3	-	I2	# TRIPS EXCESS TRAVEL TIME >PHIST2.
ZNPP1	-	I2	# PASSENGERS EXCESS TRAVEL TT <= PHIST1.
ZNPP2	-	I2	# PASSENGERS EXCESS TRAVEL TT >PHIST1 & <=PHIST2.
ZNPP3	-	I2	# PASSENGERS EXCESS TRAVEL TIME >PHIST2.
ZTVRVS	-	I2	TOTAL NUMBER OF VEHICLES ENTERING REVENUE (OCCUPIED) STATE.
ZTVDEH	-	I2	TOTAL NUMBER OF VEHICLES ENTERING EMPTY STATE.
ZTVSTO	-	I2	TOTAL NUMBER OF VEHICLES ENTERING STORAGE STATE.
ZTPSVD	-	I2	TOTAL PASSENGERS SERVED EXCLUDING ARRIVING TRANSFERS.
ZTXFLT	-	I2	MAXIMUM FLEET SIZE.
ZTMFLT	-	I2	MINIMUM FLEET SIZE.
ZNPSV	-	I2	CURRENT NUMBER OF SEATED PASSENGERS ON VEHICLES.
ZTMVLF	-	I2	MINIMUM VEHICLE LOAD FACTOR.
ZTXVLF	-	I2	MAXIMUM VEHICLE LOAD FACTOR.
SYST23	-	I2	UNDEFINED.

TABLE 3-1 (53 of 53) GLOBAL VARIABLES

NAME: ZCTR			CATEGORY: TRIP STATISTICS
VARIABLE	DIM	TYPE	DESCRIPTION
ZTTSAT	KMT	I4	SUM OF ACTUAL TRAVEL TIME.
ZTTSNT	KMT	I4	SUM OF NOMINAL TRAVEL TIME.
ZTNTXF	KMT	I4	TOTAL NUMBER OF TRANSFERS.
ZTTDIS	KMT	I4	TRAVEL INITIATION TIME.
ZTRDST	KMT	I4	DISTANCE TRAVELED ON GUIDEWAY.
ZTDDT	KMT	I4	DISPATCH TIME FROM ORIGIN STATION.
ZTXFRT	KMT	I4	SUM OF TRANSFER TIME.
ZTARRT	KMT	I4	ARRIVAL TIME AT LAST INTERMEDIATE DESTINATION.

NAME: ZCVEH			CATEGORY: VEHICLE STATISTICS
VARIABLE	DIM	TYPE	DESCRIPTION
ZVTDST	KMV	I4	TOTAL DISTANCE TRAVELED ON GUIDEWAY.
ZVDDST	KMV	I4	TOTAL DISTANCE TRAVELED EMPTY ON GUIDEWAY.
ZVRDST	KMV	I4	TOTAL DISTANCE TRAVELED OCCUPIED ON GUIDEWAY.
ZVTES	KMV	I4	TIME OF ENTRY TO STATION.
ZVSLTM	KMV	I4	TIME OF ENTRY TO STATION LINK.
ZVSQTM	KMV	I4	1) TIME OF ENTRY INTO QUEUE. 2) TOTAL MERGE SCHEDULING DELAY
ZVSTAT	KMV	I2	STATE OF VEHICLE (1 = OCCUPIED, 2 = EMPTY, 3 = STORED).
ZVMNQO	KMV	I2	MAXIMUM NUMBER OF VEHICLES IN QUEUE AT ENTRY.
ZVSEAT	KMV	I2	NUMBER OF SEATED PASSENGERS.

TABLE 3-1. GLOBAL VARIABLES (Page 54 of 63)

NAME: ZCSTN			CATEGORY: STATION STATISTICS
VARIABLE	DIM	TYPE	DESCRIPTION
ZSVANS	KMS	R4	AVERAGE NUMBER OF VEHICLES IN STATION.
ZSVATS	KMS	R4	AVERAGE TIME OF VEHICLES IN STATION.
ZSVTIS	KMS	I4	TIME INTEGRAL OF VEHICLES IN STATION.
ZSTTIS	KMS	I4	TIME INTEGRAL OF TRIPS WAITING IN STATION.
ZSPTIS	KMS	I4	TIME INTEGRAL OF PASSENGERS WAITING IN STATION.
ZSVSTS	KMS	I4	SUM OF TIMES FOR VEHICLES LEAVING STATION.
ZSVMTS	KMS	I4	MAXIMUM TIME IN STATION FOR VEHICLES LEAVING.
ZSVTMC	KMS	I4	SUM OF TIMES FOR MERGE CONFLICT RESOLUTION.
ZSTOXS	KMS	I4	MAX DELAY DEMAND TO DISPATCH FOR ANY PASSENGER (= MAX DELAY TRIP GROUP/# PASS IN GROUP).
ZSTSDS	KMS	I4	SUM DELAY DEMAND TO DISPATCH FOR ALL TRIPS.
ZSPOXS	KMS	I4	MAX DELAY DEMAND TO DISPATCH FOR ANY TRIP GROUP (= DELAY * # PASSENGERS).
ZSPSDS	KMS	I4	SUM DELAY DEMAND TO DISPATCH ALL PASSENGERS.
ZSTSAS	KMS	I4	SUM ACTUAL TRAVEL TIME FOR COMPLETED TRIPS.
ZSTSNS	KMS	I4	SUM OF NOMINAL TRAVEL TIME FOR COMPLETED TRIPS.
ZSMETS	KMS	I4	MAX EXCESS TRAVEL TIME FOR ANY COMPLETED TRIP.
ZSTCAS	KMS	I4	SUM ACTUAL TRAVEL TIME FOR ALL COALESCED TRIPS.
ZSTCNS	KMS	I4	SUM OF NOMINAL TRAVEL TIME FOR COALESCED TRIPS.
ZSMCES	KMS	I4	MAXIMUM EXCESS TRAVEL TIME FOR A COALESCED TRIP.
ZSTDST	KMS	I4	TOTAL DISTANCE TRAVELLED FOR ALL COMPLETING TRIPS.
ZSVNTS	KMS	I4	MINIMUM TIME IN STATION FOR VEHICLES LEAVING.
ZSPDMS	KMS	I4	MINIMUM DELAY DEMAND TO DISPATCH FOR A PASSENGER.
STN41	KMS	I4	UNDEFINED.
STN42	KMS	I4	UNDEFINED.
STN43	KMS	I4	UNDEFINED.
ZSVNES	KMS	I2	TOTAL # OF VEHICLES ENTERING STATION.
ZSVNLS	KMS	I2	TOTAL # OF VEHICLES LEAVING STATION.
ZSVNIS	KMS	I2	CURRENT NUMBER OF VEHICLES IN STATION.
ZSVALT	KMS	I2	TOTAL # OF VEHICLES DENIED TIMELY ENTRY.
ZSVNLN	KMS	I2	TOTAL # OF VEHICLES LAUNCHED FROM STATION.
ZSVMNS	KMS	I2	MAXIMUM NUMBER OF VEHICLES IN STATION.
ZSTNIS	KMS	I2	TOTAL # OF TRIPS IN STATION INCLUDING THOSE ON VEHICLES CURRENTLY IN STATION.
ZSTNRS	KMS	I2	TOTAL # OF TRIPS REJECTED (BOARDING QUEUE CAPACITY).
ZSTNDS	KMS	I2	TOTAL # OF TRIPS DISPATCHED FROM STATION.
ZSPNIS	KMS	I2	TOTAL # OF PASSENGERS IN STATION INCLUDING THOSE ON VEHICLES CURRENTLY IN STATION.
ZSPNRS	KMS	I2	TOTAL # OF PASSENGERS REJECTED (BOARDING QUEUE CAPACITY).
ZSPNDS	KMS	I2	TOTAL # OF PASSENGERS DISPATCHED FROM STATION.
ZSVREQ	KMS	I2	TOTAL # OF UNSATISFIED EMPTY REQUESTS.
ZSVEDC	KMS	I2	TOTAL # OF EMPTIES DISPATCHED ON CIRCUITOUS ROUTES.
ZSVELS	KMS	I2	TOTAL # OF EMPTIES DISPATCHED TO LOCAL STORAGE.
ZSVERC	KMS	I2	TOTAL # OF EMPTIES DISPATCHED TO REGIONAL CENTER.
ZSVEAN	KMS	I2	TOTAL # OF EMPTIES DISPATCHED BASED ON ANTICIPATED NEED (NOT CONSIDERING CURRENT DISTRIBUTION).
ZSVENB	KMS	I2	TOTAL # OF EMPTIES DISPATCHED BASED ON ANTICIPATED NEED (CONSIDERING CURRENT DISTRIBUTION).

TABLE 3-1. GLOBAL VARIABLES (Page 55 of 63)

NAME: ZCSTN (CUNT)			CATEGORY: STATION STATISTICS
VARIABLE	DIM	TYPE	DESCRIPTION
ZSVEDR	KMS	I2	TOTAL # OF EMPTIES DISPATCHED FOR UNSATISFIED REQUESTS.
ZSVED	KMS	I2	TOTAL NUMBER OF EMPTIES DISPATCHED FROM STATION.
ZSTNES	KMS	I2	TOTAL # OF TRIPS ENTERING STATION INCLUDING TRANSFERS & THOSE REJECTED FOR CAPACITY.
ZSTNLS	KMS	I2	TOTAL # OF TRIPS SERVED IN STATION.
ZSPNLS	KMS	I2	TOTAL # OF PASSENGERS SERVED IN STATION.
ZSPNES	KMS	I2	TOTAL # OF PASSENGERS ENTERING STATION INCLUDING TRANSFERS & THOSE REJECTED FOR CAPACITY.
ZSTNCS	KMS	I2	TOTAL # OF TRIPS COMPLETED AT STATION.
ZSPNCS	KMS	I2	TOTAL # OF PASSENGERS COMPLETED AT STATION.
ZSTCNC	KMS	I2	TOTAL # OF COALESCED TRIPS COMPLETED AT STATION.
ZSTSXS	KMS	I2	TOTAL NUMBER OF ARRIVING TRANSFERS AT STATION.
ZSTMXS	KMS	I2	MAXIMUM NUMBER OF TRANSFERS FOR COMPLETED TRIP.
ZSN TAR	KMS	I2	TOTAL # OF TRIPS ARRIVING & ENTERING BOARDING QUEUE.
ZSNPAR	KMS	I2	TOTAL # OF PASSENGERS ARRIVING & ENTERING BOARDING QUEUE.
ZSN TWT	KMS	I2	CURRENT NUMBER OF TRIPS WAITING AT STATION.
ZSXTWT	KMS	I2	MAX NUMBER OF TRIPS WAITING AT STATION.
ZSNPWT	KMS	I2	CURRENT NUMBER OF PASSENGERS WAITING AT STATION.
ZS WT	KMS	I2	MAX NUMBER OF PASSENGER WAITING AT STATION.
ZSTNSS	KMS	I2	TOTAL # OF SPLIT TRIPS CREATED.
ZSTNGS	KMS	I2	TOTAL # OF COMPLETED TRIP SUBGROUPS.
ZSVNNS	KMS	I2	TOTAL # OF ENTRAINED VEHICLES LEAVING STATION.
ZSVRNT	KMS	I2	TOTAL # EMPTIES REQUESTED NON-LOCAL STORAGE.
ZSVRLS	KMS	I2	TOTAL # OF EMPTIES REQUESTED LOCAL STORAGE.
ZSTTXS	KMS	I2	TOTAL NUMBER OF TRANSFERS FOR COMPLETED TRIPS.
ZSTPXS	KMS	I2	TOTAL NUMBER PASSENGER XFER: COMPLETED TRIPS (= TOTAL XFERS FOR TRIPS X # PASSENGERS).
ZSNPWT	KMS	I2	MINIMUM NUMBER PASSENGERS WAITING AT STATION.
ZSNPEG	KMS	I2	TOTAL # OF PASSENGERS ENTERING STATION FROM GUIDEWAY.
ZSNPLG	KMS	I2	TOTAL # OF PASSENGERS EXITING STATION TO GUIDEWAY.
ZSVXNS	KMS	I2	MINIMUM NUMBER OF VEHICLES IN STATION.

TABLE 3-1. GLOBAL VARIABLES (Page 56 of 63)

NAME: ZCSYSG			CATEGORY: OVERALL GUIDEWAY STATISTICS
VARIABLE	DIM	TYPE	DESCRIPTION
ZTQDV	-	I4	MAX AVERAGE QUEUE DELAY/VEH FOR ANY LINK.
ZTODQV	-	I4	MAX AVERAGE QUEUE DELAY/QUEUED VEH FOR ANY LINK.
GTHDWY	-	I4	SUM OF GUIDEWAY LINK HEADWAY TIMES.
ZTTIOL	-	I4	TIME INTEGRAL OF VEHICLE OCCUPANCY ON GUIDEWAY.
ZTTIQL	-	I4	TIME INTEGRAL OF VEHICLE QUEUE OCCUPANCY ON GUIDEWAY.
ZTTIPL	-	I4	TIME INTEGRAL PASSENGERS ON GUIDEWAY.
ZTTSCL	-	I4	SUM OF COMPLETED LINK TRAVEL TIMES ON GUIDEWAY.
ZTDSCL	-	I4	SUM OF COMPLETED LINK DISTANCES ON GUIDEWAY.
SYSG41	-	I4	UNDEFINED.
SYSG42	-	I4	UNDEFINED.
SYSG43	-	I4	UNDEFINED.
ZTVNQL	-	I2	CURRENT NUMBER OF VEHICLES OCCUPYING GUIDEWAY.
ZTVNQL	-	I2	CURRENT NUMBER OF VEHICLES QUEUED ON GUIDEWAY.
ZTMTQL	-	I2	MAXIMUM TIME OF VEHICLE OCCUPANCY FOR VEHICLES CURRENTLY QUEUED ON ANY GUIDEWAY LINK.
ZTMNQL	-	I2	MAXIMUM NUMBER OF VEHICLES ON THE GUIDEWAY.
ZTNEL	-	I2	TOTAL # OF VEHICLES ENTERING ALL GUIDEWAY LINKS.
ZTNLL	-	I2	TOTAL # OF VEHICLES LEAVING ALL GUIDEWAY LINKS.
ZTNVEL	-	I2	TOTAL # OF VEHICLES ENTRAINED ON GUIDEWAY LINKS.
ZTNVOL	-	I2	TOTAL # OF VEHICLES DETRAINED ON GUIDEWAY LINKS.
ZTMNQL	-	I2	MAXIMUM NUMBER OF VEHICLES QUEUED ON ANY LINK.
ZTNLQL	-	I2	TOTAL NUMBER OF VEHICLES LEAVING LINK QUEUES.
ZTMDQL	-	I2	MAXIMUM TIME DELAY FOR VEHICLES LEAVING ANY QUEUE.
ZTSDQL	-	I2	SUM OF DELAY FOR VEHs LEAVING GUIDEWAY QUEUES.
GTCAP	-	I2	TOTAL GUIDEWAY CAPACITY.
ZTMXOL	-	I2	MINIMUM NUMBER VEHICLES OCCUPYING ANY LINK.
ZTGXPL	-	I2	MAXIMUM NUMBER PASSENGERS ON ANY LINK.
ZTGMPPL	-	I2	MINIMUM NUMBER PASSENGERS ON ANY LINK.
ZTGNPL	-	I2	TOTAL NUMBER OF PASSENGERS ON GUIDEWAY LINKS.
SYSG22	-	I2	UNDEFINED.
SYSG23	-	I2	UNDEFINED.

TABLE 3-1. GLOBAL VARIABLES (page 57 of 63)

NAME: ZCSYSR		CATEGORY: OVERALL ROUTE STATISTICS	
VARIABLE	DIM	TYPE	DESCRIPTION
ZTTST1	-	I4	TIME INTEGRAL OF TRIPS ON ALL ROUTES.
ZTPST1	-	I4	TIME INTEGRAL OF PASSENGERS ON ALL ROUTES.
ZTVMDV	-	I4	MAXIMUM SCHEDULE DEVIATION FOR ANY ROUTE.
ZTVTDV	-	I4	SUM SCHEDULE DEVIATION FOR ALL ROUTES.
ZTTIDT	-	I4	SUM INTERDISPATCH TIME FOR ALL ROUTES.
ZTXIDT	-	I4	MAXIMUM INTERDISPATCH TIME FOR ANY ROUTE.
ZTMIDT	-	I4	MINIMUM INTERDISPATCH TIME FOR ANY ROUTE.
ZTVXDV	-	I4	MINIMUM SCHEDULE DEVIATION FOR ANY ROUTE.
SYSR41	-	I4	UNDEFINED.
SYSR42	-	I4	UNDEFINED.
SYSR43	-	I4	UNDEFINED.
ZTTSER	-	I2	TOTAL # OF TRIPS SERVED ON ROUTES.
ZTPSER	-	I2	TOTAL # OF PASSENGERS SERVED ON ROUTES.
ZTTNO	-	I2	CURRENT # OF TRIPS TRAVELLING ALL ROUTES.
ZTPNO	-	I2	CURRENT # OF PASSENGERS TRAVELLING ALL ROUTES.
ZTVDIS	-	I2	TOTAL # OF VEHICLES DISPATCHED ON ALL ROUTES.
SYSR21	-	I2	UNDEFINED.
SYSR22	-	I2	UNDEFINED.
SYSR23	-	I2	UNDEFINED.

TABLE 3-1. GLOBAL VARIABLES (Page 58 of 63)

NAME: ZCSYSS			CATEGORY: OVERALL STATION STATISTICS
VARIABLE	DIM	TYPE	DESCRIPTION
ZTVTIS	-	I4	TIME INTEGRAL OF VEHICLES IN STATIONS.
ZTFTIS	-	I4	TIME INTEGRAL OF TRIPS WAITING IN STATIONS.
ZTPTIS	-	I4	TIME INTEGRAL OF PASSENGERS WAITING IN STATIONS.
ZVTIIR	-	I4	TIME INTEGRAL OF VEHICLES ON INPUT RAMPS.
ZVTIIQ	-	I4	TIME INTEGRAL OF VEHICLES ON INPUT QUEUES.
ZVTIDK	-	I4	TIME INTEGRAL OF VEHICLES AT DOCKS.
ZVTIOQ	-	I4	TIME INTEGRAL OF VEHICLES ON OUTPUT QUEUES.
ZVTIOR	-	I4	TIME INTEGRAL OF VEHICLES ON OUTPUT RAMPS.
ZVTIST	-	I4	TIME INTEGRAL OF VEHICLES IN STATION STORAGES.
ZQTIIR	-	I4	TIME INTEGRAL OF VEHICLES QUEUED ON INPUT RAMPS.
ZQTIIO	-	I4	TIME INTEGRAL OF VEHICLES QUEUED ON INPUT QUEUES.
ZQTIQK	-	I4	TIME INTEGRAL OF VEHICLES QUEUED AT DOCKS.
ZQTIQQ	-	I4	TIME INTEGRAL OF VEHICLES QUEUED ON OUTPUT QUEUES.
ZQTIOR	-	I4	TIME INTEGRAL OF VEHICLES QUEUED ON OUTPUT RAMPS.
ZQTIIST	-	I4	TIME INTEGRAL OF VEHICLES QUEUED IN STORAGES.
ZTVSTS	-	I4	SUM OF TIMES IN STATIONS FOR VEHICLES LEAVING.
ZTVMTS	-	I4	MAXIMUM TIME IN A STATION FOR VEHICLES LEAVING.
ZTVTMC	-	I4	SUM OF TIME FOR MERGE CONFLICT RESOLUTIONS.
ZTTDXS	-	I4	MAX DELAY DEMAND TO DISPATCH FOR ANY TRIP GROUP.
ZTTSOS	-	I4	SUM DELAY DEMAND TO DISPATCH FOR TRIPS.
ZTPDXS	-	I4	MAX DELAY DEMAND TO DISPATCH ANY PASSENGER.
ZTPSDS	-	I4	SUM DELAY DEMAND TO DISPATCH FOR PASSENGERS.
ZTTSAS	-	I4	SUM ACTUAL TRAVEL TIME FOR COMPLETED TRIPS.
ZTTSNS	-	I4	SUM OF NOMINAL TRAVEL TIME FOR COMPLETED TRIPS.
ZTMETS	-	I4	MAXIMUM EXCESS TRAVEL TIME FOR COMPLETED TRIPS.
ZTTCAS	-	I4	SUM ACTUAL TRAVEL TIME FOR COALESCED TRIPS.
ZTTCNS	-	I4	SUM OF NOMINAL TRAVEL TIME FOR COALESCED TRIPS.
ZDTIR	-	I4	TOTAL QUEUE DELAY VEHICLES LEAVING INPUT RAMPS.
ZDTIQ	-	I4	TOTAL QUEUE DELAY VEHICLES LEAVING INPUT QUEUES.
ZDTQK	-	I4	TOTAL QUEUE DELAY VEHICLES LEAVING DOCKS.
ZDTQQ	-	I4	TOTAL QUEUE DELAY VEHICLES LEAVING OUTPUT QUEUES.
ZDTOR	-	I4	TOTAL QUEUE DELAY VEHICLES LEAVING OUTPUT RAMPS.
ZDTST	-	I4	TOTAL QUEUE DELAY VEHICLES LEAVING STORAGE AREAS.
ZTMCES	-	I4	MAXIMUM EXCESS TRAVEL TIME FOR COALESCED TRIPS.
ZTPDMS	-	I4	MINIMUM TIME DEMAND TO DISPATCH PASSENGERS.
ZTVNTS	-	I4	MINIMUM TIME IN ANY STATION FOR VEHICLES LEAVING.
ZXDIR	-	I4	MAXIMUM TIME FOR VEHICLES LEAVING IR QUEUE.
ZXDIQ	-	I4	MAXIMUM TIME FOR VEHICLES LEAVING IQ QUEUE.
ZXQDK	-	I4	MAXIMUM TIME FOR VEHICLES LEAVING QK QUEUE.
ZXDOQ	-	I4	MAXIMUM TIME FOR VEHICLES LEAVING OQ QUEUE.
ZXDOR	-	I4	MAXIMUM TIME FOR VEHICLES LEAVING OR QUEUE.
ZXDST	-	I4	MAXIMUM TIME FOR VEHICLES LEAVING ST QUEUE.
ZTTOST	-	I4	TOTAL DISTANCE TRAVELLED FOR COMPLETED TRIPS.
ZTSOST	-	I4	TOTAL STATION DISTANCE FOR VEHICLES LEAVING.

TABLE 3-1. GLOBAL VARIABLES (Page 59 of 63)

NAME: ZCSYSS (CONT)		CATEGORY: OVERALL STATION STATISTICS	
VARIABLE	DIM	TYPE	DESCRIPTION
Z1SICE	-	14	TOTAL NUMBER OF INTERMEDIATE PASSENGER STOPS.
SYSS43	-	14	UNDEFINED.
ZTVNES	-	12	TOTAL # OF VEHICLES ENTERING STATIONS.
ZTVNLS	-	12	TOTAL # OF VEHICLES LEAVING STATIONS.
ZTVNIS	-	12	CURRENT NUMBER OF VEHICLES IN STATIONS.
ZTVALT	-	12	TOTAL # OF VEHICLES DENIED TIMELY ENTRY.
ZTVJLN	-	12	TOTAL # OF VEHICLES LAUNCHED FROM STATIONS.
ZTVMNS	-	12	MAXIMUM NUMBER OF VEHICLES IN ANY STATION.
ZTTNIS	-	12	CURRENT # OF TRIPS IN STATIONS INCLUDING THOSE ON VEHICLES IN STATION.
ZTTNKS	-	12	TOTAL # OF TRIPS REJECTED (BOARDING QUEUE CAPACITY).
ZTTNDS	-	12	TOTAL # OF TRIPS DISPATCHED FROM STATIONS.
ZTPNIS	-	12	CURRENT # PASSENGERS IN STATIONS INCLUDING THOSE ON VEHICLES IN STATION.
ZTPNRS	-	12	TOTAL # OF PASSENGERS REJECTED (BOARDING QUEUE CAPACITY).
ZTPNDS	-	12	TOTAL # OF PASSENGERS DISPATCHED FROM STATIONS EXCLUDING TRANSFERS.
ZTVREQ	-	12	TOTAL # OF UNSATISFIED EMPTY VEHICLE REQUESTS.
ZTVEDC	-	12	TOTAL # OF EMPTIES DISPATCHED ON CIRCUITOUS ROUTES.
ZTVELS	-	12	TOTAL # OF EMPTIES DISPATCHED TO LOCAL STORAGE.
ZTVERC	-	12	TOTAL # OF EMPTIES DISPATCHED TO REGIONAL CENTERS.
ZTVEAN	-	12	TOTAL # OF EMPTIES DISPATCHED BASED ON ANTICIPATED NEED (NOT CONSIDERING CURRENT DISTRIBUTION).
ZTVENR	-	12	TOTAL # OF EMPTIES DISPATCHED BASED ON ANTICIPATED NEED (CONSIDERING CURRENT DISTRIBUTION).
ZTVEDR	-	12	TOTAL # OF EMPTIES DISPATCHED TO OUTSTANDING REQUESTS.
ZTVFD	-	12	TOTAL NUMBER OF EMPTIES DISPATCHED.
ZTTNES	-	12	TOTAL # OF TRIPS ENTERING STATIONS INCLUDING TRANSFERS & REJECTIONS FOR CAPACITY.
ZTTNLS	-	12	TOTAL # OF TRIPS SERVED IN STATIONS.
ZTPNLS	-	12	TOTAL # OF PASSENGERS SERVED IN STATIONS.
ZTPNES	-	12	TOTAL # OF PASSENGERS ENTERING STATIONS INCLUDING TRANSFERS & THOSE REJECTED FOR CAPACITY.
ZTTNCS	-	12	TOTAL # OF TRIPS COMPLETED AT STATIONS.
ZTPNCS	-	12	TOTAL # OF PASSENGERS COMPLETED AT STATIONS.
ZTTCNC	-	12	TOTAL # OF COALESCED TRIPS COMPLETED AT STATIONS.
ZTTSXS	-	12	TOTAL NUMBER OF ARRIVING TRANSFERS AT STATIONS.
ZTTMXS	-	12	MAXIMUM NUMBER OF TRANSFERS FOR ANY COMPLETED TRIP.
ZTNTAR	-	12	TOTAL # OF TRIPS ARRIVING AT STATIONS AND ENTERING BOARDING QUEUE INCLUDING TRANSFERS.
ZTNPAR	-	12	TOTAL # OF PASSENGERS ARRIVING AT STATIONS & ENTERING BOARDING QUEUE INCLUDING TRANSFERS.
ZTNTWT	-	12	CURRENT NUMBER OF TRIPS WAITING AT STATIONS.
ZTXTWT	-	12	MAX NUMBER OF TRIPS WAITING IN ALL STATIONS.
ZTNPWT	-	12	CURRENT NUMBER PASSENGERS WAITING IN STATIONS.
ZTXPWT	-	12	MAX NUMBER OF PASSENGER WAITING IN ALL STATIONS.
ZTTNSS	-	12	TOTAL # OF SPLIT TRIPS CREATED (SUBGROUPS).
ZTTNGS	-	12	TOTAL # OF COMPLETED TRIP SUBGROUPS AT STATIONS.

TABLE 3-1. GLOBAL VARIABLES (Page 60 of 63)

NAME: ZCSYSS (CONT)		CATEGORY: OVERALL STATION STATISTICS	
VARIABLE	DIM	TYPE	DESCRIPTION
ZTVNNS	-	I2	TOTAL # OF ENTRAINED VEHICLES LEAVING STATIONS.
ZMOIR	-	I2	MAXIMUM NUMBER OF VEHICLES ON INPUT RAMPS.
ZMOIQ	-	I2	MAXIMUM NUMBER OF VEHICLES ON INPUT QUEUES.
ZMODK	-	I2	MAXIMUM NUMBER OF VEHICLES AT DOCKS.
ZMOOQ	-	I2	MAXIMUM NUMBER OF VEHICLES ON OUTPUT QUEUES.
ZMOOR	-	I2	MAXIMUM NUMBER OF VEHICLES ON OUTPUT RAMPS.
ZMOST	-	I2	MAXIMUM NUMBER OF VEHICLES IN STORAGE AREAS.
ZMQIR	-	I2	MAXIMUM NUMBER OF VEHICLES QUEUED ON INPUT RAMPS.
ZMQIU	-	I2	MAXIMUM NUMBER OF VEHICLES QUEUED ON INPUT QUEUES.
ZMQDK	-	I2	MAXIMUM NUMBER OF VEHICLES QUEUED AT DOCKS.
ZMQOQ	-	I2	MAXIMUM NUMBER OF VEHICLES QUEUED ON OUTPUT QUEUES.
ZMQOR	-	I2	MAXIMUM NUMBER OF VEHICLES QUEUED ON OUTPUT RAMPS.
ZMQST	-	I2	MAXIMUM NUMBER OF VEHICLES QUEUED IN STORAGE.
ZNQIR	-	I2	TOTAL NUMBER VEHICLES LEAVING INPUT RAMPS.
ZNQIQ	-	I2	TOTAL NUMBER VEHICLES LEAVING INPUT QUEUES.
ZNQDK	-	I2	TOTAL NUMBER VEHICLES LEAVING DOCKS.
ZNQOQ	-	I2	TOTAL NUMBER VEHICLES LEAVING OUTPUT QUEUES.
ZNQOR	-	I2	TOTAL NUMBER VEHICLES LEAVING OUTPUT RAMPS.
ZNQST	-	I2	TOTAL NUMBER VEHICLES LEAVING STORAGE AREAS.
ZNEIR	-	I2	TOTAL NUMBER VEHICLES ENTERING INPUT RAMPS.
ZNEIQ	-	I2	TOTAL NUMBER VEHICLES ENTERING INPUT QUEUES.
ZNEOK	-	I2	TOTAL NUMBER VEHICLES ENTERING DOCKS.
ZNEOQ	-	I2	TOTAL NUMBER VEHICLES ENTERING OUTPUT QUEUES.
ZNEOR	-	I2	TOTAL NUMBER VEHICLES ENTERING OUTPUT RAMPS.
ZNEST	-	I2	TOTAL NUMBER VEHICLES ENTERING STORAGE AREAS.
ZTVRNT	-	I2	TOTAL NUMBER EMPTIES REQUESTED FROM NON-LOCAL STORAGE AREAS.
ZTVRLS	-	I2	TOTAL NUMBER EMPTIES REQUESTED FROM LOCAL STORAGE AREAS.
ZTTXS	-	I2	TOTAL NUMBER OF TRANSFERS FOR COMPLETED TRIPS.
ZTTPXS	-	I2	TOTAL NUMBER OF PASSENGER TRANSFERS FOR COMPLETED TRIPS (# XFERS X # PASSENGERS).
ZTMPWT	-	I2	MINIMUM NUMBER OF PASSENGERS WAITING IN STATIONS.
ZTMOIR	-	I2	MINIMUM NUMBER VEHICLES QUEUED INPUT RAMPS.
ZTMOIQ	-	I2	MINIMUM NUMBER VEHICLES QUEUED INPUT QUEUES.
ZTMOOK	-	I2	MINIMUM NUMBER VEHICLES QUEUED DOCK AREAS.
ZTMOOQ	-	I2	MINIMUM NUMBER VEHICLES QUEUED OUTPUT QUEUES.
ZTMOOR	-	I2	MINIMUM NUMBER VEHICLES QUEUED OUTPUT RAMPS.
ZTMOST	-	I2	MINIMUM NUMBER VEHICLES QUEUED STORAGE AREAS.
ZTXFER	-	I2	TOTAL # OF COMPLETED PASSENGERS HAVING TRANSFERRED AT LEAST ONCE.
SYSS22	-	I2	UNDEFINED.
SYSS23	-	I2	UNDEFINED.

TABLE 3-1. GLOBAL VARIABLES (Page 61 of 63)

NAME: ZCSYST			CATEGORY: OVERALL SYSTEM STATISTICS
VARIABLE	DIM	TYPE	DESCRIPTION
ZTTIRV	-	14	TIME INTEGRAL OF REVENUE SERVICE (OCCUPIED) VEHICLES.
ZTTIDH	-	14	TIME INTEGRAL OF DEADHEADING (EMPTY) VEHICLES.
ZTTISV	-	14	TIME INTEGRAL OF VEHICLES IN STORAGE.
ZTTITV	-	14	TIME INTEGRAL OF TRIPS ON VEHICLES.
ZPTITV	-	14	TIME INTEGRAL OF PASSENGERS ON VEHICLES.
ZTSEAT	-	14	TIME INTEGRAL OF SEATED PASSENGERS ON VEHICLES.
ZTSDCS	-	14	SUM OF TIMES DEMAND TO COMPLETION FOR COMPLETED TRIPS.
ZTXRTT	-	14	MAXIMUM RATIO NOMINAL TT / ACTUAL TT.
ZTMRTT	-	14	MINIMUM RATIO NOMINAL TT / ACTUAL TT.
ZDTOST	-	14	SUM OF PASSENGER DISTANCE TRAVELLED ON GUIDEWAY.
ZDDDST	-	14	SUM OF VEHICLE DEADHEADING (EMPTY) DISTANCE.
ZDRDST	-	14	SUM OF REVENUE SERVICE (OCCUPIED) DISTANCE.
ZIVEST	-	14	ACTUAL VEHICLE DISTANCE TRAVELLED (SIN & GDWY).
SYST42	-	14	UNDEFINED.
SYST43	-	14	UNDEFINED.
ZNTOV	-	12	CURRENT NUMBER OF TRIPS ON VEHICLES.
ZNPOV	-	12	CURRENT NUMBER OF PASSENGERS ON VEHICLES.
ZNVRVS	-	12	CURRENT NUMBER OF VEHICLES IN REVENUE SERVICE (OCCUPIED).
ZNVDEH	-	12	CURRENT NUMBER OF VEHICLES DEADHEADING (EMPTY).
ZNVSTO	-	12	CURRENT NUMBER OF VEHICLES IN STORAGE.
ZNTT1	-	12	# TRIPS EXCESS TRAVEL TT <= PHIST1.
ZNTT2	-	12	# TRIPS EXCESS TRAVEL TT >PHIST1 & <=PHIST2.
ZNTT3	-	12	# TRIPS EXCESS TRAVEL TIME >PHIST2.
ZNPP1	-	12	# PASSENGERS EXCESS TRAVEL TT <= PHIST1.
ZNPP2	-	12	# PASSENGERS EXCESS TRAVEL TT >PHIST1 & <=PHIST2.
ZNPP3	-	12	# PASSENGERS EXCESS TRAVEL TIME >PHIST2.
ZTVRVS	-	12	TOTAL NUMBER OF VEHICLES ENTERING REVENUE (OCCUPIED) STATE.
ZTVDEH	-	12	TOTAL NUMBER OF VEHICLES ENTERING EMPTY STATE.
ZTVSTO	-	12	TOTAL NUMBER OF VEHICLES ENTERING STORAGE STATE.
ZTPSVD	-	12	TOTAL PASSENGERS SERVED EXCLUDING ARRIVING TRANSFERS.
ZTXFLT	-	12	MAXIMUM FLEET SIZE.
ZTMFLT	-	12	MINIMUM FLEET SIZE.
ZNPSV	-	12	CURRENT NUMBER OF SEATED PASSENGERS ON VEHICLES.
ZTMVLF	-	12	MINIMUM VEHICLE LOAD FACTOR.
ZTXVLF	-	12	MAXIMUM VEHICLE LOAD FACTOR.
SYST23	-	12	UNDEFINED.

TABLE 3-1. GLOBAL VARIABLES (Page 62 of 63)

NAME: ZCTFP		CATEGORY: TRIP STATISTICS	
VARIABLE	DIM	TYPE	DESCRIPTION
Z01SAA	KMT	I4	SUM OF ACTUAL TRAVEL TIME.
Z01SNA	KMT	I4	SUM OF NOMINAL TRAVEL TIME.
Z01SNA	KMT	I4	TOTAL NUMBER OF TRANSFERS.
Z01SNA	KMT	I4	TRAVEL INITIATION TIME.
Z01SNA	KMT	I4	DISTANCE TRAVELED ON GUIDEWAY.
Z01SNA	KMT	I4	DISPATCH TIME FROM ORIGIN STATION.
Z01SNA	KMT	I4	SUM OF TRANSFER TIME.
Z01SNA	KMT	I4	ARRIVAL TIME AT LAST INTERMEDIATE DESTINATION.

TABLE 3-1. GLOBAL VARIABLES (Page 63 of 63)

NAME: 7CVEN			CATEGORY: VEHICLE STATISTICS
VARIABLE	DIM	TYPE	DESCRIPTION
ZV1DST	KMV	I4	TOTAL DISTANCE TRAVELED ON GUIDEWAY.
ZV2DST	KMV	I4	TOTAL DISTANCE TRAVELED EMPTY ON GUIDEWAY.
ZV3DST	KMV	I4	TOTAL DISTANCE TRAVELED OCCUPIED ON GUIDEWAY.
ZV1FS	KMV	I4	TIME OF ENTRY TO STATION.
ZV2FS	KMV	I4	TIME OF ENTRY TO STATION LINK.
ZV3FS	KMV	I4	1) TIME OF ENTRY INTO QUEUE. 2) TOTAL MERGE SCHEDULING DELAY.
ZVSTAT	KMV	I2	STATE OF VEHICLE (1 = OCCUPIED, 2 = EMPTY, 3 = STOPPED).
ZVMNCO	KMV	I2	MAXIMUM NUMBER OF VEHICLES IN QUEUE AT ENTRY.
ZVSFAT	KMV	I2	NUMBER OF SEATED PASSENGERS.

SECTION 4. DEBUG TOOLS

The DESM produces several types of execution related information. A user-initiated subroutine entry/exit trace and auxiliary output facility is available for obtaining information related to DESM operation. A standardized error formatting and display capability is programmed into the system to detect modeling or processing related errors. An automatic trace capability, which is invoked whenever anomalous conditions which preclude continued execution are detected is also available in the DESM.

4.1 SUBROUTINE TRACE FACILITY

Each routine executed after initialization, within each component of the DESM, contains auxiliary output statements which display execution related information under control of user-initialized logic (FLAG) variables. Each output message contains the FLAG ID which initiated the output request, a short line of text, and the name (first six characters) and value of as many as ten variables.

Auxiliary output is activated by the user through input specifications contained in the runtime input data sets. The format of the input request to initiate auxiliary output display is given in the DESM User's Manual under the description of the FLAG asynchronous input command. The logic variable assignments by routine within each component of the DESM are shown in Tables 4-1, 4-2, and 4-3.

4.2 STANDARDIZED ERROR HANDLING

Error messages are displayed by the DESM whenever anomalous conditions resulting from user action or processing are detected. All messages output have the same basic format which includes descriptive information, and indication of corrective action required or termination status, and the simulated time of error occurrence, if appropriate. Each message is formatted as follows:

XXXnnt -- Description

Action or termination status

TABLE 4-1. INPUT PROCESSOR PROGRAM MAINTENANCE FLAGS

ROUTINE	TRACE FLAG	AUXILIARY OUTPUT FLAGS
EAFLAG	115	
EIALTP	70	71
EIBWRT	3	
EICHCK	9	
EIDRSP	80	82
EIEMTY	80	84
EIFAIL	58	59
EIMORG	80	
EIMPTH	62	63
EINDPP	54	57
EINERR	7	
EINFMT	66	68, 69
EINRPT	70	
EINSLT	66	68, 69
EINTMK	50	
EIRSEL	25	
EISCFG	75	
EISCHD	85	86, 87, 88, 89, 90
EISERV	80	82
EITNIT	25	30
EITRIP	25	30
EJUDGN	35	
EIWNAM	5	

TABLE 4-2. MODEL PROCESSOR PROGRAM MAINTENANCE FLAGS

ROUTINE	TRACE FLAG	AUXILIARY OUTPUT FLAGS
EAFSM	161	162
EAASYN	113	114
EACKR	109	110
EADADD	111	112*
EAERR	155	116*
EAFLNS	157	113
EAFLAG	115	116*
EAFRPT	131	132*
EAFTRN	171	172
EAINDX	101	102*
EAINIT	147	148
EAIVEH	165	166
EANDR	141	142
EANFEL	149	150
EANMDL	119	120
EANMRG	175	176
EANRPT	121	122*
EANSCD	143	144
EANTRN	169	170
EANXTN	151	152
EAPCMP	159	160
EAPFEL	153	154
EAPLNK	123	124
EAPSTN	125	126
EARRPT	163	164*
EASAMP	127	128*
EASRPT	129	130*
EASTOR	167	168
EATORG	117	118
EAWTIX	101	102*
EAZNIT	133	134*
EGALT	245	246
EGASTN	259	260
EGCNXT	227	228
EGDSTP	249	250
EGDTRN	235	236
EGEMTY	225	226
EGETRN	255	256
EGFAIL	231	232*
EGFNTR	211	212
EGFTVL	215	216
EGGNXT	205	206
EGHTRN	251	252
EGLEAV	233	234, 298, 299
EGLMDL	201	202
EGLNTR	209	210
EGLWTQ	217	218*
EGNEXT	203	204
EGPATH	241	242
EGPRMY	243	244
EGQMRG	257	258

TABLE 4-2. MODEL PROCESSOR PROGRAM MAINTENANCE FLAGS (Cont'd.)

ROUTINE	TRACE FLAG	AUXILIARY OUTPUT FLAGS
EGQNTR	253	254
EGRESV	223	224
EGSCHD	221	222
EGTCHK	239	240
EGTCTL	237	238
EGTEST	207	208
EGTRNC	247	248
EGVALS	229	230
EGVLOG	261	262*
EGVNTR	213	214
EGVTVL	219	220
EMODEL	145	146
ESAREQ	351	352
ESBDL	315	316
ESDBL	313	314
ESDIVF	335	336*
ESDIVO	337	338
ESEVA	329	330
ESEVB	355	356
ESFAIL	321	322*
ESLDLY	343	344
ESLEAV	311	312
ESLMDL	301	302
ESLWTQ	339	340
ESMDLA	305	306
ESMDLB	307	308
ESMDLN	303	304
ESMDLY	357	358
ESNEXT	333	334
ESNSTN	317	318
ESNTRN	341	342
ESPATH	319	320
ESSDLY	309	310
ESTABQ	327	328
ESTCHK	345	346
ESTEST	325	326
ESTLOG	359	360*
ESVALS	323	324
ESVREQ	353	354
ESVRES	331	332*
EZHDR	139	140*
EZINT	135	136
EZZERO	137	138

* Flag reserved for routine but not used

TABLE 4-3. OUTPUT PROCESSOR PROGRAM MAINTENANCE FLAGS

ROUTINE	TRACE FLAG	AUXILIARY OUTPUT FLAGS
EABIN		1
EBNCHK		9
EGRAPH		1, 98
EHEADER		40
EHIST		1
ELIST		1
EOINDX		1
EOPRPT		11
EOPSUM	9	10
EOSRPT		13
EOSSUM	15	16
EOUTPT		1, 20
EOWTIX		1
EOZNIT		1
EREAD02		1, 2
EREAD03		1, 3
EREAD04		1, 4
EREAD05		1, 5
EREAD06		1, 6
EREQTLU		50
EREQU		1
ESKIPFO		1
ESTORE		2
EZPLOT		1
EZREAD		1, 3, 40

where:

XXX -- Indicates DESM processor identification:

EIP - Input Processor

EMP - Model Processor

EOP - Output Processor

nnn -- Message number

T -- Severity

I -- Information

W -- Warning

S -- Severe

Description -- A brief statement of the error condition

Action or Termination Status - A brief description of required user action or execution status as follows:

- o CONDITION MAY BE ACCEPTABLE TO THE USER
- o CONDITION MUST BE CORRECTED PRIOR TO THE NEXT RUN
- o PROGRAM EXECUTION CANNOT PROCEED BEYOND THIS POINT.

Each processor maintains a count of each message type I and W issued during a run. If the number of messages issued for either category exceeds a limit, automatic processing termination is performed with user notification of the termination action. Automatic termination also occurs in the case of a severe error occurrence. Any termination action during execution results in the display of a system traceback report as described below.

4.3 SYSTEM TRACEBACK

The DESM system trace facility is invoked upon abnormal termination of either the IP, MP, or OP. The conditions for abnormal termination can result from a system interrupt or a programmed condition within the DESM itself.

When invoked, the trace facility obtains the following information by examining the save area chains created as the result of executing a particular logic path:

1. A list of the current hierarchy of subroutines called
2. The contents of the general-purpose registers at the time each subroutine was called
3. The current values of the various arguments supplied in each subroutine call.

This information is then formatted and displayed as shown in Figure 4-1 upon processor termination. The following items of information are displayed in the traceback report:

1. Column headed "CALL BY ROUTINE"
The list of the current sequence of subroutine calls, from most recent to the oldest. In the example, the operating system (the last entry, "SYSTEM") called the main program MAIN, which in turn called DSTSUB, which called JRLOUT, which called SNAFU.
2. Column "ISN"
The internal statement number (generated by the FORTRAN compiler) from which the next subroutine call was executed, in decimal (all other numbers on the TRACBK output are in hexadecimal).
3. Column "LOCN"
The absolute core memory address of the next sequential instruction after the CALL statement.
4. Column "RELOCN"
The displacement of LOCN from the entry point address of the subroutine issuing the CALL.
5. Column "TO ROUTINE"
The name of the subroutine being called. When the traceback resulted from a program interrupt, the first line under this column will contain "*INTERRUPT*".
6. Column "LOCN"
The absolute core memory address of the entry point to the called subroutine.


```

CALL BY ROUTINE 115, LCCN, RLOCN) TO ROUTINE (LOCN)
CURRENT ARG COMMENTS AND REGS FROM SAVE AREA

SYSTEM 6 022000 000102 1MACR 019550
GR 0-4 00012FAC 00015D04 00000000 0012FAL 00013028
GR 5-9 00000004 00000002 00036080 00000078 00028018
GR10-12 00027818 00076518 00025A68
JALCOP 80 013078 000000 3MAYU 025A18
ARG 0 00012FAC => 01000300 00000000 00000000 00000000 00000000
41000000 00000000 00000000 00000000 00000000
ARG 1 00012F50 => 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
ARG 2 00012E4L => 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
GR 0-4 00000000 00013004 00013000 00000000 00000000 00000000
GR 5-9 00012E42 00000000 00000000 00000000 00000000 00000000
GR10-12 00015028 00015028 00015028 00015028 00015028 00015028
DSY308 182 012206 001A86 JALOUT 012E58
GR 0-4 00000000 00012948 00010C4C 00010C52 00017318
GR 5-9 0001003E 00000000 00016D10 00000078 00020780
GR10-12 0001F740 0001F740 0001F740 0001F740 0001F740
DSI 8 01782C 000344 DS3308 010750
GR 0-4 00000025 00012948 0001778C 00017792 00012948
GR 5-9 00012A7E 00000000 00000000 00000000 00000000
GR10-12 00019318 00019318 00019318 00019318 00019318
MAIN 24 012A88 000260 DS1 0172E8
ARG 0 00012438 => 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
ARG 1 00012E70 => 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
ARG 2 00012B4C => 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
ARG 3 00012B20 => 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
ARG 4 00012A20 => 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
GR 0-4 00000000 00012948 00000000 00000000 00000000 00000000
GR 5-9 00012A82 00000000 00000000 00000000 00000000 00000000
GR10-12 0001A978 0001A978 0001A978 0001A978 0001A978
SYSTEM 8 01750E 000260 MAIN 012858

```

FIGURE 4-1. SAMPLE TRACEBACK PRINTOUT

7. CURRENT ARG CONTENTS AND REGS FROM SAVE AREA

Two distinct types of information are provided -- the first may or may not appear while the second always appears.

a. Lines of the form "ARG @ ffaaaaa =>"

TRACBK produces one entry of this form for each argument supplied in the CALL statement (to the called routine). For all but the last argument, "ff" will be zero; for the last it will be "80". The absolute core memory address of the argument is "aaaaa." Following the arrow, the eight words of core memory beginning at the argument address are printed. Notice that the interpretation of this information depends upon the nature of the argument; for example:

- (1) The length of the argument may not be 4 bytes so grouping by units other than 4 may be necessary.
- (2) The type of information (integer, real, alphabetic) must be considered.
- (3) For call-by-value arguments and for scalar call-by-name arguments, only one unit of information is of importance; the remaining memory contents are superfluous.
- (4) For call-by-name arguments, the current values printed may not be the ones that existed when the subroutine was called, i.e., the called subroutine may have altered their values.
- (5) If the address of an argument exceeds the highest core memory address, then no argument contents can be printed, so "BADOBADO" is printed instead.

b. The lines GR 0-4, GR 5-9, and GR 10-12

These lines list what the contents of the general-purpose registers were when the branch to the called routine was executed.

Since the operation of TRACBK depends upon proper linkages, conditions could arise which would cause TRACBK to function abnormally. When an unusual condition is detected, the column "CALL BY ROUTINE" will contain a description of the error detected. These errors are:

- (1) **"*SAVE CHAIN ADR*"** -- the address of previous save-area exceeds the highest core memory address.
- (2) **"*SAVE CHAIN ER*"** -- the save-areas are not chained to one another properly.
- (3) **"*SAVE ALIGN ER*"** -- the save-area is not full-word boundary aligned.
- (4) **"*TOO MANY CALLS*"** -- the depth of subroutine calls exceeded 25.

Entry SPIEL in routine TRACBK is CALLED at the beginning of program execution to establish linkage to the traceback facility. The calling sequence is: **SPIEL(0)**

where the argument, 0, indicates that the traceback facility should be invoked upon the occurrence of any system interrupt. If only specific interrupts are to be processed by the DESM trace facility, the specific interrupt types, (by standard vs interrupt code), can be specified in the calling sequence.

SECTION 5. SUBPROGRAM LOGIC TABLES

Tables 5-1, 5-2, and 5-3 contain subprogram control section (CSECT) names, entry point names, calling subprograms, called subprograms and function for the subprograms of the IP, MP, and OP respectively.

TABLE 5-1. (1 of 5) INPUT PROCESSOR SUBROUTINE LOGIC TABLE

CSECT	ENTRY	CALLED BY	CALLS	FUNCTION
DAYTIM	DAYTIM	EIMNAM EINPUT	TIMES	GET DATE AND TIME OF DAY.
EACOMN	EACOMN	-	-	USED BY IP AND MP TO INSURE IDENTICAL ORDERING OF INPUT COMMON AREAS. NO ROUTINE CALLS EACOMN; IT ACCOMPLISHES ITS FUNCTION BY BEING LINK EDITED AHEAD OF ANY ROUTINE WHICH USES THE COMMONS BEING ORDERED.
EAFLAG	EAFLAG	EINPUT	-	SET FLAGS FOR INTERMEDIATE OUTPUT
EIADDR	EIADDR	EIINIT	EISADD EINADD	PROVIDE ADDRESSABILITY TO SYSTEM CHARACTERISTICS COMMONS SO THEY CAN BE WRITTEN BY IP TO STRUCTURED DATA FILE.
EIALTP	EIALTP	EINTWK	EINERR	ALTERNATE PATH TABLE GENERATION
EIARPT	EIARPT	EINPUT	-	WRITE ACTIVE FLEET SIZE REPORT
EIBWRT	EINADD EINRD EINWRT EISADD EISWRT	EIADDR EINTWK EINTWK EIADDR EINPUT	- - - - -	SAVE ADDRESS & LENGTH OF NETWORK STRUCTURED DATA READ NETWORK STRUCTURED DATA WRITE NETWORK STRUCTURED DATA SAVE ADDRESS & LENGTH OF SYSTEM CHARACTERISTICS STRUCTURED DATA WRITE SYSTEM CHARACTERISTICS STRUCTURED DATA
EICHCK	EICHCK	EINPUT	EINERR	DATA CHECK AND INITIALIZATION
EIDRPT	EIDRPT	EINPUT	-	WRITE TRIP GENERATION REPORT
EIDRSP	EIDRSP	EISERV	EIENTY EINERR	DEMAND RESPONSIVE SERVICE
EIENTY	EIENTY	EIDRSP	EIMNTP EINERR	EMPTY VEHICLE REDISTRIBUTION
EIFAIL	EIFAIL	EINTWK	EIMPTH EINSLT EINERR	NETWORK FAILURE/REPAIR INSERTION
EIINIT	EIINIT	EINPUT	LODCOM EIADDR EIPLST	INPUT PROCESSOR INITIALIZATION

TABLE 5-1. (2 of 5) INPUT PROCESSOR SUBROUTINE LOGIC TABLE

CSECT	ENTRY	CALLED BY	CALLS	FUNCTION
EIMNAM	EIMNAM EIWNAM	EIPLST EINPUT	DAYTIM	DECODE PARAMETER LIST WRITE MEMBER NAMES
EIMNTP	EIMNTP	EIEMTY	-	TRANSPORTATION ALGORITHM
EIMORG	EIMORG	EISERV	EINERR	TIMEOUT/GROUP DEMAND RESPONSIVE SERVICE PLANNING
EIMPTH	EIMPTH	EINTWK EIFAIL	-	MINIMUM PATH GENERATION
EINDPP	EINDPP	EINTWK	EINERR	NETWORK DEFINITION DATA PREPROCESSING
EINERR	EINERR	EIALTP EICHCK EIDRSP EIEMTY EIFAIL EIMORG EINDPP EINPUT EINTWK EISCFG EISCHD EISERV EITNIT EITRIP GDIP4	ERROR	GENERATE ERROR MESSAGE
EINFMT	EINFMT	EINTWK	EINSLT	NETWORK DATA FORMATTING
EINPUT	EINPUT	EIPARM	DAYTIM EIARPT EICHCK EIDRPT EIINIT EINTWK EAFLAG EITNIT EIUDGN EITRIP EISCFG EISERV EISRPT EISWRT EIWNAM EINERR NDBOR SPIEL	CONTROL INPUT PROCESSING READ USER INPUTS NETWORK PROCESSING TRIP GENERATION STATION CONFIGURATION SERVICE PLANNING
EINRPT	EINRPT	EINTWK	-	WRITE NETWORK INIT COND REPORT

TABLE 5-1. (3 of 5) INPUT PROCESSOR SUBROUTINE LOGIC TABLE

CSECT	ENTRY	CALLED BY	CALLS	FUNCTION
EINSLT	EINSLT	EINFMT EIFAIL	-	GEN LINK SUCCESSOR TABLE
EINTWK	EINTWK	EINPUT	EINDPP EIMPTH EINFMT EINRD EIALTP EINWRT EINRPT EIFAIL EINERR	NETWORK PROCESSING PREPROCESS NETWORK DATA MINIMUM PATH GEN NETWORK DATA FORMATTING READ/WRITE STRUC DATA FAILURE/REPAIR PROCESS
EIPARM	EIPARM EIPLST	SYSTEM JOB STEP TASK EIINIT	EINPUT EIMNAM	SAVE ADDRESS OF PARM FIELD PASSED BY SYSTEM. CALL MAIN IP ROUTINE. PASS PARM FIELD AND PARM FIELD LENGTH TO ROUTINE WHICH DIVIDES FIELD INTO FILE MEMBER NAMES.
EIPSAV	EIPSAV LODCOM	- EIINIT	- -	PROVIDE STORAGE LOCS (COMMON EIPSAV) FOR ADDRESSES OF IP COMMONS, SYS CHAR COMMONS, AND NETWORK COMMONS. CAUSES COMMON ADDRESSES TO BE INITIALIZED.
EIRNG	EIRNG	EITRIP EIRSEL	-	GENERATE RANDOM NUMBER
EIRSEL	EIRSEL	EITRIP	EIRNG	SAMPLE PROBABILITY DISTRIBUTION
EISCFG	EISCFG	EINPUT	EINERR	STATION LINK CONFIGURATOR
EISCHD	EISCHD	EISERV	EINERR	SCHEDULED SERVICE
EISERV	EISERV	EINPUT	EIDRSP EISCHD EIMORG EINERR	SERVICE PLANNING DEMAND RESPONSIVE SERVICE SCHEDULED SERVICE TIMEOUT/GROUP DR SERVICE
EISRPT	EISRPT	EINPUT	-	WRITE INITIAL CONDITIONS REPORT
EITNIT	EITNIT	EINPUT	EINERR	TRIP DEMAND INITIALIZATION
EITRIP	EITRIP	EINPUT	EINERR EIRNG EIRSEL	TRIP FILE GENERATION GENERATE RANDOM NUMBER SAMPLE PROBABILITY DIST

TABLE 5-1. (4 of 5) INPUT PROCESSOR SUBROUTINE LOGIC TABLE

CSECT	ENTRY	CALLED BY	CALLS	FUNCTION
EIUDGN	EIUDGN	EINPUT	-	UNIFORM TRIP GENERATION
ERROR	ERROR	EINERR	TRACBK	WRITE INFORMATION, WARNING, OR SEVERE ERROR MESSAGE. COUNT MSG OCCURRENCES BY TYPE AND MSG NO. AND TERMINATES WHEN COUNT(S) EXCEED LIMITS.
GDIP-SECT	GDIP4	NDBOR	EINERR GDIPF4 GDIPH4 GDIPX4	READ INTO USER DEFINED INPUT AREA USING USER SPECIFIED FORMAT.
GDIPF4	GDIPF4	GDIP4	SUDOGO	READ FULL WORD GDIP DATA
GDIPH4	GDIPH4	GDIP4	SUDOGO	READ HALF WORD GDIP DATA
GDIPX4	GDIPX4	GDIP4	SUDOGO	READ SINGLE BYTE GDIP DATA
NDBOR	NDBOR	EINPUT	GDIP4	READ FORMAT DATA FOR GDIP DATA
PFIICS	PFIICS SUDOGO	OPERATING SYS. GDIPX4 GDIPH4 GDIPF4	- -	PSEUDO I/O INTERCEPT ROUTINE ESTABLISH I/O BUFFER
TIMESECT	TIMES	DAYTIM	-	GET DAY AND DATE FROM SYSTEM CLOCK
TRACBK	TRACBK SPIEL	OPERATING SYS. ERROR EINPUT	TRCBKI TRCBKP TRCBKV TRCBKR SPIELP SPIELQ -	GET REGISTER AND ARGUMENT TRACE INFORMATION SET INTERRUPT FLAGS TO GET CONTROL AT PGM INTERRUPT TIME
TRCBKP	TRCBKP TRCBKI TRCBKV TRCBKR SPIELP SPIELQ	TRACBK TRACBK TRACBK TRACBK TRACBK TRACBK	- - - - -	PRINT 1 LINE FOR SUBROUTINE CALL PRINT PGM INT HEADING PRINT 2 LINES FOR ARGUMENT PRINT 3 LINES FOR GEN REG PRINT 6 LINES FOR PGM INTERRUPT PRINT TERMINATION MESSAGE

TABLE 5-1. (5 of 5) INPUT PROCESSOR SUBROUTINE LOGIC TABLE

NOTES:

(1)

SEVERAL OF THE CSECTS ABOVE ARE KNOWN BY DIFFERENT SOURCE NAMES. SINCE OTHER DOCUMENTATION MAY REFER TO SUBROUTINES BY SOURCE NAME RATHER THAN CSECT NAME, THE CSECTS WITH THEIR SOURCE NAMES ARE LISTED BELOW.

CSECT	SOURCE MEMBER
ERROR	EIERROR
GDIPF4	XGDIPF4
GDIPH4	XGDIPH4
GDIPSECT	EIGDIP4
GDIPX4	XGDIPX4
NDBOR	XNDBOR
PFIOCS	XPSEUDO
TIMES	DTIMEL
TRACBK	XTRACBK
TRCBKP	XTRCBKP

TABLE 5-2. (1 of 11) MODEL PROCESSOR SUBROUTINE LOGIC TABLE

CSECT	ENTRY	CALLED BY	CALLS	FUNCTION
DAYTIM	DAYTIM	EANRPT EAINDX EAFRPT EARRPT EASRPT	TIMES	GET CURRENT DATE & TIME TO YY/MM/DD/HH/MM/SS
EAAFMS	EAAFMS	EAASYN	ERROR ESLEAV EAIVEH EANTRN EASTOR EAPFEL	ACTIVE FLEET SIZE MANAGEMENT
EAASYN	EAASYN	EMODEL	NDBOR EGFAIL ESFAIL EAFLAG EACKPT EAPFEL EAAFMS ERROR	PROCESS AN ASYNCHRONOUS DATA EVENT
EACKR	EACKR EACKPT EAREST	EANSV EAASYN EMODEL EAINIT	- ERROR EAPFEL EZHDR EARRPT ERROR EAPFEL	PERFORM CHECKPOINT/RESTART PERFORM CHECKPOINT PERFORM RESTART
EACOMN	EACOMN	-	-	USED BY IP AND MP TO INSURE IDENTICAL ORDERING OF INPUT COMMON AREAS. NO ROUTINE CALLS EACOMN; IT ACCOMPLISHES ITS FUNCTION BY BEING LINK EDITED AHEAD OF ANY ROUTINE WHICH USES THE COMMONS BEING ORDERED.
EADADD	EADADD EANDTA	EANSV EAINIT	- -	INITIALIZE SYSTEM STATUS AREA ADDRESS FOR SYSTEM INITIALIZATION READ NETWORK AND SYSTEM CHARACTERISTICS
EAFINS	EAFINS	EMODEL ERROR	EAFRPT EAWTIX	SYSTEM TERMINATION PROCESSING
EAFLAG	EAFLAG	EAASYN EAINIT	-	SET FLAGS FOR INTERMEDIATE OUTPUT
EAFRPT	EAFRPT	EAFINS	DAYTIM	SYSTEM TERMINATION REPORT
EAFTRN	EAFTRN	EGSCHD ESLWTQ	-	PERFORM PUSH COUPLE ENTRAINMENT

TABLE 5-2. (2 of 11) MODEL PROCESSOR SUBROUTINE LOGIC TABLE

CSECT	ENTRY	CALLED BY	CALLS	FUNCTION
EAINDX	EAINDX EAWTIX	EAUPTX EAFINS	DAYTIM -	WRITE INITIAL INDEX FILE ENTRIES WRITE OUTPUT FILE SUMMARY TO INDEX FILE
EAINIT	EAINIT	EMODEL	EAREST SPIEL EANTSA EAUPTX EANDTA EANFEL EANXTN EANMDL EAZNIT EAFLAG NDBOR EAPFEL ERROR	SYSTEM LEVEL INITIALIZATION
EAIVEH	EAIVEH	EAFSM EAPCMP	ERROR	INITIALIZE NEW VEHICLES
EANDR	EANDR	EANMDL	EAPFEL	PERFORM DEMAND RESPONSIVE INITIALIZAION
EANFEL	EANFEL	EAINIT	-	INITIALIZE FUTURE EVENTS LIST
EANMDL	EANMDL	EAINIT	EANDR EANSCL EANMRG EANRPT ERROR	MODEL INITIALIZATION CONTROL
EANMRG	EANMRG	EANMDL	-	INITIALIZE TIMEOUT/GROUP DEMAND RESPONSIVE SERVICE
EANRPT	EANRPT	EANMDL	DAYTIM	DISPLAY INITIAL CONDITIONS REPORT
EANSV	EANSV	EANTSA	EADADD EACKR	RETRIEVE CORE ADDRESS OF IP DATA AREAS
EANSCL	EANSCL	EANMDL	EAPFEL	PERFORM SCHEDULED SERVICE INITIALIZATION
EANTIX	EANTIX EAUPTX	OPERAT- ING SYS. EAINIT	EMODEL EAINDX	RECEIVE CONTROL FROM SYSTEM AND SAVE PARM FIELD INFORMATION RETRIEVE PARM FIELD ADDRESS FOR INDEX FILE UPDATE

TABLE 5-2. (3 of 11) MODEL PROCESSOR SUBROUTINE LOGIC TABLE

CSECT	ENTRY	CALLED BY	CALLS	FUNCTION
EANTRN	EANTRN	EAAF EAPC EASTOR ESMDLB	EAPFEL	ENTRAIN AND LAUNCH SCHEDULED SERVICE ACTIVE FLEET SIZE MANAGEMENT VEHICLES
EANTSA	EANTSA	EAINIT	LODCOM EANSV	INITIALIZE SYSTEM STATUS AREA
EANXTN	EANXTN	EAINIT	-	INITIALIZE TRANSACTION DATA
EAPCMP	EAPCMP	EMODEL	EUPCMP EAI EANTRN ESVREQ EAPFEL ERROR	PERFORM PERIODIC COMPUTATION EVENT
EAPFEL	EAPFEL	EAAF EAASYN EACKR EAINIT EANDR EANS EANTRN EAPC EAPLNK EAREST EASAMP EASTOR EATORG EGDTRN EGFAIL EGFNTR EGFTVL EGHTRN EGLEAV EGLNTR EGNEXT EGQMRG EGQNTR EGSCHD EGTEST EGVNTR EGVTVL ESAREQ ESFAIL ESLEAV ESLWTQ ESMDLA ESMDLB ESMDLN ESNEXT ESTABQ ESTEST ESVREQ	ERROR	PLACE A TRANSACTION ON THE FUTURE EVENTS LIST

TABLE 5-.. (4 of 11) MODEL PROCESSOR SUBROUTINE LOGIC TABLE

CSECT	ENTRY	CALLED BY	CALLS	FUNCTION
EAPLNK	EAPLNK	EAPRMT	EGNEXT EGTEST EGLEAV EGLMDL ESNEXT ESTEST ESLMDL ESLEAV EAPFEL	DEQUEUE A VEHICLE ON A GUIDEWAY LINK
EAPSTN	EAPSTN	EAPRMT	ESNEXT ESTEST ESLEAV ESLMDL EGTEST EGLMDL	DEQUEUE A VEHICLE FROM A STATION QUEUE
EARRPT	EARRPT	EAREST	DAYTIM	SYSTEM RESTART REPORT
EASAMP	EASAMP	EMODEL	EZHDR EZINT EZZERO EASRPT EAPFEL	PERFORM PERIODIC SAMPLING
EASRPT	EASRPT	EASAMP	DAYTIM	DISPLAY PERIODIC SAMPLING REPORT
EASTOR	EASTOR	EAAFSS ESMDLB	EANTRN ESLEAV EAPFEL	ASSIGN STORED VEHICLES IN SCHEDULED SERVICE ACTIVE FLEET SIZE MANAGEMENT
EATORG	EATORG	EMODEL	ESTABQ ERROR EAPFEL	PROCESS A TRIP ARRIVAL
EAZNIT	EAZNIT	EAINIT	EZZERO EZHDR	INITIALIZE SIMULATION STATISTICS
EGALT	EGALT	EGPATH	EGPRMY	ALTERNATE PATH COST COMPUTATION
EGASTN	EGASTN	EAGCTL	-	ALTERNATE STATION ASSIGNMENT
EGCNXT	EGCNXT	EGENTY	-	NEXT STATION DETERMINATION, CIRCUITOUS EMPTY ROUTING
EGDSTP	EGDSTP	EGNEXT	EGGNXT	DEMAND STOP SERVICE PROCESSING
EGDTRN	EGDTRN	EGRESV EGTRNC	EAPFEL	GUIDEWAY LINK VEHICLE DETRAINMENT

TABLE 5-2. (5 of 11) MODEL PROCESSOR SUBROUTINE LOGIC TABLE

CSECT	ENTRY	CALLED BY	CALLS	FUNCTION
EGEMTY	EGEMTY	EGRESV	EGCNXT EGGNXT EGVALS	GUIDEWAY LINK EMPTY VEHICLE PROCESSING
EGETRN	EGETRN	EGLMDL	-	GUIDEWAY LINK DYNAMIC VEHICLE ENTRAINMENT
EGFAIL	EGFAIL	EAASYN	EAPFEL ERROR	PROCESS A GUIDEWAY LINK FAILURE
EGFNTR	EGFNTR	EGLMDL	EAPFEL	FIXED HEADWAY TRAVEL SEGMENT ENTRY
EGFTVL	EGFTVL	EGLMDL	EGLWTQ EGQMRG EAPFEL EGSCHD	FIXED HEADWAY TRAVEL SEGMENT TRAVERSAL
EGGNXT	EGGNXT	EGNEXT EGDSTP EGEMTY EGTRNC	EGPATH	GUIDEWAY--NEXT ENTITY DETERMINATION
EGHTRN	EGHTRN	EGLMDL	EAPFEL	TRAIN LINK HEADJAY TRAVERSAL PROCESSING
EGLEAV	EGLEAV	EAGCTL EAPLNK	EGVLOG EAPFEL	GUIDEWAY LINK EXIT PROCESSING
EGLMDL	EGLMDL	EAGCTL EAPLNK EAPSTN EASCTL	EGLNTR EGHTRN EGETRN EGFNTR EGVNTR EGUNTR EGFTVL EGVTVL EGUTVL	GUIDEWAY LINK MODEL CONTROL
EGLNTR	EGLNTR	EGLMDL	EAPFEL EGQNTR	GUIDEWAY LINK ENTRY PROCESSING
EGLWTQ	EGLWTQ	EAGCTL EGFTVL EGVTVL	-	ENTITY QUEUEING PROCESSING
EGNEXT	EGNEXT	EAGCTL EAPLNK	EGGNXT EGRESV EGTRNC EGDSTP EAPFEL	NEXT ENTITY DETERMINATION CONTROL

TABLE 5-2. (6 of 11) MODEL PROCESSOR SUBROUTINE LOGIC TABLE

CSECT	ENTRY	CALLED BY	CALLS	FUNCTION
EGPATH	EGPATH	EGGNXT	EGPRMY EGALT	REAL-TIME PATH SELECTION
EGPRMY	EGPRMY	EGALT EGPATH	-	PRIMARY PATH COST COMPUTATION
EGQMRG	EGQMRG	EGFTVL ESMDLN	EAPFEL	PERFORM QUASI-SYNCHRONOUS CONTROL
EGQNTR	EGQNTR	EGLNTR	EAPFEL	GUIDEWAY LINK ADVANCE POSITIONING PROCESSING
EGRESV	EGRESV	EGNEXT	EGDTRN EGEMTY EGTCTL	GUIDEWAY LINK EMPTY & RESERVED VEHICLE PROCESSING
EGSCHD	EGSCHD	EGFTVL EGTVL	ERROR EAPFEL EAFTRN	SCHEDULE VEHICLE FOLLOWER
EGTCHK	EGTCHK	EGTCTL	-	TRIP COMPATIBILITY CHECK
EGTCTL	EGTCTL	EGRESV	EGTCHK	TRIP COMPATIBILITY CHECK CONTROL
EGTEST	EGTEST	EAGCTL EAPLNK EAPSTN EASCTL	ERROR EAPFEL	GUIDEWAY LINK ENTRY TESTING
EGTRNC	EGTRNC	EGNEXT	EGGNXT EGDTRN	TRAIN COMPATIBILITY CHECK AT NETWORK DIVERGES
EGUNTR	EGUNTR	EGLMDL	ERROR	USER INTERFACE ROUTINE FOR LINK ENTRY PROCESSING
EGUTVL	EGUTVL	EGLMDL	ERROR	USER INTERFACE ROUTINE FOR LINK TRAVERSAL PROCESSING
EGVALS	EGVALS	EGEMTY	-	VEHICLE ETA AND STATION ARRIVAL LIST RECORDING
EGVLOG	EGVLOG	EGLEAV	-	WRITE VEHICLE LOG FILE ENTRY
EGVNTR	EGVNTR	EGLMDL	EAPFEL	VARIABLE HEADWAY TRAVEL SEGMENT ENTRY

TABLE 5-3. (7 of 11) MODEL PROCESSOR SUBROUTINE LOGIC TABLE

CSECT	ENTRY	CALLED BY	CALLS	FUNCTION
EGVTVL	EGVTVL	EGLMDL	EGLWTQ EAPFEL EGSCHD	VARIABLE HEADWAY TRAVEL SEGMENT TRAVERSAL
EMODEL	EMODEL	EANTIX	EAINIT EATORG EASAMP EAASYN EACKPT EAPCMP EAFINS ERROR	DISCRETE EVENT SIMULATOR CONTROL
	EARFEL	-	-	REMOVE A TRANSACTION FROM THE FUTURE EVENTS LIST
	EAGCTL	-	EGNEXT EGTEST EGLMDL EGLEAV EGLWTQ EGASTN ESTEST	GUIDEWAY LINK MODEL CONTROL (INCLUDED MEMBER IN EMODEL)
	EASCTL	-	ESLMDL ESLMDL ESNEXT ESTEST ESLEAV ESLWTQ EGTEST EGLMDL	DESM STATION MODEL CONTROL
	EAPRMT	-	EAPLNK EAPSTN	PROCESS A DEQUEUE TRANSACTION
ERROR	ERROR	EAAFSM EAASYN EACKR EAINIT EAIVEH EANMDL EAPCMP EAPFEL EATORG EGFAIL EGSCHD EGTEST EGUNTR EGUTVL EMODEL ESBDL ESEVA ESEVB ESLWTQ ESNSTN EUDIVF EUEVA EUEVB EUPCMP GDIP4	TRACBK EAFINS	ISSUE A FORMATTED ERROR MESSAGE

TABLE 5-2. (8 of 11) MODEL PROCESSOR SUBROUTINE LOGIC TABLE

CSECT	ENTRY	CALLED BY	CALLS	FUNCTION
ESAREQ	ESAREQ	ESMDLB ESVREQ	EAPFEL	TIMEOUT/GROUP DEMAND RESPONSIVE ASSIGN REQUEST TO VEHICLE
ESASAV	ESASAV LODCOM	- EANTSA	-	INITIALIZE ADDRESS OF STATUS AREA IN COMMON
ESBDL	ESBDL	ESMDLB	ESTCHK ERROR	CREATE A VEHICLE BOARDING LIST FROM WAITING TRIPS IN STATION
ESDBL	ESDBL	ESMDLB	ESTLOG	CREATE A VEHICLE DEBOARD LIST FROM TRIPS ABOARD VEHICLE
ESDIVF	ESDIVF	ESNEXT	ESDIVO EUDIVF	DIVERGE FUNCTION
ESDIVO	ESDIVO	ESDIVF	-	ORDER LINKS BY OCCUPANCY
ESEVA	ESEVA	ESNSTN	EUEVA ERROR	EMPTY VEHICLE REDISTRIBUTION
ESEVB	ESEVB	ESTEST	EUEVB ERROR	EMPTY VEHICLE BUMPING
ESFAIL	ESFAIL	EAASYN	EAPFEL	PROCESS A STATION FAILURE
ESLDLY	ESLDLY	ESMDLB	ESMDLY	STATION MODEL LAUNCH TIME DETERMINATION
ESLEAV	ESLEAV	EASCTL EAPLNK EAPSTN ESMDLA EAAFMS EASTOR ESMDLB	EAPFEL	PROCESS A VEHICLE LEAVING A STATION LINK
ESLMDL	ESLMDL	EASCTL EAPLNK EAPSTN EAGCTL	ESMDLA ESMDLB ESMDLN	DESM STATION MODEL CONTROL
ESLWTQ	ESLWTQ	EASCTL	EAFTRN EAPFEL ERROR	INSERT A VEHICLE IN A STATION LINK QUEUE
ESMDLA	ESMDLA	ESLMDL	ESNSTN ESVALS ESLEAV EAPFEL	STATION MODEL AFTER EVENT PROCESSING

TABLE 5-2. (9 of 11) MODEL PROCESSOR SUBROUTINE LOGIC TABLE

CSECT	ENTRY	CALLED BY	CALLS	FUNCTION
ESMDLB	ESMDLB	ESLMDL	ESBDL ESDBL ESPATH ESSDLY ESNTRN ESLDLY ESAREQ ESLEAV EANTRN EASTOR ESTABQ EAPFEL	STATION MODEL BEFORE EVENT PROCESSING
ESMDLN	ESMDLN	ESLMDL	EGQMRG EAPFEL	DETERMINE NEXT STATION EVENT FOR A VEHICLE AND ESTABLISH AVAILABILITY
ESMDLY	ESMDLY	ESLDLY	-	DETERMINE MERGE DELAY
ESNEXT	ESNEXT	EASCTL EAPLNK EAPSTN	ESDIVF EAPFEL	NEXT ENTITY FOR A VEHICLE IN A STATION
ESNSTN	ESNSTN	ESMDLA	ESEVA ERROR	DETERMINE NEXT STATION FOR VEHICLE
ESNTRN	ESNTRN	ESMDLB	-	PERFORM STATIC ENTRAINMENT
ESPATH	ESPATH	ESMDLB	-	DETERMINE PATHS FOR A VEHICLE
ESSEDLY	ESSDLY	ESMDLB	-	DETERMINE SCHEDULE DELAY
ESTABQ	ESTABQ	EATORG ESMDLB	ESTCHK ESVRES ESVREQ EAPFEL	ATTEMPT TRIP/VEHICLE RESERVATION
ESTCHK	ESTCHK	ESBDL ESTABQ	-	DEMAND RESPONSIVE MULTIPARTY TRIP COMPATIBILITY CHECK
ESTEST	ESTEST	EAGCTL EASCTL EAPLNK EAPSTN	ESVREQ ESEVB EAPFEL	STATION ENTRY TESTING
ESTLOG	ESTLOG	ESDBL	-	RECORD ENTR FOR TERMINATING TRIP COMPLETED TRIP LOG
ESVALS	ESVALS	ESMDLA	-	INSERT A VEHICLE IN A DOWNSTREAM ARRIVAL LIST

TABLE 5-2. (10 of 11) MODEL PROCESSOR SUBROUTINE LOGIC TABLE

CSECT	ENTRY	CALLED BY	CALLS	FUNCTION
ESVREQ	ESVREQ	EAPCMP ESTABQ ESTEST	ESAREQ EAPPFL	TIMEOUT/GROUP DEMAND RESPONSIVE PROCESS READY VEHICLE REQUEST
ESVRES	ESVRES	ESTABQ	-	RESERVE SPACE FOR TRIP ON SELECTED VEHICLE
EUDIVF	EUDIVF	ESDIVF	ERROR	USER DIVERGE FUNCTION INTERFACE
EUEVA	EUEVA	ESEVA	ERROR	USER EMPTY VEHICLE ALGORITHM
EUEVB	EUEVB	ESEVB	ERRJR	USER EMPTY VEHICLE BUMPING ALGORITHM
EUPCMP	EUPCMP	EAPCMP	ERROR	USER PERIODIC COMPUTE EVENT INTERFACE
EZHDR	EZHDR	EAREST EASAMP EAZNIT	-	WRITE A SAMPLING FILE HEADER
EZINT	EZINT	EASAMP	-	COMPUTE END POINT TIME INTEGRALS
EZZERO	EZZERO	EAZNIT EASAMP	-	ZERO PERIODIC STATISTICS AND INITIALIZE TIME INTEGRALS
GDIP- SECT	GDIP4	NDBOR	ERROR GDIPF4 GDIPH4 GDIPX4	READ INTO USER DEFINED INPUT AREA USING USER SPECIFIED FORMAT.
GDIPF4	GDIPF4	GDIP4	SUDOGO	READ FULL WORD GDIP DATA
GDIPH4	GDIPH4	GDIP4	SUDOGO	READ HALF WORD GDIP DATA
GDIPX4	GDIPX4	GDIP4	SUDOGO	READ BYTE SIZE GDIP DATA
NDBOR	NDBOR	EAASYN EAINIT	GDIP4	READ FORMAT DATA FOR GDIP DATA
PFIOCS	PFIOCS SUDOGO	OPERAT- ING SYS. GDIPX4 GDIPH4 GDIPF4	- -	PSEUDO I/O INTERCEPT ROUTINE ESTABLISH I/O BUFFER
TIMESECT	TIMES	DAYTIM	-	GET DAY AND DATE FROM SYSTEM CLOCK

TABLE 5-2. (11 of 11) MODEL PROCESSOR SUBROUTINE LOGIC TABLE

CSECT	ENTRY	CALLED BY	CALLS	FUNCTION
TRACBK	TRACBK	ERROR OPERATING SYS.	TRCBKI TRCBKP TRCBKV TRCBKR SPIELP SPIELQ	GET REGISTER AND ARGUMENT TRACE INFORMATION
	SPIEL	EAINIT	-	SET INTERRUPT FLAGS TO GET CONTROL AT PGM INTERRUPT TIME
TRCBKP	TRCBKP	TRACBK	-	PRINT 1 LINE FOR SUBROUTINE CALL
	TRCBKI	TRACBK	-	PRINT PGM INTERRUPT HEADING
	TRCBKV	TRACBK	-	PRINT 2 LINES FOR ARGUMENT
	TRCBKR	TRACBK	-	PRINT 3 LINES FOR GEN REG
	SPIELP	TRACBK	-	PRINT 6 LINES FOR PGM INTERRUPT
	SPIELQ	TRACBK	-	PRINT TERMINATION MESSAGE

NOTES:
(1)

SEVERAL OF THE CSECTS ABOVE ARE KNOWN BY DIFFERENT SOURCE NAMES. SINCE OTHER DOCUMENTATION MAY REFER TO SUBROUTINES BY SOURCE NAME RATHER THAN CSECT NAME, THE CSECTS WITH THEIR SOURCE NAMES ARE LISTED BELOW.

CSECT	SOURCE MEMBER
ERROR	EAERR
GDIPSECT	EMGDIP4
TIMES	DTIMEL
GDIPX4	XGDIPX4
GDIPF4	XGDIPF4
GDIPH4	XGDIPH4
PFIOCS	XPSEUDO
NDBOR	XNDBOR
TRACBK	XTRACBK
TRCBKP	XTRCBKP

TABLE 5-3. (1 of 5) OUTPUT PROCESSOR SUBROUTINE LOGIC TABLE

CSECT	ENTRY	CALLED BY	CALLS	FUNCTION
AADATE	AADATE	ETMERG ETRPTS	DAYTIM	PROVIDE DATE IN CHARACTER FORMAT
ABIN (1)	ABIN	STOFLO	ERROR SHIFT	BIN REALLOCATION
BNCHK (1)	BNCHK	ZHIST ZREQU	ERROR SHIFT	BIN EXPANSION
DAYTIM	DAYTIM	EOINDX AADATE	TIMES	CONVERT DATE & TIME TO YY/MM/ DD/HH/MM/SS
DBIN (1)	DBIN	EOZKIT	-	ALLOCATE BIN STORAGE
DERROR	DERROR	EOUTPT ETCOMP ETMEAS ETNMBR	-	ERROR TERMINATION ROUTINES FOR STN-STN MEASURES
DUMBIN (1)	DUMBIN	EOUTPT	-	PRINT BIN AREA HEADERS
EODATA	EODATA	-	-	BLOCK DATA INITIALIZATION OF MAJOR COMMON AREAS
EOFLAG	EOFLAG	EOUTPT	-	INITIALIZE AUXILIARY OUTPUT FLAGS
EOINDX	EOINDX EOWTIX	EOUPTX EOUTPT	DAYTIM -	DECODE PARM FIELD INFORMATION AND WRITE LOAD MODULE DATE AND TIME WRITE OUTPUT FILE SUMMARY TO INDEX FILE
EONTIX	EONTIX EOUPTX	OPER. SYSTEM EOZKIT	EOUTPT EOINDX	RECEIVE CONTROL FROM SYSTEM AND SAVE PARM FIELD INFORMATION RETRIEVE PARM FIELD ADDRESS FOR INDEX FILE UPDATE
EOPRPT	EOPRPT	EOPSUM	-	WRITE PERFORMANCE SUMMARY RPT
EOPSUM	EOPSUM	EOUTPT	EOPRPT	COMPUTE AND WRITE PERFORMANCE SUMMARY DATA
EOSRPT	EOSRPT	EOSSUM	-	WRITE SYSTEM SUMMARY STATISTICS REPORT

TABLE 5-3. (2 of 5) OUTPUT PROCESSOR SUBROUTINE LOGIC TABLE

CSECT	ENTRY	CALLED BY	CALLS	FUNCTION
EOSSUM	EOSSUM	EOUTPT	EOSRPT	COMPUTE SYSTEM SUMMARY STATISTICS
EOUTPT	EOUTPT	EONTIX	EOZNIT DUMBIN EOFLAG ZREAD ZLIST ZHIST ZPLOT ZREQU ERROR EOWTIX EOPSUM EOSSUM ETSSPM DERROR	DESM-OUTPUT PROCESSOR CONTROL
EOZNIT	EOZNIT	EOUTPT	SPIEL DBIN ZREQU ZREAD EOUTPX	OUTPUT PROCESSOR INITIALIZATION
ERROR (1)	ERROR	ZREQU ZREAD ABIN BNCHK HEADER READ02 READ03 READ04 READ05 READ06 SHIFT SKIPFO EOUTPT REQTLU SETUP	TRACBK	ISSUE A FORMATTED ERROR MESSAGE
ETACUM	ETACUM	ETCAPT	-	ACCUMULATE STATION-TO-STATION RAW STATISTICS
ETCAPT	ETCAPT	ETCOMP	ETACUM ETMEAS	CAPTURE STATION-TO-STATION RAW DATA
ETCOMP	ETCOMP	ETSSPM	ETCAPT ETTOTL ETSTAT ETRPTS DERROR	CAPTURE REQUESTED STATION-TO-STATION MEASURE
ETMEAS	ETMEAS	ETCAPT	DERROR	FUNCTION TO COMPUTE RAW DATA FOR THE MEASURE

TABLE 5-3. (3 of 5) OUTPUT PROCESSOR SUBROUTINE LOGIC TABLE

CSECT	ENTRY	CALLED BY	CALLS	FUNCTION
ETMERG	ETMERG	ETSSPM	AADATE ETNMBR	CONVERT RAW STATISTICS TO UTPS FORMAT
ETNMBR	ETNMBR	ETMERG ETSTID	DERROR	FUNCTION TO CONVERT A NUMBER TO EBCDIC
ETRPTS	ETRPTS	ETCOMP	AADATE ETSTID	WRITE STATION-TO-STATION MEASURES REPORT
ETSSPM	ETSSPM	EOUTPT	ETCOMP ETMERG	STATION-TO-STATION PERFORMANCE MEASURES CONTROL
ETSTAT	ETSTAT	ETCOMP	-	COMPUTE STATISTICS
ETSTID	ETSTID	ETRPTS	ETNMBR	FUNCTION TO CONVERT STATION ID TO EBCDIC
ETTOTL	ETTOTL	ETCOMP	-	COMPUTE RAW DATA TOTALS
GRAPH (1)	GRAPH NEWGRF SAMGRF SAMSCL	ZPLOT	- - -	PRINT TIME SERIES PLOT-1ST CALL REUSE REUSE - NO CHANGE ALPHAMERICS REUSE - NO CHANGE SCALE OR ALPHAMERICS
HEADER (1)	HEADER	ZREAD	ERROR	READ NEXT HEADER RECORD
HIST (1)	HIST	ZHIST	-	PRINT HISTOGRAM
LIST (1)	LIST	ZLIST	-	TIME SERIES LIST OF SAMPLE VALUES
MNMX (1)	MNMX	ZHIST	-	COMPUTE MINIMUM AND MAXIMUM OF OF SAMPLE VALUES STORED IN BIN
READ02 (1)	READ02	ZREAD	ERROR STOFLO	READ AND PROCESS SYSTEM STATISTICS
READ03 (1)	READ03	ZREAD	ERROR STOFLO	READ AND PROCESS STATION STATISTICS
READ04 (1)	READ04	ZREAD	ERROR STOFLO	READ AND PROCESS STATION LINK STATISTICS

TABLE 5-3. (4 of 5) OUTPUT PROCESSOR SUBROUTINE LOGIC TABLE

CSECT	ENTRY	CALLED BY	CALLS	FUNCTION
READ05 (1)	READ05	ZREAD	ERROR STOFLO	READ AND PROCESS GUIDEWAY LINK STATISTICS
READ06 (1)	READ06	ZREAD	ERROR STOFLO	READ AND PROCESS ROUTE STATISTICS
REQTLU (1)	REQTLU	ZREAD	ERROR	RECORD REQUEST CORRELATION
SETUP (1)	SETUP	ZREAD	ERROR	INITIALIZE OP TABLE VALUES
SHIFT (1)	SHIFT	BNCHK ABIN	ERROR	SHIFT DATA ITEMS IN ALLOCATED BIN
SKIPFO (1)	SKIPFO	ZREAD	ERROR	SKIP A FOLLOWER RECORD
STORE (1)	STORE STOFLO	- READ02 READ03 READ04 READ05 READ06	- ABIN	STORE DATA IN BIN STORE DATA IN BIN
TIMSECT (1)	TIMES	DAYTIM	-	ACQUIRE DAY AND DATE FROM SYSTEM CLOCK
TRACBK (1)	TRACBK SPIEL	ERROR OPERAT- ING SYS. EOZNIT	TRCBKI TRCBKP TRCBKV TRCBKR SPIELP SPIELQ -	GET REGISTER AND ARGUMENT TRACE INFORMATION SET INTERRUPT FLAGS TO GET CONTROL AT PGM INTERRUPT TIME
TRCBKP (1)	TRCBKP TRCBKI TRCBKV TRCBKR SPIELP SPIELQ	TRACBK TRACBK TRACBK TRACBK TRACBK	- - - - -	PRINT 1 LINE FOR SUBRTN CALL PRINT PGM INTERRUPT HEADING PRINT 2 LINES FOR ARGUMENT PRINT 3 LINES FOR GEN REG PRINT 6 LINES FOR PGM INTERRUPT PRINT TERMINATION MESSAGE
ZHIST (1)	ZHIST	EOUTPT	MNMX BNCHK HIST	HISTOGRAM OUTPUT CONTROL
ZLIST (1)	ZLIST	EOUTPT	LIST	TIME SERIES LIST OUTPUT CONTROL

TABLE 5-3. (5 of 5) OUTPUT PROCESSOR SUBROUTINE LOGIC TABLE

CSECT	ENTRY	CALLED BY	CALLS	FUNCTION
ZPLOT (1)	ZPLOT	EOUTPT	GRAPH	TIME SERIES PLOT OUTPUT CONTROL
ZREAD (1)	ZREAD	EOZKIT EOUTPT	HEADER SKIPFO ERROR SETUP REQTLU READ02 READ03 READ04 READ05 READ06	RAW STATISTICS DATA ACQUISITION CONTROL
ZREQU (1)	ZREQU	EOZKIT EOUTPT	ERROR BNCHK	REQUEST HANDLING

NOTES:

(1) SEVERAL OF THE CSECTS ABOVE ARE KNOWN BY DIFFERENT SOURCE NAMES. SINCE OTHER DOCUMENTATION MAY REFER TO SUBROUTINES BY SOURCE NAME RATHER THAN CSECT NAME, THE CSECTS WITH THEIR SOURCE NAMES ARE LISTED BELOW.

CSECT	SOURCE MEMBER
ABIN	EABIN
BNCHK	EBNCHK
DBIN	EDBIN
DUMBIN	EDUMBIN
ERROR	EOERR
GRAPH	EGRAPH
HEADER	EHEADER
HIST	EHIST
LIST	ELIST
MNMX	EMNMX
READ02	EREAD02
READ03	EREAD03
READ04	EREAD04
READ05	EREAD05
READ06	EREAD06
REQTLU	EREQTLU
SETUP	ESETUP
SHIFT	ESHIFT
SKIPFO	ESKIPFO
STORE	ESTORE
TIMES	DTIMEL
TRACBK	XTRACBK
TRCBKP	XTRCBKP
ZHIST	EZHIST
ZLIST	EZLIST
ZPLOT	EZPLOT
ZREAD	EZREAD
ZREQU	EREQU

SECTION 6. SUBPROGRAM DESCRIPTIONS

This section describes the components of the DESM Input Processor Model Processor, and Output Processor. These components include sub-routines and macros identified by their source library member names. The global variables used in these PARAFOR, ASSEMBLER, and PL/I components are defined in Section 3. Local variables which serve as arguments in the calling sequence of these modules are listed in the Argument Dictionary of each description. All arguments are assumed to be input only unless explicitly identified as "INPUT and OUTPUT" or "OUTPUT". Other local variables used in the modules are listed in the Local Variable Dictionary of each description.

The following is provided for each local variable:

- o Variable name: the name by which the variable is referenced. Since arguments may not be named in Assembly Language routines, an arbitrary name has been assigned.
- o Dimension: A hyphen indicates that the variable is a scalar. A number, n, e.g., 2, indicates that the variable is an array with subscripts from 1 to n, e.g., variable(1) variable(2), etc.
- o Type: FORTRAN notation is used to identify the type and length of the variable, e.g., I*2 = half word integer, R*4 = full word real number. When a character string or variable name is an argument, the letter "C" is specified. When the character string must be a specified length, that is also shown, e.g. "T" = C*1.
- o Description: A brief definition of the variable is given. Optional arguments in the calling sequence are identified and default values given.

A description of the module's logic is provided as are any supporting decision tables and algorithms. The PDL detailing the logic of each module is given in Appendix A.

6.1 INPUT PROCESSOR

This section contains the subprogram descriptions for the DESM Input Processor.

6.1.1 DAYTIM - Obtain Date and Time

6.1.1.1 Identification

- o DAYTIM - Convert Date and Time
- o IBM/FSD - July 1, 1977
- o PARAFOR

6.1.1.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
MM	-	I*2	(OUTPUT) MONTH
DD	-	I*2	(OUTPUT) DAY
YY	-	I*2	(OUTPUT) YEAR
HH	-	I*2	(OUTPUT) HOURS
MM	-	I*2	(OUTPUT) MINUTES
SS	-	I*2	(OUTPUT) SECONDS

6.1.1.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
YEAR	2	I*4	Century and year of century
HMS	3	I*4	Hours, minutes, seconds
SS	-	I*2	Seconds
LEAP	-	I*4	Indicates leap year

6.1.1.4 Description

The purpose of DAYTIM is to obtain the Julian date and time from the system clock and return the calendar date and time. DAYTIM calls DTIMEL via entry point TIMES to obtain the Julian date and time from the system clock. The returned year is then tested for leap year with the MOD function to determine which calendar conversion to use. The calendar conversion then uses the day of the year to find the month of the year and the day of the month.

6.1.1.5 PDL

See Appendix A.

6.1.1.6 Decision Tables and Algorithms

None.

6.1.2 DEBUG - Auxiliary Output Macro

6.1.2.1 Identification

- o DEBUG - Intermediate Output Macro
- o IBM/FSD - July 1, 1977
- o PL/I

6.1.2.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ID	-	C	Maximum 31 characters of text to be output in DEBUG message.
FLAG	-	I*4	Array subscript for a logical variable tested during execution to determine if message display is required.
C1	-	C	Variable name to be displayed with value when DEBUG message is issued. Must be fully qualified. (Optional)
C2	-	C	Variable name to be displayed with value when DEBUG message is issued. Must be fully qualified. (Optional)
C3	-	C	Variable name to be displayed with value when DEBUG message is issued. Must be fully qualified. (Optional)
C4	-	C	Variable name to be displayed with value when DEBUG message is issued. Must be fully qualified. (Optional)
C5	-	C	Variable name to be displayed with value when DEBUG message is issued. Must be fully qualified. (Optional)
C6	-	C	Variable name to be displayed with value when DEBUG message is issued. Must be fully qualified. (Optional)

VARIABLE	DIM	TYPE	DESCRIPTION
C7	-	C	Variable name to be displayed with value when DEBUG message is issued. Must be fully qualified. (Optional)
C8	-	C	Variable name to be displayed with value when DEBUG message is issued. Must be fully qualified. (Optional)
C9	-	C	Variable name to be displayed with value when DEBUG message is issued. Must be fully qualified. (Optional)
C10	-	C	Variable name to be displayed with value when DEBUG message is issued. Must be fully qualified. (Optional)

6.1.2.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
OUTA	-	C	Constructed FORTRAN code
OUTB	-	C	Constructed FORTRAN code
FMT	-	C	Format statement number

6.1.2.4 Description

The purpose of DEBUG is to provide a trace facility within the simulator. This macro generates: a logical IF statement which tests the state of the logical variable array element specified by argument FLAG, a WRITE statement for the output requested in the argument list, and a FORMAT statement. When the specified element of the logical variable array is set .TRUE. during simulator execution, the WRITE statement is executed, which lists the logical variable subscript (second argument), the specified text (first argument) and the first six characters of the variable and its value for up to ten variables requested in the argument list.

6.1.2.5 PDL

None.

6.1.2.6 Decision Tables and Algorithms

None.

6.1.3 DTIMEL - Read System Clock

6.1.3.1 Identification

- o DTIMEL - Read System Clock for Date and Time
- o IBM/FSD - July 1, 1977
- o ASM

6.1.3.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TIMES:			
YEAR	2	I*4	(OUTPUT) YEAR, JULIAN DAY
HMS	3	I*4	(OUTPUT) HOURS, MINUTES, AND SECONDS
SEC	-	I*4	(OUTPUT) TIME OF DAY IN SECONDS
DELT	-	I*4	(OUTPUT) ELAPSED TIME SINCE LAST CALL TO TIMES (IN SECONDS)

6.1.3.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TA	-	I*4	Seconds of the day
DBL	2	I*4	Packed decimal date and time
YIM	2	I*4	Century and year of century
HIM	3	I*4	Hours, minutes, seconds

6.1.3.4 Description

DTIMEL is called by DAYTIM to read the system clock and return the current Julian date and time. DTIMEL calls the system TIME macro to get the date and time in EBCDIC. The routine then converts the date and time to binary and returns to the calling program.

6.1.3.5 PDL

See Appendix A.

6.1.3.6 Decision Tables and Algorithms

None.

6.1.4 EACOMN - Structured Data COMMONs Ordering

6.1.4.1 Identification

- o EACOMN - Define Ordering of Structured Data COMMONs
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.1.4.2 Argument Dictionary

None.

6.1.4.3 Local Variable Dictionary

None.

6.1.4.4 Description

This routine is used by both the IP and MP to force an identical ordering of input area commons. This is done by including it as the first object module that contains the input commons at link edit time in both the IP and MP. This is necessary to ensure that the ordering of these commons in the IP from which AGT. STRUC. SYSTEM is written is the same as that in the MP into which the data is read.

6.1.4.5 PDL

See Appendix A.

6.1.4.6 Decision Tables and Algorithms

None.

6.1.5 EAFLAG - Auxiliary Output Request Processing

6.1.5.1 Identification

- o EAFLAG - Auxiliary Output Request Processing
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.1.5.2 Argument Dictionary

None.

6.1.5.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
FINI	-	L*1	Indicates last card found
TEMP	18	I*4	Flag numbers from current card
AEOF	-	L*1	End of file on asynchronous input file
L	-	I*4	Lower bound of range of flags to be set true
U	-	I*4	Upper bound of range of flags to be set true

6.1.5.4 Description

This routine sets flags associated with selecting auxiliary output generated using the DEBUG macro. This routine first resets all flags and reads cards containing the flag numbers to be set until a zero field is found. The requests can define individual flag numbers or a contiguous set of flags, e.g., 47-54. The format and contents of the request card are defined in subsection 4.1.1. of the DESM User's Manual under the description of the FLAG asynchronous input command. The requested flags remain set until another FLAG asynchronous input command is read.

6.1.5.5 PDL

See Appendix A.

6.1.5.6 Decision Tables and Algorithms

None.

6.1.6 EIADDR - Establish COMMON Addressability

6.1.6.1 Identification

- o EIADDR - Establish Addressability to Structured Data COMMONs
- o IBM/FSD - May 1, 1978
- o ASM

6.1.6.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
PARM1	-	I*4	Address of beginning of Network structured data COMMONs
PARM2	-	I*4	Length in words of Network structured data COMMONs
PARM3	-	I*4	Address of beginning of System Characteristics structured data COMMONs
PARM4	-	I*4	Length in words of System Characteristics structured data COMMONs

6.1.6.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ARGA	-	I*4	Parameter that is equivalenced with the first item in the Network structured data COMMONs
ARGB	-	I*4	Length in words of the Network structured data COMMONs
ARGC	-	I*4	Parameter that is equivalenced with the first item in the System Characteristics structured data COMMONs
ARGD	-	I*4	Length in words of the System Characteristics structured data COMMONs

6.1.6.4 Description

EIADDR establishes direct addressability to the structured data COMMON regions by defining parameters that are equivalenced to the first location in the Network structured data and in the System Characteristics structured data. These parameters and the length in words of the COMMON regions are passed to routine EIBWRT using entry points EINADD (Network data) and EISADD (System Characteristics data). This process establishes the definition of the location and size of the structured data COMMONs in routine EIBWRT for later use in writing the COMMON regions to their respective structured data files.

6.1.6.5 PDL

See Appendix A.

6.1.6.6 Decision Tables and Algorithms

None.

6.1.7 EIALTP - Alternate Path Table Generation

6.1.7.1 Identification

- o EIALTP - Alternate Path Processing
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.1.7.2 Argument Dictionary

None.

6.1.7.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
COMDIV	-	I*4	Node identification of common diverge point
ILIM	-	I*4	Index of last link to be listed on a line
LIM1	-	I*4	Index of first link to be listed on a line
LIM2	-	I*4	Index of last link of an alternate path in the concatenated list of alternate paths
LINE	-	I*4	Line counter for report generation
NID	-	I*4	Node identification.
PAGE	-	I*4	Page number for report generation
PAGLIM	-	I*4	Lines per page limit

6.1.7.4 Description

This routine establishes linkage of the user-defined alternate paths to the least cost paths at the common diverge points and translates the paths from a node to node specification to a link to link specification.

The alternate routes are defined by the user as a list of lists of node sequences to travel on from some point on the network to a specified destination station. The individual lists are separated by the value 0. Each individual list begins with the identification of the destination station, followed by a sequence of node identifications that trace a path from a common diverge point to the destination station (a common diverge point is a guideway diverge where the minimum cost path and the alternate path separate). Multiple alternate path node sequences from the same common diverge point to the same destination station are separated by the value -1.

The least cost path link connectivity table is a matrix in which each column contains the least cost path link sequences from any point in the network to a particular destination station. Current location is used to select the row of the matrix which contains the identification of the next link to travel on to a particular destination station.

This routine modifies the next link identifications to indicate the existence of an alternate path and where it is stored. The following major processing steps are performed for each alternate path.

1. If a prior definition of alternate paths exists, remove the connectivity from the least cost path link connectivity table and clear the alternate path link sequence list.
2. Determine the row and column of the least cost path link connectivity table that is to be modified. Two rows are defined if the common diverge point is an intersection.
3. Establish the connection to the alternate path

$$L = L + 1000 \times P \quad (1)$$

where

L is the contents of the least cost path link connectivity table at the row/column position determined in step 2.

P is the position of the first link in the list of lists of alternate routes from that common diverge point.

4. Put the link sequence defined by the node sequence in the alternate path link list.

After all of the alternate paths have been processed, a report is generated based on the alternate path link sequence list and the modified least cost path link connectivity table. The common diverge point, destination station, and link sequence defining each alternate path are

listed for each common diverge point encountered as the least cost path link connectivity table is scanned.

6.1.7.5 PDL

See Appendix A.

6.1.7.6 Decision Tables and Algorithms

The address of alternate path data is merged with the next link value in the least cost path link connectivity table using the algorithm described in subsection 6.1.7.4, step 3. A table value greater than 1000 indicates the existence of an alternate path. The remainder when dividing the table value by 1000 is the next link on the primary path. The integer portion of the quotient resulting when dividing the table value by 1000 is a pointer to the first link in the alternate path link sequence.

6.1.8 EIARPT - Active Fleet Size Management Report

6.1.8.1 Identification

- o EIARPT - Active Fleet Size Management Report Generation
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.1.8.2 Argument Dictionary

None.

6.1.8.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DONE	-	L*1	Logical variable indicating: 1) empty vehicle management includes anticipated need method 2) all data for a scheduled service route has been printed
LINREM	-	I*4	Number of blank lines remaining on page
OUT1	KMS	I*2	Node identification of each station
OUT5	-	R*4	Output variable for probability value
PAGE	-	I*4	Page number
PAGLIM	-	I*4	Lines per page limit
START	-	I*4	Index of first element to be printed on a page
STOP	-	I*4	Index of last element to be printed on a page

VARIABLE	DIM	TYPE	DESCRIPTION
TCUSEC	-	R*4	Time conversion, clock units to seconds
TIME	-	I*4	Time of active fleet size management request

6.1.8.4 Description

This routine formats and outputs a summary of the updated level of service characteristics when active fleet size management is requested. The report includes:

- o time of the request
- o source of level of service (user or a specified demand profile interval)
- o type of service
- o If demand responsive service
 - fleet size
 - empty vehicle management data (if it was updated and is to be used)
- o If scheduled service
 - route identification
 - number of vehicles on route
 - train length on route
 - route headway
 - station stops on route (station ID and node ID)
 - number of vehicles to initially dispatch from each station
 - initial departure time for a vehicle from each station on each route (in seconds)
 - fleet size
- o If timeout/group demand responsive service
 - fleet size

6.1.8.5 PDL

See Appendix A.

6.1.8.6 Decision Tables and Algorithms

None.

6.1.9 EIBWRT - BINARY INPUT/OUTPUT

6.1.9.1 Identification

- o EIBWRT - Binary Input/Output
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.1.9.2 Argument Dictionary

ENTRY	VARIABLE	DIM	TYPE	DESCRIPTION
EINADD	ADDR1	LEN1	I*4	First location in Network structured data area
	LEN1	-	I*4	Length in words of Network structured data area
EISADD	ADDR2	LEN2	I*4	First location in System Characteristics structured data area
	LEN2	-	I*4	Length in words of System Characteristics structured data area
EINWRT	None			
EISWRT	None			
EINRD	None			

6.1.9.3 Local Variable Dictionary

None.

6.1.9.4 Description

EIBWRT has five entry points:

1. EINADD is called by EIADDR to save the address and length of the Network structured data COMMON area.

2. EISADD is called by EIADDR to save the address and length of the System Characteristics structured data COMMON area.
3. EINWRT writes the Network structured data file using the address and length saved by EINADD.
4. EISWRT writes the System Characteristics structured data file using the address and length saved by EISADD.
5. EINRD reads the Network structured data file using the address and length saved by EINADD. It also reads the end of file record to close the file so that another member may be written to the Network structured data file by EINWRT. This occurs when alternate paths are defined for a previously processed network.

6.1.9.5 PDL

See Appendix A.

6.1.9.6 Decision Tables and Algorithms

None.

6.1.10 EICHCK - Input Parameter Checking

6.1.10.1 Identification

- o EICHCK - Parameter Checking and Initialization
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.1.10.2 Argument Dictionary

None.

6.1.10.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
BD	-	L*1	Indicates board event exists
CASE	-	I*2	Case selection parameter
DBD	-	L*1	Indicates deboard event exists
DEST	-	I*4	Destination station
DOCK	-	L*1	Indicates dock link exists
INDEX	-	I*4	Index for scanning station link lists
J	-	I*4	Miscellaneous parameter for: 1) loop index 2) indicates at least one element of a set of related data is outside limits 3) case selection parameter
K	-	I*4	Miscellaneous parameter for: 1) indicates at least one element of a set of related data is outside limits 2) list index

VARIABLE	DIM	TYPE	DESCRIPTION
LAUNCH	-	I*2	Link type that includes launch event
LINK	-	I*4	Number of link sublists found in station link lists
MGLCAP	-	I*2	Minimum guideway link capacity
ORIG	-	I*4	Origin station
PVSPR6	-	L*1	Indicates empty vehicle source option 6 selected
STANCP	-	I*2	Station total capacity counter
STORE	-	L*1	Indicates store event exists
TCNVRT	-	R*4	Time conversion, seconds to clock time
TOTVEH	-	I*2	Total number of placed vehicles
TYPE	-	I*2	Link type
WFER	-	I*4	Station to which to walk before reboarding
XFER	-	I*4	Station at which to transfer

6.1.10.4 Description

This routine checks the system characteristics data for reasonableness and consistency with the selected alternatives, performs time conversion from seconds to clock units, processes the station characteristics data if the station configurator module was not used, and converts the data from user conventions to model processor conventions.

The following guideway link parameters are processed:

- o Reaction time (GLRTIM) - Reasonableness check, time conversion.
- o Merge delay table (GLMDLY) - If heuristic merge policy selected; reasonableness check, time conversion.
- o Velocity standard deviation (GLVSD) - Reasonableness check, compatibility with longitudinal control policy, time conversion.
- o Guideway link occupancy (GLOCC) - If user - defined demand responsive service; capacity must not be exceeded, only links which precede stations can be initialized with vehicles.

dispatch policy must be non-deterministic. The sum of guideway link occupancy is saved for use in processing station link occupancy. If not user-defined demand responsive service, guideway link occupancy is set to zero.

The remaining guideway link parameters are checked during network processing in routine EINDPP.

The following station parameters are processed:

- o Number of station links (KNSL) - Reasonableness check.
- o Priority/FIFO selection (SLPF) - Reasonableness check.
- o Diverge function selection (SLDIVC) - Reasonableness check.
- o Headway zone travel time (SLHTA, SLHTB) - Reasonableness check, time conversion.
- o Station link capacity (SLCAP) - If demand responsive service; input ramp, input queue, dock, output queue and output ramp must be large enough to hold the largest train.

The following station parameters are processed only if the station configuration module has not been used:

- o Station link travel time (SLTTIM) - Time conversion.
- o Station link type (SLTYPE) - Reasonableness check.
- o Links acting as the input ramp (SLIR), output ramp (SLOR) and storage (SLSTOR) are looked for and identified.
- o Station must include at least one dock link.
- o Station link event list (SLEVL) - Pointer to sublist for each link is defined; events must be in increasing order by event type; reasonableness check; compatibility of event type with link type; type of link containing launch event is defined; deboard, board and launch events must exist; each link must have an event sublist; storage link must have a store event.
- o Station link upstream link list (SLUSL) - Pointer to sublist for each link is defined, reasonableness check, connectivity must define a valid vehicle path through the station.
- o Station link downstream link list (SLDSL) - Pointer to sublist for each link is defined, reasonableness check, connectivity must define a valid vehicle path through the station.

The following additional station parameters are processed:

- o Headway times (SLHTA, SLHTB) - Set to zero if headway zone travel event does not exist.
- o Storage link in maintenance barn (SBARN, SLSTOR) - If a station has been defined to be a maintenance barn, an available storage link must exist in that station.
- o Berthing policy (PBERTH) - Reasonableness check.
- o Boarding queue capacity (SBQCAP) - Reasonableness check.
- o Dwell factor (SMNDBT) - The minimum door open time must not be greater than the time required to completely empty and refill a vehicle.
- o Maximum boarding time (SMXDBT) - The default value and upper limit of SMXDBT is the maximum of the time to empty and fill vehicle and the minimum door open time. This check may be overridden when processing handicapped passengers.
- o Boarding time parameters (STBA, SMNDBT, STDBA, STDHFF) - Reasonableness check, time conversion.
- o Maximum number of passengers to board (SMXDBP) - If boarding time per passenger is zero, SMXDBP is set equal to vehicle capacity. In the case of handicapped passengers, this may be overridden by a high value. If boarding time per passenger is greater than zero, SMXDBP is set equal to the passenger boarding time limit divided by the boarding time per passenger and limited to the vehicle capacity.
- o Berth advancement times (STMOV1, STMOV2) - Reasonableness checks, time conversion.
- o Handicapped passenger deboard/board parameters (SHCBA, SHCBB, SHCDBA, SHCDBB) - Reasonableness checks including handicapped board and deboard times per passenger are not exceeded by ordinary passenger board and deboard times, time conversions, computation of equivalent time ordinary passenger board count for one handicapped passenger and reset handicapped board time parameters to correspond to an integer number of ordinary passengers.
- o Vehicle handicapped capacity (VHCAP) - Reasonableness check.
- o Transfer tables (PTSTN, PWSTN) - User input of transfer list is converted from node identification to station identification and checked for valid station nodes. Transfer tables are built from user input.

- o Transfer walk time (PWALKT, PWLKTS) - Reasonableness check, time conversion, and if station-station transfer walk time (PWALKT) is not provided set to default value (PWLKTS).
- o Travel time from launch to station exit (SEMTIM) - Station link travel time is summed along vehicle path from launch event to station exit.
- o Number of slots from launch to station exit (SEMSLT) - If deterministic dispatch, SEMSLT is the smallest integer that is greater than or equal to the quotient resulting from dividing the travel time from launch to station exit by the time required to travel one slot on the guideway.
- o Station link travel time factor (SLBFAC) - If at least one online station, SLBFAC is nominal speed on station links divided by user-defined bypass velocity if user did not specify bypass factor. User must specify nominal speed on station links.
- o Total station capacity (SNCAP1) - Reasonableness check. If SNCAP1 exceeds sum of available station link capacities, it is reset to that sum and a message is displayed.
- o Station link dock type (SLDTYP) - Reasonableness check, if timeout/group demand responsive service.
- o Station link platform type (SLPLAT) - Reasonableness check, if timeout/group demand responsive service.
- o Station link occupancy (SLOCC) - If user-defined demand responsive service, vehicles can only be placed on station storage links. The sum of station link occupancy is added to the sum of guideway link occupancy previously computed. The result must equal the vehicle fleet size. If not user-defined demand responsive service, station link occupancy is set to zero. If scheduled service, station link occupancy is set to zero. If timeout/group demand responsive service, vehicles can only be placed on dock links (limited by link capacity). The sum of station link occupancy must equal the vehicle fleet size.

The following system characteristics parameters are processed:

- o Empty vehicle arrival limit (PEVALM) - Time conversion.
- o Algorithmic path selection data (PMDWT, PSTWT, PSUWT) - Reasonableness check (if selected algorithm uses the parameter), time conversion (PMDWT, PSTWT only).
- o Merge window width (PMRGWW) - If synchronous control or if user did not define, PMRGWW is vehicle headway on guideway. If quasisynchronous or asynchronous control and deterministic or quasideterministic dispatch, reasonableness check, time conversion.

- o Merge reservation table width (PMRGTT) - Total time is product of window width and number of windows (compile parameter).
- o Merge threshold (PMRGTH) - If quasideterministic dispatch, reasonableness check.
- o Merge schedule limit (PNVMRG) - If synchronous control, each element equals one. Otherwise, each element is the product of the merge threshold and the merge window width divided by the headway on the link downstream from the corresponding merge.
- o Alternate station egress time (PALTET) - Time conversion.
- o Excess travel time histogram class intervals (PHIST1, PHIST2) - Time conversion.
- o Maximum slot maneuver (PARMAX) - If quasisynchronous control and advance permitted, PARMAX is limited to the number of slots on the shortest link.
- o Maximum maneuver time (PARTIM) - Product of PARMAX and time to travel one slot.

- o Entrainment time limit (PNTRLM) - Time conversion.
- o Empty vehicle source options (PVSPR) - If demand responsive service; reasonableness check, determine number of selected options, check link types and translate to link identification (for option 2), check for storage link (for option 3), reasonableness check regional center source stations and convert from node identification to station identification (for options 4 and 6), determine that empty vehicles are assigned to circuitous routes (for option 5), preclude other options and build station link list to search when a vehicle is required (for option 6).
- o Entrainment (PENTS, PENTD) - Dynamic and static entrainment not permitted with timeout/group demand responsive service.
- o Maximum train length (PMXTRL) - If demand responsive service and dynamic or static entrainment permitted reasonableness check. Set to one if neither dynamic nor static entrainment are permitted.
- o Train length on route (PRTLEN) - If scheduled service; set to zero if one was input, reasonableness check, define PMXTRL as the maximum value of PRTLEN over all routes.
- o Product of vehicle length and maximum train length must not exceed the length of any guideway link.
- o Station link travel parameters (SLTTIM, SLHTA, SLHTB) - If headway event and travel event defined, time for maximum train size to travel headway zone must not exceed station link travel time. Reasonableness check of SLTTIM.
- o Trip split size (PTSPLT) - If not defined or invalid definition, set equal to vehicle capacity.
- o Seats per vehicle (VSEAT) - If not defined or invalid definition, set equal to vehicle capacity.
- o Vehicle capacity (VCAP) - Reasonableness check.
- o Regulation/control policy combination (POLLC, POLVPR) - Synchronous or quasi-synchronous control requires fixed block regulation.
- o Dispatch/control policy combination (POLDIS, POLLC) - Synchronous control requires deterministic dispatch, deterministic dispatch not permitted with asynchronous control.
- o Entrainment/control policy combination (POLLC, PENTD) - Dynamic entrainment only permitted with asynchronous control.

- o Entrainment (PENTS, PENTD) - Dynamic and static entrainment not permitted with scheduled service.
- o Merge/control policy combination (POLMRG, POLLC) - Heuristic merge only permitted with asynchronous control, synchronous control requires FIFO merge.
- o Path selection/control policy combination (PSMETH, POLLC) - Real time path selection only permitted with asynchronous control.
- o Path selection/service policy combination (POLSER, PSMETH, PSTYPE) - Real time path selection or algorithmic path selection not permitted with demand responsive multiparty service.
- o Path selection data (PSMETH, PSTYPE, PSALGM) - Reasonableness check.
- o Vehicle spacing algorithm (PVSPAC) - If scheduled service, reasonableness check.
- o Longitudinal control policy (POLLC) - Reasonableness check.
- o Demand stop option (POLDMS) - Demand stop not permitted with synchronous control.
- o Local merge priority (PMRGL) - Reasonableness check.
- o Achievable vehicle load factor (PLDFAC) - If demand responsive multiparty or scheduled service planned by input processor, reasonableness check.
- o Maximum passenger wait time (PMAXWT) - If scheduled service planned by input processor, reasonableness check.
- o Transfer policy (PXFER) - Transfers not required when input processor defines routes for scheduled service.
- o Source of route definition (PRTDEF) - If scheduled service, reasonableness check.
- o Vehicle reservations option (PVRES) - Reservations not permitted with scheduled service.
- o Service planning basis (PLOSBS, POLSER) - User must plan level of service for timeout/group demand responsive service.
- o Berthing policy (PBERTH, POLSER) - Ripple berth advancement may be selected for timeout/group demand responsive service only.
- o Random number seed (AKSEED) - Reasonableness check.
- o Checkpoint interval (ACKPTI) - Reasonableness check, time conversion.
- o Sampling interval (ASAMPI) - Reasonableness check, time conversion.

- o Online report interval (ASTATU) - Reasonableness check.
- o Periodic computation interval (APCOM1) - If heuristic merge, reasonableness check and time conversion.
- o Time to begin reading trip file (ATREAD) - Reasonableness check, time conversion.
- o Clock units/minute (CSIZE) - Reasonableness check.
- o Number of entries per clock table entry (CLOOP) - Reasonableness check.
- o Time increment between clock table intervals (CLSMAL) - Reasonableness check.
- o Number of entries in clock table (CLSIZE) - Reasonableness check.
- o Clock table definition (CLSMAL, CLSIZE, CSIZE) - Reasonableness check on expected utilization.
- o Vehicle log request (AVLOG) - Reasonableness check, conversion from node identification to station identification.

Any errors detected in the checks described above cause a message to be printed describing the condition. Errors that preclude further processing will result in execution termination after all checks have been made.

6.1.10.5 PDL

See Appendix A.

6.1.10.6 Decision Tables and Algorithms

None.

6.1.11 EIDRPT - Trip Demand Summary Report

6.1.11.1 Identification

- o EIDRPT - Trip Demand Summary Report Generation
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.1.11.2 Argument Dictionary

None.

6.1.11.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
C1LIM	-	I*4	Limit of first column index
C2STR	-	I*4	Starting point for second column index
DES1	-	I*4	1) Destination to be printed in first column 2) Destination station index
DES2	-	I*4	Destination to be printed in second column
DONE	-	L*1	Indicates write loop is complete
FIRST	-	L*1	Indicates first call to program
IDIS	-	I*4	Trip size distribution selection index
IFMT	-	I*4	Format selection corresponding to number of trip size distributions defined
ISIZ	-	I*4	Index to an element of a trip size distribution
LINE	-	I*4	Number of lines written on page

VARIABLE	DIM	TYPE	DESCRIPTION
LINREM	-	I*4	Number of lines remaining on page
NSCFAC	-	R*4	Net demand scale factor
ORG1	-	I*4	1) Origin to be printed in first column 2) Origin station index
ORG2	-	I*4	Origin to be printed in second column
PAGE	-	I*4	Current page number
PAGLIM	-	I*4	Lines per page limit
PASS	-	I*4	Scaled number of passengers/hour for an O/D pair
PC1	-	R*4	Generated passengers as percent of total to be printed in first column
PC2	-	R*4	Generated passengers as percent of total to be printed in second column
RPTNUM	-	I*4	Report number
RTMBAS	-	R*4	Time base of demand data
STP	-	I*4	Stop point for write loop on a page
STR	-	I*4	Start point for write loop on a page
TRIP	-	R*4	Scaled trips per hour
TSCFAC	-	R*4	Scale factor resulting from time base override

6.1.11.4 Description

This routine formats and outputs a summary of the trip demand generated by routine EITRIP. If trip generation was based on new demand input and description data, the following items relating to the specified demand pattern are printed:

- o One to three trip group size distributions
- o Cumulative probability of origin station usage
- o Cumulative probability of destination selection from each origin

The following items are listed for each interval of generated trips:

- o identification of demand matrix upon which trip generation was based
- o demand scaling summary
- o scaled demand matrix input by O/D
 - trips per hour
 - applicable trip size distribution
 - mean trip size
 - passengers per hour
- o generated trips by O/D
 - number of passengers
 - percent of total

6.1.11.5 PDL

See Appendix A.

6.1.11.6 Decision Tables and Algorithms

None.

6.1.12 EIDRSP - Demand Responsive Service Planning

6.1.12.1 Identification

- o EIDRSP - Demand Responsive Service Planning
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.1.12.2 Argument Dictionary

None.

6.1.12.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
CPTR	-	I*4	Index to an element in the empty vehicle anticipated need arrays
DEST	-	I*4	Destination station index
EMTIES	-	I*4	Number of empty vehicles to be sent from one station
EMTSTR	-	I*4	A specified empty vehicle management option
EVEHCT	-	R*4	Estimated number of empty vehicles in circulation at any given time
FLEET	-	I*4	Total fleet size
INT	-	I*4	Index to concatenated list in structured data format
LINK	-	I*4	Guideway link upstream from station
LNKSPC	-	I*4	Remaining space on guideway link upstream from station after first phase of empty vehicle assignment
ORIG	-	I*4	Origin station index

VARIABLE	DIM	TYPE	DESCRIPTION
OVEHCT	-	R*4	Estimated number of occupied vehicles in circulation at any given time
PRCSTN	KMS	I*2	Indicates that a station receives empty vehicles from one or more other stations. Equivalenced with global variable DTRPTB to reduce memory requirements
PSMALL	-	I*4	Smallest entry in list of negative vehicle surplus by station
PVEPRL	LPVEPH	I*2	Assigned priority of each empty vehicle management option
STORE	-	L*1	Indicates station includes a storage link
SURPLS	-	I*4	Total surplus from all sources
TFLEET	-	I*4	Temporary location for fleet size
USR	-	I*4	Index to concatenated list in user input data format
VASGNP	-	R*4	Fraction of unassigned fleet to assign to a station storage link
VEHSTA	-	I*4	Number of vehicles to be assigned to a station
XREAL	-	R*4	Number of vehicles as a real variable to force floating point arithmetic

6.1.12.4 Description

This routine generates the data required to operate the simulation in demand responsive mode. If the user requests the Input Processor to plan the level of service, the program estimates the number of vehicles required and the redistribution of empty vehicles based on a specified trip demand pattern. For multiparty service, the fleet is estimated for single party service and then converted to multiparty using vehicle capacity, average number of passengers in a trip and estimated achievable vehicle load factor. The resulting fleet is assigned to the guideway links immediately upstream from stations and to station storage links.

This routine also checks and transforms the empty vehicle redistribution data entered by the user.

6.1.12.5 PDL

See Appendix A.

6.1.12.6 Decision Tables and Algorithms

If the routine is to plan the level of service, the following algorithms are used:

1. Compute the net scale factor to be applied to the trip demand matrix.

$$S_N = S_U \times T_D / T_U \quad (1)$$

where

S_N = net scale factor

S_U = user-defined scale factor applicable to current demand profile interval

T_D = original time base associated with current demand matrix

T_U = user-defined override to the time base applicable to current demand profile interval

2. Estimate the number of occupied vehicles traveling from station to station at any given time.

$$N_{OCC} = \sum_{j=1}^s \sum_{i=1}^s S_N \times TD_{i,j} \times T_{i,j} / 3600 \quad (2)$$

where

N_{OCC} = number of occupied vehicles

s = number of stations

$TD_{i,j}$ = demand for trips from i to j (trips/hour)

$T_{i,j}$ = station to station travel time (seconds)

3. Establish input parameters for empty vehicle allocation algorithm

$$a. \quad E_i = \sum_{j=1}^s S_N \times TD_{j,i} - \sum_{j=i}^s S_N \times TD_{i,j} \quad (3)$$

where

E_i = list of expected excess of vehicles (more arrivals than departures) by station; if an element of E_i is negative, that station is expected to have a deficit of vehicles.

- b. Determine station ID and amount of deficit for each station having a deficit of vehicles.
 - c. Determine station ID and amount of deficit for each station having a deficit of vehicles.
4. Execute the empty vehicle allocation algorithm which is implemented in subroutine EIEMPTY (see subsection 6.1.13). This algorithm is the classic transportation problem, wherein the distribution of supply from a set of sources to a set of sinks is optimized to produce a minimum total cost (vehicle travel time).
 5. Estimate the number of empty vehicles traveling from station to station at any given time.

$$N_{EMP} = \sum_{j=1}^s \sum_{i=1}^s EVM_{i,j} \times T_{i,j}/3600 \quad (4)$$

where

N_{EMP} = number of empty vehicles

$EVM_{i,j}$ = empty vehicle distribution matrix generated by subroutine EIEMPTY

6. Estimate total fleet size

$$N_V = N_{OCC} + N_{EMP} \quad (5)$$

7. If multiparty service is specified, the fleet is modified to account for multiple trips on vehicles.

- a. Calculate average group size as a weighted average of the three group size distributions.

$$A = \left(\sum_{i=1}^3 N_i \times M_i \right) / \left(\sum_{i=1}^3 N_i \right) \quad (6)$$

where

A = average group size

M_i = mean of group size distribution i

N_i = number of trips using group size distribution i

- b. Adjust the number of vehicles required.

$$N_V = N_V \times A / V_{CAP} \times L \quad (7)$$

where

N_V = total fleet size (see equation 5)

V_{CAP} = vehicle capacity

L = estimated achievable load factor

8. Calculate empty vehicle redistribution patterns from the empty vehicle distribution matrix. The patterns are defined as frequency distributions and as cumulative probability distributions. A frequency distribution is formed for each row (origin station of the empty vehicle distribution matrix by compressing to save only the non-zero values. The column numbers (receiving stations) that contained non-zero values are also saved. The frequency distributions are concatenated and the receiving station lists are concatenated. A list of pointers is produced which defines the beginning location of each frequency distribution and receiving station list. If a row of the matrix had no non-zero entries, the frequency distribution will consist of a one and the corresponding receiving station will be the closest downstream station. The frequency distributions are then formed into cumulative probability distributions.
9. Assign the vehicle fleet to guideway links and station storage.

- a. If non-deterministic dispatch was specified, vehicles can be placed on the guideway links that are immediately upstream of stations.

$$N_G = N_V/s, \quad 1 \leq N_A \leq 3 \quad (8)$$

where

N_G = number of vehicles to attempt to assign to each guideway link that is upstream from a station.

For each station until the entire fleet has been placed, set the occupancy of the guideway link upstream from the station to the minimum of N_G and the guideway link capacity. Decrement the total number of vehicles remaining to be assigned by the number of vehicles assigned to the link.

- b. If unassigned vehicles remain, they will be placed on station storage links, if a storage link exists. This is done such that the stations with the greatest expected need for empty vehicles will be assigned more vehicles.

$$D_i = -E_i, \quad i = 1 - s \quad (9)$$

$$B = \left| \text{MIN} (D_i, i = 1 - s) \right| + 3 \quad (10)$$

$$S = \sum_{i=1}^s (D_i + B) \quad (11)$$

$$N_S = F \times (D_1 + B) / S \quad (12)$$

where

D_i = expected vehicle deficit at station i (negative value is a surplus)

B = bias so that stations with an expected surplus will be assigned vehicles

S = summation over all stations of biased deficit (used to normalize each station's deficit)

N_S = number of vehicles to be assigned to the storage link in station i .

F = number of unassigned vehicles when step 9b was started.

For each station until the entire fleet has been placed, set the occupancy of the station storage link equal to N_S . If no storage link available and non-deterministic dispatch specified, add the minimum of N_S and the remaining space on the upstream guideway link to the occupancy of the upstream guideway link. Decrement the total number of vehicles remaining to be assigned by the number of vehicles assigned to the station storage link or upstream guideway link.

- c. If unassigned vehicles remain, assign them to the first station that has a storage link.
- d. If unassigned vehicles remain and non-deterministic dispatch specified, assign them to upstream guideway links without exceeding link capacity.
- e. If unassigned vehicles remain, write an error message and request execution termination.

6.1.13 EIEMTY - Empty Vehicle Allocation

6.1.13.1 Identification

- o EIEMTY Empty Vehicle Allocation
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.1.13.2 Argument Dictionary

None.

6.1.13.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
IFIRST	-	I*4	Index of first element assigned to a row
LAST	-	I*4	Index of last element assigned to a row
LCOST	-	I*4	Cost of travel from a source to a sink

6.1.13.4 Description

This routine determines the empty vehicle redistribution pattern such that the total travel time of empty vehicles is minimized. This is accomplished by using a transportation algorithm which is implemented in routine EIMNTP. Routine EIEMTY acts as the interface between the Input Processor and the transportation algorithm.

After initializing the transportation algorithm variables, the solution matrix is initialized to zero. The inputs to the transportation algorithm are established, including: number of sources, number of sinks, list of excess vehicles at each source station, list of required vehicles at each sink station, and the nominal travel time from each source station to each sink station. The transportation algorithm is invoked to determine the flow of empty vehicles from the source stations to the sink stations such that the total travel time is a

minimum. The results from the transportation algorithm are then transferred to the solution matrix.

6.1.13.5 PDL

See Appendix A.

6.1.13.6 Decision Tables and Algorithms

The results from the transportation algorithm are stored in a packed list form in variables ROW, COLUMN and X where:

ROW = cumulative total of the number of sinks receiving vehicles from each source

COLUMN = identification of the sinks which are to receive vehicles

X = number of vehicles to be received by the sinks identified in COLUMN

Then, for the i th source, the sinks are:

$COLUMN_{(ROW_{(i-1)} + 1)}$ through $COLUMN_{(ROW_i)}$

and the numbers of vehicles to be sent to those sinks are:

$X_{(ROW_{(i-1)} + 1)}$ through $X_{(ROW_i)}$

Routine EIEMTY performs this unpacking and maps the source and sink indices to the corresponding station identification values, which define the rows and columns of the solution matrix to receive the allocation amounts stored in X.

6.1.14 EIERROR - Error Message Processing

6.1.14.1 Identification

- o EIERROR - Error Message Processing
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.1.14.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
MSGNO	-	I*4	Error message number
MSG	2	L*1	Message text
MSEVER	-	I*4	Message severity code
			1 - Information
			2 - Warning
			3 - Serious

6.1.14.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
MCLOCK	-	I*4	Clock time (not used)
MSGTYP	-	L*1	Alphabetic message type designator
NUM	-	I*4	Number of characters in message text
PGM	3	I*4	Processor identifier
SCLN	-	L*1	End of text symbol (;)
TYPE	3	L*1	Message type symbols (I,W,S)

6.1.14.4 Description

Upon entry, EIERROR determines whether or not error processing was invoked while attempting to terminate IP processing due to a prior error condition. If this situation exists, a message indicating the condition is issued and immediate termination of the IP occurs. Otherwise, the required error message is formatted and displayed on the system output device.

After message display, a determination is made as to whether any specified output limits have been violated or whether system termination is required based on message severity. If the number of times a given message is issued, total number of messages issued or the number of messages by severity code exceed SYSGEN limits, termination of the IP occurs. This termination activity attempts graceful system shutdown by invoking the DESM traceback facility (TRACBK). Similar processing is performed following the issuance of a severe error message. If error processing results in IP termination for any reason, a system condition code 16 is returned to the user.

6.2.7.5 PDL

See Appendix A.

6.2.7.6 Decision Tables and Algorithms

None.

6.1.15 EIFAIL - Failure/Recovery Processing

6.1.15.1 Identification

- o EIFAIL - Failure/Recovery Processing
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.1.15.2 Argument Dictionary

None.

6.1.15.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
BADREQ	-	I*4	Indicates invalid request
COL1	-	I*2	Index for column 1 printing
COL2	-	I*2	Index for column 2 printing
EFAIL	-	I*4	Request type one (failure)
ELINK	-	I*4	End of range of affected station links
ENODE	-	I*4	End node of affected guideway link
FIRST	-	L*1	Indicates first call to routine
FOUND	-	L*1	Indicates link is on tow path
GLENT	KML	I*2	Guideway link entry status
GLEXIT	KML	I*2	Guideway link exit status
LINK	-	I*4	Affected guideway link
LINREM	-	I*4	Lines remaining to be printed on a page
NFAILS	-	I*4	Number of active guideway failures

VARIABLE	DIM	TYPE	DESCRIPTION
PAGE	-	I*4	Current page number
PAGLIM	-	I*4	Lines per page limit
REQST	-	I*4	Request type
SL	-	I*4	Index to station link type array
SLENT	KMSL, KMS	L*1	Station link entry status
SLEXIT	KMSL, KMS	I*2	Station link exit status
SLINK	-	I*4	Start of range of affected station links
SNODE	-	I*4	Start node of affected guideway link
STATN	-	I*4	Affected station
TCNVRT	-	R*4	Time conversion, seconds to clock units
TCUSEC	-	R*4	Time conversion, clock units to seconds
TIME	-	I*4	Time of failure/recovery request
TIME1	-	R*4	Link travel time in seconds for column 1 printing
TIME2	-	R*4	Link travel time in seconds for column 2 printing

6.1.15.4 Description

This routine processes user requests to insert and remove failures in the guideway or station links. It also generates a summary report for each insertion and removal.

If the request concerns a guideway link or vehicle degradation, the link that connects the nodes entered by the user is determined. A zero is placed in the first element of the Model Processor failure/recovery request array (indicating a guideway/vehicle request) and the specified link is placed in the second element of the array.

If the request is a guideway link failure and the maximum number of least cost path tables has not been reached, the request is checked to determine if the link entry, exit or the entire link is to be failed. Multiple requests for the same failure are ignored. The input defining other vehicle responses is checked for validity. A large value is added to the cost (travel time or link length) of using the specified link and the least cost path algorithm is invoked by calling routine EIMPTH to recalculate the least cost paths in the network. The number of concurrent least cost path tables and the number of active guideway link failures are incremented. The ID of the current least cost path table is set equal to the number of concurrent least cost path tables. Routine EINSLT is called to build the new least cost path table based on the results generated by routine EIMPTH.

If the request is a guideway link recovery and it is the only existing guideway failure or the maximum number of least cost path tables has not been reached, the request is checked to determine if the link entry, exit or the entire link is to be recovered. Multiple requests for the same recovery are ignored. The previous failure card matching this recovery card is found and pointed to by the 10th element of the MP failure array. The cost of using the specified link is restored to its original value. The number of active guideway failures is decremented and if guideway failures still exist, routine EIMPTH is called to recalculate the least cost paths considering the remaining failures. The number of concurrent least cost path tables is incremented and the ID of the current least cost path table is set equal to it. Routine EINSLT is called to build the new least cost path table based on the results generated by routine EIMPTH. If the recovery request removes the last guideway failure, the ID of the current least cost path table is set to one and no recalculation of paths occurs.

If the request is for vehicle degradation and the longitudinal control mode is asynchronous, the link exit or entry is marked to capture the next vehicle and the degradation factor is checked for reasonableness. The delay times are converted to clock units, and along with the recovery methods selection, are stored in the MP array. If the recovery method is tow by service vehicle, the tow vehicle speed degradation factor is checked for reasonableness and the tow path is validity checked and marked to block all merging links into the path. A large value is added to the cost of traveling these blocked links. Other vehicle responses are checked for reasonableness and passed to the MP array. If scheduled service, the closest maintenance barns are calculated for each route if they have not been found previously. A barn on the route is chosen if an available storage to dock link exists. Otherwise, the barn closest after any station stop on the route is selected. Then the closest station stop after the selected barn is found for each route. A large value is added to the cost of the link specified for degradation and the least cost path algorithm is invoked in routine EIMPTH if selected by the user to recalculate the least cost paths in the network. The number of concurrent least cost path tables and the number of active failures are incremented. The ID of the current least cost path table is set equal to the number of concurrent least cost path tables. Routine EINSLT is invoked to build the new least cost path table based on the results generated by routine EIMPTH.

If the request is for vehicle degradation recovery, the vehicle degradation factor is returned to its nominal value, and the marking of the link exit or entry to capture the next vehicle is cancelled. The previous fail card matching this recovery card is found and pointed to by the 10th element of the MP failure array. If the recovery method is tow by service vehicle, the blocking of links merging into the tow path is cancelled and the large value is subtracted from the cost of traveling these links. The large value is also subtracted from the cost of traveling the link specified for degradation. The number of active failures is decremented and, if failures still exist, routine EIMPTH is invoked to recalculate the least cost paths considering the remaining failures, if the user chooses. The number of concurrent least cost path tables is incremented and the ID of the current least cost path table is set equal to it. Routine EINSLT is called to build the new least cost path table based on the results generated by routine EIMPTH. If the recovery request removes the last guideway failure or degradation condition, the ID of the current least cost path table is set to one and no recalculation of paths occurs.

If the request concerns a station, the station located at the node specified by the user is determined. If a particular station link type was specified, the first station link of that type is found and the ID of that link is placed in the second element of the model processor failure/recovery request array. If the entire station is to be failed/recovered, the second element of the model processor failure/recovery request array is set to zero. The first element of the request array is set to the station ID. If the request is for station link degradation the degradation factor for the specified station links is set to the value entered by the user. If the request is for station link degradation recovery the degradation factor for the specified links is restored to the nominal value.

The entry/exit and request type information is transferred to the third and fourth elements of the model processor request array.

Following this request processing, a report is generated which describes the failure or recovery. If a new least cost path table was generated, it is listed in the report.

6.1.15.5 PDL

See Appendix A.

6.1.15.6 Decision Tables and Algorithms

None.

6.1.16 EIGDIP4 - Generalized Data Input Variables

6.1.16.1 Identification

- o EIGDIP4 - Generalized Data Input Variables
- o IBM/FSD - May 1, 1978
- o ASM

6.1.16.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
NAME	-	R*8	Parameter name
FMT	-	R*8	Format of data
IRAL	-	I*4	First dimension lower bound
IRAH	-	I*4	First dimension upper bound
IRBL	-	I*4	Second dimension lower bound
IRBH	-	I*4	Second dimension upper bound
IRCL	-	I*4	Third dimension lower bound
IRCH	-	I*4	Third dimension upper bound
IRDL	-	I*4	Fourth dimension lower bound
IRDH	-	I*4	Fourth dimension upper bound

6.1.16.3 Local Variable Dictionary

None.

6.1.16.4 Description

This routine provides addressability to each parameter in the COMMON areas defined in the routine. When data is read from the input device, the name of each parameter is matched to the data mapping table

and the appropriate read routine (XGDIPF4, XGDIPH4, or XGDIPX4, depending on whether the parameter is full word, half word or byte data) is invoked to read the data into the proper area in COMMON. If the parameter name is not found in the table, an error message is generated. The significant features of this routine are:

1. The statement:

```
NODIMENS 4
```

defines the maximum number of dimensions of any array to be four. This value may be changed only with corresponding changes to XNDBOR and by providing new routines named XGDIPFn, XGDIPHn, and XGDIPXn (internal routines currently provided with n = 4) to allow data formatting into higher dimensioned arrays.

2. Each COMMON area requiring GDIP data loading is defined by a set of cards, consisting of the following:

- a. A card to name the COMMON area, in the form:

```
common-name CSECT
```

- b. One or more cards to define the variables in the COMMON area. These definitions must be in precisely the same order as in the corresponding FORTRAN COMMON statement since they are used to define a data map of each variable in the common area such that addressability can be established to any data position. The format of each card is:

```
COMN 1,name,dimensions,[name-2,dimensions-2, . . .]
```

Up to 20 variables may be defined per card, provided their data item lengths are the same. The field "1" must be one of the following:

<u>1</u>	<u>Variable type(s)</u>
F	REAL*4, INTEGER*4, and LOGICAL*4
H	INTEGER*2
X	LOGICAL*1

The dimensions field is of the form:

(first-dimension, . . . , fourth-dimension)

If a variable has fewer than four dimensions, only the necessary ones are given. If the variable has no dimensions (i.e., is unsubscripted), then this field is null. However, the common must always be supplied.

3. The statement:

ENDEFS

marks the end of all the definitions and causes the data mapping to be established.

4. The pair of statements:

ERRORXIT . . .

B RETURN

defines how errors are to be handled. If GDIP encounters an undefined variable in the input stream, control branches to ERRORXIT. The branch to RETURN causes GDIP to read and process the next input card.

5. The statement:

END

terminates the definition routine code.

6.1.16.5 PDL

See Appendix A.

6.1.16.6 Decision Tables and Algorithms

None.

6.1.17 EIINIT - Input Processor Initialization

6.1.17.1 Identification

- o EIINIT - Input Processor Initialization
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.1.17.2 Argument Dictionary

None.

6.1.17.3 Local Variable Dictionary

None.

6.1.17.4 Description

This routine initiates the processing that results in the location and length of the structured data COMMONS being defined in routine EIBWRT, which writes the structured network and system characteristics files. The statement CALL LODCOM causes COMMON EIPSAV to be established in core. EIPSAV contains an array of four elements, which gives the addresses of the four overlay regions defined in the linkage editor control statements. The second element contains the address of the first parameter in the network structured data and the difference between the third and second elements divided by four is the number of full words in the network structured data. The third element contains the address of the first parameter in the system characteristics structured data and the difference between the fourth and third elements divided by four is the number of full words in the system characteristics structured data. This information is passed to routine EIADDR which establishes parameters equivalenced to the structured data COMMONS in routine EIBWRT.

EIINIT also initiates the processing that results in the names of the files defined by the user in the execution JCL being decoded from the character string in the PARM field into separate eight character member names. Entry point EIPLST in routine EIPARM is called, which passes the address of the character string and the number of characters to routine EIMNAM for decoding.

EIINIT then initializes all of the variables in the structured network and system characteristics COMMONS and most of the variables in the COMMONS used only by the Input Processor. Those variables that are not initialized are used in routines that can be called more than once during an Input Processor execution. The variables are initialized by the using routine.

6.1.17.5 PDL

See Appendix A.

6.1.17.6 Decision Tables and Algorithms

None.

6.1.18 EIMNAM - Parameter List Decode

6.1.18.1 Identification

- o EIMNAM - Parameter List Decode
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.1.18.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
COUNT	-	I*2	Length of character string in PARM field of EXEC statement
STRING	COUNT	L*1	PARM field of EXEC statement

6.1.18.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
BLANK	-	L*1	Space character
CHAR	-	I*2	One character from the PARM field
COMMA	-	I*2	Right justified comma
DAY	-	I*2	Day of month
HOUR	-	I*2	Hour of day
J	-	I*4	Index to a character in a file name
MIN	-	I*2	Minute of hour
MON	-	I*2	Month of year
PTR	-	I*2	Index to a character in the PARM field
YEAR	-	I*2	Current year

6.1.18.4 Description

The PARM field of the EXEC statement contains eleven fields separated by commas. They are:

1. Load module identification
2. System characteristics input and description file member name
3. Runtime input and description file member name
4. Network input (input and description or structured) file member name
5. Network structured data output file member name
6. Nominal travel time file member name
7. First demand input and description file member name
8. Second demand input and description file member name
9. Third demand input and description file member name
10. Fourth demand input and description file member name
11. Fifth demand input and description file member name

EIMNAM separates the PARM field into these eleven fields by scanning the PARM field for the field delimiter, a comma. Each name is stored left justified within an eight character field for later use in recording the files actually used during execution. Routine DAYTIM is called to obtain the current date and time. This information and the load module identification are written to the run index file to record the initiation of an Input Processor execution.

Entry point EIWNAM is called at the end of the Input Processor execution. This routine lists the names of the files actually read or written during Input Processor execution in the run index file and on the system printer.

6.1.18.5 PDL

See Appendix A.

6.1.18.6 Decision Tables and Algorithms

None.

6.1.19 EIMNTP - Transportation Algorithm

6.1.19.1 Identification

- o EIMNTP - Transportation Algorithm
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o FORTRAN IV

6.1.19.2 Argument Dictionary

ENTRY	VARIABLE	DIM	TYPE	DESCRIPTION
PROC	ISLACK	-	I*4	Indicates whether a slack row, column or both are to be used 0 = none 1 = row or column 2 = both row and column
	ISCIJ	-	I*4	Cost to be assigned to all slack variable assignments
ENTERC	II	-	I*4	Row number (source)
	JJ	-	I*4	Column number (sink)
	IAMT	-	I*4	Cost of assigning the row to the column

6.1.19.3 Local Variable Dictionary

ENTRY	VARIABLE	DIM	TYPE	DESCRIPTION
PROC	CE	-	I*4	Number of cost entries
	CM	-	I*4	Number of elements in cost matrix

ENTRY	VARIABLE	DIM	TYPE	DESCRIPTION
	DENSIT	-	R*4	Density of number of cost entries relative to size of cost matrix
	IDIF	-	I*4	Excess supply
	INF	-	I*4	Number of cost entries having the infinite cost value
	ITOT	-	I*4	Sum of all cost entries
	ITOTU	-	I*4	Sum of supply at all sources
	ITOTV	-	I*4	Sum of demand at all sinks
	KAVG	-	I*4	Average cost
	LIMIT	-	I*4	Index of last entry to process
BASIS	ICOST	-	I*4	Minimum cost value in a given column
	IFIRST	-	I*4	Index of first element in a given row
	IROW	-	I*4	Row that contains the minimum cost value in a given column
	IX	-	I*4	Minimum of supply and demand at the row/column containing the minimum cost value
	LAST	-	I*4	Index of last element in a given row
ENTERC	ICOL	-	I*4	Column number of cost entry
	IFIRST	-	I*4	Index of first entry in specified row

ENTRY	VARIABLE	DIM	TYPE	DESCRIPTION
	INSERT	-	I*4	Index where cost value is to be inserted
	IROW	-	I*4	Row number of cost entry
	KPOS	-	I*4	Column number of entry in specified row
	LAST	-	I*4	Index of last entry in specified row
LINC	DIJ	-	I*4	Potential reduction factor for current objective function
	IC	-	I*4	Cost assigned to specified element
	ICK	-	I*4	Index of specified element in list
	ICKTIM	-	I*4	Total run time in integer minutes (not used)
	ICLOCK	-	I*4	Clock time at start of optimization (not used)
	ICNT	-	I*4	Not used
	IHIT	-	I*4	Identification of element to be brought into basis has occurred
	II	-	I*4	Column designations of element
	INF	-	I*4	Number of elements having infinite cost
	INSERT	-	I*4	Element number to be inserted into basis
	IPATH	-	I*4	Accumulated cost of the path
	ISAVE	-	I*4	Absolute value of row designation of element

ENTRY	VARIABLE	DIM	TYPE	DESCRIPTION
	ITERA	-	I*4	Iteration number
	JCLOCK	-	I*4	Current clock time (not used)
	JSAVE	-	I*4	Absolute value of column designation of element
	KCLOCK	-	I*4	Clock time at start of iteration (not used)
	LIMIT	-	I84	Index of last entry to process
	MAJIT	-	I*4	Major iteration number
	MAXTIM	-	I*4	Maximum time permitted for optimization
	MINCEL	-	I*4	Minimum cell in loop
	MISS	-	I*4	Element brought into basis was wrong choice
	NBRC	-	I*4	Number of cells in loop
	NBRI	-	I*4	Number of iterations
	NEWOBJ	-	I*4	New value of objective function
	RUNI	-	R*4	Time required for iteration (not used)
	RUNTOT	-	R*4	Total run time in floating point minutes (not used)
	UPLUSV	-	I*4	Sum of supply and demand dual variables
DUAL	ICNT	-	I*4	Number of elements in a row
	IFIRST	-	I*4	Index of first column in a row
	ISAVE	-	I*4	Row that contained the maximum number of elements

ENTRY	VARIABLE	DIM	TYPE	DESCRIPTION
	J	-	I*4	Column designation of an element
	LAST	-	I*4	Index of last column in a row
	LIMIT	-	I*4	Maximum limit for iteration to find dual variables
	MAXCNT	-	I*4	Maximum number of elements in a row
	NOTDON	-	I*4	Indicates whether dual variables have been found
CELLNR	I	-	I*4	Row designation of element K
	J	-	I*4	Column designation of element K
	K	-	I*4	Index of an element
INDEX	I	-	I*4	Row designation of an element
	INDEX	-	I*4	Index of the element in column J of row I
	J	-	I*4	Column designation of an element
ENTER	ICOL	-	I*4	Column designation of element to be added
	IFIRST	-	I*4	Index of first element to be moved
	INSERT	-	I*4	Index where new element is to be inserted
	IROW	-	I*4	Row designation of element to be added
	IX	-	I*4	Value of solution matrix element
	IXPRIM	-	I*4	Internal value for IX

ENTRY	VARIABLE	DIM	TYPE	DESCRIPTION
	LAST	-	I*4	Index of last element to be moved
EXIT	ICOL	-	I*4	Column designation of element to be deleted
	IFIRST	-	I*4	Index of first position in list to be updated
	IROW	-	I*4	Row designation of element to be deleted
	K	-	I*4	Index of element to be deleted
	LAST	-	I*4	Index of last position in list to be updated
LOOP	INSERT	-	I*4	Number of element being inserted
	IROW	-	I*4	Row designation of inserted element
	LEAVE	-	I*4	Element that is leaving the basis
	NBRL	-	I*4	Node number of the element that is leaving the basis
FINDL	IC	-	I*4	Column designation of element
	ICELL	-	I*4	Index of element in specified row and column
	ICOL	-	I*4	Column designation of element
	IR	-	I*4	Row designation of element
	IROW	-	I*4	Row designation of element
	LOOKI	-	I*4	Row dual variable index
	LOOKJ	-	I*4	Column dual variable index

ENTRY	VARIABLE	DIM	TYPE	DESCRIPTION
	NBRB	-	I*4	Number of elements in back of loop
	NBRF	-	I*4	Number of elements in front of loop
	NUMBER	-	I*4	Working index for loop creation
MATCH	IFIRST	-	I*4	Index of first element in a given row
	KOLB	-	I*4	Column of element in back of loop
	KOLF	-	I*4	Column of element in front of loop
	LAST	-	I*4	Index of last element in a given row
	LSAVE	-	I*4	Working indicator to find loop connection
	NBRB	-	I*4	Number of elements in back of loop
	NBRF	-	I*4	Number of elements in front of loop
UPDATE	IU	-	I*4	Interim calculation of dual variable flow change for rows
	IUP	-	I*4	Dual variable flow change for rows
	IV	-		Interim calculation of dual variable flow change for columns
	IVP	-	I*4	Dual variable flow change for columns
	LEAVE	-	I*4	Element number that has left the basis

ENTRY	VARIABLE	DIM	TYPE	DESCRIPTION
	NBRL	-	I*4	Node number of the element that left the basis
TEST	None			
PARA	Not used			
PARAUP	Not used			
TIME	I	-	I*4	Current clock value (zero is returned to disable the optimization timing function)

6.1.19.4 Description

The Transportation Algorithm is a collection of routines that provide the capability to solve several transportation-related problems. One application of the algorithm is implemented in the Input Processor. The service planning function uses the algorithm to optimize the redistribution of empty vehicles in demand responsive type service.

The algorithm is based on the "stepping stone" technique, which is a special case of the general simplex method used in linear programming. There are three major inputs to any transportation problem: a supply vector, a demand vector, and a cost matrix. The objective of the solution process is to allocate the supply to the demand such that the total cost is a minimum.

In the empty vehicle redistribution application, the supply vector is a list of expected excess of empty vehicles by station (only those stations expected to have an excess are included). The demand vector is a list of expected deficit of empty vehicles by station (only those stations expected to have a deficit are included). Excess and deficit are estimated by the CALLING routine. The cost matrix is defined by entering the cost of travel between the supply stations and the demand

stations as determined in the least cost path application. The least cost solution to this application is a matrix containing the numbers of empty vehicles that are to be sent from each supply station to each demand station.

The supply and demand vectors are passed to the transportation algorithm through COMMON. They are defined as follows:

U_i = list of amount of supply by node

V_i = list of amount of demand by node.

The cost matrix is defined to the algorithm using the Enter Cost routine. The calling sequence is:

ENTERC (i, j, C_{ij})

where:

i -- Origin node

j -- Designation node

C_{ij} -- Cost of travel from i to j.

The algorithm is invoked using the calling sequence:

PROC (SLACK, SCIJ)

where:

SLACK -- Selection parameter which results in the addition of artificial supply and demand to obtain an initial feasible solution

- = 0, no additional supply and demand
- = 1, add an artificial supply node
- = 2, add both artificial supply and demand nodes.

SCIJ -- Cost to be associated with using the artificial supply and demand nodes.

6.1.19.5 PDL

None.

6.1.19.6 Decision Tables and Algorithms

There are three major inputs to any transportation problem: a supply vector (a_i), a demand vector (b_j) and a cost matrix (c_{ij}). The objective of the solution process is to allocate the supply to demand by forming a solution matrix. This allocation must be accomplished considering the minimum penalties in the cost matrix.

Assume that m origins with a supply a_i at each origin and n destinations with a demand b_j at each destination are given. In addition, a cost c_{ij} of shipping a unit from origin i to destination j is also given. The solution matrix x_{ij} is computed such that Z , the total shipping cost, is minimized:

$$\text{(minimize) } Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (1)$$

where:

$$\sum_{j=1}^n x_{ij} = a_i; \quad i = 1, 2, \dots, m \quad (2)$$

$$\sum_{i=1}^m x_{ij} = b_j; \quad j = 1, 2, \dots, n \quad (3)$$

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j \quad (4)$$

$$x_{ij} \geq 0; \quad i=1, 2, \dots, m; \quad j = 1, 2, \dots, n \quad (5)$$

The following figure illustrates the sample problem that will be used to describe the transportation algorithm.

		b_j						
		4	6	4	8	5		
	a_i	8	8	6	9	15	14	
		9	7	2	5	10	13	c_{ij}
		10	12	8	3	4	11	

The first step in the solution process is to find a basic feasible solution. This initial solution must have triangular properties, i.e., an element may exist in the solution only if there exists another element in the same row or column. In addition, the solution cannot contain any loops, i.e., a tracing through row and column elements cannot return to original row or column. The following solution matrix contains a basic feasible starting solution.

		b_j						
		4	6	4	8	5		
	a_i	8		1		2	5	
		9	4	5				x_{ij}
		10			4	6		

This solution was obtained by allocating column by column to the lowest remaining cost in each column. The objective function of this solution is 188, i.e.,

$$Z = \sum_i \sum_j c_{ij} x_{ij} = 188$$

The first step in the optimization process is to compute the simplex multipliers (dual variables). The row with the most elements is selected (e.g. row 1 with 3). The row dual variables are designated u_i and the

column dual variables are designated v_j . They are functions of the cost matrix (c_{ij}) elements that are coincident with the elements in the current solution matrix x_{ij} .

$$u_i = c_{ij} - v_j$$

$$v_j = c_{ij} - u_i$$

		v_j					
		11	6	14	15	14	
	0	8	6	9	15	14	
u_i	-4	8	2	5	10	13	c_{ij}
	-11	12	8	3	4	11	

(NOTE: indicates element is in x_{ij} matrix)

The process now requires that a new element be entered into the basis and one removed in an iterative fashion until the optimum solution is achieved. To locate the element that should be entered into the solution, it is necessary to compute d_{ij} . This value is the potential reduction factor for the current objective function if the corresponding x_{ij} element is entered.

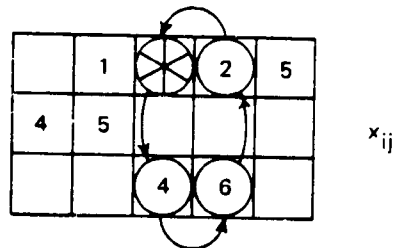
$$Z^* = Z - d_{ij} x_{ij}$$

When Z^* is the new objective function, Z is the current objective function and:

$$d_{ij} = c_{ij} - (u_i + v_j)$$

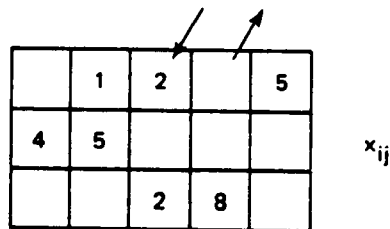
		v_j					
		11	6	14	15	14	
	0	-3	0	-5	0	0	
u_i	-4	0	0	5	-1	3	d_{ij}
	-11	12	13	0	0	8	

There are numerous solution strategies possible as to which negative d_{ij} should be selected. For the purpose of this explanation, the $d_{ij} = -5$ in row 1, column 3 is selected. The loop that this variable forms in the basis is illustrated in the following solution matrix:



The x_{ij} flow through this loop is the maximum value that will reduce one (or multiple) x_{ij} in the loop to zero and not negative. In this case, this limiting x_{ij} is at $x_{1,4} = 2$ and this variable leaves the basis.

After the loop is established with a flow of 2, the resulting solution matrix appears as follows:



The solution is still basic and feasible but the objective function is reduced by $d_{13} \cdot x_{14} = -10$. So Z now equals 178.

This iterative process of recomputing dual variables and d_{ij} and entering new variables into the basis continues until the d_{ij} calculation does not yield any negative values. This indicates that the optimum solution has been found. In this sample problem, the optimum solution exists with an objective function of 161:

		b_j					
		4	6	4	8	5	
		4				4	
a_i	8						
	9		6	3			x_{ij}
	10			1	8	1	

Optimum Solution

6.1.19a EIMORG - Timeout/Group Demand Responsive Service Planning

6.1.19a.1 Identification

- o EIMORG - Timeout/Group Demand Responsive Service Planning
- o GM TSC - July 1, 1981
- o PARAFOR

6.1.19a.2 Argument Dictionary

None.

6.1.19a.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TCNVRT	-	R*4	Time conversion, seconds to clock units

6.1.19a.4 Description

This routine performs input validation and clock conversions for parameters used in timeout/group demand responsive service. Trip platform assignments, ready request parameters, and station overfull parameters are checked for validity. The maximum wait time to request a vehicle parameter is converted to clock units.

6.1.19a.5 PDL

None.

6.1.19a.6 Decision Tables and Algorithms

None.

6.1.20 EIMPTH - Least Cost Path Determination

6.1.20.1 Identification

- o EIMPTH - Least Cost Path Determination
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.1.20.2 Argument Dictionary

None

6.1.20.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
KNLR	-	I*2	Number of links in reduced network
KNLRS	-	I*2	Current number of links when processing a diverge
KNNR	-	I*2	Number of nodes in reduced network
LCOST	-	I*4	Cost of using the link connecting a pair of adjacent nodes
LINK	-	I*4	Identification of link leaving a node
NODI	-	I*4	Node ID loop counter
NODJ	-	I*4	Node ID loop counter
NODK	-	I*4	Node ID loop counter
NRCOST	KML	I*4	Cost of using links in reduced network
NRD	KMN	L*1	Indicates that least cost paths must be found to the corresponding node
NRDEST	KML	I*2	Destination nodes of links in reduced network
NRO	KMN	L*1	Indicates that least cost paths must be found from the corresponding node to the nodes indicated in NRD
NRORIG	KML	I*2	Origin nodes of links in reduced network

VARIABLE	DIM	TYPE	DESCRIPTION
TCNVRT	-	R*4	Time conversion, seconds to clock units

6.1.20.4 Description

This routine determines the least cost paths in the network by invoking a minimum path algorithm. EIMPTH first reduces the problem to be solved by the minimum path algorithm by scanning the network connectivity data and eliminating nodes which are not major nodes. Major nodes are defined as merges, diverges or nodes immediately downstream from diverges. The costs of using the links which connect the major nodes are obtained by summing the costs of the individual links in the original network. Cost is selected by the user as either link nominal travel time or link length. The major nodes that must be considered as origins and/or destinations are also identified during the network reduction process. Only those major nodes that are diverges or immediately downstream from diverges must be considered as origins. Only those major nodes that are upstream from stations must be considered as destinations.

If the original network did not include any merges or diverges (i.e., a simple loop) the network reduces to null and no further processing is performed by EIMPTH. Otherwise, the process continues as described below.

The minimum path algorithm is then executed. The cost of traveling between each node pair in the reduced network is initialized to the cost assigned during the network reduction processing. Then all other alternative paths through the reduced network are analyzed to see if a lower cost path exists. If so, the lower cost is assigned to the station pair. When all station pairs have been evaluated the algorithm is complete.

6.1.20.5 PDL

See Appendix A.

6.1.20.6 Decision Tables and Algorithms

None.

6.1.21 EINDPP - Network Data Preprocessing

6.1.21.1 Identification

- o EINDPP - Network Data Preprocessing
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.1.21.2 Argument Dictionary

None

6.1.21.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
BD	-	L*1	Indicates a board event was found
CASE	-	I*2	Selection parameter for CASE statement
DBD	-	L*1	Indicates a deboard event was found
ENODE	-	I*2	End node of a priority link
LINK	-	I*2	A priority link
SLINK	6	L*1	Indicates which of the first six station link types have been included when tracing the vehicle path through a station
SNODE	-	I*2	Beginning node of a priority link
TCNVRT	-	R*4	Time conversion, seconds to clock units

6.1.21.4 Description

This routine converts the network definition input data from the user's nomenclature to the simulator's nomenclature, checks it for errors,

and builds network characteristic and connectivity tables that are independent of least cost path definitions. The following major steps are performed:

1. Establish mapping between the user's node labels and the simulator's node labels.
2. Check the network data for consistency, including:
 - a. A node label must be within an allowable range.
 - b. A node should not appear as an origin more than twice.
 - c. A node should not appear as a destination more than twice.
 - d. A station should not appear as an origin more than once.
 - e. A station should not appear as a destination more than once.
 - f. A node must appear as both an origin and a destination.
 - g. The number of destination nodes must equal the number of origin nodes.
 - h. A station must not exit to a guideway merge or diverge.
 - i. Reasonableness checks on link length and speed.
3. Calculate network size parameters including:
 - a. Number of nodes
 - b. Number of stations
 - c. Number of merges
4. Establish input for least cost path routine, including:
 - a. Define origin and destination of each link.
 - b. Make link length an integer number of blocks if fixed block regulation is in effect.
 - c. Assign default speed to links where speed not specified by user.
 - d. Calculate travel time on each link, using station travel time if link corresponds to an online station.

5. Establish network connectivity tables, including:
 - a. Link to merge
 - b. Link to diverge
 - c. Link to station
 - d. Merge to link
 - e. Link to link, including prioritization, if any
 - f. Link to node
 - g. Competing links at merges
6. Calculate link attributes, including:
 - a. Length
 - b. Nominal travel time
 - c. Capacity, using total station capacity if link corresponds to an online station.

6.1.21.5 PDL

See Appendix A.

6.1.21.6 Decision Tables and Algorithms

The capacity of each guideway link is calculated using the following equation when variable block regulation has been specified.

$$C = 1 + (L_g - V \times H) / L_v$$

where

C = capacity in vehicles

1 = capacity of headway zone

L_g = length of guideway link

V = nominal speed on guideway link

H = vehicle headway (time to travel the headway zone)

L_v = vehicle length

VxH = length of headway zone

For fixed block regulation, the capacity is equal to the number of control blocks on the link.

If the link corresponds to an online station, the capacity is equal to the sum of the capacity of each active station link in the station.

6.1.22 EINERR - Error Messages

6.1.22.1 Identification

- o EINERR - Error Messages
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.1.22.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
MSG	-	I*4	Message number
SEVR	-	I*4	Severity code 1 = information 2 = warning 3 = severe, terminate <0 = severe, continue

6.1.22.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
CASE	-	I*4	Internal message number within a particular group of messages
FOUND	-	L*1	Error message text is defined
GROUP	-	I*4	Message group

6.1.22.4 Description

This routine requests the output of messages when anomalous conditions are detected. The text for all messages that can be generated by the Input Processor is contained in this routine to facilitate review and update. A routine detecting an anomalous condition will call EINERR, giving the message number and the severity. If the severity is negative, EINERR sets a logical variable indicating that a serious error has occurred and then sets the severity to two (warning). This permits the calling routine to continue checking for additional errors and terminate due to the serious error when it cannot proceed. EINERR then calls routine EIERROR, giving the message number, text and severity code for the message.

6.1.22.5 PDL

See Appendix A.

6.1.22.6 Decision Tables and Algorithms

None.

6.1.23 EINFMT - Network Data Formatting

6.1.23.1 Identification

- o EINFMT - Network Data Formatting
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.1.23.2 Argument Dictionary

None.

6.1.23.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
BYLINK	-	I*4	Station bypass link ID
DNODE	-	I*4	Destination node ID
DSTA	-	I*4	Destination station ID
LINK	-	I*4	Guideway link ID
MIN	-	R*4	Minimum value
NODL	-	I*4	Node ID
ONODE	-	I*4	Origin node ID
OSTA	-	i*4	Origin station ID
STATN	-	I*4	Station ID
TCNVRT	-	R*4	Time conversion, seconds to clock units
TIME	-	R*4	Station to station travel time

6.1.23.4 Description

This routine builds the network connectivity tables that are based on least cost paths and establishes station connectivity to the guideway

network. The major processing steps are:

1. Routine EINSLT is called to build the initial successor link table, which is the first plane of a three dimensional array wherein element i, j defines the next link to travel on from link i to reach station j on a least cost path.
2. The station to station travel time matrix is built by tracing the path between each pair of stations through the successor link table and summing the link travel time.
3. The closest downstream station to each station is found by scanning each row of the station-to-station travel time table and finding the column (corresponding to station ID) that contains the smallest non-zero value.
4. The successor station table is built, which is a two dimensional array wherein element i, j defines the next station on the least cost path from station i to station j . The following steps are performed for each station:
 - a. The station ID defines the row of the successor station table that will be built.
 - b. Steps c and d are performed for each station.
 - c. If the origin station and destination station are the same, a zero is placed in the table at the position defined by the station (position is on the diagonal).
 - d. If the origin station and destination station are not the same, then beginning at the link which leaves the origin station, the successor link table is used to step through the least cost path link sequence. Each link is checked to determine if it ends in a station. If it does, the ID of that station is placed in the table at the position defined by the origin (row) and destination (column) stations.
5. A list is formed by mapping the link to station connectivity list to identify the upstream link from each station.
6. A list is formed based on the link to station connectivity list and the successor link table which identifies the downstream link from each station.

6.1.23.5 PDL

See Appendix A.

6.1.23.6 Decision Tables and Algorithms

None .

6.1.24 EINPUT - Input Processor Control

6.1.24.1 Identification

- o EINPUT - Input Processor Control
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.1.24.2 Argument Dictionary

None

6.1.24.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ABLNK	-	I*4	Four space characters
AEND	-	I*4	The characters END left justified
A3	-	L*1	Indicates that empty vehicle management option 3 was selected
A4	-	L*1	Indicates that empty vehicle management option 4 was selected
DAY	-	I*2	Current day of month
DVAR1	-	I*2	Internal variable for reading number of group two station pairs or end of file indicator
DVAR2	-	I*2	Internal variable for reading number of group three station pairs
HOUR	-	I*2	Current hour
MIN	-	I*2	Current minute
MON	-	I*2	Current month

VARIABLE	DIM	TYPE	DESCRIPTION
SUM	-	R*4	Sum of elements in a group size probability distribution
YEAR	-	I*2	Current Year

6.1.24.4 Description

EINPUT is the Input Processor main program. The first action is to call routine SPIEL, which establishes linkage to the simulator's program interrupt handling routines from the operating system. Then, if a system interrupt occurs during input processor execution, control will be given to the traceback facility which will identify the type of problem and list the current hierarchy of subroutines called with the contents of the general purpose registers at each call and any arguments supplied in each call.

Routine DAYTIM is called to obtain the current date and time, which is printed with the DESM Input Processor heading on the system output device. Routine EIINIT is called to initialize all of the parameters in COMMON. EINPUT then reads the system characteristics input and description data and the run time data that is defined for time equal zero.

Routine EINTWK is called to either read and process new network definition data or read a previously generated network structured data file. Next, if the Input Processor is to plan the transit service or if trip arrivals are to be generated, EINPUT reads the demand generation input and description data and calls routine EITNIT to generate the demand generation initial conditions. If trip arrivals are to be generated, routines EITRIP or EIUDGN, and EIDPPT are called to generate the trips and prepare a summary report.

Multiple sets of demand generation input and description data can be read and processed. EINPUT maintains synchronization between the time associated with the run time data, the time at the end of a trip demand interval, the number of the trip demand interval and the basis for defining the transit service.

If the user is defining the level of service or the Input Processor is to plan the level of service and the specified set of demand generation initial conditions exists in memory, the system characteristics processing routines are invoked. The first of these is the station configurator, EISCFG, which is called if the user has defined the station characteristics in the station configurator format. Routine EICHCK is then called to perform reasonableness checks and conversions on the system characteristics

input and description data. Following this, routine EISERV is called, which controls the generation of transit service initial conditions.

After system characteristics processing is performed, the structured data is written and summary reports are prepared. If the processing was in response to an active fleet size management request in the run time file, EINPUT writes the updated transit service data to the structured run time file and calls routine EIARPT to generate a summary report. If the processing was for the initial level of service, entry EISWRT in routine EIBWRT is called to write the structured system characteristics file and routine EISRPT is called to generate a summary report.

EINPUT continues reading the run time input and description data and performing the processing required by each set of data that has the same value of time. Any input not in time sequence causes an error message and termination. After all input data has been processed, entry EIWNAM in routine EIMNAM is called to update the run index file and EINPUT terminates.

6.1.24.5 PDL

See Appendix A

6.1.24.6 Decision Tables and Algorithms

None

6.1.25 EINRPT - Network Summary Report

6.1.25.1 Identification

- o EINRPT - Network Summary Report
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.1.25.2 Argument Dictionary

None

6.1.25.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
C1X	-	I*4	Index for printing column 1 data
C2X	-	I*4	Index for printing column 2 data
HDWY	-	R*4	Link Headway in seconds
LINECT	-	I*4	Current number of lines printed on a page
LINK	-	I*4	Link index
NLINKA	-	I*4	ID of next link, column 1
NLINKB	-	I*4	ID of next link column 2
PAGE	-	I*4	Current page number
PAGLIM	-	I*4	Lines per page limit
STAIX	-	I*4	Station index
TCNVRT	-	R*4	Time conversion, seconds to clock units
TIMEA	-	R*4	Travel time on link NLINKA
TIMEB	-	R*4	Travel time on link NLINKB

VARIABLE	DIM	TYPE	DESCRIPTION
TTIM	-	R*4	Link travel time in seconds
TYPE	2	I*4	Station Type: 1, ' ON' 2, ' OFF'
VEL	-	R*4	Link speed in meters/second

6.1.25.4 Description

This routine formats and outputs a summary of the network characteristics when network input and description data is processed. The report includes:

- o Network Configuration data for each link
 - link identification
 - node at beginning of link
 - node at end of link
 - if a station is located on the link
 - station identification
 - online or offline
 - link upstream from station
 - link downstream from station
 - if link ends at a diverge
 - diverge identification
 - identification of links leaving diverge
 - if link begins with a merge
 - merge identification
 - identification of links or link and station leading to merge
- o Characteristics data for each link
 - link identification

- length
- capacity
- nominal travel time
- headway
- line speed
- o Station summary for each station
 - station identification
 - closest downstream station
 - for each destination station
 - station identification
 - nominal travel time (excluding dwell time)
 - next station on least cost path
- o Successor link summary for each link
 - link identification
 - for each destination station
 - station identification
 - next link to travel on the least cost path to the station
 - nominal travel time on that link

6.1.25.5 - PDL

See Appendix A

6.1.25.6 Decision Tables and Algorithms

None

6.1.26 EINSLT - Generate Successor Link Table

6.1.26.1 Identification

- o EINSLT - Generate Successor Link Table
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.1.26.2 Argument Dictionary

None.

6.1.26.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
COST1	-	I*4	Cost from a diverge to a station using the first link that leaves the diverge
DLEG1	-	I*4	Node at end of first link that leaves a diverge
DNODE	-	I*4	Node at end of a link
LEG1	-	I*4	First link that leaves a diverge
LINK	-	I*4	Link identification
NODL	-	I*4	Node identification
STATN	-	I*4	Station identification
TCNVRT	-	R*4	Time conversion, seconds to clock units

6.1.26.4 Description

This routine builds the successor link table, which is a three dimensional array wherein element i, j of a given plane defines the next link to travel on from link i to reach station j on a least cost path. The first plane is built based on the nominal network configuration. Successive

planes are added as network failures are inserted or removed. The following steps are performed for each station:

1. The station ID defines the column of the successor link table that will be built
2. Steps 3 through 5 are performed for each network link
3. If the station is at the end of this link, a zero is inserted in the table at the position defined by link ID and station ID
4. If the link does not end at a diverge, the ID of the next downstream link is placed in the table at the position defined by current link ID and station ID.
5. If the link ends in a diverge, the next link is determined by selecting the first link that leaves the diverge and computing sum of that link's cost and the cost of travel from the end of that link to the major node upstream from the station on the least cost path. If the sum is equal to the cost of the least cost path from the beginning of that link to the major node upstream from the station, then the first link is the correct choice. Otherwise, the second link to leave the diverge is the correct choice. The ID of the correct link is placed in the table at the position defined by current link ID and station ID.

6.1.26.5 PDL

See Appendix A.

6.1.26.6 Decision Tables and Algorithms

None.

6.1.27 EINTWK - Network Processing Control

6.1.27.1 Identification

- o EINTWK - Network Processing Control
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.1.27.2 Argument Dictionary

None.

6.1.27.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
INDEST	4	I*4	Destination nodes of up to four links on one input record
INLLEN	4	I*4	Lengths of up to four links on one input record
INODE	4	I*4	Origin nodes of up to four links on one input record
INTYPE	4	I*4	Types of up to four links on one input record = 0, no station on link = 1, station on link

6.1.27.4 Description

This routine controls the processing of guideway network configuration data, including: reading the input and description or structured data, generation of least cost paths, failure/recovery occurrences, alternate paths, writing the structured data file and preparing the network initial conditions report. The following major steps are performed:

1. If network input and description data is to be processed:
 - o the network input and description file is read

- o routine EINDPP is called to check the input data for reasonableness and consistency and generate network configuration tables that are not based on least cost paths
 - o routine EIMPTH is called to determine the least cost paths in the network
 - o routine EINFMT is called to generate the network configuration tables that are based on least cost paths
 - o routine EINRPT is called to format and output a summary report of the network
2. If a previously processed network is to be used and the data has not been read, entry EINRD in routine EIBWRT is called to read the network structured data file.
 3. If the user has defined alternate paths for the network, routine EIALTP is called to process the alternate path data.
 4. If step 1 and/or step 3 was performed, entry EINWRT in routine EIBWRT is called to write the network structured data file.
 5. If the user entered a failure/recovery request, routine EIFAIL is called to process the request. EINTWK then writes the failure/recovery data to the structured run time file.

6.1.27.5 PDL

See Appendix A.

6.1.27.6 Decision Tables and Algorithms

None.

6.1.28 EIPARM - PARM Field Addressability

6.1.28.1 Identification

- o EIPARM - PARM Field Addressability
- o IBM/FSD - May 1, 1978
- o ASM

6.1.28.2 Argument Dictionary

ENTRY	VARIABLE	DIM	TYPE	DESCRIPTION
EIPARM	PARMAD	-	I*4	Address of parameter list from EXEC statement
EIPLST	None			

6.1.28.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ARG1	-	I*2	Length of character string in PARM field of EXEC statement
ARG2	-	L*1	Character string in PARM field of EXEC statement

6.1.28.4 Description

This routine obtains the location of the list of data set member names specified by the user in the PARM field of the EXEC statement from the operating system and invokes processing to decode the individual member names from the input character string.

EIPARM is the first routine to receive control when the Input Processor is executed. It saves the address of the parameter list and gives control to the Input Processor main routine, EINPUT.

EIPLST is called by EIINIT when parameter list decoding is required. EIPLST calls EIMNAM, passing it the length of the character string and the character string. EIMNAM scans the character string and divides it into individual member names.

6.1.28.5 PDL

See Appendix A.

6.1.28.6 Decision Tables and Algorithms

None.

6.1.19 EIPSAV - COMMON Area Addressability

6.1.29.1 Identification

- o EIPSAV - COMMON Area Addressability
- o IBM/FSD - May 1, 1978
- o ASM

6.1.29.2 Argument Dictionary

None.

6.1.29.3 Local Variable Dictionary

None.

6.1.29.4 Description

EIPSAV saves in COMMON EIPSAV the starting addresses of the Input Processor COMMONs, the network structured data COMMONs and the system characteristics structured data COMMONs and the ending address of the COMMONs. These addresses are generated at link edit time by the following overlay structure:

OVERLAY REGION1

INSERT BEGCOM Start of IP COMMONs

OVERLAY REGION1A

Input Processor COMMONs

- o
- o
- o

OVERLAY REGION2

INSERT IPNET Start of Network COMMONs

OVERLAY REGION2A

Network Structured Data COMMONs

o

o

OVERLAY REGION3

INSERT IPSYS Start of System Characteristics COMMONs

OVERLAY REGION3A

System Characteristics Structured Data COMMONs

o

o

o

OVERLAY REGION4

INSERT ENDCOM End of COMMON Area

The Linkage Editor generates the addresses (BEGCOM, IPNET, IPSYS, ENDCOM) surrounding the three sets of COMMONs. COMMON EIPSAV defines storage locations for the addresses. Entry point LODCOM provides an external reference point to COMMON EIPSAV. When called by EIINIT, the COMMON is loaded into core so that the addresses can be accessed.

6.1.29.5 PDL

See Appendix A.

6.1.29.6 Decision Tables and Algorithms

None.

6.1.30 EIRNG - Random Number Generator

6.1.30.1 Identification

- o EIRNG - Random Number Generator
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.1.30.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
MRSEED	-	I*4	Random number seed (input and output)
MRANDN	-	R*4	Random number between zero and one (output)

6.1.30.3 Local Variable Dictionary

None

6.1.30.4 Description

This routine generates a random number that is uniformly distributed between zero and one.

6.1.30.5 PDL

See Appendix A.

6.1.30.6 Decision Tables and Algorithms

None.

6.1.31 EIRSEL - Sample From Probability Distribution

6.1.31.1 Identification

- o EIRSEL - Select Point At Random From Cumulative Probability Distribution
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.1.31.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DDISTR	DENC	R*4	A cumulative probability distribution
DSTRT	-	I*2	Index of first element in DDISTR
DEND	-	I*2	Index of last element in DDISTR
DSEED	-	I*4	Random number seed (input and output)
DSLECT	-	I*2	Index of selected element in DDISTR (output)

6.1.31.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DRANDN	-	R*4	Random number returned by EIRNG

6.1.31.4 Description

This routine randomly selects a point from a cumulative probability distribution. Routine EIRNG is called to generate a random number between zero and one. EIRSEL then scans the cumulative distribution until a value larger than the random number is found. The index of that value is returned to the calling routine.

6.1.31.5 PDL

See Appendix A.

6.1.31.6 Decision Tables and Algorithms

None.

6.1.32 EISCFG - Station Configurator

6.1.32.1 Identification

- o EISCFG - Station Configurator
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.1.32.3 Argument Dictionary

None.

6.1.32.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
Pointers to Fields in Input Array SLCFIG (13, KNSL)			
LT	-	I*2	Link type field (1)
TT	-	I*2	Link travel time field (2)
LL	-	I*2	Link length field (3)
CAP	-	I*2	Link capacity (VEH) field (4)
EV1	-	I*2	Event 1 field (5)
EV2	-	I*2	Event 2 field (6) (not used)
EV3	-	I*2	Event 3 field (7) (not used)
EV4	-	I*2	Event 4 field (8) (not used)
EV5	-	I*2	Event 5 field (9) (not used)
FN	-	I*2	Diverge function field (10)
ODR	-	I*2	Upstream link ordering field (11)
HT	-	I*2	Headway field (12)
ET	-	I*2	Headway/event field (13)
Event Types			
HEVENT	-	I*2	Headway event type (1) (not used)
TEVENT	-	I*2	Travel event type (2) (not used)
DEVENT	-	I*2	Deboard event type (3)
BEVENT	-	I*2	Board event type (4)
SEVENT	-	I*2	Storage event type (5)
LEVENT	-	I*2	Launch event type (6)

Link Types

IR	-	I*2	Input Ramp (1) (not used)
IQ	-	I*2	Input Queue (2) (not used)
DK	-	I*2	Dock (3) (not used)
OQ	-	I*2	Output Queue (4) (not used)
OR	-	I*2	Output Ramp (5) (not used)
ST	-	I*2	Storage (6)
IS	-	I*2	Input to Storage (7) (not used)
SI	-	I*2	Storage to Input (8) (not used)
DS	-	I*2	Dock to Storage (9) (not used)
SO	-	I*2	Storage to output (10) (not used)

IDs of Certain Link Types Found

DBFLK	-	I*2	ID of board dock
DDBFLK	-	I*2	ID of deboard dock
DSFLK	-	I*2	ID of dock to storage link
IQFLK	-	I*2	ID of input queue link
IRFLK	-	I*2	ID of input ramp link
ISFLK	-	I*2	ID of input to storage link
OQFLK	-	I*2	ID of output queue link
ORFLK	-	I*2	ID of output ramp link
STFLK	-	I*2	ID of storage link
SIFLK	-	I*2	ID of storage to input link
SOFLK	-	I*2	ID of storage to output link

Count of Certain Link Types

IQCNT	-	I*2	Number of input queue links
DDPCNT	-	I*2	Number of deboard docks
DBCNT	-	I*2	Number of board docks
OQCNT	-	I*2	Number of output queue links

Other Variables

BD	-	L*1	Indicates that a board event was found
CCT	-	I*4	Index to list of lists arrays
DBD	-	L*1	Indicates that a deboard event was found
LAUNCH	-	I*2	Type of link that has launch event
LINK	-	I*4	Link number
SCANL	-	I*2	Index for scanning lists
SEQ	-	I*2	Used in upstream link ordering
USLOR	-	I*2	ID of link that has launch event
ZERO	-	I*2	Zero

6.1.32.4 Description

This routine builds the station link configuration tables from user defined parameters specifying the attributes of each station link. It is called if the user elects to use the station configurator capability. Each station link definition prepared by the user consists of parameters describing the station link in terms of the following attributes:

<u>Link Definition Attributes</u>	<u>Description</u>
Link Type	Numeric code for link type
Link Travel Time	Time in seconds to travel link
Link Length	Length of link in meters to compute a travel time based on station link velocity
Link Capacity	Capacity in vehicles per link
Link Events (five maximum)	Ordered events to occur on link
Link Diverge Function	Numeric code to select desired diverge function for link
Link Order for Dequeue	Numeric code to select order of upstream links for dequeue
Headway Zone Term B	Headway zone travel time constant term
Headway Zone Term A	Headway zone travel time per vehicle

The SLCFIG station link parameter values are used to build the structured data required by the DESM-MP. The processing performed and the resultant structured data prepared in the process is described in the following steps.

1. Establish the number of station links entered and define the run time number of station links in KNSL.
2. For each link travel time equal to 0, compute travel time from link length value / station velocity. When both travel time and link length are zero, then travel time is set to zero. All time values, whether given or computed, are converted into corresponding clock unit values when building the travel time table SLTTIM.

3. For each link, the capacity parameter value is used to build the capacity table SLCAP if the user did not explicitly define a capacity for that link.
4. For each link, headway times are used to build the headway zone travel time tables SLHTA and SLHTB.
5. For each link, the link type value is used to build the link type table SLTYPE.
6. For each link, the link event sequence is used to build the event sublist table SLEVL and a sublist pointer table SLEVP for each sublist in the sublist table.
7. For each link, the link diverge function value is used to build the diverge function table SLDIVC.
8. For each link, the routine computes and builds the system pointer value SLUSP and upstream link sublist SLUSL. The link IDs are ordered by the selection value given in the ORDER parameter. Possible upstream links are defined in subsection 6.1.32.6.
9. For each link, the routine computes and builds the downstream pointer and downstream link sublist SLDSP and SLDSL, respectively. Possible downstream links are defined in subsection 6.1.32.6.

6.1.32.5 PDL

See Appendix A.

6.1.32.6 Decision Tables and Algorithms

Options for ordering of upstream links:

Option

- 1 Storage link has lowest priority
- 2 Storage link has highest priority

Possible station link connectivity:

Upstream Links	Link Type	Downstream Links
None	IR	IS, (IQ or DOCK)
SI, IR, or None	IQ	DOCK (D) or DOCK (D/B)
IQ, or (IR, SI) or None	DOCK (D) only	DOCK (B)
DOCK (D), SO	DOCK (B) only	DS, (OQ or OR), or None
(IQ, SO) or (IR, SI, SO) or		
None	DOCK (D/B)	DS, (OQ or OR), or None
(Dock (B) or Dock (D/B)), SO	OQ	OR or None
SO, (OQ or Dock)	OR	None
DS, IS	ST	SI, SO
IR	IS	ST
ST	SI	IQ or DOCK
Dock (B) or Dock (D/B)	DS	ST
ST	SO	DOCK, or OQ or OR

Mnemonic-Definition

IR -- Input Ramp	OQ -- Output Queue	DS -- Dock-to-Storage
IQ -- Input Queue	OR -- Output Ramp	SO -- Storage-to-output
Dock (D) -- Deboard	ST -- Storage	
Dock (B) -- Board	IS -- Input-to-Storage	
Dock (D/B) -- Deboard/ Board	SI -- Storage-to-Input	

6.1.33 EISCHD - Scheduled Service Planning

6.1.33.1 Identification

- o EISCHD - Scheduled Service Planning
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.1.33.2 Argument Dictionary

None.

6.1.33.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DBBT	-	R*4	Time required to deboard and board
DIFF	-	I*4	Route indicator minus candidate indicator
FOUND	-	L*1	1) Indicates candidate route matches existing route 2) Indicates route is a member of a route group
GPSIZE	KMRT	I*2	Numbers of routes in route groups (equivalenced to PVSLST (KMRT + 1))
GROUP	-	I*4	Route group number
HDWY	KMR	R*4	Route headway time
IDEST	-	I*4	Intermediate destination
INL	-	I*4	Index to internally defined list
IORIG	-	I*4	Intermediate origin
IX	-	I*4	Position of origin in route list
LAUNCH	-	I*4	Time of next vehicle launch
LOOP	-	L*1	Indicates route is looping without reaching destination

VARIABLE	DIM	TYPE	DESCRIPTION
MATCH	-	L*1	Indicates stop on a candidate route matches stop on an existing route
MNHDWY	-	I*4	Minimum vehicle headway on routes
MXFLOW	-	I*4	Maximum trip flow on a route
MXHDWY	-	I*4	Maximum vehicle headway on routes
MXTRL	KMS	I*2	Maximum train length to enter each station (equivalenced to PVSLST (1))
PERIOD	-	I*4	Time to travel completely around a route
PVSLST	2 x KMRT	I*2	Station sequences on routes
PWSTN2	-	I*4	Reboard station
RTE	-	I*4	Route number
RTEND	-	I*4	Index to end of route list
RTEST	-	I*4	Index to start of route list
RTFLOW	KMRT	I*2	Station to station trip flow (equivalenced to PVSLST (1))
STATN	-	I*4	Station ID
TCNVRT	-	R*4	Time conversion, seconds to clock units
TIME	-	R*4	Station to station travel time
TIMUPS	-	I*4	Travel time from upstream station
TRL	-	I*2	Train length on route
TVLTIM	KMRT	I*4	Time to travel from first station on route to other stations on route
USR	-	I*4	Index to user-defined list
VDRTOT	-	I*2	Total vehicles dispatched on route
XFERT	-	R*4	Time required to transfer

6.1.33.4 Description

This routine analyzes a network configuration and defines a unique set of routes to serve all trips, estimates the number of vehicles required

on each route and generates time schedules for their departure from each station. The routine permits the user to define the routes and/or level of service or will perform these functions if the user requests them. The major processing steps are:

1. If the IP is to generate routes, steps 2 through 5 are performed for each origin/destination pair. Otherwise processing is begun at step 7.
2. A list is formed of all the stations on the least cost path from origin to destination and back to origin.
3. The list of stations is compared to a table of unique lists of stations defined in prior iterations to determine if a new route has been found.
4. If a new route has been found, the following steps are performed:
 - a. the list of stations is added to the table of unique lists of stations and the origin and destination stations are flagged with the value three
 - b. the number of routes is incremented
 - c. the station sequence is concatenated to the list of lists of station sequences on routes and the index of the first entry is added to the pointer list
 - d. the route number is inserted in the trip to route assignment matrix for the current origin/destination pair
 - e. the process continues at step 2 until all origin/destination pairs have been processed, then proceeds to step 6
5. If the route already exists in the table, the following steps are performed:
 - a. the origin and destination stations are flagged with the value three in the row of the table where the route exists to indicate they also generate the route and the route number is inserted in the trip to route assignment matrix for the current origin/destination pair
 - b. the process continues at step 2 until all origin/destination pairs have been processed, then proceeds to step 6

6. For each route, the stations that did not generate the route (indicated by the absence of a three in the route table) are then deleted from the concatenated list of station sequences on routes to form the concatenated list of station stops on routes. The route list pointer is revised to account for the deletions. The process then continues at step 9.
7. The user's list of lists of station stops on each route is scanned, searching for the value zero which is used to separate the lists. A set of pointers to the first station in each list is built and the zero values are deleted from the list of lists. The number of routes, length of the route list and the length of the set of pointers are computed.
8. If the user defined route groups, a set of pointers to the first route in each group is built.
9. The nominal travel time between every pair of stations is found by tracing a trip's path along the assigned route (s) including any transfers that are defined.
10. If the IP is to establish the level of service, steps 11 and 12 are performed for each route. Otherwise, the routine proceeds to step 13.
11. The average passenger demand is mapped onto pseudo links connecting, the stations on the route, considering transfers off the route, transfers onto the route and trips served by more than one route.
12. The maximum passenger flow on the route is found and used to calculate the nominal headway between trains on the route. The nominal headway is subject to a maximum wait time constraint. If no demand exists for the route, one train is placed on the route.
13. Steps 14 through 17 are performed for each route.
14. The time required to travel completely around the route, including dwell time in stations is calculated.
15. If the nominal headway is defined, the number of vehicles required is computed by dividing the period by the headway. The headway is then adjusted to account for an integer number of trains on a route. If the number of vehicles was given, the headway is calculated by dividing the period by the number of trains.

16. The total number of vehicles required is updated.
17. The departure time of the first vehicle to leave each station and the number of vehicles to be dispatched from each station is determined to achieve uniform time spacing of vehicles around the route.
18. If an active fleet size change is being scheduled, the closest maintenance barns for each route are calculated, if they have not been found previously. A barn on the route is chosen if an available storage to dock link exists. Otherwise, the barn closest after any station stop on the route is selected. Then the closest station stop after the selected barn is found for each route.
19. The station stop list is scanned and the largest train to stop at each station is determined. The station link capacities are checked to insure that they are sufficiently large to accommodate the largest train.

6.1.33.5 PDL

See Appendix A.

6.1.33.6 Decision Tables and Algorithms

When generating routes to serve a network, the following arrays are used:

Candidate Route List

PRLIST (j) = 0, station j is not on the route
 = 1, station j is on the route

Route Table

PRTABL (i, j) = 0, station j is not on route i
 = 1, station j is on route i
 = 3, station j is a generator of route i

Then, for the candidate route to be unique, the sequence of zeros and ones must not match the sequence of zeros and non-zeros in any row of PRTABL. If it is unique, PRLIST is added as a new row of PRTABL and the columns of that row corresponding to the origin and destination stations that generated PRLIT are set to three. Stations flagged with the value three will become stops on the route. Other threes may be inserted later as candidate route lists are compared to the route table and found to match an existing row in the table. The generator stations of these candidate route lists receive threes in the proper row of the route table.

When estimating the number of vehicles required on a route, the array RTFLOW is used to accumulate the total trip demand between adjacent station stops on the route. Each element in the array corresponds to a pseudo link connecting the station defined in the same element of the route list PVRLST to the next station stop on the route. The demand between any pair of stations on the route is added to each of the pseudo links that are used by the trips. Therefore, if demand exists from station A to station D on a route that has intermediate stops at stations B and C, the pseudo links corresponding to stations A, B and C will each be assigned the trip demand value. The maximum value over all pseudo links is maintained as the trip assignment is performed. Transferring trips affect only the pseudo links that they use while on the segment of their trip that uses the current route. Trips that are served by a group of routes that includes the current route are assumed to be distributed evenly over each route in the group.

The vehicle headway on the route is computed as follows:

$$H = \frac{VCAP \times L \times TLEN \times 3600}{S_N \times T_{MAX} \times A} \quad (1)$$

where:

$$S_N = S_U \times T_D / T_U \quad (2)$$

$$A = \left(\sum_{i=1}^3 N_i \times M_i \right) / \sum_{i=1}^3 N_i \quad (3)$$

VCAP = vehicle capacity

L = estimated achievable load factor

TLEN = train length on route

S_n = net scale factor

S_u = user-defined scale factor applicable to current demand profile interval

T_D = original time base associated with current demand matrix

T_U = user-defined override to the time base applicable to current demand profile interval

A = average trip group size

M_i = mean of group size distribution i

N_i = number of trips using group size distribution i

T_{MAX} = maximum trip flow over all route pseudo links

Generation of departure schedules for each station and the number of vehicles to be launched from each station on a route are computed as follows. The first station in the route list for a given route is called the home station. The time to launch the first vehicle from the home station is equal to the remainder when dividing the route period by the route headway. The number of vehicles to be launched from the home station is the largest value n such that

$$T_L = P \text{ modulo } H \quad (4)$$

$$T_L + \left(\frac{n}{T_{LEN}} - 1\right) H < T_U \quad (5)$$

where:

T_L = launch time of first vehicle

P = route period

H = route headway

T_U = travel time from upstream station

The remaining stations on the route have the initial launch and number of vehicles calculated from:

$$T_L = T_H \text{ modulo } H \quad (6)$$

$$T_L + \left(\frac{n}{T_{LEN}} - 1\right) H < T_U \quad (7)$$

where

T_H = travel time from home station to current station

This scheduling method results in a uniform time spacing on the route when all of the vehicles have been launched.

6.1.34 EISERV - Service Planning Control

6.1.34.1 Identification

- o EISERV - Service Planning Control
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.1.34.2 Argument Dictionary

None.

6.1.34.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
CASENO	-	I*4	Case selection parameter
CVRSN	-	R*4	Time conversion, seconds to clock units
DEST	-	I*4	Destination station index
EVP	-	I*4	Index to station link event list
FOUND	-	L*1	Indicates vehicle path through station completed
ORIG	-	I*4	Origin Station index
PTR	-	I*4	Index to list of lists
SL	-	I*4	Station link number
SSPTIM	KMS, KMS	I*2	Station to station travel time. Equivalenced to DTRPTB.

6.1.34.4 Description

This routine controls the generation of transit system service, including user-defined and processor-defined levels of service in either

demand responsive or scheduled modes of operation. The vehicle dwell time in a station is calculated by tracing a vehicle's path through the station and summing the time required for each event on each link. The nominal travel time between every pair of stations is calculated by adding the dwell time to the station to station time calculated by the network processing routines.

EISERV then invokes the proper routine for the type of service selected by the user. If demand responsive (single party or multiparty) service is selected, routine EIDRSP is called. If scheduled service is selected, routine EISCHD is called. If timeout/group demand responsive is selected, routine EIMORG is called unless an active fleet size change is being planned.

If the user requested a nominal travel time file to be written, the nominal travel time matrix is converted from clock units to seconds and written to the file.

6.1.34.5 PDL

See Appendix A.

6.1.34.6 Decision Tables and Algorithms

None.

6.1.35 EISRPT - System Characteristics Report Generator

6.1.35.1 Identification

- o EISRPT - System Characteristics Report Generator
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.1.35.2 Argument Dictionary

None.

6.1.35.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ALLROW	-	L*1	Indicates that all rows of a set of columns of a matrix have been printed
CASE	-	I*4	Case selection parameter
COL1	-	I*2	1) First column of a matrix to be printed on a page 2) Number of standees on a vehicle 3) Index for printing column 1 data
COL2	-	I*2	1) Last column of a matrix to be printed on a page 2) Index for printing column 2 data
DONE	-	L*1	Indicates that all elements of an array have been printed
DPTR	-	I*2	Index to station link downstream link list
EPTR	-	I*2	Index to station link event list
GROUP	-	L*1	Indicates that route groups are used
FOUND	-	L*1	Flag indicating dock type printed
FOUND2	-	L*1	Flag indicating destination printed

VARIABLE	DIM	TYPE	DESCRIPTION
GRPID	-	I*2	Route group number
LIFLAG	-	L*1	Flag indicating station ID printed
LINK1	-	I*2	Priority link at a merge
LINK2	-	I*2	Subordinate link at a merge
LINREM	-	I*4	Number of lines remaining to be printed on a page
LJFLAG	-	L*1	Flag indicating platform ID printed
LTYPE	10	I*2	Alphabetic notation for station link type: 1) IR - input ramp 2) IQ - input queue 3) D - dock 4) OQ - output queue 5) OR - output ramp 6) S - storage 7) IS - input to storage 8) SI - storage to input 9) DS - dock to storage 10) SO - storage to output
NBD	-	I*2	Capacity of dock area
NIQ	-	I*2	Number of input queues
NOQ	-	I*2	Number of output queues
NPD	-	I*2	Number of parallel docks
OUT1	KMS	I*2	Node identification of stations
OUT2	KMM	I*2	Node identification of merges
OUT3	KMDLY1	R*4	Density values corresponding to rows of merge delay table

VARIABLE	DIM	TYPE	DESCRIPTION
OUT4	KMDLY2	R*4	Density values corresponding to columns of merge delay table
OUT5	-	R*4	Internal variable for output of floating point values
PAGE	-	I*4	Current page number
PAGLIM	-	I*4	Lines per page limit
PLATID	4	I*2	Platform type
SLASH	4	L*1	Slash characters for field separation in output
START	-	I*4	Index of first element to be printed on a page
STOP	-	I*4	Index of last element to be printed on a page
TCUSEC	-	R*4	Time conversion, clock units to seconds
TIM1	-	R*4	Nominal travel time value for column 1
TIM2	-	R*4	Nominal travel time value for column 2
TYPEID	-	I*4	Station link dock type
UPTR	-	I*2	Index to station link upstream link list

6.1.35.4 Description

This routine formats and outputs a summary of the system characteristics, transit service characteristics and simulation control parameters established from the user's inputs. The routine outputs only the information that is applicable to the current simulation scenario. The first part of the report summarizes the system characteristics, listing the selected options for longitudinal control, position regulation, dispatch, merge reservation, merge control, empty vehicle management, entrainment, and path selection. Supporting data, if any, is listed with the options. Following this, the vehicle characteristics and station characteristics are summarized.

The second part of the report summarizes the initial level of service, identifying whether the level of service was specified by the user or estimated by the Input Processor and indicating the type of service. If the type is demand responsive, the following information is listed:

- o "Single Party" or "Multiparty"
- o Fleet size
- o Initial placement of vehicles
- o Transfer option
- o Vehicle reservation option
- o Vehicle diversion option
- o Nominal travel time matrix

If the type is scheduled, the following information is listed:

- o User defined/simulation defined routes
- o Fleet size
- o Vehicle spacing option
- o Transfer option
- o Demand stop option
- o Route data (for each route)
 - Station stops
 - Nominal departure times for initial dispatch
 - Number of vehicles to be dispatched from each station
 - Vehicle headway
- o Route assignment matrix
- o Route group data
- o Nominal travel time matrix

If the type is timeout/group demand responsive, the following information is listed:

- o Vehicle request parameters
- o Station platform summary
- o Nominal travel time matrix

Following this, the simulation control parameters will be listed.

6.1.35.5 PDL

See Appendix A.

6.1.35.6 Decision Tables and Algorithms

None.

6.1.36 EITNIT - Trip Demand Initialization

6.1.36.1 Identification

- o EITNIT - Trip Demand Initialization
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.1.36.2 Argument Dictionary

None.

6.1.36.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DDES	-	I*4	Destination station index
DDIS	-	I*4	trip size distribution index
DENDLP	-	I*4	Index of last origin in Trip size distribution selection data
DERR	-	I*4	Indicates whether an error was encountered
DORG	-	I*4	Origin station index
TDISTS	3	I*4	Indicates whether the corresponding trip size distribution exists

6.1.36.4 Description

This routine checks and converts the demand data and trip size distribution data to a form directly usable by the trip generation and service planning routines. The following major steps are performed:

1. A matrix containing the index to a particular trip size distribution of each O/D pair is created based on the user's input defining the O/D pairs that use the alternative distributions.

2. The mean of each trip size distribution is computed.
3. Each trip size distribution is converted to a cumulative distribution.
4. The demand matrix is converted from passengers per specified time interval to trips per hour.
5. The total number of trips and the number of trips using each trip size distribution are computed.

6.1.36.5 PDL

See Appendix A.

6.1.36.6 Decision Tables and Algorithms

The mean trip size of each trip size distribution is computed from:

$$M_j = \sum_{i=1}^m i P_{i,j} \quad (1)$$

where

M_j = mean size of distribution j ($j = 1, 2, 3$)

i = number of passengers per trip

$P_{i,j}$ = probability that number of passengers per trip is i in distribution j

m = maximum size of trip

A probability distribution is converted to a cumulative distribution using the following:

$$C_{1,j} = P_{1,j} \quad (2)$$

$$C_{i,j} = C_{i-1,j} + P_{i,j}, \quad i = 2 \text{ through } m$$

where

$C_{i,j}$ = the i th element of cumulative distribution j

The demand matrix is converted from passengers per specified time interval to trips per hour using the following:

$$TD_{O,D} = \frac{PD_{O,D} \times 60}{T \times M_{O,D}} \quad \text{for all } O,D \quad (3)$$

where

$TD_{O,D}$ = average trips per hour from O to D

$PD_{O,D}$ = total passengers departing from O for D during T

T = time interval during which $PD_{O,D}$ is valid (minutes)

O = origin

D = destination

$M_{O,D}$ = mean trip size for the O/D pair

The total number of trips per hour over the network is calculated using the following:

$$TPH = \sum_{O=1}^S \sum_{D=1}^S TD_{O,D} \quad (4)$$

where

TPH - total number of trips per hour

S - number of stations

The total number of trips per hour by trip size distribution is calculated using the following:

$$TPH1 = \sum_{O=1}^S \sum_{D=1}^S TD_{O,D} \quad \text{for those } O,D \text{ pairs using the first trip size distribution} \quad (5)$$

$$TPH2 = \sum_{O=1}^S \sum_{D=1}^S TD_{O,D} \quad \text{for those } O,D \text{ pairs using the second trip size distribution} \quad (6)$$

$$TPH_3 = \sum_{O=1}^S \sum_{D=1}^S TD_{O,D} \text{ for those } O, D \text{ pairs using the third trip size distribution (7)}$$

where

TPH_i = total trips per hour of trips using the i th trip size distribution

6.1.37 EITRIP - Trip Demand Generation

6.1.37.1 Identification

- o EITRIP - Trip Demand Generation
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.1.37.2 Argument Dictionary

None.

6.1.37.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DDES	-	I*4	Destination station ID
DDIS	-	I*4	Trip size distribution selection index
DORG	-	I*4	Origin station ID
ONE	-	I*2	The value one

6.1.37.4 Description

This routine generates a time sequence of trip arrivals. The following major steps are performed:

1. The trip tabulation array and total passengers generated counter are reset.
2. If a new demand input matrix was read, a frequency distribution of origin use is calculated from the trip per hour matrix.
3. A net scale factor for the current demand profile is calculated.
4. The mean trip interarrival time is calculated.
5. If a new demand input matrix was read, the frequency distribution of origin use is converted to a cumulative probability distribution and cumulative probability distributions of destination selection from each origin are computed.

6. Trip arrivals are generated, written to the structured demand file, and tabulated until the arrival time exceeds the specified end time.

6.1.37.5 PDL

See Appendix A.

6.1.37.6 Decision Tables and Algorithms

The frequency distribution of origin use is calculated using the following:

$$F_i = \sum_{j=1}^S TD_{i,j} \quad (1)$$

where

F_i = number of trips per hour originating at station i

S = number of stations

$TD_{i,j}$ = trips per hour from i to j

The net scale factor is computed using the following:

$$S_N = S_U \times T_D / T_U \quad (2)$$

S_N = net scale factor

S_U = user-defined scale factor applicable to current demand profile interval

T_D = original time base associated with current demand matrix

T_U = user-defined override to the time base applicable to the current demand profile interval

The mean trip interarrival time is calculated using the following:

$$\lambda = \frac{3600}{\overline{TPH} \times S_N} \quad (3)$$

where λ = mean trip interarrival time (seconds)
 TPH = total trips per hour over the network

The frequency distribution of origin use is converted to a cumulative distribution using the following:

$$CO_1 = F_1/TPH \quad (4)$$

$$CO_i = CO_{i-1} + F_i/TPH, \quad i = 2 \text{ through } S$$

where

CO_i = the cumulative probability that a trip will arrive at station i

The frequency distributions of destination selection from each origin are computed from the following:

$$CD_{i,1} = TD_{i,1}/F_i \quad (5)$$

$$CD_{i,j} = CD_{i,j-1} + TD_{i,j}/F_i, \quad j = 2 \text{ through } S$$

where

$CD_{i,j}$ = the cumulative probability that a trip will select station j as its destination given that it arrived at station i

The arrival time of a trip is calculated using the following:

$$T_i = T_{i-1} - \lambda \ln R \quad (6)$$

where

T_i = arrival time of trip i

R = a random number between zero and one taken from a uniform distribution

The origin, destination and number of passengers for the trip are obtained as follows:

O is such that CO_O is the smallest element of CO_i that is greater than or equal to R . (7)

D is such that $CD_{0,D}$ is the smallest
element of $CD_{0,i}$ that is greater than or equal
to R. (8)

N is such that CG_N is the smallest
element of CG_i that is greater than or equal
to R. (9)

where

- O = the origin
- D = the destination
- N = the number of passengers in the trip
- CG_i = the cumulative probability that a trip will consist of i
passengers (defined by routine EITNIT)
- R = random number between zero and one from a uniform distribu-
tion (a different random number is used in each of the
processes)

6.1.37a EIUDGN - Deterministic Demand Generation

6.1.37a.1 Identification

- o EIUDGN - Deterministic Demand Generation
- o GM TSC - July 1, 1981
- o PARAFOR

6.1.37a.2 Argument Dictionary

None.

6.1.37a.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DELTAT	-	R*4	Computed length of demand sub-interval
DESLST 2,KMTR	I*2		Destination station element of merge buffer
DEST	-	I*4	Destination station pointer
ETMTRX KMS,KMS	I*2		O-D "next trip" matrix
I	-	I*4	Trip buffer output pointer
INEPTR	-	I*4	Pointer to the last item in the source merge buffer
INLIST	-	I*4	Pointer to source merge buffer
INSPTR	-	I*4	Pointer to next item in source merge buffer
INTVAL	-	R*4	Length of demand interval in seconds
LSTSMP	-	I*4	Variable used for exchanging source and target buffer pointers
MNTIME	-	R*4	Time of earliest trip not yet output from O-D pairs already processed
MULT	-	R*4	Composite scaling factor for determining trip interval from O-D demand element
ORG	-	I*4	Origin station pointer

ORGLST 2,KMTR	I*2	Origin station element of merge buffer
OUTLST	- I*4	Pointer to target merge buffer
OUTPTR	- I*4	Pointer to next empty space in target buffer
RATE	- R*4	Unscaled passengers/demand interval rate for all O-D pairs together
SCALE	- R*4	Scale factor of passengers/demand interval
TIME	- R*4	End time for a demand subinterval
TIMLST 2,KMTR1	R*4	Trip arrival time element of merge buffer
TIMT	- R*4	Time of next trip for O-D pair in process
TIMT2	- R*4	Time of trip being copied back to next trip matrix
TRPINT	- R*4	Interval between successive trips from the O-D pair being processed
TRPSIZ	- I*4	Trip group size for the O-D pair being processed or output

6.1.37a.4 Description

Routine EIUDGN generates trips in time series for each O-D pair using the computed uniform interarrival time for that pair, as determined from the input O-D matrix which specifies number of passengers per demand interval. This rate is scaled by the optional scale factor and demand time base override from variable DMPROF. The trip interarrival time is the inverse of the scaled rate times the user input fixed trip size (variable TSZARR) for each O-D pair.

The initial trip for each O-D pair for each demand interval is generated at one-half the trip interarrival time past the start of the demand interval. Each subsequent trip occurs one trip interarrival time past the previous trip through the demand interval. Trips are generated for each O-D pair in turn and then sorted into arrival time sequence by a buffer exchange merge sorting algorithm. The routine uses the system-wide trip arrival rate to determine the approximate time interval needed to generate sufficient trips to fill one of the exchange buffers. Trips are then generated for each O-D pair, in turn, for this demand sub-interval, variable DELTAT. The buffer exchange algorithm is then used to merge the trips into a time-ordered trip list. The sub-interval cycle continues until all trips in the demand interval have been output. The net difference between the expected number of passengers generated and the actual number output is calculated and displayed for each demand interval. The algorithm is repeated for subsequent demand intervals.

6.1.37a.5 PDL

None.

6.1.37a.6 Decision Tables and Algorithms

None.

6.1.38 XGDIPF4 - Read Full Word GDIP Data

6.1.38.1 Identification

- o XGDIPF4 - Read Full Word GDIP Data
- o IBM/FSD - July 1, 1977
- o PARAFOR

6.1.38.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ARRAY	IDA, IDB, IDC, IDD	I*4	Array into which the data is to be read
IDA	-	I*4	Maximum value of first dimension
IDB	-	I*4	Maximum value of second dimension
IDC	-	I*4	Maximum value of third dimension
IDD	-	I*4	Maximum value of fourth dimension
FMT	2	I*4	Format of data on input record
JRAL	-	I*4	Beginning value of first dimension
JRAH	-	I*4	Ending value of first dimension
JRBL	-	I*4	Beginning value of second dimension
JRBH	-	I*4	Ending value of second dimension
JRCL	-	I*4	Beginning value of third dimension
JRCH	-	I*4	Ending value of third dimension
JRDL	-	I*4	Beginning value of fourth dimension
JRDH	-	I*4	Ending value of fourth dimension

6.1.38.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
BUFFER	20	I*4	Input buffer
DUMP	-	I*4	The characters 'DUMP'
ENT	78	I*4	Contents of input buffer in specified format
FORMAT	4	I*4	Format of data on input record
IRAH	-	I*4	Ending value of first dimension
IRAL	-	I*4	Beginning value of first dimension
IRBH	-	I*4	Ending value of second dimension
IRBL	-	I*4	Beginning value of second dimension
IRCH	-	I*4	Ending value of third dimension
IRCL	-	I*4	Beginning value of third dimension
IRDH	-	I*4	Ending value of fourth dimension
IRDL	-	I*4	Beginning value of fourth dimension
LENGTH	-	I*2	Defined length of input buffer in bytes
NBYTES	-	I*2	Number of bytes currently usable in input buffer
NCOPY	-	I*4	Number of times an input record is to be read
NOENT	-	I*4	Number of entries in input buffer

6.1.38.4 Description

This routine reads data in GDIP format into full word variables. The input record is read as a replication factor (first two columns) followed by a character string into the input buffer. A pseudo read is then performed on the input buffer to format the data and store it into the specified variable. The replication factor controls the number of times the pseudo read is performed.

6.1.39 XGDIPH4 - Read Half Word GDIP Data

6.1.39.1 Identification

- o XGDIPH4 - Read Half Word GDIP Data
- o IBM/FSD - July 1, 1977
- o PARAFOR

6.1.39.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ARRAY	IDA, IDB, IDC, IDD,	I*2	Array into which the data is to be read
IDA	-	I*4	Maximum value of first dimension
IDB	-	I*4	Maximum value of second dimension
IDC	-	I*4	Maximum value of third dimension
IDD	-	I*4	Maximum value of fourth dimension
FMT	2	I*4	Format of data on input record
JRAL	-	I*4	Beginning value of first dimension
JRAH	-	I*4	Ending value of first dimension
JRBL	-	I*4	Beginning value of second dimension
JRBH	-	I*4	Ending value of second dimension
JRCL	-	I*4	Beginning value of third dimension
JRCH	-	I*4	Ending value of third dimension
JRDL	-	I*4	Beginning value of fourth dimension
JRDH	-	I*4	Ending value of fourth dimension

6.1.39.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
BUFFER	20	I*4	Input buffer
DUMP	-	I*4	The characters 'DUMP'
ENT	78	I*2	Contents of input buffer in specified format
FORMAT	4	I*4	Format of data on input record
IRAH	-	I*4	Ending value of first dimension
IRAL	-	I*4	Beginning value of first dimension
IRBH	-	I*4	Ending value of second dimension
IRBL	-	I*4	Beginning value of second dimension
IRCH	-	I*4	Ending value of third dimension
IRCL	-	I*4	Beginning value of third dimension
IRDH	-	I*4	Ending value of fourth dimension
IRDL	-	I*4	Beginning value of fourth dimension
LENGTH	-	I*2	Defined length of input buffer in bytes
NBYTES	-	I*2	Number of bytes currently usable in input buffer
NCOPY	-	I*4	Number of times an input record is to be read
NOENT	-	I*4	Number of entries in input buffer

6.1.39.4 Description

This routine reads data in GDIP format into half word variables. The input record is read as a replication factor (first two columns) followed by a character string into the input buffer. A pseudo read is then performed on the input buffer to format the data and store it into the specified variable. The replication factor controls the number of times the pseudo read is performed.

6.1.39.5 PDL

None.

6.1.39.6 Decision Tables and Algorithms

None.

6.1.40. XGDIPX4 - Read Single Byte GDIP Data

6.1.40.1 Identification

- o XGDIPX4 - Read Single Byte GDIP Data
- o IBM/FSD - July 1, 1977
- o PARAFOR

6.1.40.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ARRAY	IDA, IDB, IDC, IDD	L*1	Array into which the data is to be read
IDA	-	I*4	Maximum value of first dimension
IDB	-	I*r	Maximum value of second dimension
IDC	-	I*4	Maximum value of third dimension
IDD	-	I*4	Maximum value of fourth dimension
FMT	2	I*4	Format of data on input record
JRAL	-	I*4	Beginning value of first dimension
JRAH	-	I*4	Ending value of first dimension
JRBL	-	I*4	Beginning value of second dimension
JRBH	-	I*4	Ending value of second dimension
JRCL	-	I*4	Beginning value of third dimension
JRCH	-	I*4	Ending value of third dimension

VARIABLE	DIM	TYPE	DESCRIPTION
JRDL	-	I*4	Beginning value of fourth dimension
JRDH	-	I*4	Ending value of fourth dimension

6.1.40.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
BUFFER	20	I*4	Input buffer
DUMP	-	I*4	The characters 'DUMP'
ENT	78	L*1	Contents of input buffer in specified format
FORMAT	4	I*4	Format of data on input record
IRAH	-	I*4	Ending value of first dimension
IRAL	-	I*4	Beginning value of first dimension
IRBH	-	I*4	Ending value of second dimension
IRBL	-	I*4	Beginning value of second dimension
IRCH	-	I*4	Ending value of third dimension
IRCL	-	I*4	Beginning value of third dimension
IRDH	-	I*4	Ending value of fourth dimension
IRDL	-	I*4	Beginning value of fourth dimension
LENGTH	-	I*2	Defined length of input buffer in bytes
NBYTES	-	I*2	Number of bytes currently usable in input buffer
NCOPY	-	I*4	Number of times an input record is to be read
NOENT	-	I*4	Number of entries in input buffer

6.1.40.4 Description

This routine reads data in GDIP format into single byte variables. The input record is read as a replication factor (first two columns) followed by a character string into the input buffer. A pseudo read is then performed on the input buffer to format the data and store it into the specified variable. The replication factor controls the number of times the pseudo read is performed.

6.1.40.5 PDL

None.

6.1.40.6 Decision Tables and Algorithms

None.

6.1.41 XNDBOR - Generalized Data Input Processing

6.1.41.1 Identification

- o XNDBOR - Generalized Data Input Processing
- o IBM/FSD - July 1, 1977
- o PARAFOR

6.1.41.2 Argument Dictionary

None.

6.1.41.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
END	-	R*8	The characters 'END'
FMT	-	R*8	Format of the data
IRAH	-	I*4	Ending value of first dimension
IRAL	-	I*4	Beginning value of first dimension
IRBH	-	I*4	Ending value of second dimension
IRBL	-	I*4	Beginning value of second dimension
IRCH	-	I*4	Ending value of third dimension
IRCL	-	I*4	Beginning value of third dimension
IRDH	-	I*4	Ending value of fourth dimension
IRDL	-	I*4	Beginning value of fourth dimension
NAME	-	R*8	Name of the variable

6.1.41.4 Description

This routine reads the name card of data that is defined in the GDIP format. The name card contains the name of the variable to be initialized, the format of the data and the lower and upper bounds of up to four dimensions which define the portion of the variable to be initialized. This information is passed to routine GDIP4 (see subsection 6.1.16), which determines the type of the variable and calls the proper routine to read the data into the variable. Routine XNDBOR continues until a record containing the characters END in the first three columns is read.

6.1.41.5 PDL

None.

6.1.41.6 Decision Tables and Algorithms

None.

6.1.42 XPSEUDO - I/O Intercept Routine

6.1.42.1 Identification

- o XPSEUDO - I/O Intercept Routine
- o IBM/FSD - July 1, 1977
- o ASM

6.1.42.2 Argument Dictionary

ENTRY	VARIABLE	DIM	TYPE	DESCRIPTION
PFIOCS	PARM 1	-	L*1	Type of I/O operation requested
	PARM 2	4	L*1	Data set reference number
	PARM 3	-	L*1	Type of I/O
SUDOGO	LENGBUF	-	I*2	Current number of characters in buffer
	BUFSIZ	-	I*2	Defined size of buffer
	BUFFER	20	I*4	The buffer

6.1.42.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ASUDOBF	-	I*4	Address of COMMON/SUDOBF/
BUFLOC	-	I*4	Current I/O position in buffer
CURUNIT	-	L*1	Current data set reference number
FIOCSB	-	I*4	Address of true FIOCS routine
I256	-	I*4	The value 256

VARIABLE	DIM	TYPE	DESCRIPTION
ONE	-	I*4	The value 1
SAVE	16	I*4	Register save area
SIX	-	I*4	The value 6

6.1.42.4 Description

This routine provides pseudo input operations for the simulation, which are performed during the read of data in the Generalized Data Input Package (GDIP) format. Entry SUDOGO is called by one of the GDIP read routines (XGDIPF4, XGDIPH4, XGDIPX4) to receive the location and size of the pseudo input buffer. SUDOGO also alters IBCOM so that all calls to FIOCS will come to entry point PFIOCS in this routine first. Then, for each FORTRAN I/O request, entry PFIOCS will be called. If the data set reference number in the READ or WRITE statement is not zero, PFIOCS calls the normal FIOCS routine. If the data set reference number in a READ statement is zero, pseudo input is required and PFIOCS supplies the character string in the pseudo input buffer as the result of the read operation. The character string is formatted as in a normal read to convert it to numerical data. The following major steps occur during a GDIP read:

1. A GDIP read routine calls SUDOGO to define the pseudo input buffer and intercept normal FORTRAN I/O calls.
2. The GDIP read routine performs a normal read (data set reference equal to five) of one record which consists of a replication factor (columns 1 and 2) and 78 columns of data. The data is read as a character string into the pseudo input buffer.
3. The GDIP read routine then performs a read with the data set reference number equal to zero. PFIOCS recognizes this as a pseudo read and returns the contents of the pseudo input buffer in the specified format as the result of the read operation. The GDIP read routine performs the pseudo read the number of times indicated by the replication factor and then continues at step 2 until all of the data for the specified array has been read.

6.1.42.5 PDL

None.

6.1.42.6 Decision Tables and Algorithms

None.

6.1.43 XTRACBK - Interrupt Handler

6.1.43.1 Identification

- o XTRACBK - Interrupt Handler
- o IBM/FSD - July 1, 1977
- o ASM

6.1.43.2 Argument Dictionary

None.

6.1.43.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
AREA	18	I*4	Save area for subroutine call
BAD	16	L*1	The characters 'BADO' repeated four times
BLANKS	16	L*1	Sixteen space characters
BOTSAV	-	I*4	Address of earliest subroutine linkage save area
CALLADR	-	I*4	Address of entry point in called routine
CALLEDR	16	L*1	Name of called routine
CTR	16	L*1	Count of program interrupts by type
ENTRYAD	-	I*4	Address of beginning of calling routine
FLAGIT	-	I*4	Indicates error encountered during traceback
HIGHADRS	-	I*4	Highest memory address

VARIABLE	DIM	TYPE	DESCRIPTION
INST	12	L*1	Instruction image at program interrupt
LOWADRS	-	I*4	Lowest memory address
OLDPSW	-	R*8	Program Status Word at program interrupt
PARMADRS	-	I*4	Address of arguments in a subroutine call
PARMCONT	32	L*1	First 32 bytes of arguments in a subroutine call
QUITFLAG	-	I*4	Logical flag indicating termination required
RELOCN	-	I*4	Relative address of call in calling routine
RETURNAD	-	I*4	Absolute address of call in calling routine
ROUTINE	16	L*1	Name of calling routine
SPA	16	I*4	Save area for general registers when program interrupt occurs
SPC	12	R*8	Contents of general and floating point registers at program interrupt
STMTNO	-	I*4	Internal statement number in calling routine (not active)
XLIST	14	I*4	List of areas having interrupts to be ignored.

6.1.43.4 Description

This routine retrieves the Program Status Word, General and Floating Point Register contents, Instruction Image and subroutine call hierarchy when a program interrupt occurs. This information is passed to FORTRAN

routine XTRCBKP through various entry points to be formatted and printed. A call to entry SPIEL initializes the program to intercept subsequent program interrupts. Entry SPIELA is called by the system when a program interrupt occurs, which initiates the retrieval and printing of a program interrupt report. If entry TRACBK is called, a listing of the current hierarchy of subroutine calls will be generated. Section 4.3 describes the system trace facility of which this routine is a component.

6.1.43.5 PDL

None.

6.1.43.6 Decision Tables and Algorithms

None.

6.1.44 XTRCBKP - Save Area Trace Formatting

6.1.44.1 Identification

- o XTRCBKP - Save Area Trace Formatting
- o IBM/FSD - July 1, 1977
- o PARAFOR

6.1.44.2 Argument Dictionary

ENTRY	VARIABLE	DIM	TYPE	DESCRIPTION
TRCBKP	I	12	I*4	Identity and address information associated with the call of one routine by another routine.
TRCBKV	J	9	I*4	Address and values of up to eight words of an argument supplied in a sub-routine call
TRCBKR	K	13	I*4	Contents of the general registers at the time of the call
SPIELP	OLDPSW	2	R*4	Program Status Word
	SPD	24	R*4	Contents of general purpose and floating point registers
	INST	3	R*4	Instruction image
SPIELQ	MSG	4	R*4	Message associated with abnormal termination.

6.1.44.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
IOUT	-	I*4	Data set reference number in write statement (=6)

6.1.44.4 Description

This routine formats and outputs the information associated with program interrupts and the current hierarchy of subroutine calls. Section 4.3 describes the system trace facility of which this routine is a component.

Entry TRCBKP writes the information contained in argument I, which defines a subroutine call including: calling routine, internal statement number of call (not active), absolute address and relative address of call in calling routine, called routine, and absolute address of entry point in called routine.

Entry TRCBKI writes the column headings for the subroutine call hierarchy.

Entry TRCBKV writes the address of an argument supplied during a subroutine call and the eight words of core beginning at that address as defined in argument J.

Entry TRCBKR writes the contents of the general registers that existed at a subroutine call as defined in argument K.

Entry SPIELP writes the output associated with a program interrupt including: the Program Status Word, the general purpose registers and floating point registers, and the instruction image. This information is provided as input arguments.

Entry SPIELQ writes a message provided as an input argument when the program interrupt results in termination of the program.

6.1.44.5 PDL

None.

6.1.44.6 Decision Tables and Algorithms

None.

6.2 MODEL PROCESSOR

This section contains the subprogram descriptions for the DESM Model Processor.

6.2.1 DAYTIM - Obtain Date and Time

See subsection 6.1.1, DAYTIM

6.2.2 DEBUG - Auxiliary Output Macro

See subsection 6.1.2, DEBUG

6.2.3 DQUE - Remove First Transaction From A Queue

6.2.3.1 Identification

- o DQUE - Dequeue Macro
- o IBM/FSD - May 1, 1978
- o PL/I

6.2.3.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TYPE	-	C	Transaction type to be dequeued (X-System, V-Vehicle, T-Trip)
HEAD	-	C	Queue list pointer (fully qualified)
INDEX	-	C	(INPUT) Variable name to be assigned transaction ID removed from list (OUTPUT) Transaction ID removed from list, zero if list empty
YCHAIN	-	C	Transaction chain word (Optional: default standard chain (QUECH/FELCH))

6.2.3.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
OUT	-	C	Generated text string
M	-	C	Output page margin indicator
XCHAIN	-	C	Default chain word (QUECH)

6.2.3.4 Description

DQUE provides a standardized programming procedure for generating a FORTRAN code sequence to remove the first transaction from a simulation queue. The transaction removed can be the first or last to en-

ter the queue, depending upon whether the queue organization is FIFO or LIFO, respectively.

The code generated by the macro checks the indicated queue for the existence of queued transactions, and if none are available returns a zero transaction ID. Otherwise, the transaction pointed to by the transaction indicated in the queue list pointer is removed from the queue and its ID is returned.

6.2.3.5 PDL

None.

6.2.3.6 Decision Tables and Algorithms

The structure and dequeuing procedure for a simulation queue is described in Section 2.2.1.

6.2.4 DQUEM - Remove a Specified Transaction From a Queue

6.2.4.1 Identification

- o DQUEM - Dequeue a Particular Entity Macro
- o IBM/PSD - May 1, 1978
- o PL/I

6.2.4.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TYPE	-	C*1	Transaction type to be dequeued (X-System, V-Vehicle, T-Trip)
HEAD	-	C	Queue list pointer (fully qualified)
INDEX	-	C	(INPUT) Variables containing ID of transaction to be de- queued (OUTPUT) >0, if transaction not found =0, if transaction dequeued
YCHAIN	-	C	Transaction chain word (Optional: default standard chain QUECH/FELCH)

6.2.4.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
OUT	-	C	Generated text string
M	-	C	Output page margin indicator
XCHAIN	-	C	Default chain word (QUECH)
LOC1	-	C	Current transaction ID
LOC2	-	C	Next transaction ID

6.2.4.4 Description

DQUEM provides a standardized programming procedure for generating a FORTRAN code sequence to search a simulation queue and remove a specific transaction. The code generated traces the circular transaction chain, indicated by the queue list pointer, until either the specified transaction is found or the end of the queue is encountered. If the desired transaction is found, it is removed from the queue by adjusting the transaction chain pointers and the successful dequeuing operation is indicated by zeroing the input transaction ID. If the queue is empty or the specified transaction cannot be located in the queue, the input transaction ID is left unchanged to indicate the unsuccessful dequeue attempt.

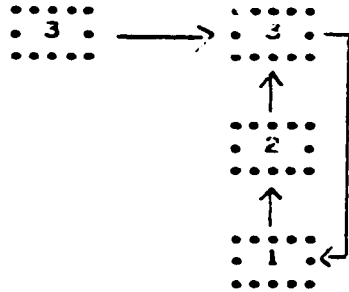
6.2.4.5 PDL

None

6.2.4.6 Decision Tables and Algorithms

The transaction removal procedure implemented by the macro involves resetting the chain points of the preceding transaction in the queue to the value of the chain pointer of the transaction being removed as shown in Figure 6-1. If transaction removal results in the queue being empty, the queue list pointer is set zero.

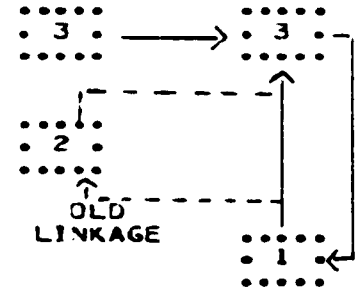
QUEUE LIST
POINTER



ORIGINAL CHAIN

TRANSACTION
CHAIN
POINTER

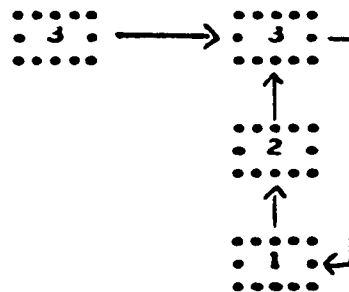
REMOVED
TRANS-
ACTION



MODIFIED CHAIN

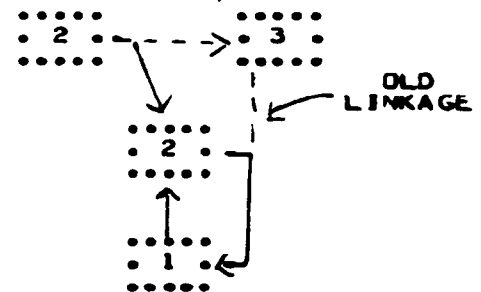
FIFO DEQUEUE

QUEUE LIST
POINTER



ORIGINAL CHAIN

REMOVED
TRANS-
ACTION



MODIFIED CHAIN

LIFO DEQUEUE

FIGURE 6-1. TRANSACTION DEQUEUEING

6.2.5 DTIMEL - Read System Clock

See subsection 6.1.3, DTIMEL.

6.2.6 EAAFSM - Active Fleet Size Management

6.2.6.1 Identification

- o EAAFSM - Perform Fleet Size Modification
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.6.2 Argument Dictionary

None.

6.2.6.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
CAP	-	L*1	Logical variable indicating storage capacity exceeded
CURVEH	-	I*4	Number of vehicles currently assigned to route
DIRSTN	-	I*4	Directional station ID
EVP	-	I*4	Event list pointer
HWAY	-	L*1	Flag indicating headway event on input ramp
MRATIO	-	R*4	Maximum ratio of vehicle need on route
MTR	-	I*4	Old number of trains on route
NDP	-	I*4	Loop counter - directional platforms
NDS	-	I*4	Loop counter - directional stations
NEWVEH	-	I*4	Number of vehicles intended to travel route
NTR	-	I*4	Number of trains to terminate
NUM	-	I*4	Number of transition vehicles to initialize
NI	-	I*4	Vehicles needed on route

N2	-	I*4	Vehicles intended on route
RATIO	-	R*4	Ratio of vehicle need on route
RNEED	-	I*4	Route with maximum need ratio
RTE	-	I*4	Loop counter - route ID
SL	-	I*2	Loop counter - station link ID
SLP	-	I*4	Loop counter - station link dock ID
STN	-	I*4	Station ID
SUM	-	I*4	Number of vehicles marked for termination
VEH	-	I*4	Vehicle transaction ID

6.2.6.4 Description

EAAFSM performs adjustment of the vehicle fleet in response to user requests entered as asynchronous input via the time dependent runtime data file. Fleet adjustments can be made to either increase or decrease the number of vehicles available for service. The implementation of a fleet size change by EAAFSM differs for the several service types. Fleet size changes for Demand Responsive Service are performed by iteratively cycling through each station area and either increasing or decreasing the number of vehicles in storage by one, each time a particular station is encountered. As a vehicle is removed or added to storage, the current fleet size is adjusted. Each vehicle removed from storage is freed and returned to the available list of vehicle transactions. For fleet reductions this process is repeated until either all station storage areas have been depleted or the requested fleet change has been satisfied (desired fleet size is equal to actual). If the necessary fleet size reduction cannot be satisfied, solely by removing vehicles from storage, the next n vehicles, given by the distance between the actual and desired fleet sizes, are removed from the network by routine ESMOLA as they become empty. If an attempt is made during processing to increase the fleet beyond available storage capacity, an information message is issued and simulation processing is continued without attempting further fleet size increases.

In the scheduled service case, EAAFSM can increase or decrease the number of trains traveling each route, or can change the train consist of the fleet of trains traveling a route, or any combination of these. The following functions are performed in initiating the scheduled service AFSM response by routine EAAFSM.

- o Validates the number of transition vehicles input for each route
- o Calculates total vehicles required for new level of service
- o Calculates total vehicles available as sum of previous vehicles plus user-input transition vehicles

- o If more vehicles are necessary, additional transition vehicles are assigned to the routes having the maximum need ratio.
- o For each route, if the train consist is unchanged and number of vehicles decreases, enough trains are marked to terminate for AFSM to meet the new requirement.
- o For each route, if the train consist is changed, a number of vehicles equal to the number of excess vehicles on the route (upper bounded by the number of trains currently on the route) are marked to terminate for AFSM.
- o For each route, transition vehicles are initialized by invoking routine EAIVFH. The count of vehicles assigned to the route is then set to the number of current vehicles plus the number of transition vehicles. Then, if the train consist is changed, the count of current vehicles is set back to zero (and relevant statistics changed) to control the reconstituting process. That is, ANVRTE now counts the number of vehicles traveling the route at the correct train consist.
- o For each route, EANTRN is invoked to attempt to form and launch new trains on the route in the barn (from transition vehicles) and EASTOR is invoked to identify and reassign excess vehicles in storage for the route.

The active fleet size change is subsequently completed when vehicles marked to terminate arrive at the maintenance barn and are either reconstituted and launched on the route, reassigned to another route, or removed from service.

In the timeout/group demand responsive case, fleet size increases are effected by introducing new vehicles on input ramps of stations by iteratively cycling through each station given capacity constraints until the new desired fleet size is reached. Station link statistics and inventory management parameters are updated. Fleet size reductions are accomplished by routine ESMOLA by removing vehicles from services as they become empty.

6.2.6.5 PDL

See Appendix A.

6.2.6.6 Decision Tables and Algorithms

None.

6.2.7 EAASYN - Asynchronous Event Processing

6.2.7.1 Identification

- o EAASYN - Asynchronous Event Processing
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.7.2 Argument Dictionary

None.

6.2.7.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ATYPES	KMDHR	I*4	Header card array
CU	-	I*4	Time of next event in clock units
ENDI	-	I*4	The character string "END" (left justified)
EOF	-	L*1	End file indicator
FIND	-	L*1	End data indicator - true upon reading 'END' header
LINK	-	I*4	Link ID
NLSLT	-	I*2	Current minimum path routing table number
R	-	I*4	Loop counter - route ID
SKIP	-	L*1	End asynchronous input indicator - set time after reading 'STOP' or 'EOD'
STN	-	I*4	Station ID - loop counter
STNI	-	I*4	Station ID - inner loop counter
TEXT	18	I*4	Text or comment follower cards read and echoed to SYSOUT
TIME	-	R*4	Time of next asynchronous event occurrence
TRNS	-	I*4	Temporary transaction ID
TRTIME	-	I*4	Temporary estimated time of arrival

6.2.7.4 Description

EAASYN is invoked by the main routine, EMODEL, to process an asynchronous event occurrence as indicated by the previous header read from the time dependent runtime data file. Upon entry, the minimum path routing table ID is saved for use in determining whether the estimated time of arrival table for travel between network stations must be recomputed. The type of processing required is then determined by comparing the mnemonic of the last asynchronous header statement read against a table of valid asynchronous input commands. The relative displacement of the mnemonic in the command list serves as an index to invoke required processing as follows:

1. Checkpoint Command (CKPT) - A demand checkpoint of system status is taken by invoking routine EACKPT to write a record to the checkpoint file.
2. End of Data or Stop (EOD, STOP) - The occurrence of this header causes a termination transaction to be scheduled for graceful termination of the simulation system at the time indicated in the header statement.
3. Index File Input (INDEX) - The next n card images are read from the asynchronous input file and written to the run index file. Asynchronous processing is terminated upon encountering an END header statement in the input stream.
4. Text Input (TEXT) - This header indicates a card image formatted line of descriptive text follows in the asynchronous runtime input. The line of text is read and transferred to the system output device.
5. Data Read Command (DATA/OPTION/PARAM/SELECT) - This command causes routine NDBOR to be invoked to initiate the reading of asynchronous input by the Generalized Data Input Processor (GDIP). The input data read represents data change requests which update global common area variables and parameters. Termination of input occurs upon encountering an 'END' header which signifies end of data.
6. Failure Command (FAIL) - Failure events in the DFSS encompass both occurrence and failure of station and guideway links and vehicles. Failures are introduced into the system when processing of a FAIL command occurs. The failure is processed by invoking NDBOR to initiate reading of failure related data contained in the asynchronous runtime data file. This data identifies the location (station or guideway) and type of failure as follows:
 - 1 -- guideway link entry
 - 2 -- guideway link exit
 - 3 -- station link entry
 - 4 -- station link exit

5 -- entire station (entry
and exit of station are
failed/recovered simultaneously)

6 -- degradation (vehicle failure)

In addition, for guideway and vehicle failures, this data may include routing information used in subsequent vehicle dispatching from network stations and guideway link processing. Failures have routing table information that attempts to route future vehicles around the failure location, while recovery data includes specification of the routing table to become effective once the failure is cleared. Information defining the recovery method, delay times, and other vehicle responses associated with the failure are also included in this data. Degraded vehicle or station link failures have a penalty factor (fraction of velocity at which the vehicle or vehicles are to operate) as data which is read during the recognition of failures.

Once all failure data has been read by GDIP, a determination is made as to whether station or guideway processing is required and the appropriate failure processing routine is invoked, EGFAIL for guideway links, ESFAIL for station links, respectively.

7. **Active Fleet Size Management (AFSM)** - Processing of this command begins with determining if a previously entered AFSM request is currently active. If this condition exists, a serious error message is issued and simulation termination occurs. Otherwise, the count of vehicles currently circulating on scheduled routes, if they exist, is saved and NDBOR is invoked to initiate reading of AFSM related data from the asynchronous runtime file. Upon completion of data input processing, routine EAAFSM is invoked to implement the Active Fleet Size Management request.
8. **Auxiliary Output Request (FLAG)** - This header card is used to initialize the auxiliary output feature of the simulator for tracing execution flow through the simulation system. Routine EAFLAG is invoked to process the auxiliary output indicator setting contained in the asynchronous runtime file.

9. Comment Data (COMMENT) - The following n card images contained in the input stream are read and transferred to the system output device until an 'END' command is encountered.

Upon completion of the processing associated with the current header request, the next header is read from the asynchronous input stream, and the asynchronous input request transaction is rescheduled to occur at the time specified in the header data. If the current request caused the scheduling of simulation termination, this rescheduling process is not performed. If a new routing table was input as the result of asynchronous processing, the estimated time of arrival between each station pair in the network (PETATM), is recomputed. The determination of the routing table change is made by comparing the saved table ID to the currently active table ID.

6.2.7.5 PDL

See Appendix A.

6.2.7.6 Decision Tables and Algorithms

None.

6.2.8 EACKR - Checkpoint/Restart Processing

6.2.8.1 Identification

- o **EACKR - Checkpoint/Restart Initialization and Processing**
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.8.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
CAREA	-	I*4	Starting address of System Status Area
LEN1	-	I*4	Length in full words of global common data contained in System Status Area
TIME	-	R*4	Time of requested restart (E.P. EAREST)

6.2.8.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ATIME	-	R*4	I/O parameter containing time of data input
ATYPES	14	I*4	List of asynchronous header types
END	-	I*4	Keyword 'END' for recognizing end of asynchronous input
EOF	-	L*1	Logic variable for End File
FIND	-	L*1	Logic variable for legal header found
TTIME	-	I*4	Requested restart time in C.U.

6.2.8.4 Description

Checkpointing is performed to save the status of a simulation experiment at any point during the simulation run. Checkpointing can occur at periodic intervals or via an asynchronous data request. The checkpoint data can be used to restart the simulation by reinitialization of system status as saved by the checkpoint. This routine contains entry points to perform both checkpointing (EACKPT) and restart (EAREST). The writing of system status during a checkpoint involves a sequential binary write of core storage beginning at the symbolic address defining the beginning of global common data for the simulation system. The address of this area and its length are defined to checkpoint processing during system initialization. This is accomplished via a call to the primary entry point of the routine. The result of this call is to establish an equivalence between the I/O array defined in the routine and the physical addresses of core memory containing the global common area data. The actual writing of the checkpoint data is performed by a call to entry point EACKPT.

System restart is accomplished via a call to entry point EAREST. The restart process involves reading the checkpoint file until a record containing a clock time which is greater than or equal to the requested restart time is encountered. If no such record is found, a serious error message is issued and processing is terminated. Otherwise, the trip arrival and asynchronous input files are spaced forward to the time of restart and scheduling of the first trip arrival and asynchronous input event is performed. The initial header record containing run time constants is then written to the sampling file.

6.2.8.5 PDL

See Appendix A.

6.2.8.6 Decision Tables and Algorithms

See description of EAINIT for discussion of System Status Area.

6.2.9 EACOMN - Structured Data Commons Ordering

See subsection 6.1.4, EACOMN.

6.2.10 EADADD - Initialize System Status Area

6.2.10.1 Identification

- o EADADD - Initialize System Characteristics, Network Characteristics and Message Common Area Data
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.10.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
IPARE1	LEN1	I*4	Array equivalenced to network characteristics data in System Status Area
IPARE2	LEN2	I*4	Array equivalenced to system characteristics data in System Status Area
LEN1	-	I*4	Length of network characteristics data in full words
LEN2	-	I*4	Length of system characteristics data in full words

6.2.10.3 Local Variable Dictionary

None.

6.2.10.4 Description

EADADD performs the transfer of I.P. generated data contained in the System Characteristics and Network Structured Data files to the System Status Area. This area containing the global common data used by the MP. The regions within the Status Area, which are to contain the data, are established during initialization by a call to the primary entry point of the routine. The result of this call is to establish an equivalence between the physical memory addresses in the Status Area and locally defined arrays into which the System Characteristics and Network data are read. The actual reading of the Structured data is performed by

entry point EANDTA. After reading the data, message output limits are established.

6.2.10.5 PDL

See Appendix A.

6.2.10.6 Decision Tables and Algorithms

See description of EAINIT for discussion of System Status Area.

6.2.11 EAERR - Error Message Processing

6.2.11.1 Identification

- o EAERR - Standardized Error Handling
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.11.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
MSEVER	-	I*4	Message severity code 1 - Information 2 - Warning 3 - Serious
MSG	2	L*1	Message Text
MSGNO	-	I*4	Error message number

6.2.11.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
MCLOCK	-	R*4	Clock Time in seconds
MSGTYP	-	L*1	ALPHA message type designator
NUM	-	I*4	Count of characters in text string
PGM	2	I*4	Processor mnemonic
SCLN	-	L*1	End of text symbol ';'
TYPE	3	L*1	Message type symbols 'I', 'W', 'S'

6.2.11.4 Description

Upon entry, EAERR determines whether or not error processing was invoked while attempting to terminate MP processing due to a prior error condition. If this situation exists, a message indicating the condition is issued and immediate termination of the MP occurs. Otherwise, the required error message is formatted and displayed on the system output device.

After message display, a determination is made as to whether any specified output limits have been violated or whether system termination is required based on message severity. If the number of times a given message is issued, total number of messages issued, or the number of messages by severity code exceed SYSGEN limits, termination of the MP occurs. This termination activity attempts graceful system shutdown by invoking the DESM traceback facility (TRACBK) and model termination activity (EAFINS). Similar processing is performed following the issuance of a severe error message. If error processing results in MP termination for any reason, a system condition code 16 is returned to the simulation user.

6.2.11.5 PDL

See Appendix A.

6.2.11.6 Decision Tables and Algorithms

None.

6.2.12 EAFINS - System Termination

6.2.12.1 Identification

- o EAFINS - System Termination
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.12.2 Argument Dictionary

None.

6.2.12.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
AVG	-	R*4	Mean time/scheduled event
STD	-	R*4	Standard Deviation for scheduled event time
TIME	-	I*4	Beginning time for scheduling interval
TIME	-	I*4	Ending time for scheduling interval
TOTAL	-	I*4	Sum total of scheduled events
TOTAL1	-	R*4	Summation of event scheduling times
TOTAL2	-	R*4	Summation of event scheduling times squared

6.2.12.4 Description

EAFINS is invoked to perform model termination activity. This termination can be initiated as the result of error processing (EAERR) or the occurrence of a simulation STOP or EOD command.

Termination activities begin by calling EAFRPT to display termination status of the simulation experiment. EAWTIX is then invoked to update the run index to reflect output files generated during the

simulation experiment. Event timing usage statistics are gathered and displayed in the Event Scheduling Summary Report. The number of messages issued by type are then summarized and displayed prior to terminating model execution.

Simulator efficiency as given by the ratio of simulated time to CPU time is effected by the usage demands placed upon the event scheduling mechanism of the simulation system. This ratio can be improved by adjusting parameters which control operation of the simulation event scheduling mechanism. The relationship between event scheduling parameters defined by the user and organization of the Future Events List is shown in Figure 6-2. CLSMAL establishes the width in clock units of each clock table (C/T) entry given by $CSIZE * 10$. CLSIZE defines the length of the clock table in scheduling intervals. CLBIG is derived within the simulator as $CLSIZE * CLSMAL$ and represents the total time in clock units encompassed by the clock table and each multiple thread (M/T) list. The structure of the Future Events List is such that if the Δt of a TXN is less than CLBIG, then the TXN is put into the C/T while if $\Delta t > CLBIG$, the TXN is placed on the M/T.

If the interval width of the clock table is specified too large, the overhead required to chain transactions in time order within the interval increases. This situation occurs by overspecifying clock granularity as given by CSIZE. Similarly, specifying the interval too small can result in inefficiencies by causing underutilization of consecutive clock table entries and increasing linear search time in the table for transaction insertion and removal. Extensive use of the multiple thread also increases scheduling overhead since reloading of the C/T from the M/T is required for each interval of time given by CLBIG. Each transaction transferred from the M/T to the C/T must be sorted and placed within the appropriate C/T interval.

The event scheduling summary displayed at simulation termination is intended to provide insight into event scheduling efficiency such that the simulation user can fine tune the scheduling process for specific classes of model application. The summary contains a distribution of event scheduling activity within the Future Events List. The number of events scheduled and the fraction of total is displayed in ten frequency classes as defined by CLBIG. The first class corresponds to the C/T portion of the list while the other classes reflect M/T utilization. The mean and standard deviation for all scheduled events is also displayed to provide insight into the transaction distribution within the C/T.

The objective in using this data for adjusting event scheduling parameters should be to obtain a large proportion of scheduled events within the clock table. The distribution of Δt 's for scheduled transactions should encompass the entire C/T interval. However, the required

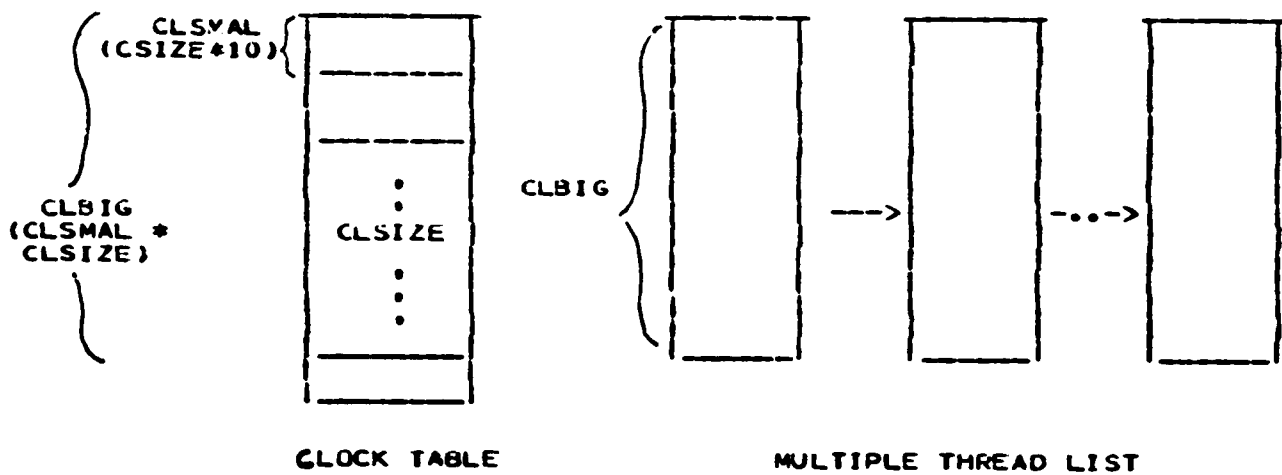


FIGURE 6-2. FUTURE EVENTS LIST DEFINITION

balance between values of the event scheduling parameters for a particular class of simulation experiments can only be determined by direct experimentation.

6.2.12.5 PDL

See Appendix A.

6.2.12.6 Decision Tables and Algorithms

None.

6.2.13 EAFLAG - Auxiliary Output Request Processing

See subsection 6.1.5, EAFLAG.

6.2.14 EAFRPT - Termination Report

6.2.14.1 Identification

- o EAFRPT - Display Model Termination Status
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.14.2 Argument Dictionary

None.

6.2.14.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DAY	-	I*2	Calendar day
HOUR	-	I*2	Hour of day
MIN	-	I*2	Minute of hour
MONTH	-	I*2	Month of year
NOV1	-	I*4	Data accumulation
NOV2	-	I*4	Data accumulation
TCAP	-	I*4	Total capacity
TIME	-	I*4	Termination time in seconds
TYPE	10	I*2	Alpha station link designations
YEAR	-	I*2	Year

6.2.14.4 Description

Upon entry, EAFRPT calls DAYTIM to acquire the current date. The termination status of the simulation is formatted and displayed in report form as follows:

- o General system statistics related to vehicle and passenger activity
- o Individual guideway link occupancy statistics
- o Scheduled service route statistics, if scheduled operation being modelled. If route groups have been defined, passenger summary statistics are displayed for each group.
- o Individual station performance statistics
- o Individual station link occupancy statistics displayed for each station

6.2.14.5 PDL

See Appendix A.

6.2.14.6 Decision Tables and Algorithms

None.

6.2.14a EAFTRN - Push Couple Vehicle Entrainment

6.2.14a.1 Identification

- o EAFTRN - Push Couple Vehicle Entrainment
- o GM TSC - July 1, 1981
- o PARAFOR

6.2.14a.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
V1	-	I*4	Lead vehicle ID of first train
V2	-	I*4	Lead vehicle ID of second train

6.2.14a.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DDIRST	-	I*4	Destination directional station of push vehicle
DSTN	-	I*4	Destination station of push vehicle
LASTV	-	I*4	Vehicle ID of last vehicle in first train
NUM	-	I*4	Number of vehicles in second train
OSTN	-	I*4	Origin station of push vehicle
RTE	-	I*4	Route assignment of second train
SL	-	I*4	Station link number
TIME	-	R*4	Current time in seconds
TRL	-	I*4	Train length parameter

6.2.14a.4 Description

This routine performs the entrainment of a train headed by a degraded vehicle to a following train during push coupling recovery from a vehicle failure. The entrainment process involves extending the length of an already existing train or establishing a new train from two consecutive unentrained vehicles. Thus, a single vehicle or train may be entrained to another single vehicle or train establishing a new train as described in Section 6.2.45.4. This is accomplished by setting all of the appropriate

linkage pointers and length fields for each member of the new train as described in Section 6.2.45.6.

In addition, for scheduled service, the second train is removed from the scheduled service route count (it will return after pushing is completed). For timeout/group demand responsive service, the second vehicle is removed from the inventory counts of its destination station and the affected inventory parameters are recalculated.

6.2.14a.5 PDL

None.

6.2.14a.6 Decision Tables and Algorithms

See Section 6.2.45.6.

6.2.15 EAGCTL Guideway Link Modeling Control

6.2.15.1 Identification

- o EAGCTL - Guideway Link Modeling Control
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.15.2 Argument Dictionary

None. Included segment in EMODEL.

6.2.15.3 Local Variable Dictionary

None.

6.2.15.4 Description

EAGCTL provides the architectural interface for directing event processing associated with vehicles traversing the automated guideway portion of the network. This control involves invoking model processing components which perform a fixed set of macro events associated with moving vehicles along automated guideway links and performing the transition between guideway traversal and network station entry. The control process is initiated by the main routine when a vehicle transaction requiring guideway link event processing is removed from the FEL.

Processing begins by invoking the guideway link model (EGLMDL) for processing the required model level event (micro event) for the vehicle transaction. Upon completion of model event processing, a determination is made as to whether all required processing for the current guideway link assigned to the vehicle transaction has been completed. If all events in the model associated with link traversal have been completed, the next network entity (guideway or station) determination process is invoked for the vehicle (EGNEXT). Otherwise control returns to the control loop of the simulator for processing of the next available transaction from the FEL.

Once the next entity has been determined, it is checked for availability by invoking either station (ESTEST) or guideway (EGNEXT) link next

entity entry testing. A station entity is indicated by a negative value returned by EGNEXT, and a guideway link by a positive value. If the next entity is available, as indicated by checking an entry flag returned by the test routine, vehicle link exit processing is performed (EGLEAV). The guideway link model (EGLMDL) or station model (ESLMDL) is then invoked, as required, to perform initial event processing and scheduling on the FEC of entry event completion for the vehicle.

If next entity availability checking results in the vehicle being precluded from entry to its next entity, transaction queuing is invoked (EGLWIQ) for the vehicle. Subsequent vehicle processing is then delayed until a prompt or dequeue event is scheduled as the result of the other modeling actions which alleviate the source of congestion. Control is then returned to the simulator control loop. In the case of station entry failure an additional check is made to determine if alternate station assignment is possible. As the result of station entry testing, the next entity for the vehicle can be changed to indicate the guideway link immediately downstream of the station entry, if an alternate station assignment is possible. If this entity change occurs, the guideway link is tested for entry availability by invoking EGTEST. If entry can be performed, EGASTN is invoked to assign the alternate station to the vehicle and the guideway link model is invoked to perform event processing and scheduling.

6.2.15.5 PDL

See Appendix A.

6.2.15.6 Decision Tables and Algorithms

None.

6.2.16 EAINDX - Update Index File

6.2.16.1 Identification

- o EAINDX - Decode Parm Field and Update Index
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.16.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
COUNT	-	I*2	Number of characters contained in PARM field string
STRING	2	I*1	PARM field character string

6.2.16.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
BLANK	-	L*1	EBCDIC blank constant
CHAR	-	I*2	Character variable for comparison against COMMA
COMMA	-	I*2	EBCDIC COMMA constant
DAY	-	I*2	Day of month
HOUR	-	I*2	Hour of day
PTR	-	I*2	Index into STRING array
YEAR	-	I*2	Year

6.2.16.4 Description

EAINDX is called during initialization to decode PARM field information and write the initial entries reflecting MP execution to the Run Index file. The routine is invoked via an indirect call to entry point

EAUPTX in routine EANTIX which initially saved the address of the PARM information, after receiving control from the Operating System. EAUPTX **retrieves** the address of the PARM field data and its length, and passes them as calling arguments to EAINDX.

Upon entry, EAINDX decodes the PARM field information containing input and output file member names and saves the data for use in updating the Run Index file. The current date and time is then obtained by invoking DAYTIM. The initial entries containing date and time, processor ID, and load module name are then written to the Run Index.

During system termination the routine is invoked via entry point EAWTIX to update the Run Index file with entries reflecting output files created during MP execution. A summary of the Run Index information is also displayed on the system output device.

6.2.16.5 PDL

See Appendix A.

6.2.16.6 Decision Tables and Algorithms

None.

6.2.17 EAINIT - System Initialization

6.2.17.1 Identification

- o EAINIT - Model Processor System Initialization
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.17.2 Argument Dictionary

None.

6.2.17.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
CU	-	I*4	Transaction scheduling time in C.U.
DATA	-	I*4	EBCDIC constant 'DATA'
EOF	-	L*1	End of file indicator
FLAG	-	I*4	EBCDIC constant 'FLAG'
NAME	-	I*4	I/O variable containing asynchronous input command identifier
OPTI	-	I*4	EBCDIC constant 'OPTI'
PARA	-	I*4	EBCDIC constant 'PARA'
REST	-	I*4	EBCDIC constant 'REST'
SELE	-	I*4	EBCDIC constant 'SELE'
TIME	-	R*4	I/O variable containing trip arrival time in seconds
TIMEA	-	R*4	I/O variable containing asynchronous input command time in seconds

6.2.17.4 Description

EAINIT is invoked by the main routine, EMODEL, to establish initial conditions for the simulation system. Upon entry, the interface to the traceback (interrupt handling) facility is established via a call to entry point SPIEL in routine TRACBK. The effect of this call is to define the system level interrupts which are to be excluded from Operating System control and processed by the internal DESM traceback facility. The interrupt specification passed via the call to SPIEL results in any system level interrupt being intercepted and processed by the traceback facility.

Once the interrupt handling interface has been established, initialization proceeds by defining the system status area Addressability.

The System Status Area is the region within core memory which contains all global common area data used by the simulation system. Within this area, all common areas reside in contiguous core locations beginning and ending with symbolic names which are converted to physical memory addresses by defining dummy control sections (CSECTS) to the linkage editor. The physical addresses defined in this manner delineate the data associated with Network and System Characteristics, as defined within the IP and communicated via the AGT data base, and the checkpoint region which must be saved to accomplish subsequent system restart. The structuring of the system status area is shown in Figure 6-3. During this phase of system initialization, EANTSA is invoked to establish **FORTTRAN** addressability to each region within the System Status Area.

The first asynchronous data command is then read to determine if the simulation experiment is to be restarted from a previously recorded checkpoint. If required, a system restart is performed by invoking EAREST and subsequent steps of the initialization process are bypassed. Otherwise initialization proceeds by performing the following:

1. EANDTA is called to initialize the System Characteristics and Network data within the System Status region.
2. Zero time data and auxiliary output requests are processed from the asynchronous data input file to complete system definition processing.
3. Transaction data associated with vehicles, trips, and system service requests are initialized by invoking EANXTN.
4. EANFEL is called to establish the Future Events timing mechanism for the simulation run.

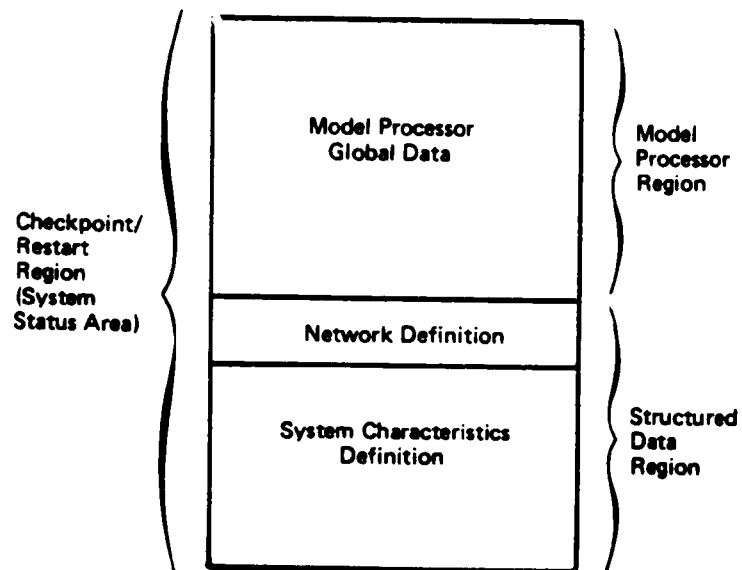


FIGURE 6-3. SYSTEM STATUS AREA ORGANIZATION

5. Sampling data initialization is performed by invoking EAZNIT.
6. System level events are initialized by the initial scheduling of trip arrivals, and periodic processes within the simulation system. The first trip record is read from the demand file and scheduled on the FEL for processing at its recorded arrival time. If periodic sampling, checkpoint or computation is required, necessary system service request transactions are obtained and scheduled as required to invoke processing at the end of the first periodic processing interval.
7. Modelling subsystem initialization is performed by calling EANMDL.
8. The first asynchronous simulation event is scheduled on the FEL.
9. EAUPTX is invoked to write the initial entries to the Run Index file.

6.2.17.5 PDL

See Appendix A.

6.2.17.6 Decision Tables and Algorithms

None.

6.2.17a EATIVEH - Initialize New Vehicles

6.2.17a.1 Identification

- o EATIVEH - Initialize New Vehicles
- o GM TSC - July 1, 1981
- o PARAFOR

6.2.17a.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
RTE	-	I*4	Route assignment for new vehicles
NUM	-	I*4	Number of new vehicles to initialize

6.2.17a.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
CAP	-	L*1	Flag indicating storage link capacity has been reached
NUMV	-	I*4	Number of vehicles initialized
STN	-	I*4	Station ID of maintenance barn
VEH	-	I*4	Transaction ID for new vehicle

6.2.17a.4 Description

This routine initializes the user input number of transition vehicles for scheduled service active fleet size management for each route at the calculated closest barn to any station on the route. The vehicles are placed on the barn's storage link awaiting formation of trains and subsequent launch. If insufficient storage is available only as many vehicles as will fit are initialized on the storage link and an informational message is generated. For scheduled service failed vehicle replacement, one vehicle is initialized at the closest barn to any station on the route. Overall station, station link, and storage statistics are updated.

6.2.17a.5 PDL

None.

6.2.17a.6 Decision Tables and Algorithms

None.

6.2.18 EANDR - Demand Responsive Service Initialization

6.2.18.1 Identification

- o EANDR - Initialize for Demand Responsive Operation
- o IBM/FSD - May 1, 1978 modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.18.2 Argument Dictionary

None.

6.2.18.3 Local Variable Dictionary

<u>VARIABLE</u>	<u>DIM</u>	<u>TYPE</u>	<u>DESCRIPTION</u>
PRMT	-	L*1	Flag indicating no headway event at station entry
VEH	-	I*4	Vehicle ID index

6.2.18.4 Description

EANDR performs initial fleet definition for Demand Responsive service. This involves the placement of vehicles in station storage areas and on guideway links at the approach to network stations.

Processing begins by cycling through each defined station in the network and for each station requiring initial vehicle allocation; obtaining the necessary vehicle transactions and queuing them in the station storage area. The required vehicle allocation is given by the initial occupancy value specified for the station storage link. Each vehicle queued in the storage area is assigned storage status (ZVSTAT = 3).

Once station storage has been initialized, the guideway approach links to stations are checked for initial occupancy requirements. If vehicles are required on a given link, the specified number of vehicle transactions are obtained and placed in the link exit queue. Each vehicle is assigned dead-heading status (ZVSTAT = 2) with its next destination being the station immediately downstream of the guideway link. As initialization is completed for a link, a prompt system transaction is scheduled for zero time to dequeue the first vehicle and begin movement into network stations.

After fleet definition has been accomplished, the first empty vehicle selection option (PVSPR(1)) is examined to determine if the earliest arriving vehicle is to be assigned for initial trip servicing. If this is the case, the empty vehicle selection list is modified to contain the following prioritized selection options:

1. Upstream in station
2. Local storage
3. Arrival list of station
4. Regional storage

6.2.18.5 PDL

See Appendix A.

6.2.18.6 Decision Tables and Algorithms

None.

6.2.19 EANFEL - Future Events List Initialization

6.2.19.1 Identification

- o EANFEL - Initialize Future Events List
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.19.2 Argument Dictionary

None.

6.2.19.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
INFINY	-	I*4	Largest integer full word value

6.2.19.4 Description

Upon entry, the multiple thread portion of FEL is established by obtaining a system transaction, assigning it an infinite time value, and chaining the transaction to itself to form a FIFO queue list. Another system transaction is obtained and also assigned an infinite time value with an event specification of 10, indicating system termination. This transaction is inserted in the transaction chain for the infinite time multiple thread loop, just created, to provide guaranteed termination of the simulation system in the absence of user specification. The clock table portion of the FEL is then zeroed as are the statistics associated with event scheduling and the value of CLBIG, specifying the size of the clock table in C.U., is computed.

6.2.19.5 PDL

See Appendix A.

6.2.19.6 Decision Tables and Algorithms

The length of the clock table is given by $CLBIG = CLSMAL * CLSIZE$

Where:

$CLSMAL$ = clock table interval in C.U.

$CLSIZE$ = number of entries in the clock table.

6.2.20 EANMDL - Modeling Subsystem Initialization

6.2.20.1 Identification

- o EANMDL - Modeling Subsystem Initialization
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.20.2 Argument Dictionary

None.

6.2.20.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
FOUND	-	L*1	Logic variable for indicating point station defined
FOUND1	-	L*1	Logic variable indicating input ramp has board or deboard event

6.2.20.4 Description

The global status data associated with guideway links, stations and station links is initialized. A determination is then made as to whether vehicle arrival list recording is required during the modeling experiment to satisfy initial vehicle assignment for arriving trips. If reservations are required or an assignment option (PVSPR) requiring knowledge of arriving vehicles is found, a global arrival list recording indicator (VALIST) is set to true.

Since local station storage is required for initial vehicle placement, the station configuration is checked to ensure proper storage definition and connectivity exists. If a station storage link has been explicitly defined, the upstream and downstream connectivity pointers are traced to verify that explicit paths into and out of the storage area have been defined. If these paths are not found, as identified by the absence of an input to store, dock to store, store to input, or store to output link, a warning message is issued. In the case of Demand Responsive Service, this message indicates potential problems exist for empty vehicle distribution. If any option requires the distributions of empty vehicles to either local or regional storage, the absence of paths from input to store or dock to store will produce unpredictable simulation

results. Similarly, the absence of paths from storage can cause failure of the vehicle assignment process for arriving trips, depending upon the options specified by the user. In the case of Scheduled Service, the message indicates a potential problem with initialization. Since initial vehicle scheduling on assigned routes involves movement from storage to the station docking areas, the absence of path from storage to input will result in unpredictable simulation results.

If no explicit definition of a storage link is found, processing is performed to implicitly define a fully connected pseudo storage link to satisfy initialization requirements. The use of this link is transparent to the simulation user and adds no additional processing time. Each link defined to establish the pseudo storage area is assigned an initial capacity of 100 and a zero time headway event.

Once station storage has been verified or established, all vehicle modeling status data is initialized to zero. The estimated time of arrival containing the guideway link traversal time between O/D pairs in the network is computed for use in the arrival list recording process. Service type initialization is then performed by invoking EANDR or EANSCD or EANMRG to perform fleet size definition for the simulation run. Finally, routine EANRPT is invoked to display simulation initial conditions.

6.2.20.5 PDL

See Appendix A.

6.2.20.6 Decision Tables and Algorithms

None.

6.2.20a EANMRG - Timeout/Group Demand Responsive Service Initialization

6.2.20a.1 Identification

- o EANMRG - Timeout/Group Demand Responsive Service Initialization
- o GM TSC - July 1, 1981
- o PARAFOR

6.2.20a.2 Argument Dictionary

None.

6.2.20a.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
NDP	-	I*4	Directional platform ID
NDS	-	I*4	Directional station ID
SL	-	I*4	Station link ID
STN	-	I*4	Station ID
V	-	I*4	Number of vehicles
VEH	-	I*4	Vehicle transaction ID

6.2.20a.4 Description

This routine performs service initialization for the timeout/group demand responsive service mode. The required number of vehicles are initialized and placed on the dock links of each station. The first vehicle on each dock link is placed at the load berth and subsequent vehicles are placed at the most downstream unload berths. Then inventory control parameters are calculated for each directional station and platform based on the initial vehicle placements.

6.2.20a.5 PDL

None.

6.2.20a.6 Decision Tables and Algorithms

None.

6.2.21 EANRPT - Initial Conditions Report

6.2.21.1 Identification

- o EANRPT - Display Initial Conditions Report
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.21.2 Argument Dictionary

None.

6.2.21.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
CONV1	-	R*4	Time in seconds
CONV2	-	R*4	Time in seconds
CONV3	-	R*4	Time in seconds
CUSEC	-	R*4	Clock units/second
DAY	-	I*4	Day of month
DONE	-	L*1	Logic variable for processing complete
END	-	I*4	Output control
FNOV	-	R*4	Station link headway in seconds
FNOV1	-	R*4	Station link headway in seconds
FNOV2	-	R*4	Station link travel time in seconds
HOUR	-	I*4	Hour of day
MIN	-	I*4	Minute of hour
MONTH	-	I*4	Month of year

VARIABLE	DIM	TYPE	DESCRIPTION
NOV1	-	I*4	Occupancy total
NOV2	-	I*4	Capacity total
NUM	-	I*4	Number of defined station links
NUMO	-	I*4	Output control

6.2.21.4 Description

Upon entry, DAYTIM is called to acquire the current date and time which is displayed on the initialization report. The starting conditions for the simulation are then formatted and displayed in report form as follows:

- o General simulation characteristics and options
- o Scheduled Service route definitions, if scheduled operation being modeled
- o Individual guideway link characteristics
- o Generalized station configuration summary and event timing parameters
- o Specific link configuration by station

6.2.21.5 PDL

See Appendix A.

6.2.21.6 Decision Tables and Algorithms

None.

6.2.22 EANSAV - Establish Status Area Addressability

6.2.22.1 Identification

- o EANSAV - Establish Status Area Addressability
- o IBM/FSD - May 1, 1978
- o Assembler

6.2.22.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
A	-	I*4	Address of System Status Area
B	-	I*4	Length of System Status Area
C	-	I*4	Address of System Characteristics
D	-	I*4	Length of System Characteristics
E	-	I*4	Address of Network definition data
F	-	I*4	Length of Network data
ARGA	-	I*4	Parameter equivalenced to beginning of System Status Area
ARGB	-	I*4	Length of System Status Area in words
ARGC	-	I*4	Parameter equivalenced to beginning of System Characteristics data
ARGD	-	I*4	Length of Systems Characteristics data
ARGE	-	I*4	Parameter equivalenced to beginning of Network data
ARGF	-	I*4	Length of Network data in words

6.2.22.3 Local Variable Dictionary

None.

6.2.22.4 Description

EANSAV provides direct **FORTRAN** addressability to the System Status Area. The routine is invoked during model initialization to retrieve and pass as calling arguments the address and length of the System Status Area to EACKR and the System and Network Characteristics region to EADADD. The effect of these calls is to establish an equivalence between locally defined **FORTRAN** arrays, used for I/O in the routines, and the System Status Area.

6.2.22.5 PDL

See Appendix A.

6.2.22.6 Decision Tables and Algorithms

None.

6.2.23 EANSCD - Scheduled Service Initialization

6.2.23.1 Identification

- o EANSCD - Initialize Scheduled Service Routes
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.23.2 Argument Dictionary

None.

6.2.23.3 Local Variable Dictionary

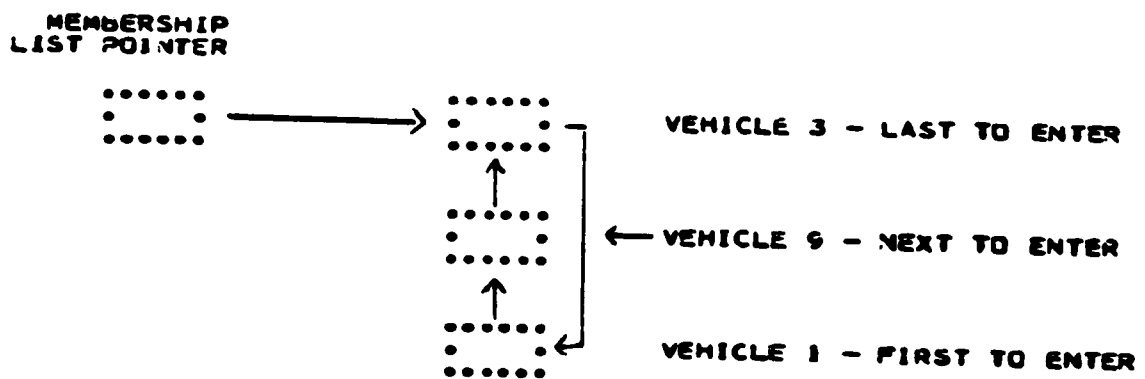
VARIABLE	DIM	TYPE	DESCRIPTION
ENTRY	-	I*4	Pointer into scheduled route list
EVP	-	I*2	Pointer to station link event list
FIRSTV	-	I*4	ID of first vehicle initialized at a Scheduled Service station
FOUND	-	L*1	Logical variable for launch event located
NUMV	-	I*2	Number of vehicles initialized at a Scheduled Service station
NVL	KMS	I*2	Number of trains scheduled for station departure
SL	-	I*2	Station link ID used in tracing station path
TIMEX	-	I*2	Station time from storage to launch processing
V	-	I*2	Loop counter - vehicle in train

6.2.23.4 Description

Prior to initializing the vehicles on each scheduled route, the time from station storage to the beginning of the launch event is computed. This time is used to adjust the scheduled dispatch times on each route to account for vehicle movement from storage to the launching area of the station. This adjustment is required since the schedule, as computed by the IP, is based upon immediate vehicle dispatch from network station assuming a steady state condition.

Once the schedule adjustment time has been computed, initialization proceeds for each defined route by initializing the required number of vehicles to be dispatched from each station contained in the route sequence as follows:

1. A number of vehicle transactions sufficient to satisfy train length on the route are obtained and assigned to storage in the station
2. If trains are required on the route, the vehicle transactions obtained for the current group are coupled to form an entrained sequence. Single vehicles are entrained to themselves and assigned a train length of one. Trains created during this process remain static for the length of the simulation run.
3. The single vehicle or lead vehicle of the entrained group is placed in the station storage membership queue in time sequence order. This involves tracing the chain of vehicles already assigned link membership until the proper time position for the current vehicle or train is found, and inserting the single vehicle or lead vehicle of the entrained group into the membership list. The membership list for a station link is structured as FIFO queue in the following manner:



Where:

1. The membership list pointer contains the ID of the last vehicle to enter
2. The chain pointer for the last vehicle points to the first
3. The chain pointer of subsequent vehicles in the list point to the next vehicle in a circular manner
4. The single vehicle or lead of the entrained group is scheduled on the FEL for movement from the storage area to the dock at its required departure time

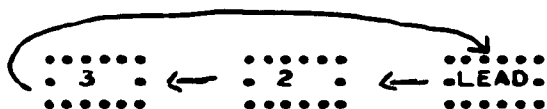
6.2.23.5 PDL

See Appendix A.

6.2.23.6 Decision Tables and Algorithms

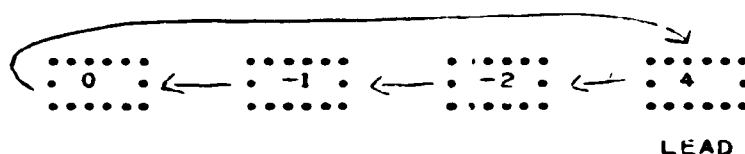
A) Vehicle entrainment

Vehicles are grouped to form trains via a chain word or pointer and a train length field. The chain word for each vehicle in the entrained group points to the following vehicle of the train, with the last vehicle pointed to the lead to form a circular chain as follows:



The train length assigned to each vehicle in the group reflects the number of following vehicles in the train. This value is positive for the lead vehicle of the group and reflects the train length assigned to the scheduled service route. The train length for intermediate vehicles of the train are maintained as negative integers. For example, an entrained

group of four vehicles would contain the following train length relationship:



Where:

- (1) $L = V_N - M_{TR}$ and
 V_N = relative train position
 M_{TR} = train length assigned to route

B) Entrained group occupancy requirements

In addition to the train length maintained for the lead vehicle of an entrained group, a block occupancy count is maintained. This count is used during guideway movement to determine the number of occupancy positions required to accommodate the train based upon the position regulation scheme in effect. In variable block mode, this count is equal to the train length. However, the fixed block occupancy requirement for the train is computed as

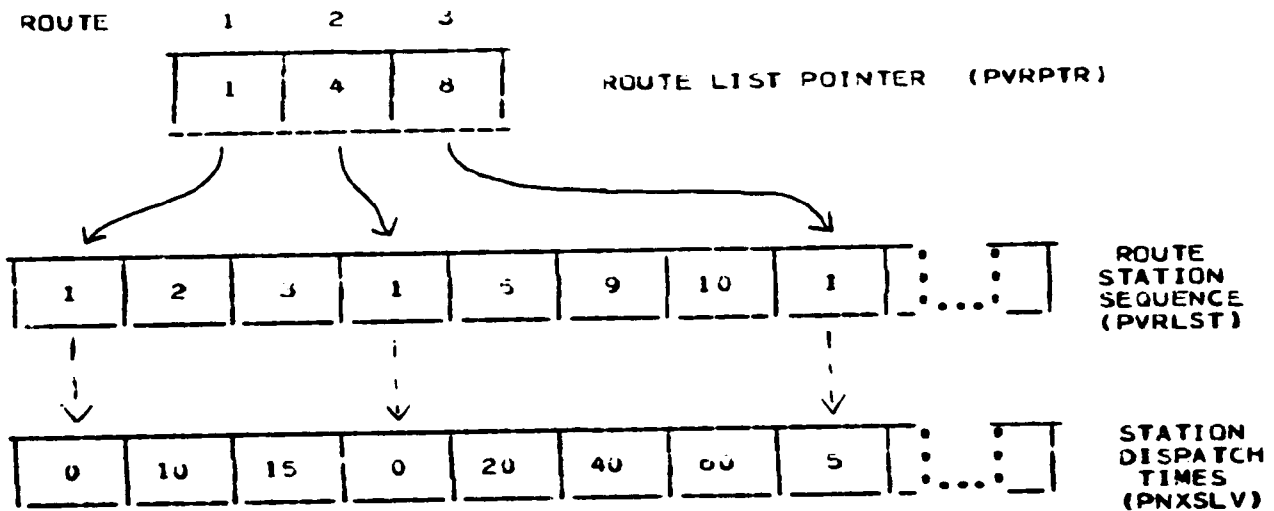
$$(2) T_0 = \frac{(T_L - 1) \times V_L}{B_L} + 1 \quad (\text{rounded up})$$

Where:

- T_L = train length or number of vehicles in the entrained group
 V_L = vehicle length (meters) and
 B_L = block length (meters).

C) Scheduled Service Dispatch

The initial dispatch times for each station by route are computed by the IP as a sequential list organized as follows:



As Vehicles are initialized for each station assigned to the route, the dispatch time of the vehicle is computed as:

$$(3) T_0 = \text{PNXSLV}_S + (N_{RS} \times H_R)$$

where:

PNXSLV_S = as defined above

N_{RS} = number of vehicles or trains previously dispatched from station for current route

H_R = Assigned route headway

After all vehicles have been initialized for a particular station on a route, the initial dispatch time given by PNXSLV is adjusted to reflect the time for vehicle movement from

storage to the beginning of launch processing. Thus, the original schedule computed by the IP is maintained and not affected by the model initialization process.

6.2.24 EANTIX - Save PARM Field Information

6.2.24.1 Identification

- o EANTIX - Save PARM Field Information
- o IBM/FSD - May 1, 1978
- o Assembler

6.2.24.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ARG1	-	I*2	(OUTPUT) Length of PARM Field in bytes
ARG2	-	L*1	(OUTPUT) PARM Field Character String

6.2.24.3 Local Variable Dictionary

None.

6.2.24.4 Description

EANTIX receives control from the Operating System and saves the address of the PARM field information coded on the EXEC JCL statement. The address is passed to the routine via standard linkage in general purpose register 1. Control is then passed to the MP main routine, EMODEL, to begin the simulation process.

Entry point EAUPTX is invoked by EAINIT during initialization to retrieve the saved PARM field information for decoding and use in updating the Run Index file via EAINDX. Return from the entry point to EAINIT is accomplished via non-standard linkage in order that a direct control transfer occur between EAINDX and EAINIT after PARM field decoding.

6.2.24.5 PDL

See Appendix A.

6.2.24.6 Decision Tables and Algorithms

None.

6.2.24a EANTRN - Entrain and Launch Scheduled Service AFSM Vehicles

6.2.24a.1 Identification

- o EANTRN - Entrain and Launch Scheduled Service AFSM Vehicles
- o GM TSC - July 1, 1981
- o PARAFOR

6.2.24a.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
RTE	-	I*4	Route ID to launch on
STN	-	I*4	Station ID of launching maintenance barn

6.2.24a.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
FIRSTV	-	I*4	Vehicle ID of first vehicle in train
NUMV	-	I*4	Number of vehicles in train
NUMVEH	-	I*4	Number of stored vehicles assigned to route
PRIOR	-	I*4	Vehicle ID of previous vehicle in train
TIME	-	R*4	Clock time in seconds
TRL	-	I*4	Train length for follower vehicles
TRN	-	I*4	Train length on route
V	-	I*4	Vehicle ID looping through stored vehicles (outer loop)
VEHN	-	I*4	Temporary vehicle ID
VEHNXT	-	I*4	Next vehicle ID looping through stored vehicles (outer loop)

6.2.24a.4 Description

This routine entrains and schedules for launch, vehicles from maintenance barn storage areas to accomplish reconstituting and service increases during scheduled service active fleet size management. If

sufficient vehicles are stored and assigned to the route, a train is formed by setting the appropriate linkage pointers and length fields for each member of the new train as described in Section 6.2.45.6. If more than one route uses the station as a maintenance barn some vehicles may be switched from one route to another in order to allow the new train to form from the most downstream vehicles. However, for every downstream vehicle in storage switched from route A to route B, an upstream vehicle in storage is switched from route B to route A. Storage and route statistics are updated and the new train is prompted to begin movement out of the storage link to its next station stop on the route.

6.2.24a.5 PDL

None.

6.2.24a.6 Decision Tables and Algorithms

See Section 6.2.45.6.

6.2.25 EANTSA - Initialize System Status Area Addresses

6.2.25.1 Identification

- o EANTSA - Initialize System Status Area Addresses
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.25.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ACORE(1)		I*4	(OUTPUT) Beginning address System Status Area
(ACORE(4)-ACORE(1))/4		I*4	(OUTPUT) System Status Area Length
ACORE(2)		I*4	(OUTPUT) Address System Characteristics
(ACORE(3)-ACORE(2))/4		I*4	(OUTPUT) System Character- istics Length
ACORE(3)		I*4	(OUTPUT) Address Network Definition
(ACORE(4)-ACORE(3))/4		I*4	(OUTPUT) Network Data Length

6.2.25.3 Local Variable Dictionary

None.

6.2.25.4 Description

EANTSA contains a FORTRAN common area which is automatically equivalenced during Link Edit to the identically named CSECT, ESASAV, which contains the address constants for each region in the System Status Area. Since the length of these regions and their addresses are not directly accessible via FORTRAN, EANTSA invokes the assembler

language routine EANSV to establish direct System Status Area addressability. This addressability is established by retrieving the addresses within the equivalenced common area and the associated length of each region, and passing them as calling parameters to EACKR and EADADD to define the checkpoint/restart and data initialization regions, respectively.

6.2.25.5 PDL

See Appendix A.

6.2.25.6 Decision Tables and Algorithms

See EAINIT for description of System Status Area.

6.2.26 EANXTN - Initialize Transaction Data

6.2.26.1 Identification

- o EANXTN - Zero Transaction data and Establish Available Chains
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.26.2 Argument Dictionary

None.

6.2.26.3 Local Variable Dictionary

None.

6.2.26.4 Description

EANXTN initializes the transaction data associated with vehicles, trips and system service requests. The data associated with each transaction is zeroed and a pointer relationship is established between successive transaction entries by chaining them to one another. The transactions are then allocated by class, in response to requirements contained in the System Characteristics data. This allocation involves establishing the available list pointers by transaction type. The first unused transaction ID is pointed to by the available list and the chain pointer of the last allocated transaction is zeroed to establish the end of the available chain. These available lists are used during model initialization and simulation processing for acquisition, definition and maintenance of vehicles, trips and system service requests.

6.2.26.5 PDL

See Appendix A.

6.2.26.6 Decision Tables and Algorithms

None.

6.2.27 EAPCMP - Periodic Computation Processing

6.2.27.1 Identification

- o EAPCMP - Invoke Periodic Algorithmic Computation
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.27.2 Argument Dictionary

None.

6.2.27.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DDIRST	-	I*4	Destination directional station ID
DDPA	-	I*4	Destination directional platform A ID
DELAY	-	I*4	Sum of travel times along tow path
DEN1	-	R*4	Traffic density on guideway link 1 approaching merge
DEN2	-	R*4	Traffic density on guideway link 2 approaching merge
DIRSTN	-	I*4	Destination directional station ID
DSTN	-	I*4	Destination station ID
EDIRST	-	I*4	Closest eligible origin directional station ID
ESTN	-	I*2	Closest eligible origin station ID
FOUND	-	L*1	Flag indicating search object found
FOUND2	-	L*1	Flag indicating secondary search object found
KLNK	-	I*4	Link ID of merging link
LASTV	-	I*4	Last vehicle ID in first train
LINK	-	I*4	Guideway link ID
LINKC	-	I*4	Merging guideway link ID

VARIABLE	DIM	TYPE	DESCRIPTION
LNK	-	I*4	Link ID of two path link
MIN	-	I*4	Minimum nominal travel time found for eligible origin station
N	-	I*4	Number of vehicles to initialize
NUM	-	I*4	Length of following train
ODIRST	-	I*4	Origin directional station ID
OSTN	-	I*4	Origin station ID
RTE	-	I*4	Route ID
SIDECH	-	L*1	Flag indicating side channel inventory request
SL	-	I*4	Station link ID
SLD	-	I*4	Station link dock ID
SLP	-	I*4	Station link pointer to upstream link table
STN	-	I*4	Station ID
TIM	-	I*4	Event scheduling delta time in clock units
TIME	-	R*4	Current simulation clock time in seconds
TRL	-	I*4	Train length for follower vehicles
TRNID	-	I*4	Vehicle request transaction ID
ULNK	-	I*4	Upstream link ID
V	-	I*4	Vehicle ID - loop counter
VEH	-	I*4	Vehicle ID of lead vehicle in first train
VEHN	-	I*4	Temporary vehicle ID
VEH2	-	I*4	Vehicle ID of lead vehicle in second train
XSTN	-	I*2	Station ID of ready vehicle request
XTN	-	I*4	New inventory request transaction

6.2.27.4 Description

EAPCMP provides the ability to invoke algorithms which modify modeling status based upon simulation conditions on a time dependent basis. The algorithm to be executed is indicated by the system service request transaction processed by the main routine to invoke the periodic computation process. The algorithms included in EAPCMP are heuristic merge delay computation, failure detection event processing, failure restart event processing, failure replacement vehicle initialization, timeout/group demand responsive maximum passenger wait time event processing, timeout/group demand responsive inventory management processing. An interface to an optional user periodic computation algorithm is provided by the ability to invoke routine EUPCMP.

The heuristic merging of vehicle strings, in the DESM, is accomplished by applying a delay factor to the travel time of vehicles as they enter the links upstream of a network merge. The delay factor to be applied is selected from a merge delay table, input by the user, based upon the density of traffic approaching the network merge. All vehicles traversing the approach links are subject to the selected links travel time adjustment until the next periodic computation interval when a new merge delay factor is computed. The computation process is performed for each guideway link which is immediately upstream of a network merge. The density of traffic in vehicles/unit length of guideway on both the link and the opposing link entering the merge are computed as follows:

(1) Fixed block regulation

$$D = ((O_L - Q_L)/(L_L - (B_L \times Q_L)))/(L_V \times L_H)$$

Where:

O_L = occupancy of link

Q_L = occupancy of link queue (stopped vehicles at link exit)

L_L = length of link in meters

B_L = fixed headway block length in meters

Q_L = occupancy of link queue

L_V = link velocity in m/sec

L_H = headway time in sec

(2) Variable block regulation

$$D = ((O_L - Q_L)/(L_L - (V_L \times Q_L))$$

Where:

O_L , Q_L , L_L , L_V and L_H , as defined above

V_L = vehicle length in meters

These density factors are converted to table indices for selecting the appropriate delay factor for each link as GLMDLY (I,J) where:

$$(3) \quad I = D_1 \times \text{KMDLY1} + .5$$

Where:

D_1 = density on first link as defined above

KMDLY1 = number of rows in merge delay table

$$(4) \quad J = D_2 \times \text{KMDLY2} + .5$$

Where:

D_2 = density on opposing link as defined above

KMDLY2 = number of columns in merge delay table

At the time of failure detection, member EAPCMP2 of routine EAPCMP is invoked and performs the following functions:

- o Activates the new minimum path provided the push coupling failed vehicle response is not chosen
- o Initiates the other vehicle responses by marking each affected vehicle via the appropriate global parameter
- o For scheduled service, removes the failed train from the route service count
- o For timeout/group demand responsive service, the failed vehicle is removed from the inventory counts of the destination directional station corresponding to a turnaround.
- o If the vehicle can restart, schedules the restart event after the user-input time delay.
- o If push coupling specified, looks to see if another train is queued directly behind the failed train. If so, the entrainment is accomplished, the restart is scheduled after the user-input coupling delay, and the new minimum path table is activated. If not, the failed vehicle simply remains queued until the trailing vehicle arrives.
- o If tow by service vehicle specified, the tow path is processed to both check if it is now clear and close all other entrances to the tow path such as merges and off-line stations. In the case of on-line stations, one and only one path through the station is kept open and checked for being clear. If the path is now clear, the restart is scheduled at a time computed as the degraded travel time along the tow path plus the time (user-input) to get the tow vehicle ready and to the first link on the path.

- o Schedules the replacement vehicle to be initialized after a user-input time delay

At the time of vehicle restart (by whatever method), member EAPCMP3 of routine EAPCMP is invoked and performs the following functions:

- o Activates the next (or original if no other failures active) minimum path table
- o Terminates the other vehicle responses by unmarking each affected vehicle via the appropriate global parameter. Also, in the case of vehicles held at the dock, issues prompts to station links to restart the held vehicles.
- o Issues a prompt to the failed vehicle's link in order to restart the failed vehicle
- o If tow by service vehicle was specified, reopens the other entrances to the tow path and issues prompts to all links which had been closed in order to restart any vehicles which may have stopped

At the time of replacement vehicle initialization, member EAPCMP4 of routine EAPCMP initializes a replacement vehicle. For the demand responsive case, a new vehicle is initialized at the first station with available storage capacity, while, in scheduled service, a vehicle is initialized at the route's maintenance station and, if possible, launched on the route at that time.

At the time of maximum passenger wait event occurrence for timeout/group demand responsive service, member EAPCMP5 of routine EAPCMP is invoked to mark the vehicle request transaction as ready and invokes ESVREQ to try to find an available vehicle to service the trip group, if the group has not yet been served. If the trip has been served, the transaction is marked to indicate the maximum passenger wait event has occurred, or, if boarding is complete, the transaction is freed.

Member EAPCMP6 of routine EAPCMP is invoked to perform inventory management for timeout/group demand responsive service. If the destination station requires vehicles a search is performed to find the closest eligible origin, issue an inventory request, and, if required, issue a side channel request. If the origin and destination are at the same physical station, only the side channel request is required. A directional station is an eligible origin if:

- o Its inventory surplus is positive
- o There are no inventory requests for the same destination waiting to be assigned
- o Either the origin and destination dispatch toward each other, or the origin and destination are at the same physical station and the origin station-on-way count is positive, or the origin and

destination are at adjacent physical stations and dispatch in the same direction. If the shortage is at an A platform then the origin and destination must dispatch toward each other.

- o When a side channel request is required, all previous requests for the same side channel have been satisfied.

If an eligible origin station is found, an inventory request is either added to an existing vehicle request that is pending or a new inventory request is created. In either case, routine ESVREQ is then invoked to try to find a vehicle to service the inventory request.

6.2.27.5 PDL

See Appendix A.

6.2.27.6 Decision Tables and Algorithms

A. Merge Delay Table

The Merge Delay Table (GLMDLY) contains the travel time adjustment factors to be applied to vehicles approaching network merges in order to accomplish congestion free merging. The table is organized as a two dimensional array with each row and column representing a range of densities as follows:

LINK 2 DENSITIES

	0.0	0.2	0.4	1.0
0.0	0				
0.1					
0.2		2			
0.3					
⋮	⋮				
⋮					
⋮					
1.0					

INDICATES DELAY OF 2 ON LINK 1, GIVEN DENSITY RANGE, $.1 \leq R < .2$ ON LINK 1 AND DENSITY RANGE $.2 \leq R < .4$ ON LINK 2

LINK 1 DENSITIES

6.2.28 EAPFEL - Place a Transaction on the Future Events List

6.2.28.1 Identification

- o EAPFEL - Place a Transaction on the Future Events List
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.28.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
XTN	-	I*4	ID of transaction to be scheduled
GOTO	-	I*4	Next system event to be performed for transaction
DELTA	-	I*4	Delta time in C.U. until next event for transaction
PRTY	-	I*4	Priority order indicator (0 = highest, 9 = lowest)

6.2.28.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TIME	-	I*4	Maximum of new time of XTN and CLMINI
FMTHRD	-	I*4	Holds ID of current XTN when looping thru M/T
NMTHRD	-	I*4	Holds ID of next XTN when looping thru M/T
FIRST	-	I*4	Holds ID of current XTN when looping thru C/T
NEXT	-	I*4	Holds ID of next XTN when looping thru C/T

6.2.28.4 Description

EAPFEL performs the scheduling of a transaction on the future events list. EAPFEL is invoked by either the scheduling of a transaction via the SCHED macro or via a direct call. The transaction to be placed on the future events list is either placed in the clock table or on the multiple thread list depending upon whether the schedule time is within the current clock table interval or at some extended time in the future. Scheduling on the clock table involves finding the correct position for insertion and adding the transaction ID to the clock table. Multiple thread scheduling requires either the addition of the transaction to an existing multiple thread loop or the creation of a new multiple thread loop. Upon entry, FEL utilization statistics related to clock table and multiple thread usage are collected. The time position for placing the transaction on the FEL is as follows:

$$(1) T = \text{CLOCK} \times 10 + \text{PRTY}$$

Where:

CLOCK = current simulation time in C.U.

PRTY = scheduling priority

Thus, the priority indicator is used within the FEL to obtain a priority ordering of transactions which are scheduled for event completion at identical simulation times. The time of event occurrence is corrected upon transaction removal from the FEL as described in EARFEL.

6.2.28.5 PDL

See Appendix A.

6.2.28.6 Decision Tables and Algorithms

See 2.2.1.2 for description of Future Events List.

6.2.29 EAPLNK - Dequeue Vehicles from Upstream Guideway Links

6.2.29.1 Identification

- o EAPLNK - Dequeue on Guideway Vehicles
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.29.2 Argument Dictionary

None.

6.2.29.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
LINK	-	I*4	Link ID downstream of link(s) to be dequeued
LNK	-	I*4	Link ID of vehicle to be dequeued
LNK1	-	I*4	Link ID of vehicle to be dequeued
NPOS	-	I*4	Number of possible vehicles for dequeuing
POS	2	I*2	Vehicle ID's for possible dequeue
PSAVE	-	I*4	Temporary storage of vehicle ID
STATUS	2	I*2	Empty/Full status indicator for local merge priority ordering
SW	-	L*1	Logical variable for indicating vehicle order at local exit merge

6.2.29.4 Description

Upon entry, the type of dequeue event to be performed is determined based upon a type indicator associated with the System Service Request transaction which initiated the dequeuing process. If a prompt of a guideway link immediately upstream of a station is required, the link is checked for an unfailed exit and the existence of a queued vehicle.

If both conditions are satisfied, the queued vehicle is identified as a possibility and dequeue processing is performed. If a more typical prompt of upstream entities is required, regardless of type, the following is performed to identify the possible vehicles for dequeuing:

1. The guideway link or station exit ramp queues immediately upstream of the specified link are examined for the existence of queued or waiting vehicle transactions. If the exit of the guideway link or station is not failed, the transaction ID of the first vehicle present in each of the possible upstream queues (one in the case of a network diverge, two in the case of merge or intersection) is recorded for possible dequeuing and event scheduling.
2. Prior to attempting dequeue processing, a determination is made as to whether dequeuing should be performed or the re-ordering of vehicle possibilities is required. If only one possibility is found, a determination is made as to whether the vehicle is located on a guideway link approaching leg to a network merge. If this is the case and the opposing network entity is a guideway link, merge priority requirements are checked to determine if vehicle dequeuing must be rescheduled to allow a competing vehicle to traverse the merge first. If a vehicle is scheduled for immediate link exit on a prioritized opposing link, (travel completion), this rescheduling is performed and dequeue processing for the vehicle is terminated. In the case of a station being the opposing network entity, the simultaneous arrival of an exiting vehicle is checked to determine if local merge priorities must be examined. If a competing vehicle is scheduled for immediate arrival a station exit, priority vehicle determination is based upon a 2 x 2 local merge table which considers both occupancy status (empty vs. non-empty) and current vehicle location. The priority for each combination is defined by the simulation user. If the on guideway vehicle is determined to have priority, the dequeue process continues, otherwise vehicle dequeuing is rescheduled to allow merge traversal by the vehicle exiting the network station. Similar processing is performed if the vehicle possibility for dequeuing is currently queued at station exit and the opposing network entity is a guideway link.

If two possibilities for dequeuing exist, similar processing for the ordering of the queued vehicles is performed based upon guideway or local exit merge priority or FIFO merging or earliest arrival at merge requirements. The vehicle having priority or the earliest queued or arriving vehicle is placed first in the list of possibles. If

rescheduling of the dequeue process was not performed, dequeuing is attempted for the first possible vehicle by determining the next entity to which the vehicle is to move, and ensuring that entity is currently available. If vehicle entry is allowed, the vehicle is dequeued from the exit queue of its current guideway or station link, enters its next entity (guideway link or station) and is scheduled for the time of its first event completion on the next entity. If the vehicle cannot proceed to its next entity, it remains queued until a subsequent dequeue operation is scheduled in response to upstream vehicle movement or a failure recovery event. The dequeuing process is repeated for the second possibility since the prompting action may take place at a guideway intersection. In this case the competing vehicles could have different next entities and the movement of the first vehicle would not preclude entry of the next vehicle on its new entity.

The rescheduling of the dequeuing process results in a retry time attempt in zero time. This retry is attempted without clock advance, since simultaneously occurring events, still scheduled on the FEL at the time of retry determination, could preclude entity exit of the priority passage vehicle.

6.2.29.5 PDL

See Appendix A.

6.2.29.6 Decision Tables and Algorithms

None.

6.2.30 EAPRMT - Dequeue Control

6.2.30.1 Identification

- o EAPRMT - Dequeueing Control
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.30.2 Argument Dictionary

Included member EMODEL.

6.2.30.3 Local Variable Dictionary

None.

6.2.30.4 Description

EAPRMT is invoked by the main routine (EMODEL) in response to the occurrence of a dequeuing request scheduled by the Link or Station Model. EAPRMT invokes the appropriate prompting mechanism as required by the scheduled event type to resume vehicle movement on guideway links (EAPLNK) or stations (EAPSTN).

6.2.30.5 PDL

See Appendix A.

6.2.30.6 Decision Tables and Algorithms

None.

6.2.31 EAPSTN - Dequeue Vehicles from Upstream Station Queues

6.2.31.1 Identification

- o EAPSTN - Dequeue Vehicles from Upstream Station Queues
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.31.2 Argument Dictionary

None.

6.2.31.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
NPOS	-	I*4	Number of possible vehicles for dequeue
POS	6	I*4	Possible vehicles for dequeue
PSAVE	-	I*4	Temporary storage of vehicle ID
SL	-	I*4	Station Link ID

6.2.31.4 Description

Upon entry, a determination is made as to whether a specific station link or the upstream queues from a specified station link are to be prompted. If prompting of a specific station link is required, the specified link is examined for the existence of a queued vehicle. This situation arises due to the preemption of link completion within the **Station Model** caused by differences in event processing times for vehicles undergoing parallel event processing on the same station link. If a queued vehicle is found, it is recorded as a possibility for dequeue processing. The prompting of upstream links involves the examination of up to six upstream link queues for the existence of queued vehicles, depending upon the specific station configuration, and each queued vehicle is recorded as a possibility for dequeue processing. The list of possibles is then ordered based upon the station link ordering option specified the user. If priority ordering is required, the list remains the same since upstream queues are automatically examined

in priority sequence as listed by the I.P. If FIFO ordering is required, the possible vehicles are reordered by earliest queue entry time. After the list of possibles has been developed, dequeue processing is attempted for each potential vehicle. This process begins by determining the next entity for the vehicle by invoking ESNEXT. The next entity for the vehicle is then tested for entry availability (ESTEST - station link, EGTEST - guideway link). If entry is permitted, the appropriate model for processing the first event completion for the vehicle is invoked (EGLMDL - guideway link, ESLMDL - station link).

6.2.31.5 PDL

See Appendix A.

6.2.31.6 Decision Tables and Algorithms

None.

6.2.32 EARFEL - Obtain Next Most Imminent Transaction

6.2.32.1 Identification

- o EARFEL - Removes Next Most Imminent Transaction from FEL
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.32.2 Argument Dictionary

Included segment in EMODEL.

6.2.32.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
XMTHRD	-	I*4	Holds ID of next XTN when looping thru M/T
XFIRST	-	I*4	Holds ID of current XTN when looping thru C/T
XNEXT	-	I*4	Holds ID of next XTN when looping thru C/T

6.2.32.4 Description

The purpose of EARFEL is to obtain the next imminent event to be performed from the Future Events List and update the clock table and multiple thread list as necessary. A sequential scan of successive entries in the clock table, beginning with the currently active interval, is performed until a non-empty interval or the end of the clock table is reached. If a non-empty interval pointer is found, the first transaction chained within the interval is removed and returned as the currently active transaction requiring event processing. The time associated with the transaction is used to increment the simulation clock to the current event time. Since the priority indicator used in the original scheduling of the transaction is appended to the transaction time, the time is divided by 10 prior to updating the simulation clock. If the end of the table is reached during the scan, the first available multiple thread FEL list is removed from the multiple thread chain and reloading of the

FEL is performed. The base time value of the FEL is reestablished to the time of the multiple thread transaction.

Each transaction chained on the multiple thread list is removed and chained in time order within the clock table interval given by:

$$I = 1 + \frac{X_{TIME_{XTN}} - CLBASE}{CLSIZE}$$

Where:

$X_{TIME_{XTN}}$ = Scheduled time of transaction

$CLSIZE$ = Time value encompassed by a clock table interval

$CLBASE$ = Base time for the clock table.

Once loading of the clock table is complete, the first available transaction within the current table interval (first reloaded clock interval) is returned as the currently active transaction requiring event processing.

6.2.32.5 PDL

See Appendix A.

6.2.32.6 Decision Tables and Algorithms

None.

6.2.33 EARRPT - Restart Conditions Report

6.2.33.1 Identification

- o EARRPT - Display Restart Conditions
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.33.2 Argument Dictionary

None.

6.2.33.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
CONVI	-	R*4	Time in seconds
CUSEC	-	I*4	C.U. Second
DAY	-	I*4	Day of month
HOUR	-	I*4	Hour of day
MIN	-	I*4	Minute of hour
MONTH	-	I*4	Month of year
NOV	-	I*4	Data accumulation variable
NOV1	-	I*4	Data accumulation variable
NOV2	-	I*4	Data accumulation variable
TCAP	-	I*4	Accumulated capacity
TYPE	10	I*2	SL generic type designators
YEAR	-	I*4	Year

6.2.33.4 Description

EARRPT is invoked during system restart to display the initial conditions of the simulation run. The initial conditions displayed include:

- o General system parameters and options
- o Vehicle occupancy status by guideway link
- o Scheduled route activity summary, if scheduled service being modeled
- o Passenger and vehicle summary by station
- o Station link vehicle occupancy summary by individual station

6.2.33.5 PDL

See Appendix A.

6.2.33.6 Decision Tables and Algorithms

None.

6.2.34 EASAMP - Periodic Sampling

6.2.34.1 Identification

- o EASAMP - Periodic Sampling Event
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.34.2 Argument Dictionary

None.

6.2.34.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ALOCC	KML	R*4	Guideway link occupancy statistic
ASOCC	KMS	R*4	Station boarding queue occupancy statistic
FOLLOW	-	R*8	The character string "FOLLOWER"
NBY	-	I*2	Number of bytes in sampling record
NL	-	I*4	Number of defined guideway links
NR	-	I*4	Number of defined routes and route groups
NS	-	I*4	Number of defined stations
NSL	-	I*4	Number of defined station links
TIME	-	R*4	Simulation clock time in seconds

6.2.34.4 Description

Statistical data collection within the DESM is accomplished by recording modelling status and history data for entities and transactions at key event points in the modelling subsystem. The accumulation of statistical data is performed over a sampling interval. The occurrence of a sampling event causes the accumulated statistics to be written to the raw statistics file. The statistics are written to the file as a series of header (EZHDR) and data follower records. Each record contains the time of the sample and a type designator for the entity or transaction to which the follower statistics apply. This header information is used in output processing to determine the format and position of data in the sampling records.

Prior to accomplishing the processing above, EASAMP invokes EZINT to correct time integral accumulations recorded during the sampling period. Time integrals are maintained for all statistical data reflecting averages over the sampling interval. Averages are obtained by:

1. Initializing the time integral for a particular statistic at the start of the sampling interval to the negative of the product of the current time (clock) value and the current occupancy of the element to which the integral is associated. This value is reinitialized after each periodic sample is taken.
2. Whenever the occupancy decreases during the sampling interval, the integral is increased by the product of the current time value and the size of the occupancy decreases.
3. Whenever the occupancy increases during the sampling interval, the integral is decreased by the product of the current time and the size of the increase.

At the end of the interval, the integral value is corrected by increasing it by the product of the current time value and the current occupancy. At this point, the integral contains the desired value. The average value of the statistic can then be obtained by dividing the computed time integral by the periodic sampling interval. If required, a "snapshot" or intermediate sampling report is then written to the system output device by invoking EASRPT. The report summarizes selected statistics related to overall system performance and modelling subsystem (guideway and station) status to provide a basis for "quick look" run analysis. Once the required statistics have been recorded in the sampling file, EZZERO is called to reset sampling statistics for the next data accumulation interval and the system service transaction which initiated the sampling process is rescheduled for occurrence at completion of the next periodic sampling interval.

If requested by the user, EASAMP calculates and writes to the output device, the link statistics log containing the average link occupancy for each link over the sampling interval, and/or the station boarding queue statistics log containing the average boarding queue occupancy for each station over the sampling interval.

6.2.34.5 PDL

See Appendix A.

6.2.34.6 Decision Tables and Algorithms

None.

6.2.35 EASCTL - Station Link Modeling Control

6.2.35.1 Identification

- o EASCTL - Station Link Modeling Control
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.35.2 Argument Dictionary

None. Included segment in EMODEL.

6.2.35.3 Local Variable Dictionary

None.

6.2.35.4 Description

EASCTL is invoked in response to a transaction from the Future Events List (FEL) which requires station link event processing. Upon entry, the Station Model (ESLMDL) is invoked to perform required event processing for the vehicle. Upon completion of station link event processing, a determination is made as to whether the vehicle has completed all required events on its current station link. If all events have been completed, the next entity (station or guideway link) for the vehicle is determined. Otherwise, control returns to the control loop of the simulator for processing of the next available transaction from the FEL.

Once the next entity for the vehicle has been determined, it is checked for availability depending on type, guideway or station link. If the next entity for the vehicle is a station link, and entry is not inhibited due to congestion or failure, station link exit processing is performed for the current link and the station model is invoked to accomplish required processing and scheduling on the FEL of event completion for the vehicle on its next station link. If, on the other hand, the next link is a guideway link and vehicle progress is not inhibited due to congestion or failure, station link exit processing is performed and the guideway link model is invoked to perform required event processing.

In either case, when progress is inhibited, the vehicle queues at the exit to its current station link waiting to be prompted as a result of downstream vehicle movement.

6.2.35.5 PDL

See Appendix A.

6.2.35.6 Decision Tables and Algorithms

None.

6.2.36 EASRPT - Intermediate Sampling Report

6.2.36.1 Identification

- o EASRPT - Display Intermediate Sampling Report
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.36.2 Argument Dictionary

None.

6.2.36.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
AL	-	R*4	Delay time seconds
AL1	-	R*4	Average occupancy
DAY	-	I*2	Day of month
DELAY	-	R*4	Time in seconds
DEV	-	R*4	Computational variable for average
DEV1	-	R*4	Computational variable for average
DEV2	-	R*4	Computational variable for average
HOUR	-	I*2	Hour of day
KM	-	R*4	Distance in KM
LOAD	-	R*4	Link load factor
MIN	-	I*2	Minute of hour
MONTH	-	I*2	Month of year
NOV3	-	R*4	Total link load factor

VARIABLE	DIM	TYPE	DESCRIPTION
TCAP	-	I*4	Total link capacity
TOT	-	I*4	Total empties dispatched
TOT1	-	I*4	Total empties requested
TYPE	10	I*2	SL generic type designations
VLOAD	-	R*4	Vehicle load factor
YEAR	-	I*2	Year

6.2.36.4 Description

EASRPT displays a report of simulation status at multiples of the sampling interval as specified by the simulation user. The report contains selected measures of overall system performance and statistics related to individual guideway links, routes, stations and station links. The statistics displayed represent a subset of both status and historical items maintained by the DESM and recorded to the sampling file at periodic intervals. Each statistic representing a time average is divided by the sampling interval in c.u. to obtain a non time based value prior to display. All time values listed in the report are converted to seconds prior to display and all distance measurements are converted from meters to kilometers.

6.2.36.5 PDL

See Appendix A.

6.2.36.6 Decision Tables and Algorithms

None.

6.2.36a EASTOR - Assign Stored Vehicles in Scheduled Service AFSM

6.2.36a.1 Identification

- o EASTOR - Assign Stored Vehicles in Scheduled Service AFSM
- o GM TSC - July 1, 1981
- o PARAFOR

6.2.36a.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
RTE	-	I*4	Route ID

6.2.36a.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
FOUND	-	L*1	Flag indicating route using same barn needs vehicles
FOUND2	-	L*1	Flag indicating route using different barn needs vehicles
MNEED	-	I*4	Maximum number of vehicles needed by any route
NEED	-	I*4	Number of vehicles needed by route
NUMVEH	-	I*4	Number of stored vehicles assigned to route but not needed
R	-	I*4	Route ID (loop counter) for assigning excess vehicles
RNEED	-	I*4	Route ID having maximum need
STN	-	I*4	Maintenance barn station ID
V	-	I*4	Vehicle ID loop counter (inner loop)
VEH	-	I*4	Vehicle ID loop counter (outer loop) through stored vehicles
VEHN	-	I*4	Next vehicle in storage
XTN	-	I*4	Transaction ID for vehicle prompt

6.2.36a.4 Description

This routine reassigns excess vehicles to other routes during scheduled service active fleet size management. First, the number of vehicles stored and assigned to the route less the number of vehicles still needing to be launched on the route is calculated. If the result is positive, then excess vehicles exist in storage for the route and need to be reassigned. Reassignment precedence is given to other routes using the same barn as the current route. For each excess vehicle, the route with the greatest need for vehicles (intended number of vehicles traveling the route less number of vehicles assigned to the route) using the same barn is chosen. If there are none, then the route with the greatest need for vehicles using a different barn is chosen. If the chosen route uses the same barn the vehicle's route is changed, route statistics are updated, and routine EANTRN is called to attempt entrainment and launch on the new route. If the chosen route uses a different barn, the vehicle's route is changed, route statistics are changed and a prompt is scheduled to move the vehicle out of storage to the chosen route's maintenance barn. If no other route needs vehicles, the vehicle is excess to the total fleet; thus, it is removed from service and the appropriate statistics are updated.

6.2.36a.5 PDL

None.

6.2.36a.6 Decision Tables and Algorithms

Number of excess vehicles on route =
Number of stored vehicles assigned to route +
Number of vehicles traveling route: ANVRTE(RTE) -
Number of vehicles intended to travel route: PNV RTE(RTE)

Number of vehicles needed by other routes =
Number of vehicles intended to travel route: PNV RTE(RTE) -
Number of stored vehicles assigned to route: PNTVEH(RTE)

6.2.37 EATORG - Trip Origination

6.2.37.1 Identification

- o EATORG - Process a trip arrival
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.37.2 Argument Dictionary

None.

6.2.37.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
END	-	L*1	Logical variable for end file
FOUND	-	L*1	Logic variable indicating vehicle found by ESTABO
NGRP	-	I*4	Number of subgroups created for a trip
REJ	-	L*1	Logical variable for trip rejection
STN	-	I*2	Origin station
TIME	-	R*4	Clock time in seconds of next trip arrival
TRPID	-	I*4	Trip transaction ID
TRPNXT	-	I*4	Subgroup transaction ID

6.2.37.4 Description

EATORG performs placement of an arriving trip into a specific station boarding queue and initial vehicle selection for servicing of the trip demand. The arriving trip may represent either a new demand request or a transfer currently enroute to its requested destination.

Upon entry, a determination is made as to whether the trip arrival was previously scheduled on the FEL during a checkpointed experiment from which a restart was performed. If this is the case, the trip is ignored since it is invalid for the current run and the trip transaction is freed and control is returned to the main routine.

Arrival processing for a valid arriving trip begins by determining the origin station. If the arrival time of the trip is equal to the current clock-time, the origin station is set equal to the arrival origin of the trip. Otherwise, the trip is a transfer and the origin station is set equal to the trip's intermediate destination. The trip transaction is then placed in the boarding queue of its origin station. In the case of a new arrival, capacity constraints must be satisfied before boarding queue entry is performed. If capacity is exceeded, the trip arrival is recorded as a rejection and the trip transaction is returned to the transaction available list.

If a transfer is required for the trip to reach its destination, an intermediate or transfer destination is then assigned to the trip. This destination becomes the next scheduled stop for the trip. Upon arrival at its transfer destination, the trip is required to disembark its current vehicle and enter the boarding process after it has undergone transfer time scheduling. If Scheduled Service is being modeled, the trip is also assigned a route which can provide transportation to its next destination. This ID can correspond to a specific route number or a route group encompassing several specific routes available for trip servicing.

If the number of patrons associated with the trip exceeds the maximum trip size, it may be subdivided into subgroups such that size of any one subgroup does not exceed a maximum specified trip group size. This is accomplished by obtaining additional trip transactions to form subgroups associated with the original trip. This association is made by chaining the subgroup trip transactions to the original arriving trip. As each subgroup transaction is created, the number of patrons associated with the original trip is decremented by the maximum trip size, and the subgroup trip assumes a number of patrons equal to the maximum trip size. The subgroup division process is terminated when the number of patrons remaining in the original trip falls below the maximum trip subgroup size. Each subgroup transaction is placed in the boarding queue at the origin station of the original trip.

Once the arriving trip and any subgroups have been placed in the station boarding queue initial vehicle selection for servicing of the trip is performed by invoking ESTABQ. Upon return, the next trip arrival is read from the Structured Demand file. The arriving trip is assigned to an available trip transaction whose ID is recorded within the service request transaction which is scheduled for occurrence at the required arrival time for the trip. If an end of file is encountered in reading the file, the demand input is reused by rewinding the file and reading the first trip arrival record. The arrival time of the trip, in the case of demand reuse serves as a delta time, which is added to the current clock for scheduling trip origination.

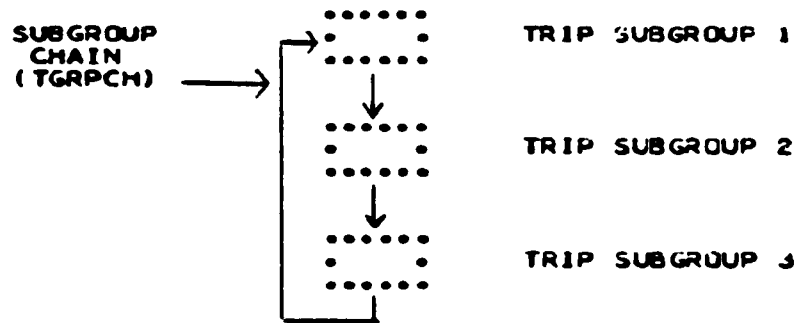
6.2.37.5 PDL

See Appendix A.

6.2.37.6 Decision Tables and Algorithms

a. Trip Subgroups

The subgroup relationships between trip segments created to satisfy vehicle capacity constraints are maintained via a trip group chain or pointer associated with each trip transaction as follows:



The circular nature of this chain permits the coalescing of trip subgroups at the destination station by tracing the subgroup chain beginning with the current trip as described in ESMOLA.

6.2.38 EAZNIT - Statistics Initialization

6.2.38.1 Identification

- o EAZNIT - Initialize DESM Sampling Statistics
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.38.2 Argument Dictionary

None.

6.2.38.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
FOLLOW	-	R*8	Keyword 'Follow'
NBY	-	I*2	Number of bytes in follower record
NL	-	I*4	Number of guideway links
NS	-	I*4	Number of stations
NSEAT	-	I*4	Seat Capacity
NSL	-	I*4	Number of station links
NR	-	I*4	Number of routes
NRG	-	I*4	Number of route groups
NV	-	I*4	Fleet Size

6.2.38.4 Description

EAZNIT performs initialization of sampling statistics and writes the system descriptor record to the sampling file. Upon entry, status statistics maintained as running sums during the simulation experiment are set to their initial values. The historical statistics which reflect activity over a particular sampling interval are then zeroed to establish

the starting conditions for the first sampling period by invoking EZZERO. The initial header record, identifying the occurrence of the system descriptor record is written to the sampling file by EZHDR. The system descriptor follower record containing characteristics of the simulation experiment is then formatted and written to the sampling file for use during output processing.

6.2.38.5 PDL

See Appendix A.

6.2.38.6 Decision Tables and Algorithms

None.

6.2.39 EGALT - Alternate Path Cost Computation

6.2.39.1 Identification

- o EGALT - Alternate Path Cost Computation
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.39.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
PTR	-	I*4	Pointer to alternate path table entry

6.2.39.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ARRVD	-	L*1	Arrived at destination indicator
GLNK	-	I*4	Temporary variable for link ID
GLOCCR	-	REAL*4	Temporary variable for link occupancy
LNK	-	I*4	Starting link to use in tracing primary path

6.2.39.4 Description

EGALT computes the cost of an alternate route contained in the alternate route table, starting at 'PTR', by tracing through the table until the last entry for the particular path is reached (or the destination is encountered). As each link along the path is identified, the cost as determined by the user-specified algorithm option of

- a) Link nominal travel time
- b) Link length

- c) Utilization or
- d) A weighted combination of (a) and (c)

is accumulated.

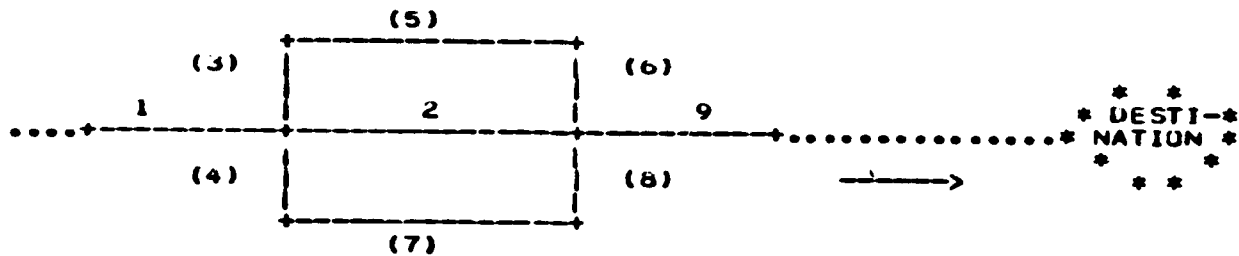
If a failed link is encountered, the path is excluded from further consideration by adding an artificially high failure factor to the cost of the route. If the vehicle destination is encountered in the tracing process, cost accumulation ceases and 'PTR' is updated to point to the last entry of the particular path in the alternate route table. If, however, the end of the alternate path entry is reached prior to finding the vehicles's destination, cost accumulation continues with the primary path from the most recently processed link, 'LNK', to the destination, by invoking EGPRMY.

6.2.39.5 PDL

See Appendix A.

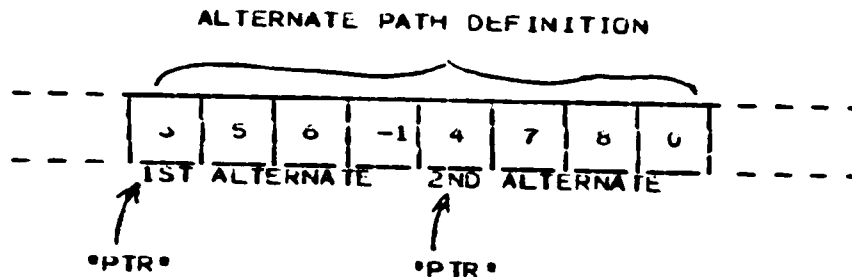
6.2.39.6 Decision Tables and Algorithms

The following diagram illustrates a sample network segment with a common diverge and alternate path definition.



The primary path through this sample network segment from the common diverge at the end of link 1 would consist of links 2, 9, etc. The alternate paths would be links 3, 5 and 6 and links 4, 7 and 8 respectively. In tracing both of these alternate paths, it is clear that the end of each alternate definition, in the example, would be found prior to encountering the vehicle's destination. Cost accumulation, as described above, must then continue with link 9, etc., until the destination is reached.

A sample of an alternate path definition within the alternate route table corresponding to the above network segment is shown below:



As described in the description of EGGNXT, the table entries or link sequence from '3' to '0' consist of an alternate path definition, pointed to by the common diverge value in the PLSLT. The '-1' indicates the end of one alternate possibility from the diverge point and further indicates that another path description follows. The '0' indicates the end of an alternate path description and the end of the entire definition within the alternate path table. 'PTR' is the pointer used by EGALT to locate and trace one alternate possibility. In the above example, if the value of 'PTR', pointing to table entry '3' was '10' then the common diverge value within the PLSLT in the entry following the entry for link 1 would be 10002 which is equal to

$$'10' \times 1000 + '2'$$

with '10' being the pointer to the alternate path definition within the alternate path table and '2' being the next guideway link from link 1 along the primary or minimum path.

The algorithms used in accumulating the alternate path segment costs are the same as those described, by algorithm type, in the description of EGPRMY.

6.2.40 EGASTN - Alternate Station Assignment

6.2.40.1 Identification

- o EGASTN - Assign Alternate Station to a Vehicle
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.40.2 Argument Dictionary

None.

6.2.40.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
GL	-	I*4	Current guideway link for vehicle
NEXT	-	I*4	Assigned next station
STN	-	I*4	Current next station
TRP	-	I*4	Trip transaction ID
VEH	-	I*4	Vehicle ID

6.2.40.4 Description

Alternate station assignment is invoked for non-synchronous operation by guideway link modeling control (EAGCTL) when vehicle entry to a destination station is blocked and continued guideway traversal to an alternate is possible. The destination station for which entry is precluded can correspond to the next stop for the vehicle or a bypassed station on the vehicle path for which diversion to service waiting trips was attempted.

Upon entry, the alternate station for the vehicle and all entrained followers is determined. If the vehicle was attempting to divert into a bypassed station, the next station for the vehicle is reset to its original destination. Otherwise, an alternate is assigned based upon the type of service being simulated. In Demand Responsive the vehicle is

assigned the next downstream station in the network as an alternate destination. If multi-party operation is permitted, the assigned downstream station is selected as the next station on the current path of the vehicle to its ultimate destination. Scheduled service processing results in redirection of the vehicle to the next station on its scheduled route.

Once the alternate station has been assigned to the vehicle, all onboard trips originally scheduled to complete at the bypassed station, are reassigned the alternate station as their current destination. Upon arrival at the alternate station, all reassigned trips are considered to have completed required travel, and their actual travel time is increased by an egress time which is used for compensation of off-guideway travel time associated with reaching their originally scheduled destinations. If the vehicle is travelling empty, and assigned to a circuitous route, it is removed from that route and must be reassigned to a new circuitous route at its alternate destination. This is required since the assigned alternate may not be on the route currently being traversed by the vehicle.

The alternate station assignment process is concluded by cancelling any reservations on the vehicle for trips waiting at the bypassed station.

6.2.40.5 PDL

See Appendix A.

6.2.40.6 Decision Tables and Algorithms

None.

6.2.41 EGCNXT - Next Station Determination on Circuitous Empty Route

6.2.41.1 Identification

- o EGCNXT - Next Station Determination on Circuitous Empty Route
- o IBM/FSD - May 1, 1978
- o PARAGOR

6.2.41.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
STNID	-	I*2	(OUTPUT) next station

6.2.41.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ROUTE	-	I*2	Beginning entry number for route in circuitous route table

6.2.41.4 Description

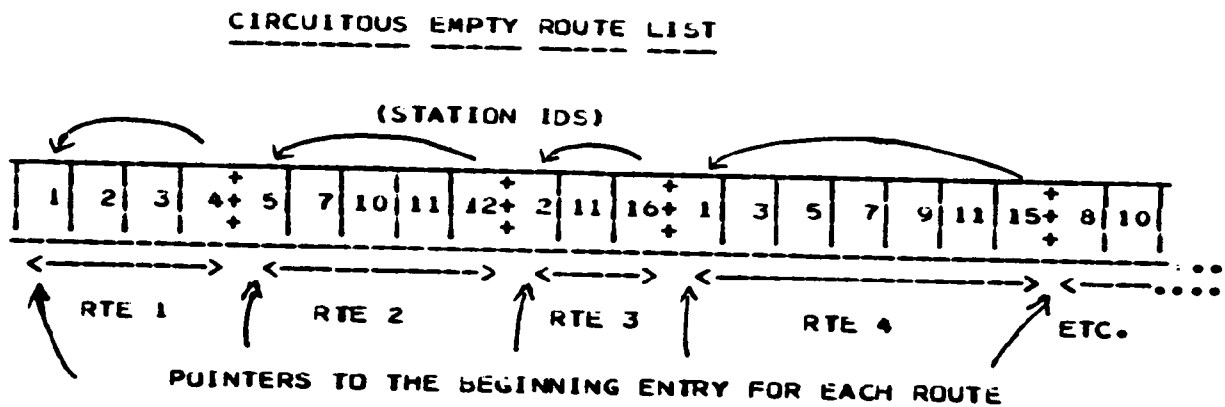
EGCNXT determines the next station destination for a circuitous empty vehicle, being rerouted around a scheduled stop when entry is not required along its circulation path. To accomplish this, the particular list of station IDs associated with the vehicle's assigned route is searched for the current station ID, which is followed by the identification of the next station along the route. This following entry is assigned to the vehicle's next station destination unless it is the first entry of the next circulation route within the circuitous empty route list (see diagram in following section). In this case, the destination assignment will instead be the first station ID in the list, since the list of stations for a route must identify a circular path.

6.2.41.5 PDL

See Appendix A.

6.2.41.6 Decision Tables and Algorithms

The circuitous empty route list (as shown in the example below) is a concatenation of empty vehicle circulation routes, which are lists of station IDs defining circular paths. The station to be followed along the path after the last station entry for a route will again be the first station in the specific route list. A system pointer exists to the first entry for each route and thus is available for use when a vehicle is assigned a particular circuitous route.



6.2.42 EGDSTP - Demand Stop Service Processing

6.2.42.1 Identification

- o EGDSTP - Demand Stop Service Processing
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.42.2 Argument Dictionary

None.

6.2.42.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DSTOP	-	L*1	Stop at station indicator
PTR	-	I*2	Next position in scheduled route list
RTE	-	I*2	Next route identification
STN	-	I*2	Current station
TRPID	-	I*2	ID of station trip being processed
VEH	-	I*4	Temporary variable for vehicle ID
VTRPID	-	I*2	ID of vehicle trip being processed

6.2.42.4 Description

EGDSTP determines, as a result of user specification of demand stop processing, if it is necessary for a particular vehicle to stop at the current station which is a station along the vehicle's scheduled service route. Normal entry into the station is required if any of the following conditions exist:

- o The vehicle is empty and is scheduled to terminate due to active fleet size management,
- o At least one trip onboard the vehicle is destined for the current station or
- o At least one trip exists in the station trip queue (is waiting) that would fit on the vehicle and which has the same route assignment as the vehicle.

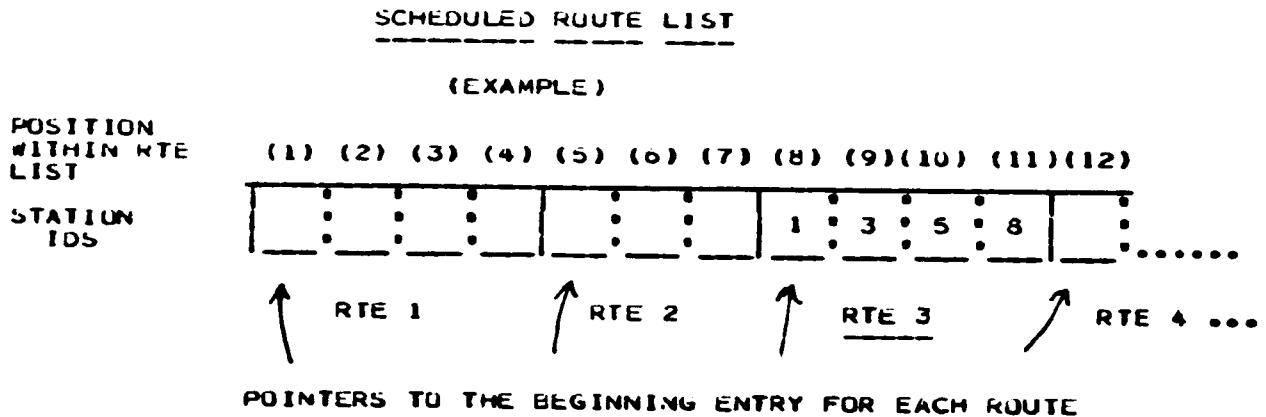
If it is found that a need to stop does not exist for the vehicle and if the vehicle is a lead vehicle of a train, then similar processing is done for each successive vehicle until one is found that must stop at the station. If, after doing the above, it is clear that it is still not necessary for the vehicle or train to stop, then the vehicle(s) may bypass the station and proceed to its next stop. This is accomplished by reassigning the next vehicle stop to that of the next entry in the scheduled route list (see below). If the station being bypassed is an off-line station then EGGNXT is invoked to determine the next guideway link the vehicle must traverse enroute to its new next station assignment. (A vehicle must still pass through any on-line station, but as long as its destination is no longer the current station, the vehicle will not stop.)

(This demand stop processing is not valid and would thus not be invoked in simulation exercises using synchronous longitudinal control.)

6.2.42.5 PDL

See Appendix A.

6.2.42.6 Decision Tables and Algorithms



The scheduled route list, used in order to retrieve the next station for a vehicle along its current route, is a concatenated list of all the scheduled routes. Each route consists of a list of station IDs defining a circular path. Thus, the station to follow along the path after the last station entry of the route will again be the first station in the specific route definition. A system pointer exists to the beginning entry of each route. Each vehicle, when travelling, is assigned a route number. In addition, a current position pointer within the route list is maintained for each vehicle and must be incremented every time a vehicle is assigned its next station destination (as described above). If the position pointer, after being updated, is the same as the pointer to the beginning entry of the next scheduled route in the route list, then the current station must be the last entry in the route. For this case, the position pointer, pointing to the entry with the next station destination, is reset to point to the beginning entry of the route.

6.2.43 EGDTRN - Guideway Link Vehicle Detrainment

6.2.43.1 Identification

- o EGDTRN - Dynamic Vehicle Detrainment
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.43.2 Argument Dictionary

None.

VARIABLE	DIM	TYPE	DESCRIPTION
VEH	-	I*4	Vehicle being detrained

6.2.43.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
BLKOCC	-	I*2	Old block occupancy requirement for train
GLBLKR	-	R*4	Block size (real format)
NUMVEH	-	I*2	Number of vehicles left in original train
VEH1	-	I*4	Temporary variable for vehicle ID in vehicle chain
VLENTM	-	I*4	Time it takes to travel a vehicle length

6.2.43.4 Description

EGDTRN accomplishes line-speed detrainment at network diverges, including the diverge points at the entry to stations. Detrainment occurs whenever the next entity for traversal differs between an intermediate vehicle and the lead vehicle of a train. EGDTRN assumes that next entity incompatibility has already been determined (see EGRESV

for detrainment decision at station diverge points and EGTRNC for the decision at network diverges) and the vehicle identified to the module, 'VEH', is the incompatible vehicle.

At the time of and just immediately after detrainment, one of the following situations, as pictured and described, should exist: (Assume for the purpose of the discussion, the existence of the following train prior to detrainment.)



The incompatible vehicle, VEH, is the second vehicle of the original train. The lead vehicle must then travel alone and will no longer be associated with the train. The detrained vehicle, VEH, will become the lead vehicle of a second train (as in the example above) if detrainment results in division of the train at an intermediate point. Otherwise, it too will travel alone.



Detrainment results in division of the train at an intermediate point, such that a smaller front train still exists along with a following train of which VEH is the lead vehicle.



The detrained vehicle, VEH, was the last vehicle of the original train and thus must now travel alone. If more than one vehicle exists in front of VEH it will continue to travel as a train.

Since the identifying information of an entrained group consists of pointers or chains from vehicle to vehicle along with train length data, all pointers and length fields for each member of the detrained vehicles or trains must be reestablished (see in the section below). In addition, when vehicles are travelling under a fixed headway positioning scheme, the block occupancy requirement for the detrained vehicles and the resulting link occupancy value must be adjusted to reflect the new train status on the guideway (Equations (2) and (3)). The block requirement must now be determined for two single vehicles, a single vehicle plus a train, one vehicle length shorter, or for two smaller trains. The link occupancy value will be adjusted by the difference between the old train block occupancy requirement and the sum of the requirements for the new detrained vehicles or trains (see below). The adjustments described are necessary since individual members of trains, unlike single travelling vehicles, are not required to occupy an entire fixed length of guideway or slot. They may assume a position immediately behind the preceding vehicle within the train, with the sum of the vehicles or train occupying a calculated number of these fixed blocks. Detrainment may force the number of required blocks to be greater than than required for a single entrained group.

Once the vehicle(s) is detrained it is scheduled to reenter the model at delta time '0' to continue travel, independently, to the end of the link (unless the vehicle is queued). Its current distance on the link and its link exit time, which are dependent on the position the detrained vehicle occupied in the old train, are established (Equations (4) and(5)) so as the vehicle enters the model, it can be determined how much time the vehicle(s) will require to complete link travel.

The detrainment of queued vehicles requires that the queue pointers be adjusted to include linkage with the new member, now identified independently. Also, queue occupancy is similarly adjusted in fixed headway cases to reflect new block occupancy requirements of the detrained vehicles.

6.2.43.5 PDL

See Appendix A.

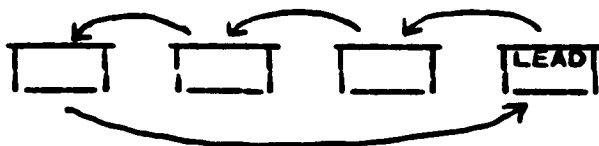
6.2.43.6 Decision Tables and Algorithms

a. Trains/detrained vehicle chains and length fields

Vehicle entrainment/detrainment is maintained through two pieces of data per entrained member, a chain word or pointer and a length field.

The chain word:

if > 0 points to the following vehicle in the train, with the last vehicle in the train pointing to the lead vehicle (as shown),



if = 0, implies the vehicle is now unentrained.

The length field is:

> 0 for the lead vehicle of the train, which indicates the current length of the train (would be '4' above).

= 0 for an unentrained vehicle or for last vehicle of an entrained group at train length capacity.

< 0 for an intermediate vehicle, which is the negative value of the number of vehicles that can further be added behind the vehicle to the entrained group. This intermediate vehicle length field is calculated as:

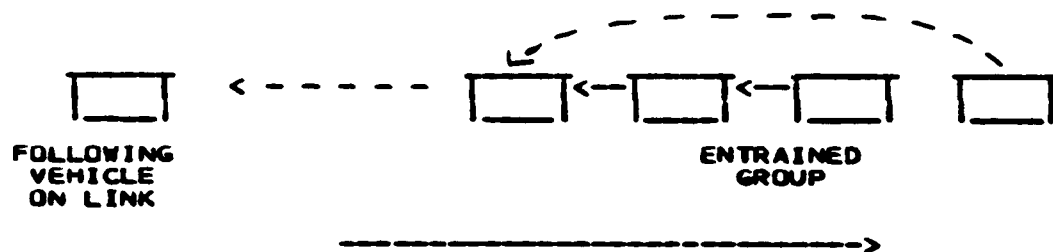
$$(1) \quad L = V_N - M_{TR} \quad (\# \text{ of vehicles})$$

Where:

V_N = relative train position of vehicle or vehicle number and

M_{TR} = maximum train length allowed.

Another piece of data, similar to the chain word, which is maintained as part of the detraining process is the vehicle follower pointer. A 'follower' is the ID of the vehicle, if any, which is anywhere behind the particular vehicle on the same link. The 'follower' does not need to be a part of the train. The one exception to the general follower rule applies to the lead vehicle of a train. The follower ID maintained with the lead is the ID of the last vehicle of the current train. Thus, if a train definition changes, so will this follower ID. The 'follower' relationship is depicted with the dotted lines in the following diagram.



b. Train/vehicle fixed block occupancy requirement

The fixed block or slot occupancy requirement for the detrained vehicles or trains is computed by the following:

$$(2) T_0 = \frac{(T_L - 1) \times V_L}{B_L} + 1 \quad (\text{rounded up})$$

Where:

T_L = train length or number of vehicles in the entrained group,

V_L = vehicle length (meters) and

B_L = block length (meters).

The occupancy requirement for a newly detrained vehicle is thus '1'.

The adjustment made to the current link occupancy value to reflect the new number of blocks required for both of the detrained portions of the train is accomplished as follows:

$$(3) L_0 = L_{T0} - T_0 + (T_1 + T_2)$$

Where:

L_{T0} = former link occupancy, before detrainment, in # of fixed blocks,

T_0 = block occupancy associated with train before detrainment, in # of fixed blocks

T_1 = block occupancy of detrained portion 1 (= 1 for single vehicle) and

T_2 = block occupancy of detrained portion 2 (= 1 for single vehicle).

c. Detrained vehicle link measurements

The current position of the detrained vehicle(s) or the distance travelled on the link is determined as:

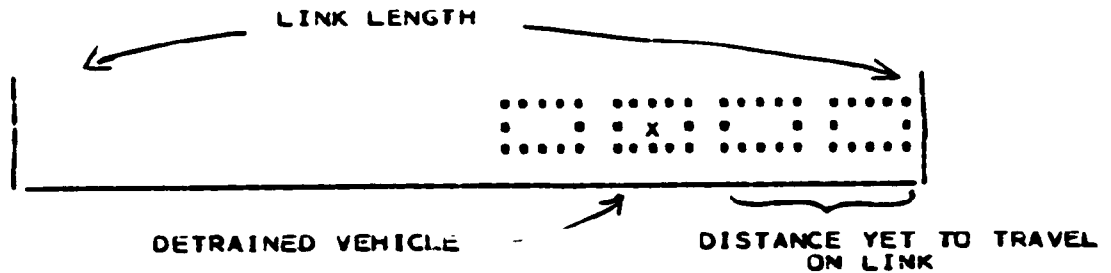
$$(4) L_D = L_L - (V_1 \times V_L)$$

Where:

L_L = link length (meters),

V_1 = # of vehicles associated with detrained portion 1 of the original train, those vehicles in front of the recently detrained vehicle and

V_L = vehicle length (meters).



The projected exit time of the detrained vehicle(s) from the link is given by:

$$(5) T_X = C_T + (V_1 \times V_T)$$

Where:

C_T = current clock time (cu)

V_T = time it takes to travel a vehicle length along the guideway link (cu) and

V_1 as described above.

6.2.44 EGEMTY - Guideway Link Empty Vehicle Processing

6.2.44.1 Identification

- o EGEMTY - Guideway Link Empty Vehicle Processing at Station Diverge Locations
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.44.2 Argument Dictionary

None

6.2.44.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
LNKSTN	-	I*2	station ID at end of link
STNID	-	I*2	variable returned from finding next station
VEHDQ	-	I*4	vehicle to be removed from station arrival list

6.2.44.4 Description

EGEMTY determines if an empty vehicle circulating on the guideway and currently at a station diverge point must stop at the station. The vehicle may not be scheduled to enter the station, but if the proper conditions exist, the vehicle will be diverted from its current path into the station. On the other hand, a circuitous empty vehicle scheduled to enter the station, which is a stop associated with its circuitous path assignment, may be rerouted around the station stop and allowed to continue along its circulation path.

Empty Vehicle Diversion

For the diversion of an empty vehicle into a station to occur, the following conditions must exist:

1. the vehicle must not have any reservations (at a downstream station),

2. the vehicle diversion policy must be enabled
3. unserviced (unreserved) trips must be waiting at the current station and
4. the vehicle must not be in station by-pass failure response mode

A vehicle is diverted into a station by resetting its next entity and destination equal to the station ID. Its former station assignment is maintained in case it is later determined that entry into the immediate station is not possible. In addition, as part of this diversion process, the number of unserviced station trips is decremented, to assure that a following vehicle is not diverted to service the same trip. If the vehicle has a reserved station assignment and it is something other than the station being diverted to, the vehicle ID is then removed from the arrival list of the former reserved station.

Empty Vehicle Rerouting

For a vehicle to be rerouted around a scheduled station stop, the following conditions must exist:

1. the vehicle must be an unreserved circuitous empty. Thus the vehicle must have a circuitous path assignment and must not have any reservations related to waiting trips at the current station,
2. no unserviced (unreserved) trips exist in the station or the vehicle diversion policy is not enabled,
3. the station is not an on-line station and
4. non-deterministic dispatch is in effect. This is necessary since deterministic dispatch requires the delaying of vehicle launch until clear passage is found through all enroute merges to the vehicle's next destination.

If it is found that station entry is not required for the circuitous empty, EGCNXT is invoked to obtain the next station assignment along the vehicle's circulation path. Following this, EGGNXT is called to determine the next guideway link the vehicle must traverse enroute to its new station assignment. Finally, if the reservations or arrival list recording option is enabled, the next station becomes the vehicle's reserving station and the vehicle ID is recorded in the arrival list of that station (in module EGVALS).

6.2.44.5 PDL

See Appendix A

6.2.44.6 Decision Tables and Algorithms

None

6.2.45 EGETRN - Dynamic Vehicle Entrainment

6.2.45.1 Identification

- o EGETRN - Guideway Link Dynamic Vehicle Entrainment
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.45.2 Argument Dictionary

None

6.2.45.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
BLKOC1		I*2	old block occupancy requirement for train 1
BLKOC2		I*2	old block occupancy requirement for train 2
ENTRN		L*1	entrainment status flag
FIRSTV		I*2	ID of first vehicle of entire train
GLBLKR		R*4	block size (real format)
LASTV		I*2	ID of last vehicle of entire vehicle train
LEADV		I*2	temporary variable for lead vehicle ID
TOCC		I*2	# of new vehicles attempting to entrain
VEH		I*2	temporary vehicle ID
VLENTM		I*4	vehicle length travel time

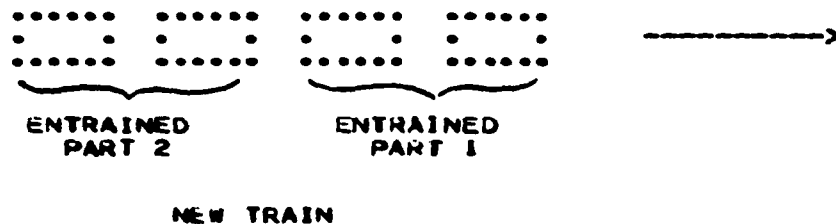
6.2.45.4 Description

EGETRN performs on-guideway entrainment of vehicles at the be-

gining of a guideway link following headway traversal, whenever the following conditions are satisfied:

1. the current link is a merge output link,
2. the dynamic entrainment option is enabled,
3. an adjacent vehicle(s) exists in front of the current vehicle(s) on the link, such that the time separation between the vehicles is within a vehicle length travel time,
4. the following (or current) vehicle is not degraded,
5. the preceding vehicle is not degraded or member of a degraded train and
6. entrainment of the vehicles can be performed without violating the maximum train length allowed.

The entrainment process involves extending the length of an already existing train or establishing a new train from two consecutive unentrained vehicles. Thus a single vehicle or train may be entrained to another single vehicle or train, establishing a new entity as shown below.



This is accomplished by setting all of the appropriate linkage pointers and length fields for each member of the new train (see the Algorithm section below).

When vehicles are travelling under a fixed headway positioning scheme, the block occupancy requirement for the entrained vehicles (see Equation (2)) and the resulting link occupancy value (see Equation (3)), in number of occupied blocks, must be adjusted to reflect the new train

status on the guideway. The block requirement must now be determined for the single entrained group, while negating that previously identified for each of the entrained parts (single vehicles or trains). The link occupancy value will be adjusted by the difference between the sum of the requirements of the unentrained vehicle(s) and the requirement, for the new entrained group. The adjustments described are necessary since individual members of trains, unlike single travelling vehicles, are not required to occupy an entire fixed length of guideway or slot. They may assume a position immediately behind the preceding vehicle within the train, with the sum of the vehicles or the train occupying a calculated number of these fixed blocks. Entrainment may cause the number of required parts to be less than that required for the individual unentrained parts.

6.2.45.5 PDL

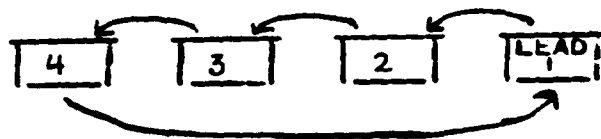
See Appendix A.

6.2.45.6 Decision Tables and Algorithms

a. Entrained vehicle chains and length fields

Vehicle entrainment is maintained through two pieces of data per entrained member, a chain word or pointer and a length field:

The chain word points to the following vehicle in the train, with the last vehicle in the train pointing to the lead vehicle (as shown).



The length field is:

- > 0 for the lead vehicle of the train, which indicates the current length of the train,
- < 0 for each intermediate vehicle, which is the negative value of the number of vehicles that can further be added to the entrained group behind the vehicle.

This intermediate vehicle length field is calculated as:

$$(1) L = V_N - M_{TR} \quad (\# \text{ of vehicles})$$

where:

V_N = relative train position of vehicle or vehicle number and

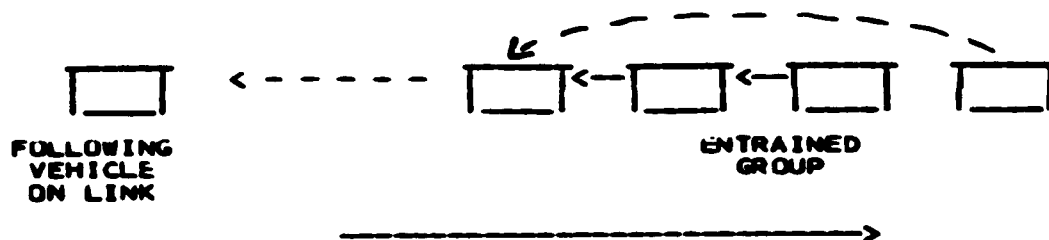
M_{TR} = maximum train length allowed.

= 0 for the last vehicle of a train at train length capacity.

The length fields for the train example pictured above, with a maximum train length allowance of six vehicles would be:

- 4 for the lead vehicle
- 4 for vehicle 2, (4 vehicles may yet be entrained)
- 3 for vehicle 3, (3 vehicles may yet be entrained)
- 2 for vehicle 4, (2 vehicles may yet be entrained)

Another piece of data, similar to the chain word, which is maintained as part of the entrainment process is the vehicle follower pointer. A 'follower' is the ID of the vehicle, if any, which is behind the present vehicle, somewhere on the link. It is not necessary that the 'follower' be a part of the same entrained group. The one exception to the general rule describing this relationship applies to the lead vehicle of an entrained group. The follower ID maintained with the lead is the ID of the last vehicle of the train. (The normal follower of the lead vehicle can be found using the train vehicle chain word.) Thus, if a train definition changes, so will this follower ID. (The 'follower' relationship is depicted with dotted lines in the diagram below.)



b. Entrained group fixed block occupancy requirement

The fixed block or slot occupancy requirement for the train is computed by the following:

$$(2) \quad T_0 = \frac{(T_L - 1) \times V_L}{B_L} + 1 \quad (\text{rounded up})$$

where:

T_L = train length or number of vehicles in the entrained group,

V_L = vehicle length (meters) and

B_L = block length (meters).

The adjustment made to the current link occupancy value to reflect the new number of blocks required for the entrained group is accomplished by:

$$(3) \quad L_0 = L_1 - (T_1 + T_2) + T_0$$

where:

L_1 = former link occupancy value, before entrainment, in # of blocks,

T_1 = block occupancy requirement associated with the former unentrained front portion (single vehicle or train) of the new train (= 1 for single vehicle),

T_2 = block occupancy requirement associated with the former unentrained rear portion (single vehicle or train) of the new train (= 1 for single vehicle) and

T_0 as given above.

6.2.46 EGFAIL - Guideway Link Failure/Recovery

6.2.46.1 Identification

- o EGFAIL - Guideway Link Failure/Recovery Processing
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.46.2 Argument Dictionary

None

6.2.46.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
LINK	-	I*4	Link downstream of recovered link.

6.2.46.4 Description

EGFAIL is invoked by EAASYN in response to the occurrence of an asynchronous guideway link failure event. Prior to invoking EGFAIL the data associated with the implementation of the failure/recovery condition is read from the asynchronous data stream as generalized data input formatted variable definitions. This data identifies the location (guideway link) of the failure and type of failure as follows:

1. Entire guideway link failure/recovery
2. Link entry failure/recovery
3. Link exit failure/recovery
4. Link degradation

In addition, this data includes, for any guideway link or vehicle failure or recovery event, routing information which is used in subsequent vehicle dispatching from network stations and guideway link traversal, routing table input associated with the failure which attempts to route future vehicles in the simulation around the failed link, and recovery method selection and delay time information. Data associated with the recovery condition includes specification of the routing table ID which is to become effective now that the failure condition is cleared (minimum

path routing table ID in use prior to failure occurrence). Degraded guideway or station link failures have a penalty factor (fraction of velocity at which the vehicle or vehicles are to operate) as data which is read during the recognition processing of failures.

Upon entry to the routine a determination is made as to whether a failure, degradation or recovery event has been entered. This processing results in the setting of link entry or exit flags which are used in determining next entity availability within the link modelling subsystem. Total guideway link failure results in both the entry and exit flags being set simultaneously.

Specification of a failure recovery for a non-degradation failure results in the resetting of guideway link entry or exit status indicators and the scheduling of a system service transaction to invoke appropriate prompting or dequeuing actions for restarting any vehicles preempted by the failure condition. In the case of entry recovery, the recovered link is specified as the prompt entity to cause dequeuing of waiting upstream vehicles. Exit recovery results in a link immediately downstream of the recovered link, as obtained from the minimum path routing table (PLSLTN), being prompted.

Degradation processing causes the next vehicle entering or exiting a specified link to assume that an operation failure has occurred which will require subsequent operation at a user-specified fraction of line speed and eventual removal from service. The vehicle stops until restarted by the user input method after the specified time delays. The vehicle travels at the degraded speed to its next station stop, deboards all of its passengers as transfers, and is then routed to its maintenance barn. In EGFAIL, the link entry or exit specified is simply marked to capture the next vehicle.

Degradation recovery processing only verifies that a vehicle has been captured by the degradation failure conditions. If not, a message is displayed on the system output device and the degradation condition is removed.

6.2.46.5 PDL

See Appendix A.

6.2.46.6 Decision Tables and Algorithms

The values of the global variables shown below determine the status of a guideway or station link entry or exit.

GLENT - Guideway Link Entry

0: No blockage
1-99: Link failure
100+: Link marked for vehicle failure

GLEXIT - Guideway Link Exit

0: No blockage
1-9: Link failure
10-99: Link blocked to clear tow path
100+: Link marked for vehicle failure

SLENT - Station Link Entry

F: No blockage
T: Link failure

SLEXIT - Station Link Exit

0: No blockage
1-9: Link failure
10+: Link blocked to clear tow path

6.2.47 EGFNTR - Fixed Headway Travel Segment Entry

6.2.47.1 Identification

- o EGFNTR - Fixed Headway Travel Segment Entry
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.47.2 Argument Dictionary

None

6.2.47.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
LNK		I*2	current link
MEANV		R*4	distribution mean value
OCCLEN		I*4	occupancy length of queue at end of link
PENFAC		R*4	penalty factor
SCDTME		I*4	computed travel delta time
VVAR		R*4	velocity variance factor

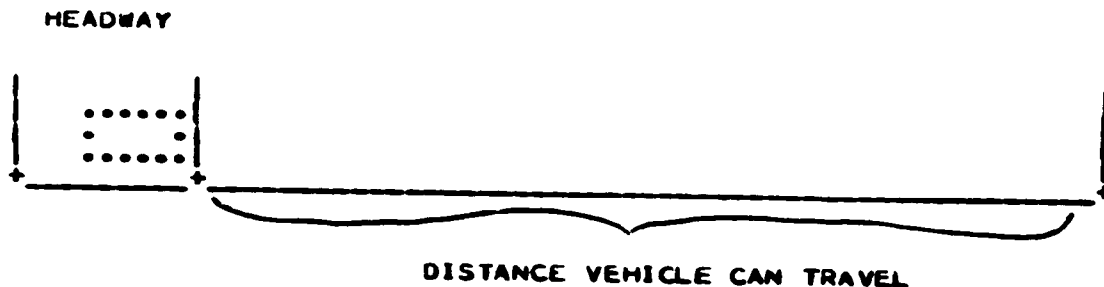
6.2.47.4 Description

EGFNTR processes a vehicle within the Guideway Link Model, upon its completion of headway zone travel, as it is to begin the traversal of the link travel segment. Travel time and the distance to where a vehicle is to move along the link is determined, followed by the scheduling of the vehicle for this computed travel time and the next link event. Traversal of the link is accomplished under control of a fixed headway positioning scheme. Thus a vehicle, if scheduled to travel, moves to the first unoccupied block position available behind the preceding vehicle, if any, on the link.

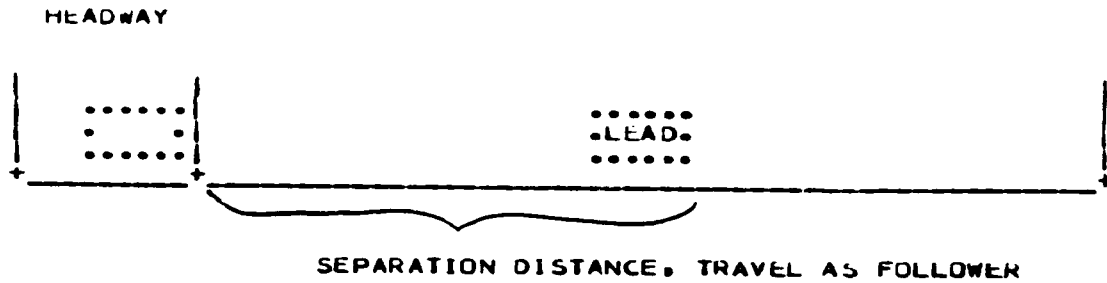
The travel segment entry process begins by determining the velocity variation, if any, that is to be assigned to the vehicle (valid only in asynchronous mode). This variation factor, which is a normally

distributed random number based on a user-supplied standard deviation, along with any existing degradation or penalty factor for the vehicle and merge delay factor for the link is applied to the computation that determines the time at which the vehicle should complete link traversal, based on its current location and the link velocity. The penalty factor adjustment to the time, if any, is performed to reflect degraded vehicle operational status. As long as a vehicle is operating on the guideway in degraded mode or is a member of a degraded entrained group, travel time adjustment will be made. The merge delay factor, used to model increases or decreases in vehicle velocity, is applied to the computed travel time in order to adjust vehicle flow into a merge (for asynchronous mode only). This factor is computed as a result of applying a heuristic merge algorithm on a periodic basis (see description of EAPCMP).

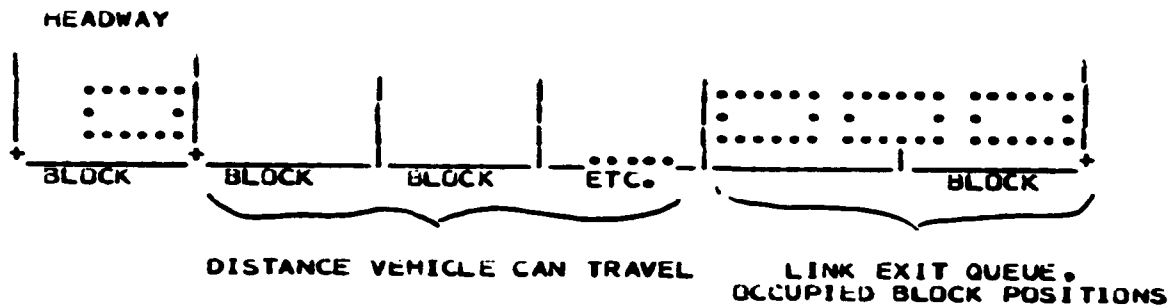
Following the computation of the vehicle's projected exit time from the link (see equation (1) below), it is determined whether scheduling of the vehicle for independent rather than vehicle follower movement is required. If no prior vehicle exists on the link, the vehicle, identified as the next vehicle to exit the link, is scheduled to travel the distance of the entire travel segment (see diagram below). The travel



time required for this event is the difference between the computed projected link exit time for the vehicle and the current clock time (see equation (2) below). If a prior vehicle exists on the link and is in the process of travelling the link, the current vehicle is not scheduled for independent travel but instead travels as a follower (see diagram below) until



the leading vehicle completes travel on the link. If, however, a prior vehicle exists on the link, but it is queued due to congestion, independent vehicle travel is scheduled. The current entering vehicle in this case moves to the block position on the link immediately preceding the blocks occupied by the link exit queue (as shown in the example below). The time required



to travel to the queue is the difference between the computed projected link exit time of the vehicle and the sum of the current clock time and the time required to travel the block lengths occupied by the queue (see equations (3) and (4) below).

Any time a vehicle is scheduled for link travel during link travel segment entry, the next modelling event indicator is set to indicate that final link travel processing is required at completion of the entry process time.

6.2.47.5 PDL

See Appendix A.

6.2.47.6 Decision Tables and Algorithms

The projected link exit time of the vehicle is given by:

$$(1) \quad T_X = C_T + \left(\frac{L_L - L_D}{L_V + V_V} \right) / P + M$$

where:

C_T = current clock time (cu),

L_L = length of link (meters),

L_D = distance traveled by vehicle on link (meters),

L_V = link velocity (m/cu),

V_V = velocity variation (m/cu), (default = 0),

P = penalty factor assigned to the vehicle, (default = 1)
and

M = merge delay factor (cu), (default = 0).

If no prior vehicle exist on the link, the vehicle is scheduled to travel the distance of the entire travel segment, or the remaining link length, in time, T , given by:

$$(2) \quad T = T_X - C_T$$

where:

T_X and C_T as described above.

Any adjustment to the travel time for application of a merge delay, penalty or velocity variation factor is assumed to be imbedded in the projected link exit time.

If a prior vehicle exists on the link, but is queued, the time, T , scheduled for vehicle travel and the distance, D , along the link to where the vehicle can move is given by:

$$(3) \quad T = T_X - C_T - \frac{Q_O \times B_L}{L_V}$$

where:

Q_O = queue occupancy (# of blocks)

B_L = block length (meters) and

T_X , C_T and L_V as described above.

$$(4) \quad D = L_L - (Q_O \times B_L)$$

where:

L_L , Q_O and B_L as described above.

6.2.48 EGFTVL - Fixed Headway Travel Segment Traversal

6.2.48.1 Identification

- o EGFTVL - Fixed Headway Travel Segment Traversal
- o IBM/FSD - May 1, 1978
- o PARAFOR.

6.2.48.2 Argument Dictionary

None

6.2.48.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ENDLNK		L*1	indication that vehicle has reached end of link
GOFIRST		L*1	indication if current vehicle to go first before competing vehicle
LNK		I*2	current link
LNKX		I*2	competing link
OCCLEN		I*4	occupancy length of queue at end of link
SCDNXT		L*1	indication that follower vehicle may be scheduled for travel
SCDTME		I*4	computed travel delta time
STATUS	2	I*2	passenger status of competing vehicles
STN		I*2	station ID at merge
VEH		I*2	competing vehicle at merge

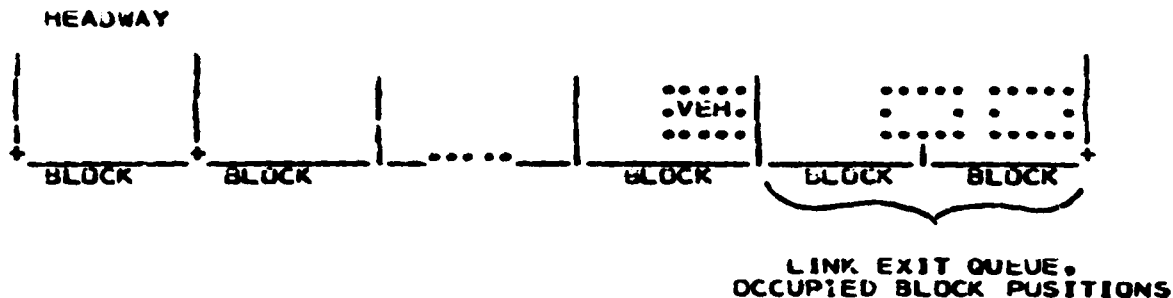
6.2.48.4 Description

EGFTVL provides travel segment completion processing, under fixed headway control, in response to completion of a link travel event by a vehicle. Depending on the conditions that exist, the vehicle is either again made to be a non-scheduled follower, scheduled for a subsequent travel event on the link, queued due to link congestion or noted that

travel to the end of the link has been completed.

All vehicles completing a link travel event initially require processing to determine whether link traversal is complete or the scheduling of another travel event is required. This determination is based on the status of the preceding vehicle, if any, to travel the link with the four possible conditions and appropriate processing summarized as follows:

- a. No lead vehicle was identified to the current vehicle as it travelled the link. It never travelled as a follower and was thus scheduled to traverse the entire link. Link travel completion for the vehicle has therefore occurred.
- b. A lead vehicle was originally identified to the current vehicle but is no longer on the same link, which implies that the current vehicle has already been scheduled to travel the separation distance between the vehicles. Therefore, link completion processing is based on whether or not the current vehicle has travelled the entire length of the link. This is determined by comparing the projected vehicle link exit time to the current clock time. If the difference is greater than zero, the vehicle is scheduled to travel for the remaining delta time (see Equation 1). Otherwise, link travel completion has occurred.
- c. A leading vehicle exists on the same link as the current vehicle and is queued. Thus, it must be determined if the vehicle has also encountered the end of the existing queue of stopped vehicles or if the vehicle may continue to travel on the link towards the queue. The vehicle has reached the queue if it is within a block length of the preceding vehicle or blocks occupied by the queue (as shown below). This determination



is made by comparing the projected vehicle link exit time to the sum of the current time and the time required to travel the occupied queue block positions. If the computed difference is zero, then the vehicle vehicle(s) has reached the queue and must join it (via module EGLWTQ). Otherwise, the vehicle is scheduled to travel for the delta time difference. Its position or distance travelled on the link at the end of that time would be the block location immediately preceding the link exit queue as pictured above (see Equations (2) and (3)).

- d. A leading vehicle which is not queued, still exists on the same link as the current vehicle. For this condition to occur, the preceding vehicle had to have been recently detrained from a train which had completed the link and thus had scheduled any follower (which was the current vehicle) to travel the separation distance. The preceding vehicle, at detrainment, was also scheduled to travel and must currently be in the process of completing travel to the end of the link. When this vehicle reaches the end of the link it will schedule the current vehicle to once again catch up or travel the separation distance. Therefore, the current vehicle need not be scheduled for travel at this time and can again assume a follower relationship.

For the cases above in which it is found that the vehicle has completed travel on the link, one more processing step is required at specific locations and under certain policy constraints in order to signify the full completion of link traversal. At network merges, it must further be determined if any adjustment in vehicle position (in quasi-synchronous mode) or merge planning (in the case of asynchronous mode and priority merging) must be performed as the vehicle plans to leave the link and merge onto the next. As mentioned, under quasi-synchronous control, all vehicles completing traversal of a guideway link which serves as an entry to a network merge, are subject to merge control processing as accomplished in routine EGQMRQ. This process involves the application of a slot or fixed block advance/retard criterion which attempts to eliminate congestion and ensure merge availability for subsequently arriving vehicles. Thus a vehicle may be required to spend additional time on the link.

Merge planning on the other hand, is required if two unqueued vehicles are to arrive at a merge simultaneously when priority merging is to be applied. If the opposing vehicle is completing travel on another guideway link which does not have priority then the current vehicle may plan to continue to move through the merge. Similarly, if the competing vehicle is completing its last event in a station, it must also be determined which vehicle has priority. This is done by referencing the user-supplied local merge priority table, once the

"empty" or "occupied" status is known for each vehicle (see the Decision Table discussion below). If, however, it is determined that for either case the current vehicle does not have the necessary priority to continue movement along its path, it is scheduled to reenter the model at a zero delta time, allowing the priority vehicle to enter the headway of the merge output link.

If, at the conclusion of all of the preceding processing, the vehicle has not been queued or scheduled for any further event on the link, for movement, stalling or retardation purposes, it is assumed that final link travel processing for the vehicle is complete. This results in the signalling of link model completion to the architecture in order that processing for moving the vehicle to its next required entity may be invoked. Additionally, if the vehicle has completed travel or has joined the link exit queue, EGSCHD is invoked in order that any non-travelling vehicle assuming a vehicle follower position behind the current vehicle or train, may be scheduled to independently traverse the link.

6.2.48.5 PDL

See Appendix A.

6.2.48.6 Decision Tables and Algorithms

Travel Times and Distances

If no prior vehicle exists on the link, the vehicle is scheduled to travel the distance to the end of the link in time T given by:

$$(1) T = T_x - C_T$$

where:

T_x = projected link exit time of the vehicle (cu),

C_T = current clock time (cu) and

$T_x > C_T$.

If the computed value of T is less than or equal to zero, then the vehicle is assumed to have completed travel of the link. A negative value for T implies that the vehicle took longer to travel the link than was expected. This could happen if it was following a vehicle that was degraded or travelling at a decreased link velocity.

If a prior vehicle exists on the link, but is queued, the vehicle is scheduled to travel for time, T , and will thus travel to a distance, D , along the link as given by:

$$(2) \quad T = T_x - C_T - \frac{Q_o \times B_L}{L_v}$$

where:

Q_o = queue occupancy (# of blocks),

B_L = block length (meters),

L_v = link velocity (m/cu) and

T_x and C_T as described above.

$$(3) \quad D = L_L - (Q_o \times B_L)$$

where:

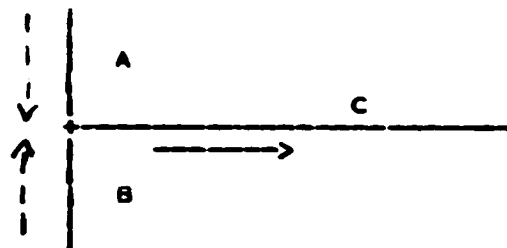
L_L = length of link (meters) and

Q_o and B_L as described above.

A vehicle is scheduled to travel time, T , only if it is a time > 0 . Otherwise it is assumed that the vehicle has encountered and must join the link exit queue.

Merge Priority Decision Making

To determine the priority of vehicles merging from two guideway links onto another, the 1×2 matrix of upstream link pointers assigned to each merge output link is used. If, in the diagram below, links A and B merge onto C, then



two upstream link pointers are identified for link C (as shown) with the first one listed, in this



case 'A', being the link with priority.

Priority determination for vehicles merging at a juncture of a station exit ramp and a guideway link is somewhat more complicated. Here the user-supplied local merge priority table is used. This table, as pictured in the 2 x 2 matrix below, defines the priorities to be assumed by merging vehicles, which are dependent on the vehicle location as well as its "empty" or "occupied" status.

(1,1)	A	C	(1,2)
(2,1)	B	D	(2,2)

- A = table entry (1,1), is the priority of an empty vehicle on the guideway,
- B = table entry (2,1), is the priority of an occupied vehicle on the guideway,
- C = table entry (1,2) is the priority of an empty vehicle on the station exit ramp and

D = table entry (2,2), is the priority of an occupied vehicle on the station exit ramp.

The higher the number assigned to A, B, C or D, the lower the priority. Thus, if the guideway vehicle is empty and the station vehicle is occupied, then a comparison would be made with A and D, with the lower value dictating the vehicle with priority.

6.2.49 EGGNXT - Guideway -- Next Entity

6.2.49.1 Identification

- o EGGNXT - Guideway -- Next Entity
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.49.2 Argument Dictionary

None.

6.2.49.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
LNK	-	I*4	tempory variable for next link assignment

6.2.49.4 Description

The purpose of EGGNXT is to determine the standard next guideway entity for a vehicle traversing a particular path. If the vehicle is currently assigned an alternate route, the next entity becomes the following entry in the alternate path table for as long as the alternate path is defined, at which time the vehicle again assumes a primary path. If, however, the vehicle is currently assigned a primary path as defined in a Successor Link Table (PLSLT), the next entity is selected from the appropriate primary path which is dependent on the path selection method (apriori or real-time) and type (table look-up or algorithmic) in effect. Further description of this entity selection process for each of the four combinations of path method and type is as follows:

1. Real-Time Algorithmic - The next entity is selected from the currently active path table (PLSLT). Additional path selection processing is performed, however, if this next entity indicates that the vehicle is currently at a common diverge point in the network. These diverge points are

places in the network where an alternate path definition exists and the arrival at a common diverge is noted by an entity value >1000 in the first primary path table. This value is of the form

$$X = A \times 1000 + B \text{ where,}$$

A = the pointer to the start of the list of alternate routes that may be used at this point

B = next guideway link to enter on the minimum or primary path.

Alternate path evaluation is then performed, in EGPATh, to determine the best path for the vehicle. This involves comparing the "cost" for the primary path and all alternates defined from the common diverge point to the vehicle's next destination.

2. Real-Time Table Look-Up - The next entity is selected from the currently active minimum path table (PLSLT). Care must be taken to use only the "B" portion of the value (see above) in case a common diverge is encountered. This could occur if a network with alternate path definitions is being used even though it is not appropriate for this particular path selection type.
3. Apriori Table Look-Up - The next entity is selected from the primary path table assigned for the vehicle's use as it was dispatched from a station. It is possible that this path table may not be the currently active PLSLT (a new minimum path table is generated at the time of a link failure) but is the table that is to be used until the vehicle reaches its destination, even though it may encounter congestion or failure along its path. Again the "B" portion of the value (see above), or the next link along the minimum path must be used in case a common diverge is encountered.
4. Apriori Algorithmic - The next entity is selected from the primary path table or the apriori alternate path assigned at the last station. If use of an apriori alternate is indicated, it must be determined at any common diverge point (again denoted by a value >1000 in the first PLSLT) if it is the appropriate location along the guideway to begin use of the alternate. The alternate route is assumed if the current link for a vehicle is an upstream link of the link

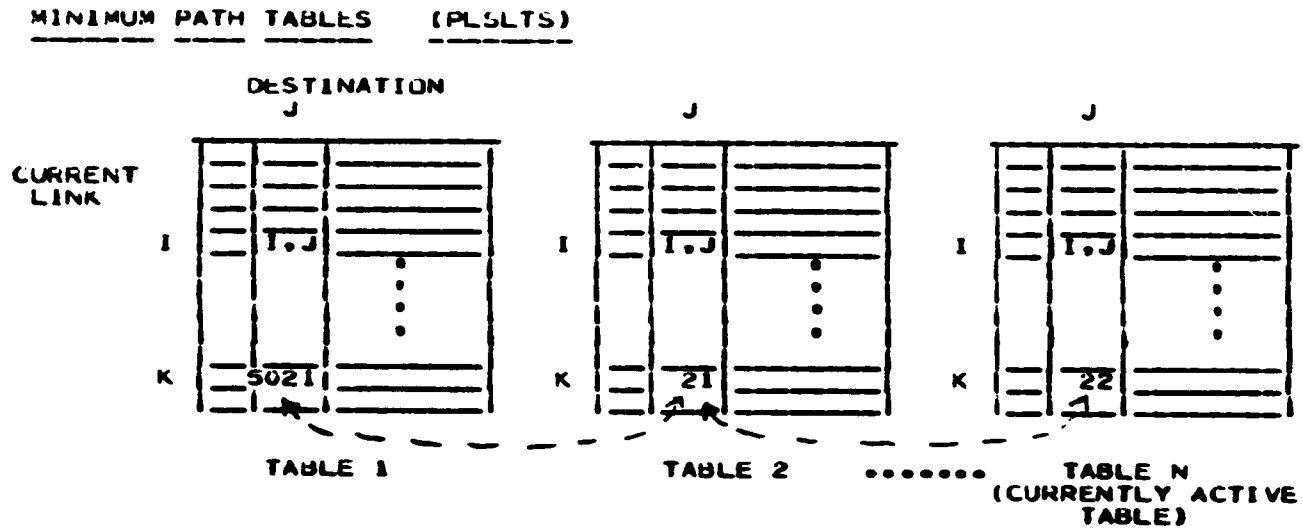
pointed to in the alternate route list by the apriori alternate assignment. The next entity is then set to the first entry of the alternate route in the alternate route table.

6.2.49.5 PDL

See Appendix A.

6.2.49.6 Decision Tables and Algorithms

A description of the tables used for path selection, minimum path tables and the alternate route table, and the method used in selecting the next entity from the appropriate table is outlined below along with examples:



Element (i, j) contains the ID of the next link to travel after link on the minimum or primary path enroute to station j. If element (i, j) is zero, then the station j entry ramp has been reached.

Element (k, j), in the examples above, represents the ID of the next link to travel at the common diverge point at the end of link k for the case where an alternate path definition has been entered. The element ID, if >1000, is of the form:

$A \times 1000 + B$ where

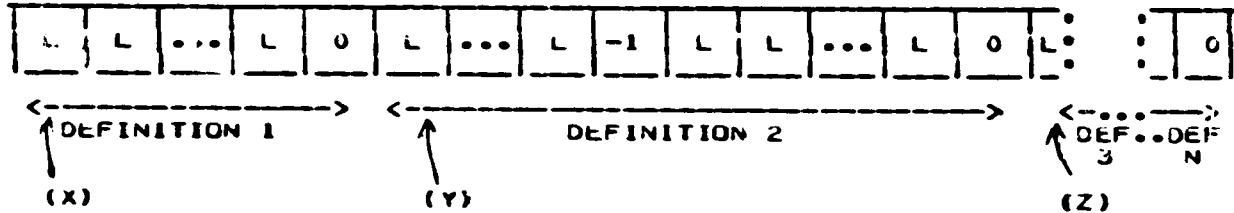
A = the position (relative address) of the start of the list of alternate routes within the alternate route table that may be used at this point,

B = old element (k, j), which is the next guideway link ID on the minimum path.

In table 1, for example, element (k, j) = 5021 ($5 \times 1000 + 21$), indicates that the 5th element in the alternate path table is the starting location of the appropriate alternate path definition to use from the end of link k, with link 21 being the next link along the primary path. Since the common diverge locations are noted only in the first primary path table, which always remains available for reference, it is clear why the corresponding element (k, j) in a succeeding generated PLSLT, Table 2, is merely the value '21'.

Table n, represents an example of what can occur to any element (k, j) should a link failure occur somewhere along the path between link k and station j. The new value, '22' indicates that a different "minimum" path has been determined for use until the recovery of the failed link.

ALTERNATE PATH TABLE



The alternate path table consists of a concatenated list of alternate route definitions (1-n) where:

Ls - the sequence of link ID's comprising an alternate path from a common diverge mode to the destination or back to a location along the minimum path to the destination.

0 - indication of the end of alternate path data (one or more paths) from the common diverge point to the specified destination or point along the minimum path.

-1 - indication that an additional alternate path from the common diverge point to the specified destination exists in the link sequence which follows the -1. The additional link sequences are assumed to follow from the same common diverge point as the first link sequence in the set or definition.

The 'A' portion of the a PLSLT common diverge entry (described above) is equivalent to the entry number in this alternate path table which begins the definition or choice of alternate paths to be considered, which in the above diagram is represented by pointers (x), (y) and (z).

6.2.50 EGHTRN - Train Headway Traversal Processing

6.2.50.1 Identification

- o EGHTRN - Train Headway Traversal Processing
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.50.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
HTIME	-	I*4	(OUTPUT) entrained group headway travel time

6.2.50.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
LNK	-	I*2	temporary link assignment
SYSID	-	I*4	system transaction ID used to schedule travel completion of the headway zone

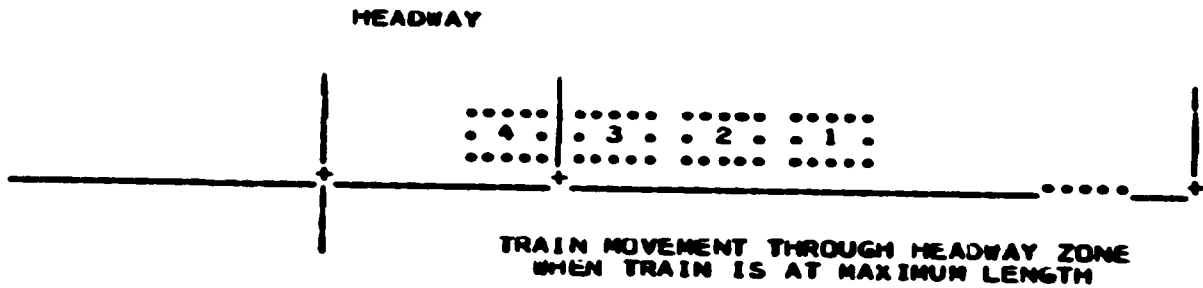
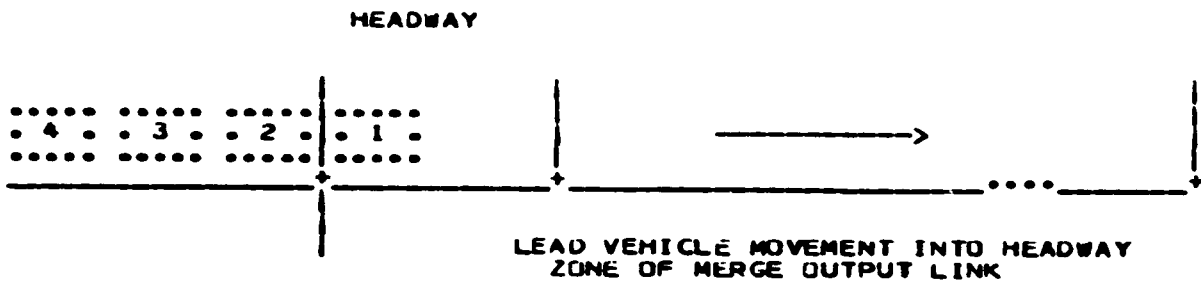
6.2.50.4 Description

EGHTRN processes train entry onto a guideway link for traversal of the headway segment. This involves determining the required headway zone travel time, 'HTIME', for the remainder of the entrained group (the lead vehicle already having been processed), while ensuring proper spacing between the train and any subsequent vehicle to enter the link.

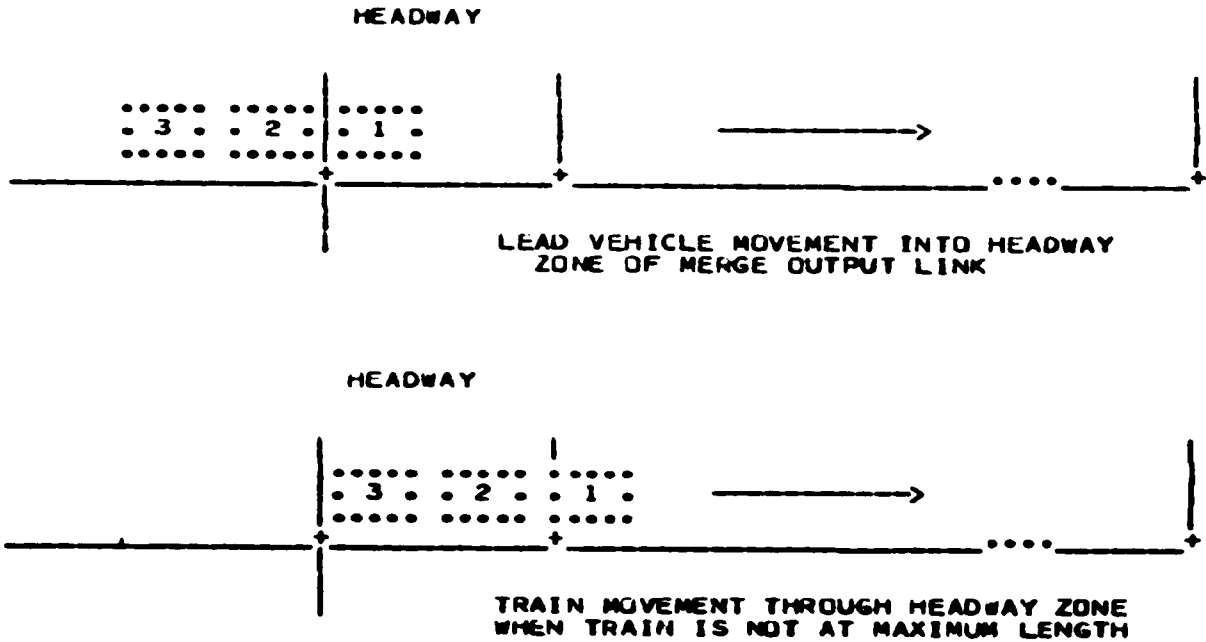
The determination of the required completion time is based on the location and size of the train and the position regulation scheme (fixed or variable headway) in effect. The particular combination of these conditions will result in one of the following:

- a. The train is at a merge output link and dynamic entrainment is enabled. Thus, the lead vehicle, as shown in the first illustration below, has only moved a vehicle length distance into the headway (see EGLNTR).

1. If the train is already at maximum size then further entrainment will not be attempted. Thus the train may be scheduled to move such that the last vehicle will be positioned at the full nominal headway distance (see Equation (1)), as shown in the second example.

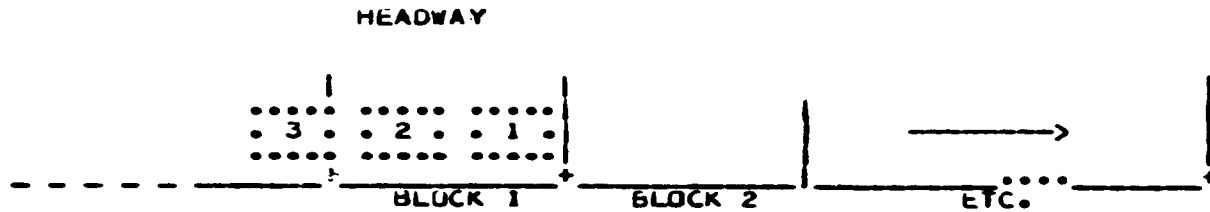


2. If the train is not at maximum size, as shown in the first example below, then it is necessary to allow for further entrainment. Thus the train must be scheduled to move such that the last vehicle is moved to the reduced headway position on the link, (see Equation (2)), as shown in the second example below.

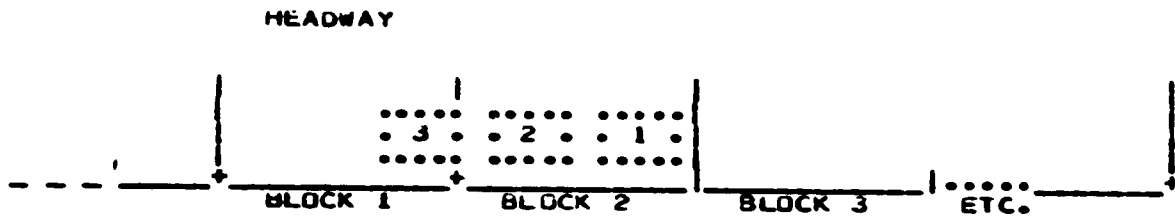


- b. The link is not a merge output link and /or dynamic entrainment is not enabled. Therefore the lead vehicle has already been scheduled to travel the full headway distance (see EGLNTR).
1. If fixed headway position regulation is in effect, it can be assumed that the lead vehicle has entered one full block distance, as shown in the first example below. Since a block occupancy requirement

exists for the train, the train can be scheduled to move for the remainder of the block requirement distance, (see Equation (3)), as indicated in the second example below.



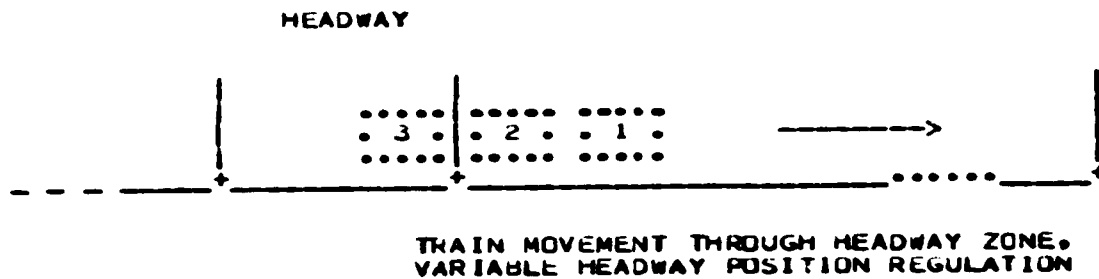
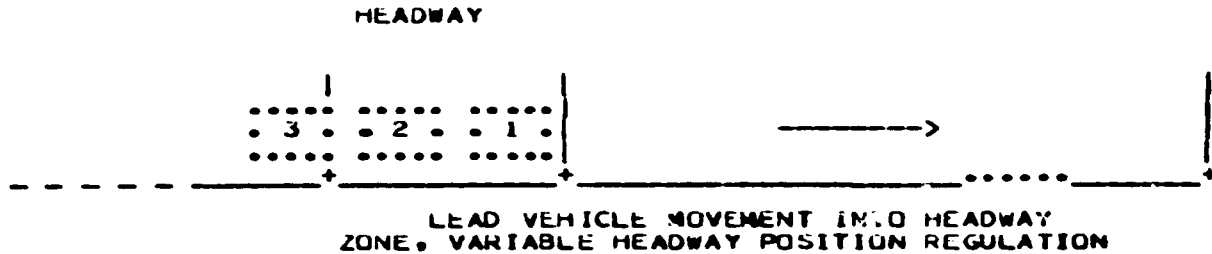
LEAD VEHICLE MOVEMENT INTO HEADWAY ZONE.
FIXED BLOCK POSITION REGULATION



TRAIN MOVEMENT THROUGH HEADWAY ZONE.
FIXED BLOCK POSITION REGULATION

2. If variable headway position regulation is in effect, it can be assumed that the lead vehicle has moved to the nominal headway distance, as shown in the first example below. The train must then be scheduled to

move for the remainder of the train length distance (minus one vehicle length for the lead - see Equation (2)), as indicated in the second example below.



Uninterrupted movement of vehicle trains in the DESM is ensured by continual occupancy of the headway zone until the last vehicle of the train has entered the link for traversal. This is accomplished by scheduling a special system service transaction to free the headway zone at the computed train headway traversal time, as described above and as listed in the algorithms below. This is performed to ensure that competing vehicles do not become intermixed with the train on merge output links. If the competing vehicles are eligible for entrainment, they are considered after passage of the last vehicle of the entrained group through the merge.

6.2.50.5 PDL

See Appendix A.

6.2.50.6 Decision Tables and Algorithms

Train Headway Traversal Times

In the case of a train entering a merge output link when dynamic entrainment is enabled, headway traversal time for the remainder of the train is given by:

$$(1) H_{TR} = \frac{(T_L - 2) \times V_L}{L_V} + L_H$$

if the entrained group is already at the maximum train size where:

T_L = train length or number of vehicles in the entrained group,

V_L = vehicle length (meters),

L_V = link velocity (m/cu) and

L_H = nominal link headway time (cu).

$$(2) H_{TR} = \frac{(T_L - 1) \times V_L}{L_V}$$

if the entrained group is not at maximum train size and

T_L , V_L , and L_V as described above.

Equation (1) above, implies that the last vehicle of the train is assumed to move for the full nominal link headway time with the intermediate vehicles assuming a continual reduced headway movement. The headway travel time shown by equation (2), however, would position the last vehicle of a train such that further vehicle entrainment is possible (next vehicle to enter link is within a vehicle length).

If the link is not a merge output link or if dynamic entrainment is not enabled, then the headway traversal time for the remainder of the train is given by:

$$(3) H_{TR} = (T_0 - 1) \times L_H$$

for a fixed headway positioning scheme

where:

T_0 = train occupancy (blocks), and

L_H as described above.

Exact spacing between vehicles in this case is not critical because line speed entrainment is not possible. Since the number of blocks required for the entrained group (block occupancy) is already established it is assumed that the train continues to travel as a block.

Variable headway processing time for the remainder of a train when the link is not a merge output link or if dynamic entrainment is not enabled is satisfied by Equation (2) above.

6.2 51 EGLEAV - Guideway Link Exit

6.2.51.1 Identification

- o EGLEAV - Process Vehicle Link Exit
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.51.2 Argument Dictionary

None

6.2.51.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
FOUND	-	L*1	Flag indicating path found through on-line station
GL	-	I*2	Guideway link for vehicle
NPASS	-	I*2	Number passengers aboard vehicle
NTRPS	-	I*4	Number trips aboard vehicle
RTE	-	I*2	Assigned vehicle route
STN	-	I*4	Station ID at link exit
TIME	-	R*4	Current simulation clock time in seconds
TOCC	-	I*4	Occupancy count for vehicle or train
TRP	-	I*4	Trip ID
VEH	-	I*4	Vehicle ID
VEH1	-	I*4	Vehicle ID
VTRPID	-	I*4	Trip ID of trips on vehicle

6.2.51.4 Description

Guideway link exit processing is invoked by the architecture for processing vehicles which have completed traversal of their current link and must begin travel on the next link of their route.

Upon entry, guideway link and individual vehicle travel statistics are updated to reflect completion of the travel distance and time associated with the link. A determination is then made as to whether vehicle station entry logging is required. Logging is performed by invoking EGVLOG if a station entry diverge exists at link exit and the station corresponds to the user specified station for vehicle arrival recording. A prompt event is then scheduled for the link to cause dequeuing of any waiting upstream vehicles to occur. The next vehicle to exit the guideway link is updated to reflect the vehicle immediately following the exiting vehicle or train. The follower pointer of the vehicle or train is then zeroed in preparation for entry onto the next network entity. If no following vehicle exists on the link, the ID of the last vehicle to enter the link is also zeroed. The occupancy of the link is decreased by the number of occupancy positions required for the vehicle and any entrained followers based upon the position regulation scheme in effect for the simulation run. Statistics related to guideway link occupancy are updated to reflect exit of the vehicle or train.

If an unclear tow path exists, EGLEAV tests whether the tow path will become clear if this vehicle is leaving a link on the tow path but not entering another link on the tow path. If the path does clear, the restart is scheduled at the time calculated for the tow vehicle to reach and restart the failed vehicle.

6.2.51.5 PDL

See Appendix A.

6.1.51.6 Decision Tables and Algorithms

- a. Vehicle occupancy requirements - See EANSCD

6.2.52 EGLMDL - Guideway Link Model Control Routine

6.2.52.1 Identification

- o EGLMDL - Guideway Link Model Control
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.52.2 Argument Dictionary

None.

6.2.52.3 Local Vocabulary Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
HTIME	-	I*4	entrained group headway zone travel time
LNK	-	I*2	temporary link assignment

6.2.52.4 Description

EGLMDL controls the overall movement of vehicles along the guideway network links or within the guideway link model subsystem. Each vehicle transaction in the simulator contains a micro event indicator which is used in controlling the event processing. Upon entry into the guideway link model, this event indicator is interrogated to determine the processing required for the vehicle as follows:

1. Initial link or headway zone entry
2. Headway zone travel completion and travel segment entry
3. Travel segment traversal.

Each time the vehicle is rescheduled, as the result of model processing, the event indicator is reset to indicate the vehicle processing required upon reentry into the model.

As listed, the first link event required for a vehicle is the traversal of the headway segment. All vehicles are required to travel this segment of the link which is used to ensure proper spacing between vehicles. EGLNTR is the module invoked to provide this link entry processing for a vehicle.

Upon vehicle completion of the headway zone and before the link traversal event is begun, a number of preliminary functions are performed (unless the vehicle(s) assumed an advance maneuver as it entered the link, in which case the appropriate functions have already been handled). If the vehicle being processed is a single vehicle, then the headway zone for the link is indicated as available. In addition, a prompt is scheduled which will initiate the movement of any queued vehicles on the upstream link(s), (since movement on to the current link would not have been possible as long as the headway zone had been occupied). Special processing is applied for link entry of all of the following vehicles of an entrained group in EGHTRN. The purpose of this module is to ensure proper spacing between the vehicle train and the next vehicle to enter the link and to provide uninterrupted movement of the train onto the link. Thus, continual occupancy of the headway zone must be achieved until the last vehicle of the train has entered the link. Also, as part of the unique processing for trains, the link status variables containing the ID and projected entry time onto the travel segment of the last vehicle to enter the link, are assigned the ID of the last vehicle in the train and the computed train headway completion time, respectively.

All vehicles about to enter the link travel segment assume a vehicle follower relationship with the "lead" or prior vehicle to enter the link, if still present. This is established by identifying the presence of the following vehicle to the vehicle immediately ahead on the link and vice versa. When this is complete, dynamic entrainment is attempted, by invoking EGETRN.

Once the above headway zone maintenance and vehicle relationships have been established, the vehicle or train may begin the second link model micro event, the traversal of the travel segment. Travel of this segment can be accomplished under control of either a variable (in module EGVNTR) or fixed (in module EGFNTR) headway positioning scheme. (In addition, the provision for recognition and application of a user-defined position regulation scheme for vehicle link movement is supported.) The difference between the positioning schemes exists in the relative position on the link to which a vehicle can move when link traversal is required. In the modelling of variable headway movement, vehicles can move to within a headway (safe stopping distance) of the preceding vehicle on the link to account for possible vehicle deceleration requirements, although not explicitly modelled in the DESM.

In the case of fixed headway modeling the vehicle moves to the first unoccupied block position (fixed length of guideway) available behind the preceding vehicle on the link. This difference is illustrated in Figure 6-4.

Since the modeling of travel on the guideway links in the DESM is designed to minimize transaction scheduling requirements and at the same time reflect realism in vehicle movement, the processing of the travel event may or may not require transaction scheduling. This will depend on the existence of and status of an immediately preceding vehicle or "lead" on the link. Independent movement or scheduling of a travel event of the "follower" is only required when it is determined that the time separation between vehicles can no longer be maintained. This action may be required immediately upon link travel segment entry or in the future as the result of the "lead" vehicle encountering link congestion or completing link traversal. Thus, at any one time, only one vehicle on the guideway link requires travel event scheduling, thereby reducing simulation overhead requirements.

As vehicles reenter the link model for the processing of the third event, a determination is made in module EGTVL or EGFTVL for variable or fixed headway regulation schemes, respectively, as to whether link travel for the vehicle is complete. If the vehicle has reached the end of the link exit queue or traversed the entire link, the scheduling of any following vehicle travelling as an unscheduled follower, is performed. Otherwise, the vehicle is rescheduled to travel as far as it can on the link to either reach the end of the link exit queue or the end of the link. If at completion of a link traversal event, the vehicle encounters the end of the link queue, it joins the queue and moves on the link at the rate of dissipation of the queue. No rescheduling of the vehicle is required until it becomes the lead vehicle in the queue ready for link exit and entry of its next modelling entity.

Link traversal of a train is similar to that discussed above with the exception that all intermediate or follower vehicles never require independent link traversal processing. They assume a constant follower relationship as long as they are entrained. Allowance is made, at the time of lead vehicle processing, for the passage of followers in the entrained group through the headway zone as well as along the link travel segment. When the train completes link traversal or encounters an exit queue, any following vehicle chained to the rear vehicle of the train is scheduled for travel to close up the gap.

In order to accommodate train traversal of the headway zone, the guideway link model processes headway zone completion via recognition of a system service transaction which causes the headway zone of the specified link to become unoccupied or "free". The event, which appears as the fourth model micro event, is scheduled as a train begins movement through the headway zone. As the headway zone becomes unoccupied, similar to single vehicle headway zone completion, a prompt request is issued to cause vehicles, which have been delayed due to headway constraints, to be scheduled for entry.

6.2.52.5 PDL

See Appendix A.

6.2.52.6 Decision Tables and Algorithms

None.

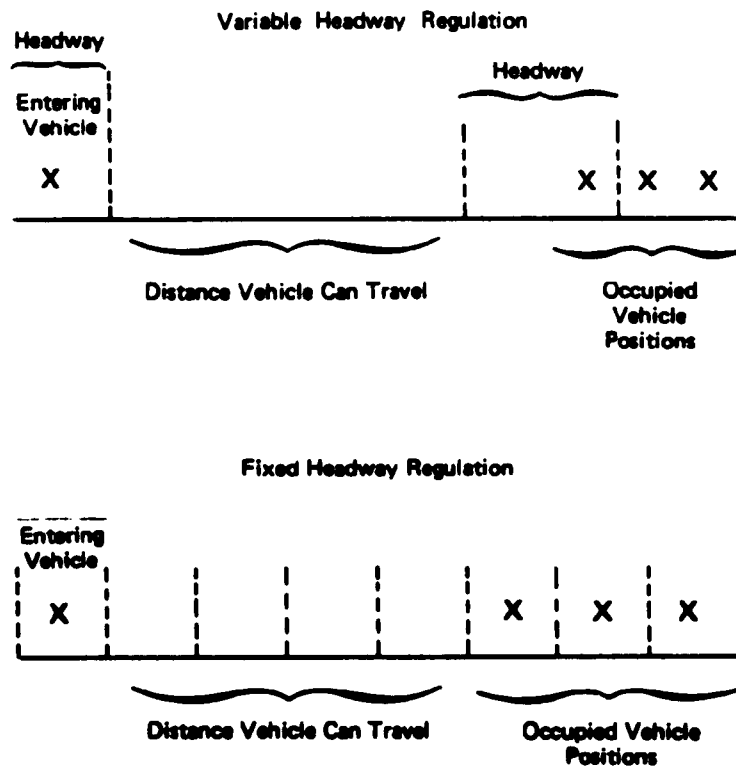


FIGURE 6-4. DESM HEADWAY REGULATION

6.2.53 EGLNTR - Guideway Link Entry Processing

6.2.53.1 Identification

- o EGLNTR - Guideway Link Entry Processing
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.53.2 Argument Dictionary

None.

6.2.53.3 Local Variable Dictionary

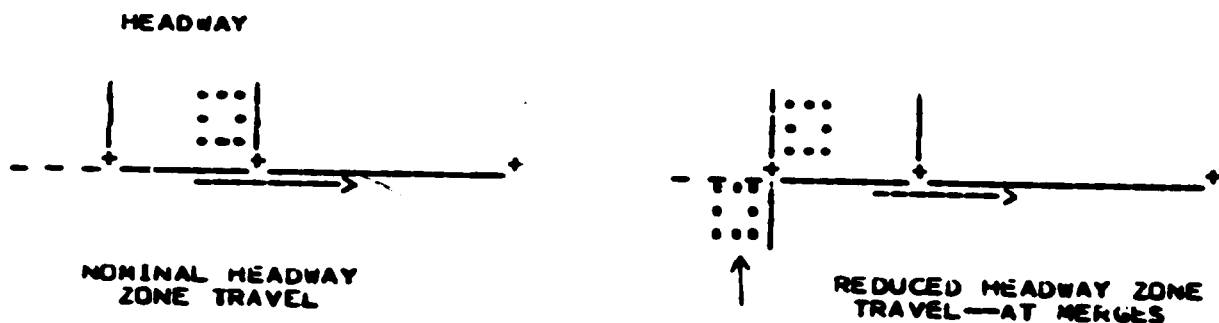
VARIABLE	DIM	TYPE	DESCRIPTION
HDTIME		R*4	headway zone travel time
LNK		I*2	current link
NPASS		I*2	number of passengers entering the link on current vehicles
NUMVEH		I*2	number of vehicles entering the link
RTE		I*2	vehicle route assignment
SCDTME		I*4	travel delta time used for schedule macro
VEH		I*2	temporary variable for vehicle ID

6.2.53.4 Description

EGLNTR is invoked by the Guideway Link Model upon initial entry of a vehicle or train onto a network link. Entry processing involves determining the required headway zone travel time for the vehicle, establishing the presence of the vehicle(s) on the link and scheduling the vehicle for its headway zone travel event completion.

Headway zone travel time is used in the DESM to ensure proper spacing between vehicles entering a guideway link. Thus, subsequent vehicle entry is precluded during occupancy of the headway zone by a previous entering vehicle. In the typical case, this travel time is given as the time required to maintain safe stopping distances between vehicles travelling at nominal link velocity. For variable headway travel, this headway time is a user-specified value and for fixed headway travel, it is the time required to travel the user-specified fixed block length at link travel speed (see Equation 1). However, headway zone travel is adjusted for certain cases in the DESM to reduce spacing between vehicles on merge output links. This is done in order to allow line-speed entrainment of merging vehicles, if enabled, and to ensure proper spacing between vehicles of an entrained group moving from one guideway link to another.

The reduction of headway, in the case of any vehicle's entry onto a merge output link (if dynamic entrainment is allowed), is achieved by assigning to the headway travel time the time it takes to travel a single vehicle length. By assuming this reduced time, the vehicle will, at the time of its headway completion, be positioned on the link such that any succeeding vehicle immediately entering the link will be able to entrain to the current vehicle. Notice the position of the vehicles, in the diagrams below, after completion of the nominal and reduced headway zone travel times.



(The special processing applied at link entry to all of the following vehicles of an entrained group, in order to ensure proper spacing between each vehicle of the entrained group and between the last vehicle of the train and the next vehicle to enter the link, is described in EGHTRN.)

Prior to scheduling the vehicle for headway traversal, the headway distance is computed (see Equation (2)) followed by a further adjustment to the headway zone travel time only if the vehicle (single or lead vehicle of a train) is currently stopped in a queue. This adjustment is the addition of a reaction time used for modeling acceleration to line speed from a stop. The queued vehicle(s), now ready to move into the headway zone is noted as no longer queued.

Routine EAPFEL is invoked to provide the actual scheduling of the vehicle for the computed headway zone travel time. The occupied indicator for the headway is then set and the link occupancy value is incremented by the number of vehicle positions to be occupied by the vehicle(s) entering the link. The remaining specific link data items assigned are the ID and projected entry time onto the travel segment of the last (current) vehicle to enter the link.

The vehicle data assignments made as part of link entry include the following:

- o new current link assignment (also assigned to each member of an entrained group),
- o link entry clock time (also adjusted and assigned to each member of an entrained group, see Equation (3) below),
- o leading vehicle ID, if last vehicle to enter the link is still present and
- o separation time between the travel segment entry of the current and leading vehicle, if any (see Equation (4)).

Link entry processing will vary from the above if the vehicle or train is to assume an advance maneuver as it enters a merge output link. This instantaneous advance adjustment or repositioning onto the link, accomplished by invoking EGQNTN, is applied to a vehicle(s) in an attempt to ensure merge availability for subsequently entering vehicles under quasi-synchronous control, as described in EGQNTN.

As each vehicle or train enters or "advances" onto a link EGLNTR accumulates a number of status and historical link entry statistics (see Table C-1 in Appendix C).

6.2.53.5 PDL

See Appendix A.

6.2.53.6 Decision Tables and Algorithms

The reduced headway travel time, computed for vehicles entering merge output links when dynamic entrainment is enabled, is given by:

$$1. \quad H_T = \frac{V_L}{L_V}$$

where:

V_L = vehicle length (meters) and

L_V = link velocity (m/cu).

The vehicle headway distance traveled is given by:

$$2. \quad H_D = H_T \times L_V$$

where:

H_T = the nominal or computed reduced headway time and

L_V as described above.

The link entry time or the clock time at which the vehicle(s) started into the headway zone is given as (for each vehicle):

$$3. \quad V_E = C_T + \left((V_N - 1) \times \frac{V_L}{L_V} \right)$$

where:

C_T = current clock time,

V_N = relative train position of vehicle or vehicle number (=1 for single vehicle or lead vehicle of train, =2 for second vehicle, etc.) and

V_L and L_V as described above.

The separation time between travel segment entry or headway zone completion time of the current vehicle and its predecessor, if still on the link, is given by:

$$4. \quad V_S = C_T + (H_T + R_T) - E_V$$

where:

R_T = reaction time from a stopped position, if any (cu),

E_V = clock time of last vehicle entry along travel segment and

C_T and H_T as described above.

6.2.54 EGLWTQ - Guideway Link Queuing

6.2.54.1 Identification

- o EGLWTQ - Enter a Vehicle into a Guideway Link Exit Queue
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.54.2 Argument Dictionary

None.

6.2.54.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
GL	-	I*2	Guideway Link ID
VEH	-	I*4	Vehicle ID
TOCC	-	I*4	Vehicle or train occupancy requirement

6.2.54.4 Description

Guideway link queuing is invoked by the architecture to queue as waiting, vehicle transactions which are precluded for next event processing due to network failures or congestion. The queuing process involves the following:

- o Indicating queued status for the vehicle and any entrained followers
- o Enqueuing the vehicle or lead vehicle of a train in the link exit queue
- o Updating link queue occupancy based upon block occupancy requirements or train length depending upon the vehicle positioning scheme in effect.

6.2.54.5 PDL

See Appendix A.

6.2.54.6 Decision Table and Algorithms

None.

6.2.55 EGNEXT - Next Guideway Entity Determination Control

6.2.55.1 Identification

- o EGNEXT - Next Guideway Entity Determination Control
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.55.2 Argument Dictionary

None.

6.2.55.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
FOUND	-	L*1	Flag indicating fail card match
LNK	-	I*2	Current link
TIME	-	R*4	Current simulation clock time in seconds
VEH	-	I*2	Computational variable for vehicle ID

6.2.55.4 Description

The purpose of EGNEXT is to control the processing of the determination of the next entity which a vehicle(s), currently on the guideway or about to enter the guideway, must enter to continue traversal to a required destination. The type of entity is identified to the architecture by returning a positive value for next link in the case of a guideway segment, a negative value for station ID, if station entry is required, or a "0" value if it is not feasible to continue with the next entity determination due to the exit failure of the current guideway link.

After first verifying that the current link exit is not failed, EGNEXT is invoked to determine the next vehicle entity from the appropriate tables based on the vehicle's current location, destination and the path selection method and type in effect. At certain network locations, EGNEXT must further determine if the given standard next entity is to be altered for the vehicle or other vehicles of an entrained group, depending on the existing conditions, service policy and system options (e.g., dynamic entrainment, etc.) in effect as follows:

a. Demand responsive service. Further processing for final next entity determination may be required in this mode of operation for vehicles at network and station diverge points. If a station, on-line or off-line, does exist at the end of a current link, EGRESV is invoked to determine if any unscheduled diverting into the station or rerouting around a scheduled station is required. At network diverge points, EGTRNC is invoked when the current vehicle being processed is the lead vehicle of a train and when dynamic detrainment is in effect. This routine will execute detrainment of the vehicles if a difference exists in the next entity for members of the entrained group.

b. Scheduled service or timeout/group demand responsive service. In this mode of operation it is necessary to further check, at station diverge points only, if the next entity for the vehicle or entrained group is to be altered. If the vehicle(s) is to enter a station and demand stop processing is enabled, EGDSTP is invoked to determine if the vehicle may be rerouted around the station. This processing is not invoked, however, if the vehicle is failed. If the vehicle is to bypass stations as part of a response to a failure condition, the vehicle will enter the station only if it has passengers onboard so it may deboard all passengers at the first stop as transfers. On the other hand, vehicles not scheduled to stop, at on-line stations only, are diverted into the station, by resetting their next station stop indication, if they are failed and have passengers on board.

The next entity assigned to the vehicle, and each following vehicle in the case of a train, after the above processing has been accomplished, will be:

- o the guideway link number, if a guideway link is the next segment to be traversed or,
- o The station ID (negated) if the vehicle is to either immediately enter an off-line station or stop at an on-line station.

If the link exit is failed or marked to degrade the next vehicle, EGNEXT searches the active FAIL cards to find data for this failure. The captured vehicle is linked to the failure conditions, the degradation factor is applied for the vehicle failure case, and the failure detection event is scheduled.

6.2.55.5 PDL

See Appendix A.

6.2.55.6 Decision Tables and Algorithms

None.

5.2.56 EGPATH - Real-Time Path Selection

6.2.56.1 Identification

- o EGPATH - Real-Time Path Selection
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.56.2 Argument Dictionary

None.

6.2.56.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
BCOST	-	R*4	Cost of current best path
LNK	-	I*4	Current link, starting link to use in tracing primary path
NLNK	-	I*4	PLSLT common diverge value
PTR	-	I*4	Computational variable for alternate path table entry number
PTRSV	-	I*4	Starting pointer for the particular alternate route

6.2.56.4 Description

The real-time path selection processing in EGPATH is performed if the next entity for the vehicle, selected from the PLSLT (minimum path table), indicates that the vehicle is currently at a common diverge in the network. EGPATH controls the cost computation and comparison for each alternative path available at the diverge in order to determine the best path for a vehicle to its destination.

The first path evaluated from the current link is the primary path (EGPRMY). The "cost" of this path becomes the current best or lowest path cost which is the basis for further cost comparison. The starting

point of the alternate route definition to use in the alternate route table is then extracted from the common diverge value (see description in EGGNXT) and the pointer is given to EGALT in order to compute the cost of the particular alternate path. As the cost is determined for this path and for any other alternate defined as part of the alternate path definition, it is checked against the previously lowest computed cost to determine if that route is now the current best alternative. Once the cost computation process is completed for all possible paths from the common diverge to the vehicle's next destination, the next entity is reselected based on the resulting minimum cost path. This then becomes either the next guideway link along the primary path or the first link listed in the appropriate alternate path in the alternate path table.

(Note: If other common diverges are encountered during the cost computation processing for a possible route, the alternate paths indicated at these points are ignored.)

6.2.56.5 PDL

See Appendix A.

6.2.56.6 Decision Tables and Algorithms

None.

6.2.57 EGPRMY - Primary Path Cost Computation

6.2.57.1 Identification

- o EGPRMY - Primary Path Cost Computation
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.57.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
LNK	-	I*4	Starting link and current link to use in tracing primary path

6.2.57.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
GLOCCR	-	REAL*4	Temporary variable for link occupancy

6.2.57.4 Description

EGPRMY computes the cost of a primary route contained in the current Successor Link Table (PLSLT), by tracing through the table from 'LNK' to the vehicle's destination. This primary route computation may be done either from a common diverge point or from the end of an alternate path definition, at which point the primary path is then assumed to the destination. As each link along the path is identified, the cost as determined by the user-specified algorithm option of:

- a. link nominal travel time
- b. link length
- c. utilization or
- d. a weighted combination of (a) and (c) is accumulated. This accumulated cost is merely added to the cost already determined, in the case of initial alternate path processing.

If a failed link is encountered, the path is excluded from further consideration by adding an artificially high failure factor to the cost of the route. If another common diverge is encountered in tracing the path, it is ignored. Only the primary path is used.

6.2.57.5 PDL

See Appendix A.

6.2.57.6 Decision Tables and Algorithms

The cost accumulation, by link, is based on the path selection algorithm type specified. The following is a further description of each algorithm type as it is used in determining the cost:

- a. Link nominal travel time
 1. Cost = Σ nominal travel times of each link along the primary path
- b. Link length
 2. Cost = Σ link lengths of each link along the primary path
- c. Link utilization
 3. Cost = Σ current link occupancy/link capacity, of each link along the primary path
- d. Weighted combination of travel time and utilization
 4. Cost = Σ (nominal travel time * weighting factor for travel time) + (utilization * weighting factor for utilization) of each link along the primary path.

6.2.58 EGOMRG - Quasi-Synchronous Guideway Control

6.2.58.1 Identification

- o EGQMRG - Perform Quasi-Synchronous Control
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.58.2 Argument Dictionary

None.

6.2.58.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
AREQ	-	L*1	Logic variable for indicating advance required
LINK	-	I*2	Link downstream of merge
NUM	2	I*2	Occupancy slots approaching merge on input legs
OCC	-	I*2	Occupied slot position count
POS	-	I*2	Index into merge priority table for current vehicle entity
POS1	-	I*2	Opposing entity index into merge priority table
RREQ	-	L*1	Logic variable for indicating retardation required
STATUS	2	I*2	Local exit merge empty/occupied indicators
SW	-	L*1	Logic variable for indicating opposing vehicle priority at local merge

VARIABLE	DIM	TYPE	DESCRIPTION
TIME	-	I*4	Slot time in C.U.
VEH	-	I*2	ID of opposing vehicle
VEHICLE	-	I*4	ID of following vehicle

6.2.58.4 Description

In quasi-synchronous control mode, all vehicles completing traversal of a guideway link or station exit ramp which serves as an entry to a network merge, are subject to merge control processing performed by EGQMRG. This control process involves the application of a slot advance/retard criteria which attempts to eliminate congestion and ensure merge availability for subsequently arriving vehicles. If uninterrupted traversal of the merge at current line speed is not possible, the vehicle is rescheduled to retry link entry at a delta time given by the time for traversing a slot distance on the guideway as defined by the fixed headway interval. This rescheduling corresponds to a slot retardation maneuver which causes the vehicle to decelerate rather than queue on the guideway. Vehicles which can successfully traverse the network merge, are subject to a slot advance criteria which attempts to ensure merge availability by increasing the separation distance between itself and subsequently arriving vehicles at the merge. The amount of advance or vehicle acceleration is based upon the density of traffic flow approaching the merge on competing input legs to the merge.

Upon entry to the routine, the immediate follower of the vehicle attempting merge traversal and the ID of any competing vehicle approaching the merge are identified for subsequent use in examining merge traversal requirements. A determination is then made as to whether unimpeded merge traversal can be accomplished based upon merge output headway requirements, capacity constraints and merge priority assignments. If the headway zone of the merge output link is occupied and simultaneous headway zone completion is not indicated, vehicle retardation is required to ensure proper headway separation through the network merge. If simultaneous headway zone travel completion is found and the vehicle currently traversing the headway zone is unentrained, zero time rescheduling of the merge process is performed to allow completion of processing for the prior vehicle to occur before attempting merge entry for the current vehicle. If headway constraints are satisfied, output link capacity is checked to determine if entry of the current vehicle and any entrained followers can be performed. If entry causes capacity constraints to be violated, current merge traversal is prohibited and the requirement for vehicle retardation is indicated. Otherwise, merge priority requirements are examined to determine if the current vehicle is eligible for

immediate merge entry. If the vehicle has already performed the maximum number of retardation maneuvers, it assumes priority merge status and subsequent priority checking is bypassed. If a competing vehicle exists on the opposing input leg of the merge and it has performed the maximum number of retardation maneuvers allowed, current vehicle merge entry is precluded. Otherwise, normal merge priority assignments based upon the merge policy in effect are examined as follows:

- o FIFO - The earliest arriving vehicle at the merge, as indicated by the number of previous retardation maneuvers performed, is determined. If the opposing vehicle has waited longer for merge entry, the current vehicle is precluded from merge entry and a retardation maneuver is indicated.
- o Priority - If either vehicle waiting at the merge is located at the exit of a network station, local exit merge priority determination is performed. This involves establishing the empty/occupied status of the competing vehicles and examining the 2 x 2 user specified local merge priority table for priority assignment based upon vehicle location and occupancy status. If the competing vehicle is assigned higher priority, the current vehicle is precluded from merge entry and must undergo a retardation maneuver. If competing guideway links form the network merge, then priority assigned to each competing link is examined to determine retardation requirements. In this case the priority link is indicated as the first entry in the upstream connectivity specification for the merge output link.

The merge traversal checking performed above results in the setting of either a retardation or advance maneuver indicator. If retardation is indicated the retardation maneuver count for the vehicle is incremented and checked against the **maximum maneuver limit**. The retardation process can be applied to any vehicle precluded from merge entry up to a maximum slot retardation or 'slip' limit. The occurrence of successive retardation maneuvers can occur if one link is prioritized and simultaneity of vehicle positioning on the merge approach link exists. Once the maximum retardation limit is reached for a vehicle it is prioritized to allow it to attempt merge traversal on its retry attempt. If the vehicle cannot enter the merge on its next retry attempt, a congestion situation has developed and it is forced to queue, awaiting the dissipation of congestion at the network merge. Following vehicles on the link are affected by the retardation in the following manner. If the following vehicle is separated by a distance greater than one headway, the separation time between it and the retarded vehicle is adjusted to reflect the effect of headway reduction between the vehicles produced by the retardation maneuver. If the headway separation between the

vehicles is currently at a minimum, the following vehicle is assumed to have also been subject to a retardation maneuver as are any subsequent contiguous following vehicles also separated by minimum headway. Since, in the guideway link model, consecutively entering vehicles on a link assume a vehicle follower relationship, this adjustment process need only be applied until either all contiguous vehicles on the link have recorded a retardation maneuver or a headway separation adjustment has been made between consecutively following vehicles. The effect of the retardation process is shown in Figure 6-5.

If the current vehicle is eligible for merge traversal, as indicated by the advance requirement indicator, vehicle advance computation is performed if not precluded by user option. The density of the vehicles following the merging vehicle on its merge approach link and the density of vehicles on the competing approach link are examined to determine if congestion free merging is possible. If the combined density on the two competing links requires the vehicle to advance on the merge output link in order to allow conflict free merging, it enters the output link and assumes a position corresponding to the computed slot advance. The advance computation is constrained by the separation distance between the entering vehicle and the preceding vehicle on the link and the specified maximum slot advance/retard limit. The desired advance maneuver for the merging vehicle is computed as

$$\sum_{i=1}^M V_s$$

where:

M - Maximum advance limit

V_s - Predicted simultaneous merge arrival occurrence

The maximum advance interval is used to determine the number of fixed headway intervals or 'slots' occupied on the two approach links in computing the desired merge advance as shown in Figure 6-6. This desired maneuver reflects the number of occupancy positions necessary following the vehicle on the output link to accommodate the uncongested merging of other vehicles approaching the guideway merge. Once the desired maneuver has been computed, the actual maneuver which can be performed is chosen as the minimum of the desired maneuver and the number of unoccupied occupancy positions available at entry to the merge output link as shown in Figure 6-7. The effect of the actual advance maneuver on relative positioning of the vehicle is modeled as the vehicle begins traversal of the merge output link.

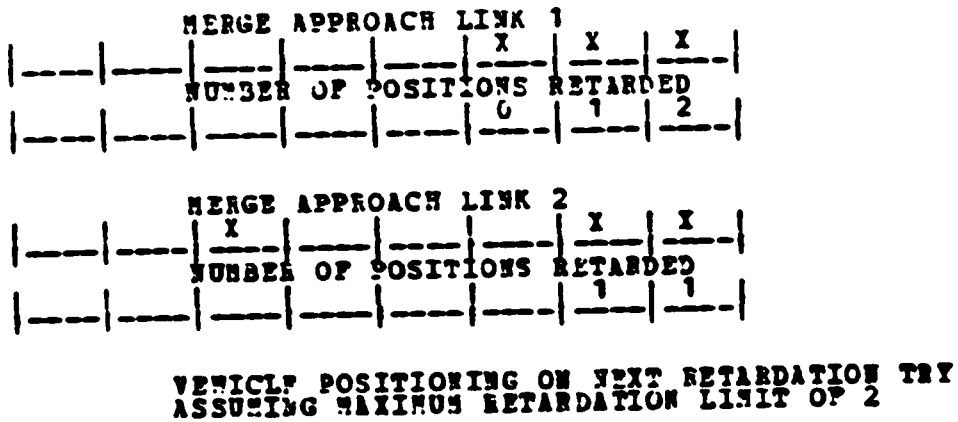
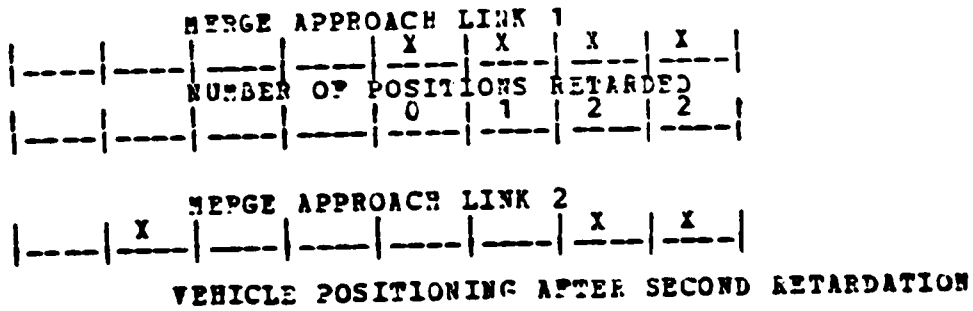
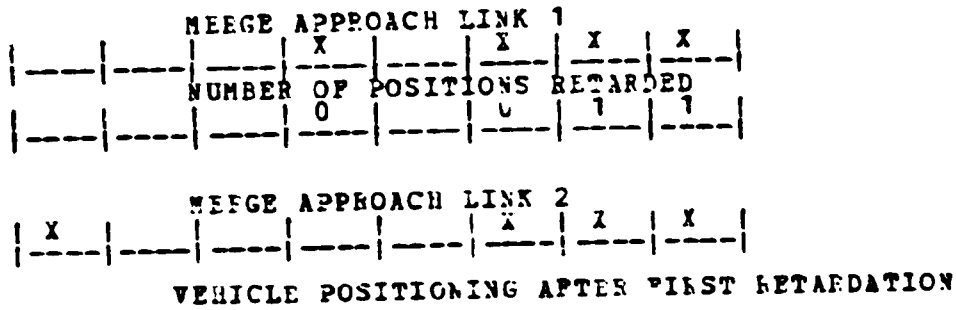
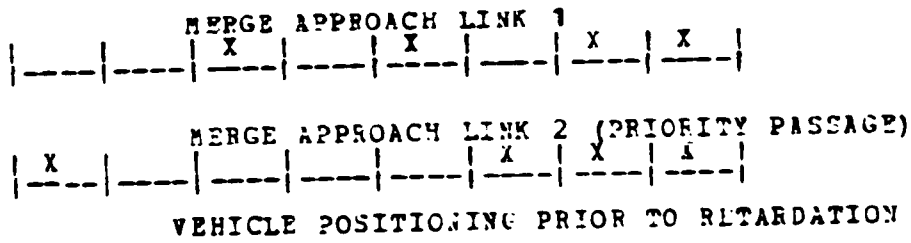


FIGURE 6-5. VEHICLE RETARDATION PROCESS

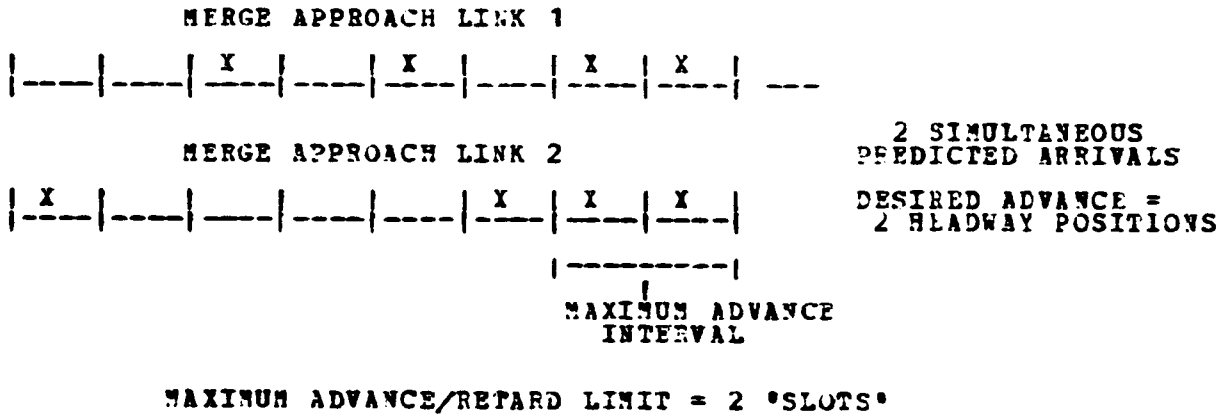


FIGURE 6-6. DESIRED ADVANCE COMPUTATION

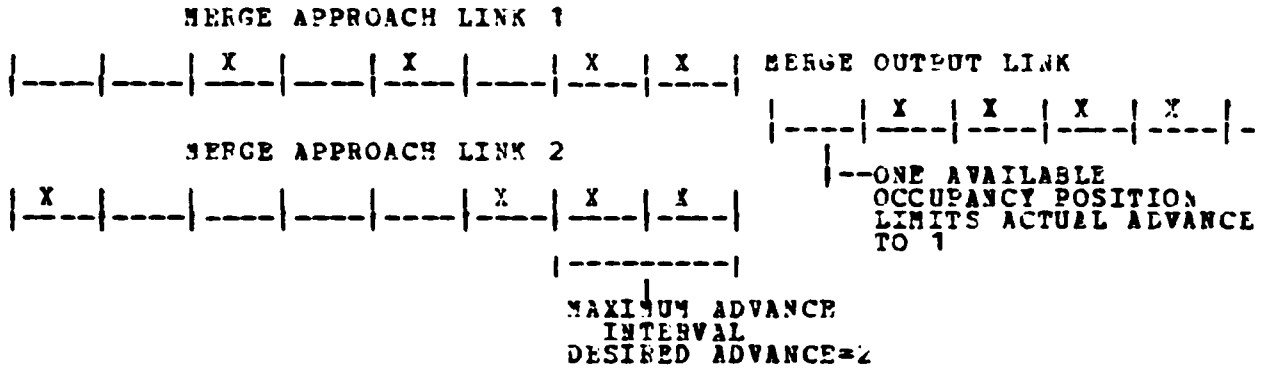


FIGURE 6-7. ACTUAL ADVANCE COMPUTATION

6.2.58.5 PDL

See Appendix A.

6.2.58.6 Decision Tables and Algorithms

None.

6.2.59 EGQNTR - Advance Positioning Processing

6.2.59.1 Identification

- o EGQNTR - Guideway Link Advance Positioning Processing
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.59.2 Argument Dictionary

None.

6.2.59.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
HTIME	-	I*4	entrained group headway travel time
LNK	-	I*2	current link ID
SYSID	-	I*4	system transaction ID used to schedule completion of the headway zone
TOCC	-	I*2	number of block positions to be occupied by vehicle(s) entering link
VEH	-	I*2	temporary variable for vehicle ID

6.2.59.4 Description

EGQNTR provides the link entry processing for a vehicle or train which is to assume an advance maneuver as it enters a merge output link. This instantaneous advance is applied to a vehicle(s) in an attempt to ensure merge availability for subsequently entering vehicles under quasi-synchronous control. Thus as a vehicle enters the link, it must assume a position or distance traveled on the link corresponding to the computed advance (supplied in EGQMRG), which is a multiple of a fixed block length.

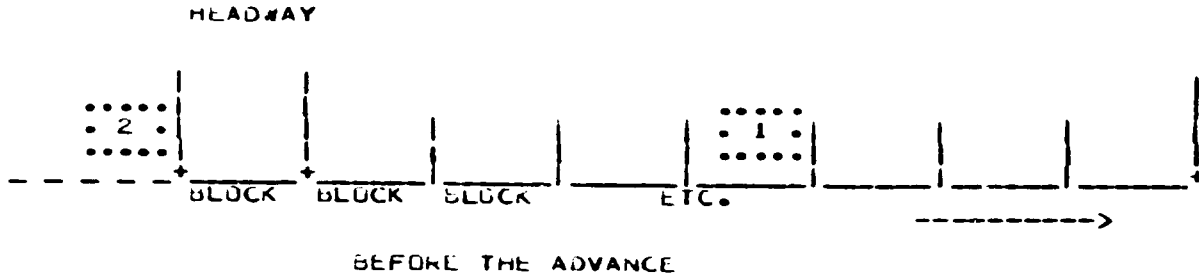
Since EGQNTR is invoked in lieu of the normal link entry routine for this particular condition, similar preliminary link status and vehicle data assignments are made prior to the "advance" processing in order to establish the existence of the vehicle on the link. The status information maintained for the link includes:

- o the link occupancy value which is incremented by the number of vehicle positions to be occupied by the vehicle(s) entering the link and,
- o the ID of the last vehicle to enter the link, which will be the identification of the single vehicle or that of the last vehicle of a train.

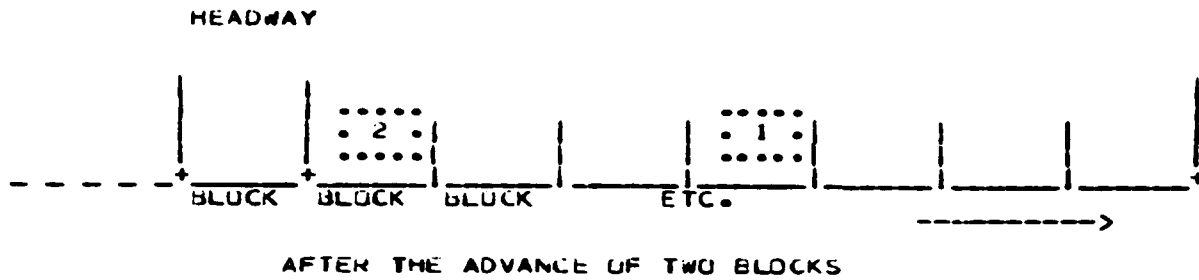
The vehicle data assignments made as part of link entry include:

- o new current link assignment (also assigned to each member of an entrained group),
- o link entry clock time (also adjusted and assigned to each member of an entrained group, Equation (1)) and,
- o "lead" and "follower" relationship pointers, if last vehicle to enter the link is still present.

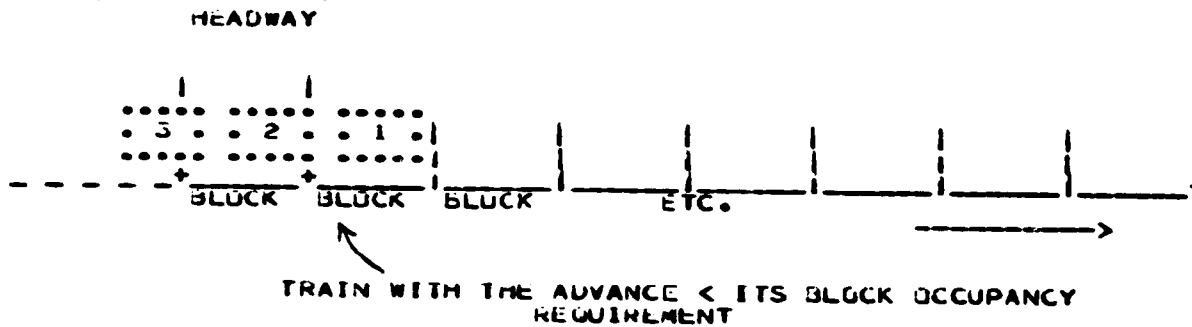
The advance of the vehicle(s) is applied by positioning the vehicle (by determining its new "assumed" distance traveled, Equation (2)) along the link at the calculated number of advance positions or fixed block slots. As a consequence, the time typically assigned as the time a vehicle will complete the headway zone and begin traversal of the travel segment must reflect the adjustment, thus reducing the separation time between the entering vehicle(s) and its predecessors (Equation (3) and (4)). The diagrams below illustrate the positional relationships of two vehicles on a link as a vehicle is about to enter or advance onto the link and after the instantaneous advance of two blocks has taken place.



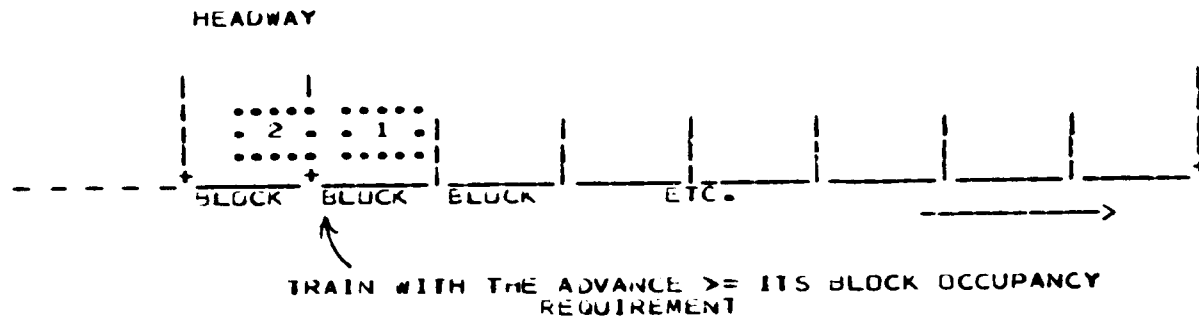
The distance traveled by vehicle 2 on the link at this point is '0' and its separation distance between vehicle 1 is equivalent to five block lengths. After the advance (which must be at least one block, which would place the vehicle through the headway zone), the vehicle distance "traveled" is two full block lengths and the separation distance between the vehicles is reduced to three block lengths.



If an advance is to be applied to a train which is not sufficient for movement of the entire train through the link headway zone (see example below), a system



service transaction to "free" the headway zone is scheduled, by invoking EAPFEL, for the time it will take to complete the link entry maneuver (Equation (5)). However, when the number of fixed block positions required for a train (see below) is less than or equal to the advance in blocks to be assumed (as for a single vehicle), then the headway zone for the link is noted as "free" and a prompt is scheduled to initiate the movement of any queued vehicles on the upstream link(s).



Once all of the link positioning is accomplished to accommodate the advance, the vehicle or train is scheduled to immediately re-enter the model for link traversal processing.

6.2.59.5 PDL

See Appendix A.

6.2.59.6 Decision Tables and Algorithms

The link entry time (unaffected by the advance) or the clock time at which the vehicle(s) would have normally started into the headway zone is given as (for each vehicle):

$$1. \quad V_E = C_T + ((V_N - 1) \times \frac{V_L}{L_V})$$

where:

C_T = current clock time

V_N = relative train position of the vehicle or vehicle number (=1 for single vehicle or lead vehicle of the train, =2 for second vehicle, etc.),

V_L = vehicle length (meters) and

L_V = link velocity (m/cu).

The distance along the link that the vehicle will assume as a result of the advance is given by:

$$2. \quad L_D = A_B \times B_L$$

where:

A_B = advance in number of blocks

B_L = block length (meters).

The adjusted separation time between travel segment entry or headway zone completion time of the vehicle and its predecessor, if still on the link, is given by:

$$3. \quad V_S = C_T + ((A_B - 1) \times L_H) - E_V$$

where:

L_H = normal link headway time or the time it takes to travel a block (cu),

E_V = clock time of last vehicle entry along travel segment (cu) and,

C_T and A_B as described above.

The time that the last vehicle (single vehicle or last vehicle of the train) should have or will begin link traversal or complete the headway zone, as adjusted by the advance, is given as:

$$4. \quad E_{VN} = C_T - ((A_B - T_0) \times L_H)$$

where:

T_0 = train occupancy, in # of blocks (=1 for single vehicle) and,

C_T , A_B and L_H described above.

The delta time required for headway zone completion when the applied advance is not sufficient for movement of the entire train through the link headway zone, is given by:

5. $H_{TR} = (T_0 - A_B) \times L_H$

Where:

T_0 , A_B and L_H are described above.

6.2.60 EGRESV - Guideway Link Empty and Reserved Vehicle Processing

6.2.60.1 Identification

- o EGRESV - Next Station Decision Control at Station Diverge Locations
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.60.2 Argument Dictionary

None.

6.2.60.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
FOUND	-	L*1	Flag indicating failed vehicle has passengers on board
NLINK	-	I*2	next entity of lead vehicle
STN	-	I*2	next station stop for lead vehicle
VEH	-	I*4	computational variable for vehicle ID
VEHDQ	-	I*4	vehicle ID to be removed from station arrival list
VEHX	-	I*4	temporary variable to store lead vehicle ID

6.2.60.4 Description

EGRESV controls the required processing for a vehicle(s) which encounters a network station (on-line or off-line) along its path in

demand responsive operation. In addition to its arrival list maintenance processing, EGRESV also determines if any unscheduled vehicle diverting into the station or rerouting around a scheduled station is required. If the vehicle to be processed is the lead vehicle of an entrained group and dynamic entrainment/detrainment is enabled, then the processing is repeated for each vehicle of the train or until a next entity incompatibility is encountered between a vehicle and the lead vehicle.

The first step performed in this module is the check to determine if the vehicle is at its reserving station which, if it exists, must also be its next station destination. If it is, then the ID of the vehicle is removed from the arrival list associated with the station. A failure check is then made on any vehicle not scheduled to enter the station at the end of the current link. If it is found that the vehicle is degraded and has passengers on board, then the vehicle is diverted into the station, by resetting its next entity and destination equal to the station ID, and by removing the vehicle ID from any downstream reserving station arrival list to which it may be identified. The diversion for failure is also accomplished for any failed circuitous empty vehicle, regardless of its next station destination, this station or any other. Whenever a diversion is made, the former station assignment is maintained in case it is later determined that entry into the immediate station is not possible.

For those unfailed vehicles, not destined for the current station or for any unfailed circuitous empty, it is then determined if the vehicle must be diverted into the station to satisfy unserved trip requests. (This checking is not accomplished for any member of an entrained group when dynamic entrainment/detrainment is not enabled. The only unscheduled diversion for vehicles of static trains is for degraded vehicle removal. The entire train must enter the station.) If the vehicle is empty, EGEMTY is invoked to process any further diversion checks (and to possibly reroute a circuitous empty which is not required to enter the station). If, however, the vehicle has passengers, the policy enabling the diversion of vehicles is active and the service policy in effect is demand responsive multi-party, then EGTCTL is invoked to determine whether the vehicle should be diverted to enable any additional unreserved trips to board the vehicle.

The final determination of the next entity for a vehicle is complete when all of the above possibilities have been queried. If the destination has been altered for a member of a dynamically entrained group, it is not necessary that each succeeding vehicle be diverted/rerouted as well. However, after the leading vehicle of such a train is processed, its next station destination is used to compare against that determined for each succeeding vehicle. If, as each of the intermediate vehicles are processed and compared it is found that one vehicle (the lead vehicle or the current one) is to enter the immediate station and the other is not, then EGDTRN is invoked to detrain the vehicles at the point of

incompatibility. Notice that vehicles may have different destinations but if none are to enter the current station, then detrainment will not be required at this diverge point.

In summary, a vehicle(s) is required to enter a station if:

- a. The vehicle is already scheduled to enter the station due to the destination of its passengers or reservations made for the vehicle at the station,
- b. The vehicle is degraded and has passengers on board. Only the lead vehicle of a train is considered to be degraded if the train is traveling with a penalty factor. If the degraded vehicle is a member of a train and dynamic detrainment is enabled, only the lead vehicle needs to enter the station. If, however, dynamic entrainment is not enabled, then the entire static train must enter.
- c. The vehicle is empty, it has no reservations at any downstream station, unreserved trips exist at this station and the diverting of vehicles to service such service requests is enabled or,
- d. The vehicle has passengers, demand responsive multi-party service is enabled and the remainder of (c) above is true.

The detailed checking required to determine if any of the above conditions exist, is found in EGEMTY and EGTCTL, as mentioned above, as well as EGRESV.

6.2.60.5 PDL

See Appendix A.

6.2.60.6 Decision Tables and Algorithms

None.

6.2.61 EGSCHED - Schedule Vehicle Follower

6.2.61.1 Identification

- o EGSCHED - Schedule Vehicle Follower
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.1.61.2 Argument Dictionary

None.

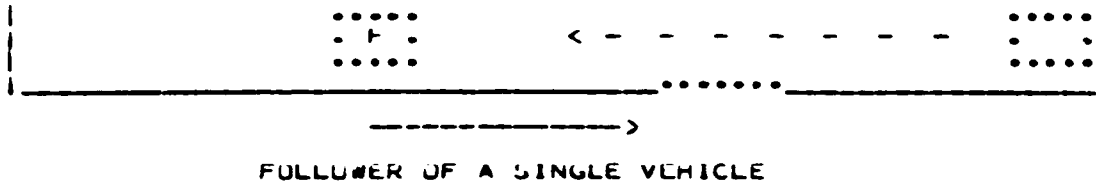
6.2.61.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
FOUND	-	L*1	Flag indicating FAIL card match found
LNK	-	I*2	current link
OCCLEN	-	I*4	occupied blocks or positions at end of link
RSCDTM	-	R*4	delta travel time (real format)
SCDTME	-	I*4	delta travel time (integer format)
VEHF	-	I*4	ID of follower vehicle

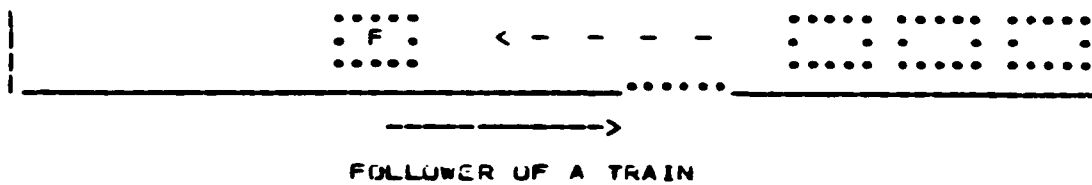
6.2.61.4 Description

EGSCHED determines if independent travel must be scheduled for a following vehicle, once link travel is completed for the current vehicle or train. If a non-traveling vehicle assuming a vehicle follower position exists, it is scheduled to traverse the link and move to an adjacent position to the current vehicle(s).

The details of the processing begins by determining the identity of the follower. The follower is the succeeding vehicle identified to the single vehicle (as shown below), or is the vehicle identified to the



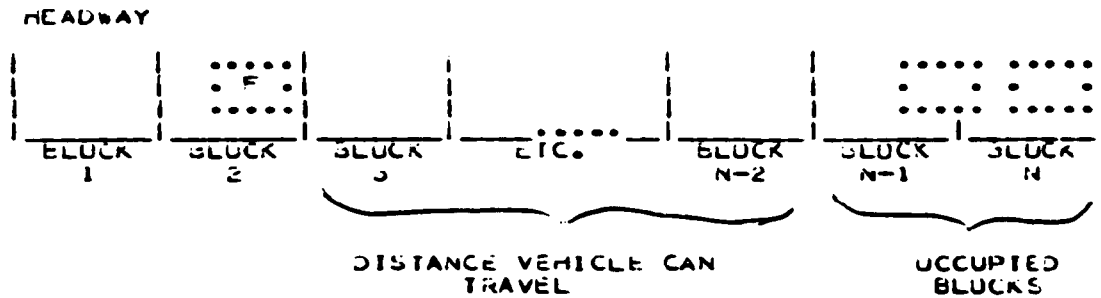
last vehicle of the entrained group or train (see below).



If it is found that a follower doesn't exit on the link or if the identified follower currently has a scheduled event, then the remaining procedures within the routine are ignored. A follower vehicle would only have an event scheduled if the current vehicle was recently detrained from a train which had completed the link and thus had scheduled the same follower to travel the separation distance. Since a vehicle can have only one scheduled event at a time, no further processing for the follower is accomplished.

A follower with no scheduled event must travel to a position behind the single vehicle or train which it follows. The time and distance to travel is based on the position regulation scheme in effect (see Equations (1)-(4) below).

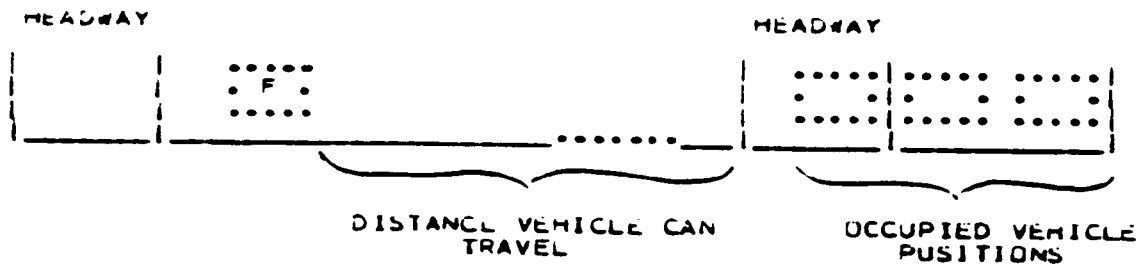
For fixed block movement the vehicle is scheduled to travel to the adjacent block position to that on the link occupied by the preceding vehicle (as shown in the example below). The number of occupied block



positions at the end of the link, used to determine the location or block position to where the follower can move, is:

- o a single block, if preceding vehicle is not queued and is unentrained,
- o the number of fixed block positions required for the train if the preceding vehicle is not queued but is a member of an entrained group or,
- o the number of block positions currently occupied by the link exit queue, if the preceding vehicle is queued.

For variable headway movement, the follower vehicle is scheduled to travel to within a headway or safe stopping distance of the preceding vehicle on the link (as shown in the example below). The number of occupied



vehicle positions at the end of the link, used to determine the location to where the follower can move, is:

- o A single vehicle position, if the preceding vehicle is not queued and is not entrained,
- o The number of vehicles associated with the train if the preceding vehicle is a member of an entrained group or
- o The number of vehicles currently in the link exit queue, if the preceding vehicle is queued.

(Provision is made in the routine for recognition of a user-defined headway model for accomplishing travel along the link.)

The computed delta time for which the vehicle is scheduled to travel (by invoking EAPFEL) may vary from the separation time or "gap" established between the vehicles as the follower began traversal of the travel segment. This may be caused by differences in velocity or link travel time resulting from vehicle degradation, retardation or velocity variance, assumed by the consecutively traveling vehicles. For the computed travel time result to be less than zero, the follower must be within a headway or block and must have been traveling at a faster rate than the preceding vehicle. Since the follower may not overtake the prior vehicle, its projected link exit time is incremented by the delta time difference (its absolute value). The follower is then scheduled to re-enter the model at 'zero' delta time for the last model event or link traversal completion.

If the lead vehicle has entered a queue and is directly behind a train led by a degraded vehicle, it may be necessary to form a push coupling train. If the failure has been detected and the recovery method is push coupling, then the coupling is effected by invoking routine EAFTRN, the n.r.w minimum path table is activated, and the restart is scheduled for the user input delta time.

6.2.61.5 PDL

See Appendix A

6.2.61.6 Decision Tables and Algorithms

For fixed block movement, the vehicle (follower) is scheduled to travel to the adjacent block position on the link to that occupied by the preceding vehicle with the time for travel, T, and the position or distance, D, on the link given by:

$$1. \quad T = T_X - C_T - \frac{B_0 \times B_L}{L_V}$$

where:

T_X = projected link exit time of follower (cu),

C_T = current clock time (cu),

B_0 = occupied blocks in front, = T_0 or Q_0 and

T_0 = train block occupancy requirement (=1 for single vehicle), if preceding vehicle is not queued,

Q_0 = queue occupancy in # of blocks, if preceding vehicle is queued,

B_L = block length (meters) and,

L_V = link velocity (m/cu).

$$2. \quad D = L_L - (B_0 \times B_L)$$

where:

L_L = length of link (meters) and

B_0 and B_L as described above.

For variable headway movement, the follower is scheduled to travel to within a headway or safe stopping distance of the preceding vehicle in time 'T' and to distance 'D' on the link as given by:

$$3. \quad T = T_X - C_T - \frac{(B_0 - 1) \times V_L}{L_V} - L_H$$

where:

B_0 = occupied vehicle positions in front, = T_0 or Q_0 and

T_0 = train length in # of vehicles (=1 for single vehicle), if the preceding vehicle is not queued and,

Q_0 = number of vehicles in the link exit queue, if preceding vehicle is queued,

V_L = vehicle length (meters),

L_H = nominal link headway travel time (cu) and,

T_X , C_T and L_V as described above.

4.
$$D = L_L - ((B_0 - 1) \times V_L) - (L_H - L_V)$$

where:

L_L , B_0 , V_L , L_H and L_V as described above.

6.2.62 EGTCHK - Trip Compatibility Check

6.2.62.1 Identification

- o EGTCHK - Multi-Party Trip Compatibility Check
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.62.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TRPID	-	I*4	station trip being processed for compatibility check

6.2.62.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
STNX	-	I*2	temporary variable used for bypassed station ID
VTRPID	-	I*4	vehicle trip being compared with station trip

6.2.62.4 Description

EGTCHK checks the basic criteria for adding a new station trip 'TRPID' to a partially loaded vehicle which is currently on the guideway at a station diverge location. The criteria is available space and destination compatibility. Thus if the station trip can fit,

THE # OF PASSENGERS ASSOCIATED WITH THE PARTICULAR STATION TRIP + THE # OF PASSENGERS CURRENTLY ON THE VEHICLE <= THE VEHICLE CAPACITY

then the destination of the trip is further compared to the destination/route of each current trip on the vehicle until a compatibility, if any is found. The station trip will be approved or found compatible if any one of the following conditions is met:

- a. the station trip destination matches that of one of the trips already on the vehicle,
- b. the station trip destination is a bypassed station along the route implied by any one of the current trips on the vehicle or,
- c. the most remote destination among the trips already on the vehicle is a bypassed station along the route that would be used for the new station trip.

To trace the paths implied in conditions (2) and (3), the Successor Station Table is used. (A description of this table and of how it is used is found in the section below.)

6.2.62.5 PDL

See Appendix A.

6.2.62.6 Decision Tables and Algorithms

The Successor Station Table used in the processing of this routine is a matrix (as shown below) of station ID's that identifies the path (of stations only) from any station in the network to any other station.

Therefore,

STATION DESTINATION
(TO)

J

	I	J	
			⋮

STATION LOCATION
(FROM) I

element (I,J) is the next station on the minimum path from I to J.

6.2.63 EGTCTL - Trip Compatibility Check Control

6.2.63.1 Identification

- o EGTCTL - Multi-Party Trip Compatibility Check Control
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.63.2 Argument Dictionary

None.

6.2.63.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
LNKSTN	-	I*2	station ID at end of current link
TRPID	-	I*4	station trip being processed for compatibility checking
VEHDQ	-	I*4	vehicle ID to be removed from station arrival list

6.2.63.4 Description

EGTCTL provides the control for determining if an unreserved loaded vehicle circulating on the guideway and currently at a station diverge point must make an unscheduled stop at the station in order to board an additional "compatible" trip(s), in demand responsive multiparty service operation. If unserved (unreserved) trips exist in the boarding queue and space exists on the vehicle, then each trip in the station trip queue is queried until a trip is found (if any) that can be added to the vehicle. As each station trip is processed it is determined first if the trip already has a reservation. If not, EGTCHK is invoked to see if vehicle space is available for the trip and if the trip destination is compatible with the existing vehicle trips.

If a station trip is found that can be added to the vehicle, preparation is made to divert the vehicle into the station to pick up the trip(s). The next entity and destination of the vehicle is set to the ID of the station. Its former station assignment is maintained in case it

is later determined that station entry is not possible. In addition, as part of this diversion process, the number of unserved station trips is decremented, to assure that a following vehicle is not diverted to service the same trip. If the vehicle has a reserving station assignment and it is something other than the station being diverted to, the vehicle ID is then removed from the arrival list of the former reserving station.

6.63.5 PDL

See Appendix A.

6.2.63.6 Decision Tables and Algorithms

None.

6.2.64 EGTEST - Guideway Link Entry Testing

6.2.64.1 Identification

- o EGTEST - Guideway Link Entry Testing
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.64.2 Argument Dictionary

None.

6.2.64.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
FOUND	-	L*1	Flag indicating FAIL card match found
LNKOCC	-	I*2	Computational variable for link occupancy
TIME	-	R*4	Current simulation time clock in seconds
TOCC	-	I*2	Computational variable for number of block positions to be occupied by vehicle entering link

6.2.64.4 Description

The purpose of EGTEST is to determine if a vehicle(s) can exit its current entity and enter the next assigned guideway link. Link entry testing therefore begins by verifying that the vehicle(s) may leave its current link by checking for the existence of a failure at the link exit. If the exit is failed, a negative entry acknowledgement is indicated. Otherwise, conditons on the next link are checked and entry is also precluded, with indication made, if any of the following is found to exist:

- o If the link has a competing link at a merge which is open but has a non-empty queue (in this case, this link also queues until a prompt is issued to dequeue one of the competing up-stream links).
- o the entry of the link is blocked by a failure occurrence,
- o the headway zone of the link is currently occupied by another vehicle or
- o capacity constraints on the link would be violated by entry of the vehicle or entrained group.

The temporary occupancy value used in determining if capacity exists on the next link depends upon whether the vehicle is entrained (since entry testing is only provided at the time the lead vehicle is entering the link) and the position regulation scheme in effect for the simulation experiment. If the vehicle is unentrained, the increment in occupancy required on the next link is "one". The occupancy requirement for an entrained group is given by the length of the train for variable headway modeling, and as the number of block intervals required to accommodate the train as computed at the time of train formation, for fixed headway control.

If the link entrance is failed or marked to degrade the next vehicle, EGNEXT searches the active FAIL cards to find data for this failure. The captured vehicle is linked to the failure conditions, the degradation factor is applied to the vehicle failure case, and the failure detection event is scheduled.

Provision is made in the entry testing process for recognition of a user-defined procedure for computing train occupancy requirements to satisfy a user-defined position regulation scheme.

6.2.64.5 PDL

See Appendix A.

6.2.64.6 Decision Tables and Algorithms

None.

6.2.65 EGTRNC - Train Compatibility Check at Network Diverges

6.2.65.1 Identification

- o EGTRNC - Train Compatibility Check at Network Diverges
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.65.2 Argument Dictionary

None.

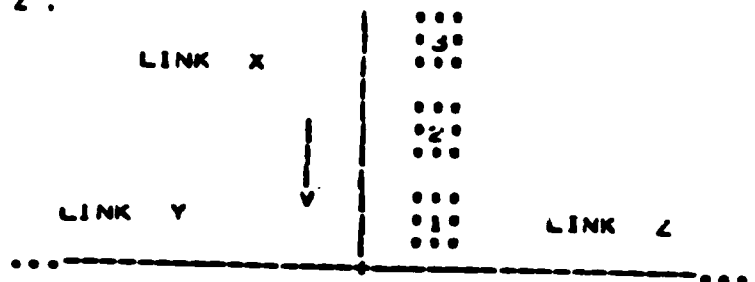
6.2.65.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
VEH	-	I*4	computational variable for vehicle ID
VEHX	-	I*4	temporary variable to store lead vehicle ID

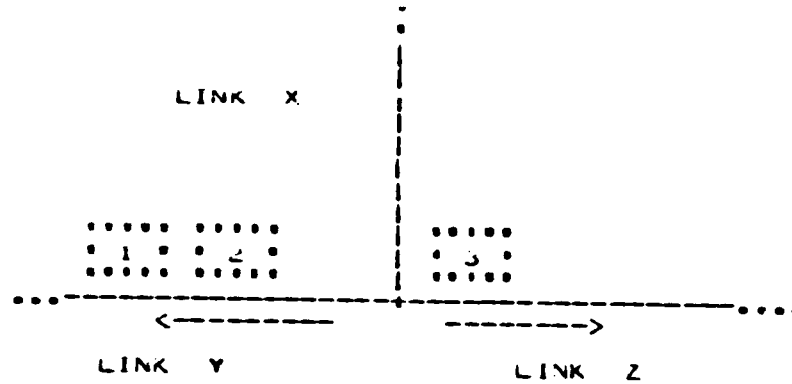
6.2.65.4 Description

EGTRNC is used in nonsynchronous demand responsive operation to check each vehicle of an entrained group at a network diverge in order to determine if linespeed detrainment is required. At the time EGTRNC is invoked, the next entity to be traversed by the lead vehicle has already been established. This next entity value therefore becomes the point of reference or the basis for comparing the next entity, determined by invoking EGGNXT for each succeeding vehicle in the train, until each vehicle is processed or until a difference is found between the next entity of the lead vehicle and the next entity of an intermediate vehicle. If such a difference is found then EGDTRN is invoked in order to detrain the vehicles at the point in the train of incompatibility.

For example, assume as shown in the diagram below, that it is found that vehicle 1 and vehicle 2 of the train of three vehicles are to travel link "y" enroute to their respective destinations, but vehicle 3 is to travel link "z".



EGTRNC would therefore invoke EGTRN with the information that vehicle 3 is to be detained from the other vehicles. The scene would thus soon look like this.



One exception to the above processing exists. If vehicles have the same next entity but are assigned to alternate paths which are different, then detrainment is also initiated. However, if one vehicle is assigned an alternate path and the other vehicle being compared is not, then it is assumed that the vehicles may continue traveling together, at least until another network diverge is encountered.

6.2.65.5 PDL

See Appendix A.

6.2.65.6 Decision Tables and Algorithms

None.

6.2.66 EGUNTR - User Headway Model, Link Travel Segment Entry

6.2.66.1 Identification

- o EGUNTR - Interface Routine for User - Specified Guideway Headway Model Travel Segment Entry Processing
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.66.2 Argument Dictionary

None.

6.2.66.3 Local Variable Dictionary

None.

6.2.66.4 Description

EGUNTR is a module provided to facilitate the recognition of a user-defined headway model for accomplishing link travel segment entry. The routine currently consists only of an error message indicating the non-existence of such a model. When the model is defined, the message processing may be replaced by the user-defined requirements for a vehicle, upon its completion of headway zone travel, as it is to begin traversal of the link travel segment.

6.2.66.5 PDL

See Appendix A.

6.2.66.6 Decision Tables and Algorithms

None.

6.2.67 EGVTVL - User Headway Model Link Travel Segment Traversal

6.2.67.1 Identification

- o EGVTVL - Interface Routine for User - Specified Guideway Headway Model Travel Segment Traversal Processing
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.67.2 Argument Dictionary

None.

6.2.67.3 Local Variable Dictionary

None.

6.2.67.4 Description

EGVTVL is a module provided to facilitate the recognition of a user-defined headway model for accomplishing link travel segment traversal. The routine currently consists only of an error message indicating the non-existence of such a model. When the model is defined, the message processing may be replaced by the user-defined requirements for travel segment completion, in response to completion of a link travel event by a vehicle.

6.2.67.5 PDL

See Appendix A.

6.2.67.6 Decision Tables and Algorithms

None.

6.2.68 EGVALS - Vehicle ETA Calculation and Station Arrival List Recording

6.2.68.1 Identification

- o EGVALS - Vehicle ETA Calculation and Station Arrival List Recording
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.68.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
VEH	-	I*4	Vehicle to be processed

6.2.68.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
FIRST	-	I*2	temporary vehicle ID used for comparison
NEXT	-	I*2	temporary vehicle ID used for comparison - ID following 'FIRST' in arrival list
TIME	-	I*4	ETA time of vehicle being added to arrival list

6.2.68.4 Description

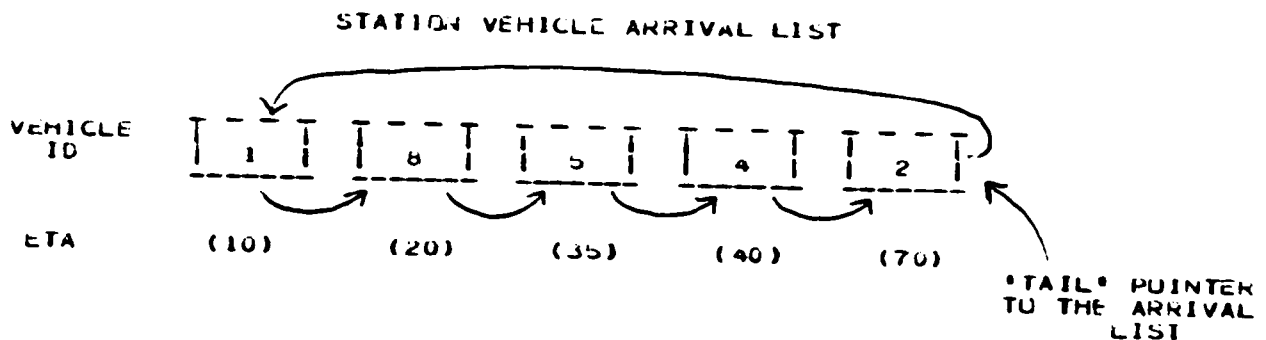
EGVALS uses the estimated time of arrival (ETA) table to project the time at which a vehicle will arrive at its new destination assignments (reserving station). Since this routine is invoked at the time a vehicle(s) is being rerouted around a scheduled station stop and since the table times are from station exit to station entrance points, the time requirement to travel the station by-pass link must be added to the new ETA. Once the time is determined the vehicle ID is placed on the downstream station arrival list in ETA time order. In addition, since the vehicle is not going to stop at the present station, the current minimum path table pointer must be assigned to the vehicle, if apriori routing is enabled. This assignment is usually done at the time a vehicle is launched from the station.

6.2.68.5 PDL

See Appendix A.

6.2.68.6 Decision Tables and Algorithms

The vehicle arrival list, if any, associated with each station is a chain of vehicle IDs (as shown in the example below). A pointer exists to the last, or 'tail', vehicle in the chain with the latest ETA and this vehicle has a pointer to the first vehicle or the vehicle with the earliest ETA. The first vehicle then points to the second, the second to the third etc.



When inserting a vehicle into the arrival list chain, the following checks are made until the arrival list position for the vehicle is determined:

- a. if the 'tail' is zero, then no list as yet exists for the station. Therefore the vehicle ID becomes the 'tail' and the vehicle in turn must point to itself.
- b. if the ETA of the vehicle is larger than the ETA of the 'tail', the vehicle becomes the new 'tail' vehicle, now to be pointed to by vehicle 2, in the example. The new 'tail', must similarly point to vehicle 1.
- c. if the ETA of the vehicle is less than the ETA of the 'tail' then its identification must be inserted at the proper time-ordered position within the list. This is done by adjusting the preceding vehicle pointer to point to the new insertion and assigning that vehicle's former pointer to the current vehicle.

Note: Station arrival lists are maintained in demand responsive operation whenever the arrival list option is specified or the reservations policy is in effect.

6.2.69 EGVLOG - Vehicle Logging

6.2.69.1 Identification

- o EGVLOG - Log Vehicle Arrival at Station Entry
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.69.2 Argument Dictionary

None.

6.2.69.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ARRT	-	R*4	Time in seconds
RTE	-	I*4	Assigned Scheduled Service route ID
SINK	-	I*4	Vehicle destination - 1-Guideway
TRN	-	I*4	Train Length

6.2.69.4 Description

EGVLOG is invoked by EGLEAV to record the arrival of a vehicle at a designated station entry in the Vehicle Log file. The recording process involves the writing of a vehicle summary record followed by a series of trip follower records representing each trip on board the vehicle and any entrained followers. The format of these records is compatible with the vehicle demand input accepted by the Detailed Station Model (DSM). The file number can be used as direct input to the DSM for modeling individual station operation and performance.

6.2.69.5 PDL

See Appendix A.

6.2.69.6 Decision Tables and Algorithms

None.

6.2.70 EGVNTR - Variable Headway Travel Segment Entry

6.2.70.1 Identification

- o EGVNTR - Variable Headway Travel Segment Entry
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.70.2 Argument Dictionary

None.

6.2.70.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
LNK	-	I*2	current link
MEANV	-	R*4	distribution mean value
OCCLEN	-	I*4	occupancy length of queue at end of link
PENFAC	-	R*4	penalty factor
SCDTME	-	I*4	computed travel delta time
VVAR	-	R*4	velocity variance factor

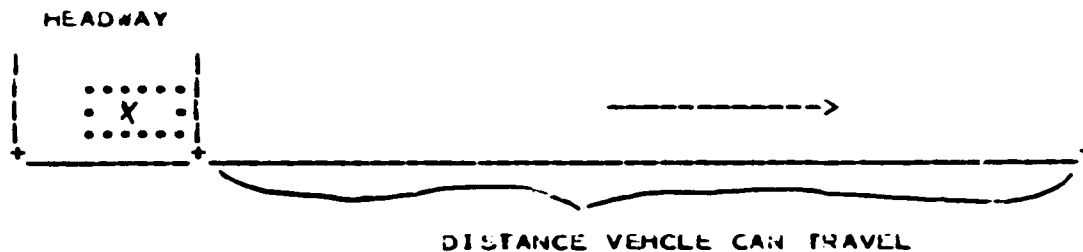
6.2.70.4 Description

EGVNTR processes a vehicle within the Guideway Link Model, upon its completion of headway zone travel, as it is to begin the traversal of the link travel segment. Travel time and the distance to where a vehicle is to move along the link is determined, followed by the scheduling of the vehicle for this computed travel time and the next link event. Traversal of the link is accomplished under control of a variable headway positioning scheme. Thus a vehicle, if scheduled to travel, moves to within a headway of the preceding vehicle, if any, on the link.

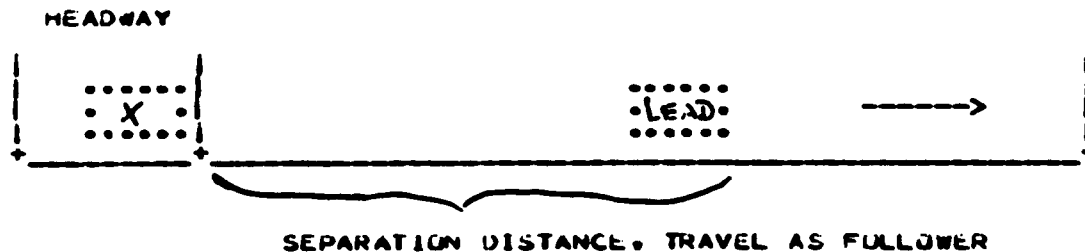
The travel segment entry process begins by determining the velocity variation, if any, that is to be assigned to the vehicle (valid only in asynchronous mode). This variation factor, which is a normally

distributed random number based on a user-supplied standard deviation, along with any existing degradation or penalty factor for the vehicle and merge delay factor for the link is applied to the computation that determines the time at which the vehicle should complete link traversal, based on its current location and the link velocity. The penalty factor adjustment to the time, if any, is performed to reflect degraded vehicle operational status. As long as a vehicle is operating on the guideway in degraded mode or is a member of a degraded entrained group, travel time adjustment will be made. The merge delay factor, used to model increases or decreases in vehicle velocity, is applied to the computed travel time in order to adjust vehicle flow into a merge for asynchronous mode only). This factor is computed as a result of applying a heuristic merge algorithm on a periodic basis (see description of EAPCMP).

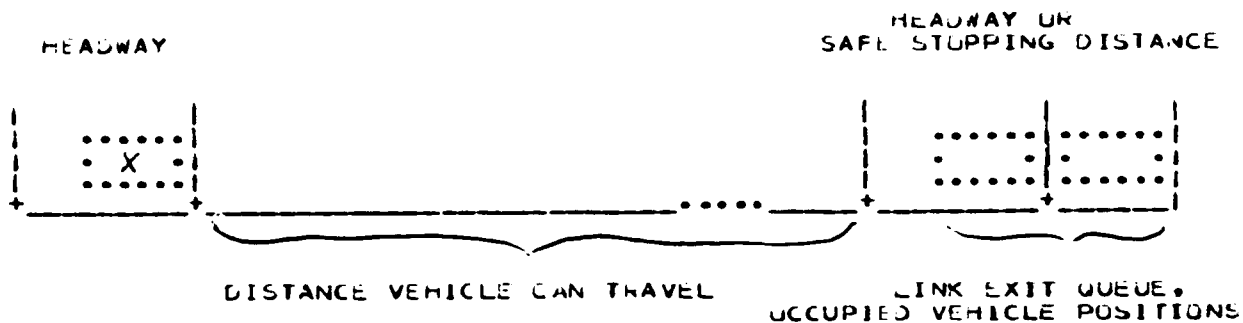
Following the computation of the vehicle's projected exit time from the link (see Equation (1) below), it is determined whether scheduling of the vehicle for independent rather than vehicle follower movement is required. If no prior vehicle exists on the link, the vehicle, identified as the next vehicle to exit the link, is scheduled to travel the distance of the entire travel segment (see diagram below). The travel time



required for this event is the difference between the computed projected link exit time for the vehicle and the current clock time (see Equation (2) below). If a prior vehicle exists on the link and is in the process of traveling the link, the current vehicle is not scheduled for independent travel but instead travels as a follower (see diagram below) until the leading



vehicle completes travel on the link. If, however, a prior vehicle exists on the link, but it is queued due to congestion, independent vehicle travel is scheduled. The current entering vehicle in this case moves to a position on the link which is within a headway of the link exit queue (as shown in the example below). The time required to travel



to the queue is the difference between the computed projected link exit time and the sum of the current clock time and the time required to travel the vehicle positions (including a headway separation) occupied by the queue (see Equations (3) and (4) below)

Anytime a vehicle is scheduled for link travel during link travel segment entry, the next modeling event indicator is set to indicate that final link travel processing is required at completion of the entry process time.

6.2.70.5 PDL

See Appendix A.

6.2.70.6 Decision Tables and Algorithms

The projected link exit time of the vehicle is given by:

$$1. \quad T_X = C_T + \frac{(L_L - L_D)}{(L_V + V_V)} / P + M$$

where:

C_T = current clock time (cu),

L_L = length of link (meters),

L_D = distance traveled by vehicle on link (meters),

L_V = link velocity (m/cu),

V_V = velocity variation (m/cu), (default = 0),

P = penalty factor assigned to the vehicle, (default = 1)
and

M = merge delay factor (cu), (default = 0).

If no prior vehicle exists on the link, the vehicle is scheduled to travel the distance of the entire travel segment, or the remaining link length, in time T given by:

$$2. \quad T = T_X - C_T$$

Where:

T_X and C_T as described above.

Any adjustment to the travel time for application of a merge delay, penalty or velocity variation factor is assumed to be imbedded in the projected link exit time.

If a prior vehicle exists on the link, but is queued, the time, T , scheduled for vehicle travel and the distance, D , along the link to where the vehicle can move is given by:

$$3. \quad T = T_X - C_T - \frac{((Q_0 - 1) \times V_L)}{L_V} - L_H$$

Where:

Q_0 = number of vehicles queued,

V_L = vehicle length (meters),

L_H = link nominal headway travel time (cu)
and

T_X , C_T and L_V as described above.

$$4. \quad D = L_L - ((Q_0 - 1) \times V_L) - (L_H \times L_V)$$

Where:

L_L , Q_0 , V_L , L_H and L_V as described above.

6.2.71 EGVTVL - Variable Headway Travel Segment Traversal

6.2.71.1 Identification

- o EGVTVL - Variable Headway Travel Segment Traversal
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.71.2 Argument Dictionary

None.

6.2.71.3 Local Variable Dictionary

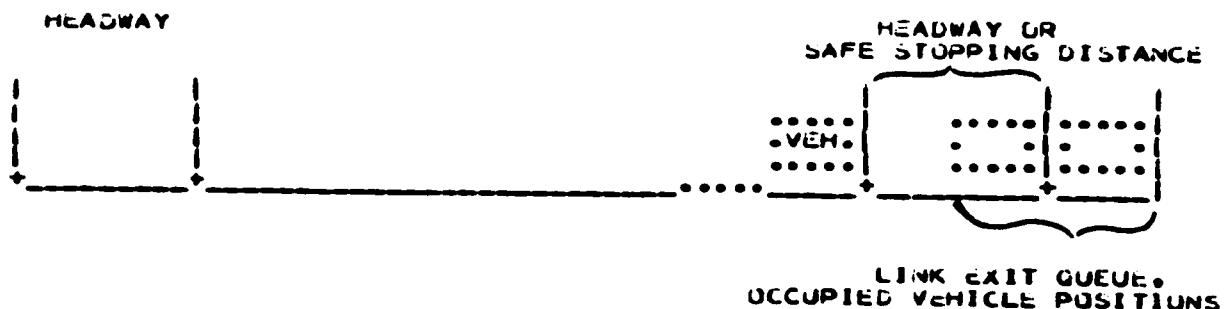
VARIABLE	DIM	TYPE	DESCRIPTION
ENDLNK	-	L*1	indication that vehicle has reached end of link
GOFIRST	-	L*1	indication if current vehicle to go first before competing vehicle
LNK	-	I*2	current link
LNKX	-	I*2	competing link
OCCLEN	-	I*4	occupancy length of queue at end of link
SCDNXT	-	L*1	indication that follower vehicle may be scheduled for travel
SCDTME	-	I*4	computed travel delta time
STATUS	2	I*2	passenger status of competing vehicles
STN	-	I*2	station ID at merge
VEH	-	I*2	competing vehicle at merge

6.2.71.4 Description

EGVTVL provides travel segment completion processing, under variable headway control, in response to completion of a link travel event by a vehicle. Depending on the conditions that exist, the vehicle is either again made to be a non-scheduled follower, scheduled for a subsequent travel event on the link, queued due to link congestion or noted that travel to the end of the link has been completed.

All vehicles completing a link travel event initially require processing to determine whether link traversal is complete or the scheduling of another travel event is required. This determination is based on the status of the preceding vehicle, if any, to travel the link with the four possible conditions and appropriate processing summarized as follows:

- a. No lead vehicle was identified to the current vehicle as it traveled the link. It never traveled as a follower and was thus scheduled to traverse the entire link. Link travel completion for the vehicle has therefore occurred.
- b. A lead vehicle was originally identified to the current vehicle but is no longer on the same link, which implies that the current vehicle has already been scheduled to travel the separation distance between the vehicles. Therefore link completion processing is based on whether or not the current vehicle has traveled the entire length of the link. This is determined by comparing the projected vehicle link exit time to the current clock time. If the difference is greater than zero the vehicle is scheduled to travel for the **remaining delta time** (see Equation 1). Otherwise link travel completion has occurred.
- c. A leading vehicle exists on the same link as the current vehicle and is queued. Thus, it must be determined if the vehicle has also encountered the end of the existing queue of stopped vehicles or if the vehicle may continue to travel on the link towards the queue. The vehicle has reached the queue if it is within a headway or safe stopping distance of the preceding queued vehicle (as shown below). This determination is



made by comparing the projected vehicle link exit time to the sum of the current time and the time required to travel the vehicle positions (including a headway separation) occupied by the queue. If the computed difference is zero, then the vehicle(s) has reached the queue and must join it (via module EGLWTQ). Otherwise, the vehicle is scheduled to travel for the delta time difference. Its position or distance traveled on the link at the end of that time would be a headway distance behind the front end of the last queued vehicle, as pictured above (see Equations (2) and (3)).

- d. A leading vehicle which is not queued, still exists on the same link as the current vehicle. For this condition to occur the preceding vehicle had to have been recently detrained from a train which had completed the link and thus had scheduled any follower (which was the current vehicle) to travel the separation distance. The preceding vehicle, at detrainment, was also scheduled to travel and must currently be in the process of completing travel to the end of the link. When this vehicle reaches the end of the link it will schedule the current vehicle to once again catch up or travel the separation distance. Therefore the current vehicle need not be scheduled for travel at this time and can again assume a follower relationship.

For the cases above in which it is found that the vehicle has completed travel on the link, one more processing step is required at specific locations and under certain policy constraints in order to signify the full completion of link traversal. At network merges, it must further be determined if merge planning (in the case of asynchronous control and priority merging) must be performed as the vehicle plans to leave the link and merge onto the next. Merge planning is required under these conditions if two unqueued vehicles are to arrive at a merge simultaneously. If the opposing vehicle is completing travel on another guideway link which does not have priority then the current vehicle may plan to continue to move through the merge. Similarly, if the competing vehicle is completing its last event in a station, it must also be determined which vehicle has priority. This is done by referencing the user-supplied local merge priority table, once the "empty" or "occupied" status is known for each vehicle (see the Decision Table discussion below). If, however, it is determined that for either case the current vehicle does not have the necessary priority to continue movement along its path, it is scheduled to reenter the model at a zero delta time, allowing the priority vehicle to enter the headway of the merge output link.

If, at the conclusion of all of the preceding processing, the vehicle has not been queued or scheduled for any further event on the

link (including an event scheduled for delta time zero), it is assumed that final link travel processing for the vehicle is complete. This results in the signaling of link model completion to the architecture in order that processing for moving the vehicle to its next required entity may be invoked. Additionally, if the vehicle has completed travel or has joined the link exit queue, EGSCHD is invoked in order that any non-traveling vehicle assuming a vehicle follower position behind the current vehicle or train may be scheduled to independently traverse the link.

6.2.71.5 PDL

See Appendix A.

6.2.71.6 Decision Tables and Algorithms

Travel Times and Distances

If no prior vehicle exists on the link, the vehicle is scheduled to travel the distance to the end of the link in time, T , given by:

$$1. \quad T = T_X - C_T$$

where:

T_X = projected link exit time of the vehicle (cu),

C_T = current clock time (cu) and

$$T_X > C_T.$$

If the computed value of T is less than or equal to zero, then the vehicle is assumed to have completed travel of the link. A negative value for T implies that the vehicle took longer to travel the link than was expected. This could happen if it was following a vehicle that was degraded or traveling at a decreased link velocity.

If a prior vehicle exists on the link, but is queued, the vehicle is scheduled to travel for time, T , and will thus travel to a distance, D , along the link as given by:

$$2. \quad T = T_x - C_T - \frac{((Q_0 - 1) \times V_L)}{L_V} - L_H$$

where:

Q_0 = number of vehicles queued,

V_L = vehicle length (meters),

L_V = link velocity (m/cu),

L_H = link nominal headway travel time (cu)
and

T_x and C_T as described above.

$$3. \quad D = L_L - ((Q_0 - 1) \times V_L) - (L_H \times L_V)$$

where:

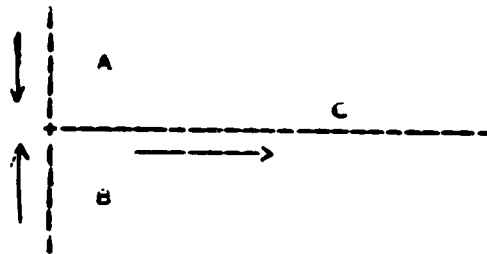
L_L = length of link (meters) and

Q_0 , V_L , L_H and L_V as described above.

A vehicle is scheduled to travel time, T , only if it is a time > 0 . Otherwise it is assumed that the vehicle has encountered and must join the link exit queue.

Merge Priority Decision Making

To determine the priority of vehicles merging from two guideway links onto another, the 1 x 2 matrix of upstream link pointers assigned to each merge output link is used. If, in the diagram below, links A and B merge onto C, then two upstream link pointers are identified for



link C (as shown) with the first one listed, in this case 'A', being



the link with priority.

Priority determination for vehicles merging at a juncture of a station exit ramp and a guideway link is somewhat more complicated. Here the user-supplied local merge priority table is used. This table, as pictured in the 2 x 2 matrix below, defines the priorities to be assumed by merging vehicles, which are dependent on the vehicle location as well as its "empty" or "occupied" status.

(1,1)	A	C	(1,2)
(2,1)	B	D	(2,2)

A = table entry (1,1), is the priority of an empty vehicle on the guideway,

B = table entry (2,1), is the priority of an occupied vehicle on the guideway,

C = table entry (1,2) is the priority of an empty vehicle on the station exit ramp and

D = table entry (2,2), is the priority of an occupied vehicle on the station exit ramp.

The higher the number assigned to A, B, C or D, the lower the priority. Thus, if the guideway vehicle is empty and the station vehicle is occupied, then a comparison would be made with A and D, with the lower value dictating the vehicle with priority.

6.2.72 EMGDIP4 - Generalized Data Input Variables

6.2.72.1 Identification

- o **EMGDIP4** - Generalized Data Input Variables
- o IBM/FSD - May 1, 1978
- o ASM

6.2.72.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
NAME	-	R*8	Parameter name
FMT	-	R*8	Format of data
IRAL	-	I*4	First dimension lower bound
IRAH	-	I*4	First dimension upper bound
IRBL	-	I*4	Second dimension lower bound
IRBH	-	I*4	Second dimension upper bound
IRCL	-	I*4	Third dimension lower bound
IRCH	-	I*4	Third dimension upper bound
IRDL	-	I*4	Fourth dimension lower bound
IRDH	-	I*4	Fourth dimension upper bound

6.2.72.3 Local Variable Dictionary

None.

6.2.72.4 Description

This routine provides addressability to each parameter defined in the Model Processor input data commons for Generalized Data Input

Processing (GDIP). The processing sequence performed by the routine is identical to that described for EIGDIP4.

6.2.72.5 PDL

See Appendix A.

6.2.72.6 Decision Tables and Algorithms

None.

6.2.73 EMODEL - Model Processor Main Control

6.2.73.1 Identification

- o EMODEL - DESM M.P. Main Routine
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.73.2 Argument Dictionary

None.

6.2.73.3 Local Variable Dictionary

None.

6.2.73.4 Description

EMODEL serves as the main routine for the DESM Model Processor. It provides the architectural mechanism for directing event processing within the simulation system. EMODEL is invoked by the Operating System via an indirect entry to EANTIX. Upon entry, system initialization is performed by invoking EAINIT. The basic control loop for accomplishing the recognition, scheduling and processing of transaction events is then started. This control process provides for obtaining the next transaction to be processed (EARFEL), updating the simulation clock, and invoking required architectural components to perform processing in response to a transaction event occurrence as follows:

1. Asynchronous Data Input - EAASYN is invoked to process data input.
2. Periodic Sampling - EASAMP is invoked to record statistics to the Sampling file.
3. Trip Arrival - EATORG is invoked to process the trip arrival event.
4. Checkpoint - EACKPT is invoked to record a periodic checkpoint of system status.
5. Prompt - EAPRMT decodes the dequeue request and initiates appropriate station (EAPSTN) or guideway (EAPLNK) dequeue processing.

6. **Guideway or Station Modeling Event-** EAGCTL and EASCTL are invoked as appropriate to sequence transaction processing through the modeling subsystem. This sequencing process is continued until the transaction is rescheduled for event completion on the Future Events List.
7. **Periodic Computation -** EAPCMP is invoked to process the required event.
8. **Simulation Termination -** The occurrence of this event results in termination of the MP after EAFINS is called to perform simulation termination activity.

6.2.73.5 PDL

See Appendix A.

6.2.73.6 Decision Tables and Algorithms

None.

6.2.73a ESAREQ - Assign Request to Vehicles

6.2.73a.1 Identification

- o ESAREQ - Assign Request to Vehicles
- o GM TSC - July 1, 1981
- o PARAFOR

6.2.73a.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TRNID	-	I*4	Transaction ID of request
VEH	-	I*4	Vehicle ID assigning vehicle
OSTN	-	I*2	Origin station of request
SLDOCK	-	I*4	Station dock link of vehicle
DSTN	-	I*4	Destination station of request

6.2.73a.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DDIRST	-	I*4	Destination directional station ID
ODIRPL	-	I*4	Origin directional platform ID
SL	-	I*4	Station link loop counter
XTN	-	I*4	Transaction ID for inventory management scheduling

6.2.73a.4 Description

This routine assigns a ready request to an available vehicle at the load berth during timeout/group demand responsive service. The request is assigned to the vehicle and then marked assigned. Inventory parameters for the origin directional platform are updated. Inventory parameters for the destination directional station corresponding to turnaround are updated. Inventory management for the destination directional station is scheduled.

6.2.73a.5 PDL

None.

6.2.73a.6 Decision Tables and Algorithms

None.

6.2.74 ESASAV - Initialize System Status Area Addresses

6.2.74.1 Identification

- o ESASAV - Initialize System Status Area Addresses
- o IBM/FSD - May 1, 1978
- o ASM

6.2.74.2 Argument Dictionary

None.

6.2.74.3 Local Variable Dictionary

None.

6.2.74.4 Description

ESASAV is a dummy CSECT which provides the ability to retrieve the addresses associated with the structured data and model processor common data regions within the System Status Area. These addresses are retrieved during initialization by referencing an identically named FORTRAN common which is equivalenced to the CSECT during link edit.

6.2.74.5 PDL

See Appendix A.

6.2.74.6 Decision Tables and Algorithms

a. System Status Area Addressability

See EADADD and EANSVA

6.2.75 ESBDL - Board List Generation

6.2.75.1 Identification

- o ESBDL - Generate a List of Boarding Trips
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.75.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TIM	-	I*4	(OUTPUT) Board event time in clock units
TMAX	-	I*4	(OUTPUT) Maximum number of boarding passengers
VEH	-	I*4	Vehicle ID available for boarding

6.2.75.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
BRDFLG	-	L*1	Flag indicating that trip is boarded
ENDQFL	-	L*1	Flag indicating that end of queue hit or board time expired
LSTTRP	-	I*4	Last trip in station boarding queue
RTE	-	I*4	Pointer to route group list
STN	-	I*4	Current station of vehicle
STN1	-	I*4	Next station on vehicle path to alternate destination
STOTPS	KMV	R*4	Number of passengers boarding a vehicle
STOTPW	KMV	I*4	Time needed to board and secure all handi-capped passengers boarding
STOTPY	-	I*4	Time from start of board to completion of boarding current trip, including secure time
TOFLAG	KMV	L*1	Indicates vehicle "timeout"
TOFTST	-	L*1	Indicates a vehicle timed out on trip being tested

VARIABLE	DIM	TYPE	DESCRIPTION
TPASSX	-	I*4	Trip door time passenger equivalent count
TRP	-	I*4	Temporary trip ID
TRPID	-	I*4	Trip ID
TRPID1	-	I*4	Trip ID
TRPNTR	KMV	I*4	Vehicle pointer to trip considered for boarding
TRPNXT	-	I*4	Next trip in boarding queue
TRTE	-	I*4	Trip's route group ID
ULT	-	L*1	Flag indicating completion of boarding list generation
VEHSKP	KMV	I*4	Counter of number of cycles to skip a vehicle in boarding operation
VEH2	-	I*4	Loop counter - vehicles
VEH3	-	I*4	Loop counter - vehicles
YHCBRD	KMV	I*4	Number of handicapped passengers boarding a vehicle
VOCC	KMV	I*4	Number of passengers on vehicle
VTRCH2	KMV	I*4	Temporary vehicle train chain pointer

6.2.75.4 Description

ESBDL is invoked to initiate the boarding process for a single vehicle. The selection procedure used in determining the eligibility of waiting trips for possible boarding differs for the different service types as follows:

- a. In the case of demand responsive single party service, the trip selected for boarding is either a trip with a confirmed reservation or the trip at the head of the station's boarding queue (longest waiting). The selected trip is removed from the station boarding queue and assigned to the boarding list of the vehicle. The number of boarding passengers is given by the trip size (associated number of patrons). In either case, the boarding trip determines the vehicle's ultimate destination and adjustments to boarding times and special occupancy are made for boarding handicapped passengers.
- b. In the case of demand responsive multiparty service, trips with reservations are boarded first and the ultimate destination of the vehicle is determined considering all trips currently on the vehicle and those selected for boarding as reserved trips. This ultimate destination corresponds to the most remote station stop required for the vehicle and serves as a limiting factor in selecting other potential trips for boarding. If space remains on the vehicle after initial trip selection, additional trips waiting at the station, regardless of whether they have reserved other vehicles, are entered in the vehicles boarding list as long as they can be accommodated. The trip selection process is performed by cycling through the station boarding queue until the last trip has been processed or vehicle capacity is reached. Acceptability for boarding depends on space available and trip compatibility with the vehicle's destination as determined by invoking ESTCHK. An additional requirement is placed on trips which have outstanding vehicle reservations. Such trips are only considered for boarding if the vehicle on which a reservation was made has not yet entered the station. If this is the case the vehicle reservation is cancelled and the trip is entered into the boarding list. Otherwise, the trip must wait for servicing by the reserved vehicle. Adjustments to boarding times are made for boarding handicapped passengers.
- c. In the case of scheduled service, all of the vehicles in a train are boarded evenly and the overlap of deboarding and boarding within a train is modeled. The algorithm cycles through a train boarding at most one trip on a vehicle in a cycle. Each trip in the boarding queue is evaluated to determine if the patrons can be accommodated by the vehicle's route. If the trip can be serviced by a defined route group, the route group list is searched to determine if the vehicle route is assigned to the specific group. Compatibility results in the trip being assigned to the route on which the boarding vehicle is traveling. Compatible trips which can be accommodated while space and boarding time permits on a vehicle are added to the vehicle's boarding list.

Upon entry to ESDL, a test is made for existence of trips in the boarding queue. If the queue is empty, the time of board is computed using the maximum deboard time and the minimum door open time and placed on the FEL, and the subroutine exited.

If the trips are present, copies of the train pointers are made into variable VTRCH2(VEH), and the number of "board cycles" that must be spent in completing the deboard event for each vehicle is computed as follows: the board overlap for the vehicle is subtracted from the maximum board overlap for a vehicle in the train to determine how much time the earliest vehicle boards before the current vehicle starts, and divided by STBA to get the number of passengers the earliest starting vehicle can board in this time.

$$VEHSP(VEH) = STBOV(VEH) - STDOV(VEH)/STBA$$

(In the event STBA=0, no overlap is modeled (VEHSP=0). Since STBA=0 causes fixed board time, the number of passengers boarding each vehicle causes no variation in board time.) A pointer into the trip queue TRPNTR for each vehicle is initialized to the earliest arriving trip.

The process then begins the actual board loop. The loop is exited when the station queue becomes empty or when the board terminate flag ULT is set.

The cycle is started with the first vehicle. A test is made to determine if the vehicle has scanned the entire trip queue, or reached time or available space limits. If so the vehicle has completed boarding, and is removed from further board processing by dechaining it from the VTRCH2 list and advancing to the next vehicle on the list. If it was the last boarding vehicle, the flag ULT is set and boarding terminates. A test is then made as to whether the vehicle is still deboarding or boarding a multi-passenger trip.

If VEHSP(VEH)=0, the vehicle is not available to board this cycle, so the VEHSP counter is decremented, and the next vehicle tested. If the vehicle can board, it scans the trip queue until it finds a trip which can board, hits the end of the queue, or finds a trip unable to board only because of the time limit. In the latter case, the procedure defined in the "timeout check" description is executed. If the last trip in the queue is scanned, (whether or not it boards), the loop is exited and the vehicle is removed from the VTRCH2 board cycle list as above. Also, if end of queue is hit, the value of LSTRP, the end of queue indicator, is redefined, in case the trip was boarded. If a trip is boarded, and the end of queue is not hit, the cycle goes to the next vehicle and performs the same procedure.

Note the use of separate pointers for each vehicle. This is done so a vehicle can reject a trip and not reconsider it, but another vehicle can board it. This is important especially when vehicles are nearly full and some vehicles may not have space for large trips. This process requires that a vehicle boarding a trip first shift down any pointers of other vehicles that are pointing at the same trip, to keep them on the board queue list. Note also the use of VEHSKIP(VEH) to skip board cycles for remaining deboarding passengers, and also to see that a vehicle uses as many cycles to board a trip as there are passengers in the trip.

Two logical*1 local variables TOFLAG(KMV) and TOFTST have been defined in ESDL. TOFLAG(VEH) is initialized to F in the initialization loop for each vehicle prior to starting the scheduled service board cycle.

In ESDL3, if a vehicle "times out" on a trip, a test is made over all vehicles of the train (using link pointer VTRNCH, not VTRCH2 which links only vehicles still boarding) to see if any vehicle "timeout" flag is set, with trip pointer TRPNTR for the vehicle pointing at this trip. If so, the loop flag TOFTST is set, and the trip is skipped by the boarding vehicle (next trip is tested). If no vehicle has "timed out" on this trip, then the vehicle timeout flag TOFLAG is set to T, and the vehicle pointer is reset to point at the "timeout" trip (it was advanced to the next trip) and the vehicle is removed from the board list.

- d. In timeout/group demand responsive service, only passengers who are part of the ready group assigned to the vehicle are added to the vehicle's board list. The boarding time is adjusted for handicapped passengers.

In all types of service, the board time is computed by first determining the minimum remaining "door open" time after completing the deboard event. This is determined by subtracting the board overlap time (STBDOV(1)) from the minimum remaining door open time STDBMN(1) for the lead vehicle. This time interval is the same for all vehicles in the train. Then the algorithm computes the total board time for each vehicle in the train (STBA*Y1 number of boarding passengers) and subtracts the overlap value computed in ESDL. The time for completion of the board event is the maximum of the minimum remaining door open time and the total board time. The value of SPBDMX(VEH) is reduced to a maximum value of VCAP which is the maximum number of passengers that can possibly board a vehicle in scheduled service.

6.2.75.5 PDL

See Appendix A.

6.2.75.6 Decision Tables and Algorithms

The following algorithm is used for processing a deboard/board sequence on the same link (i.e., deboard and board on the same dock).

$T_{im} = \max (SMNDBT, \min(\max(SMXDBT, STDBA \cdot X), STDBA \cdot X, STBA \cdot Y))$.
taken over the vehicles of a train (see Figure 6-8).

STDHFF is unused except in the computation of route headways.
Also $Y_{max} = (SMXDBT - STDBA \cdot X) / STBA$ if STBA is not zero (see Figure 6-9).

In addition, in the case of scheduled service:

A. Each vehicle's deboard time is separately subtracted from the total dwell time limits to determine the amount of time available for board, and the number of passengers that can board. If one or more vehicles deboard past the time limit, other vehicles are permitted to extend their board times up to the time all vehicles have completed deboarding.

B. Passengers are boarded on the vehicles simultaneously up to the time limit, or until a vehicle is full or the station queue empty. If a vehicle is full, other vehicles continue to board in the same manner. Also, if a vehicle completes deboarding first, it is allowed to board ahead of other vehicles that are still deboarding.

For each vehicle "i" in the train, the number of passengers y_i to be boarded on the vehicle is computed as:

$$0 \leq y_i \leq (\max(SMXDBT, STDBA \cdot x_m) - STDBA \cdot x_i) / STBA,$$

(for $STBA > 0$)

where x_i = number of deboarding passengers from vehicle i.

x_m = maximum number of deboarding passengers from a vehicle (i.e., vehicle m) of the train.

The total time for deboard and board for the train is computed as,

$$TIM = \max (\max(SMNDBT, \text{over all vehicles in the train } STDBA \cdot x_i + STBA \cdot y_i))$$

In addition, the following condition is also true:

C. In the event that a trip met compatibility and space constraints attempting to board a vehicle, but the vehicle did not have time to board a trip, this trip will not be boarded on this vehicle.

However, the station model test for trips that have been previously rejected because of the time limit and will not allow them to time out on other vehicles of the train.

In the case of deboard and board on separate links, the minimum and maximum times are applied against the board event time only, as is the passenger limit. No deboard/board overlap occurs. The rest of the algorithm (even boarding, all vehicles starting together) is the same. The time for the deboard event is given by

$$T_d(\text{Veh}) = X * STDBA$$

and the board time

$$T_b = \max(\text{SMNDBT}, \min(\text{SMXDBT}, Y * \text{STBA}))$$

with passenger board limit

$$Y_{\max} = \text{SMNDBT} / \text{STBA}, \text{ with no time constraint for } \text{STBA} = 0$$

(the vehicle available space is also tested).

Handicapped Passenger Boarding Algorithm

Board time = $\text{Max}((n \cdot \text{SHCBA}) + \text{SHCBB}, n \cdot \text{SHCBA} + z \cdot \text{STBA})$

n = # of boarding handicapped passengers

z = # of boarding ordinary passengers

Note that SHCBA and SHCBB are rounded to an integer multiple of STBA in the input processor.

6.2.76 ESCHED - Schedule a Transaction on Future Events List

6.2.76.1 Identification

- o ESCHED - Schedule a Transaction Macro
- o IBM/FSD - May 1, 1978
- o PLI

6.2.76.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DELTA	-	C	event duration time
INDEX	-	C	transaction ID
MEVNT	-	C	modeling event identifier
TYPE	-	C	architecture event control indicator (G=guideway, S=station, T-trip)
XRTY	-	C	priority, $\phi \leq XRTY \leq 9$ where ϕ is highest

6.2.76.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
GOTO	-	C	system event number
M	-	C	margin control
OUT	-	C*1	generated FORTRAN code

6.2.76.4 Description

ESCHED provides the capability for placing a non system transaction on the Future Events List. In the case of a zero time event, the macro bypasses Future Events List scheduling and sets a global modeling variable (AZSCHD) to indicate the occurrence. This variable is checked by the simulation architecture to determine if immediate processing can be performed for a given transaction. If the indi-

cator is set, the transaction is repetitively entered into the appropriate modeling subsystem until a non zero time activity is required. This capability, adds to efficiency of the simulator by eliminating unnecessary processing overhead. In the case of a non zero scheduling time, EAPFEL is invoked to place the transaction in the proper time ordered position on the FEL.

6.2.76.5 PDL

None

6.2.76.6 Decision Tables and Algorithms

None

6.2.77 ESDBL - Deboard List Generation

6.2.77.1 Identification

- o ESDBL - Generate a List of Deboarding Trips
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.77.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ATERM	-	L*1	Flag indicating vehicle marked for AFSM termination
VDBTIM	-	I*4	(OUTPUT) total deboard time for a vehicle
VEH	-	I*4	ID of vehicle undergoing deboard processing

6.2.77.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DONE	-	L*1	Logic variable for indicating end of trip list
LSTTRP	-	I*4	ID of last trip onboard vehicle
RATIO	-	R*4	Computed ratio of nominal to actual travel time
STN	-	I*4	Station ID for vehicle
STOTPS	-	I*4	Actual passenger board count for a vehicle
TERM	-	L*1	Logical variable to indicate termination of degraded vehicle
TRP	-	I*4	Deboarding trip ID
TRPID	-	I*4	Deboarding trip ID
TRPID1	-	I*4	ID of next trip to consider for deboarding
TRVT	-	I*4	Excess travel time
VDBMIN	-	I*4	Minimum time needed to deboard all deboarding handicapped passengers from a vehicle (including release time)

6.2.77.4 Description

ESDBL is invoked for each vehicle entering the deboard process to identify onboard trips which must deboard at the current station. The selection process involves examining each trip to determine if its final, intermediate or alternate destination is the current station. If the vehicle is degraded or in by-pass mode as part of a failure response or has been marked to terminate for active fleet size management, all onboard trips are required to deboard, regardless of destination compatibility.

Each trip which must disembark is removed from the onboard vehicle queue and placed in either the deboard or transfer list of the vehicle. The deboard list contains all trips deboarding at their termination station or their alternate termination station. The transfer list contains all trips using the station as a transfer point as given by their intermediate destination or as forced by failure or active fleet size conditions. As each trip is placed in the appropriate vehicle deboard queue, the count of deboarding passengers is incremented for use in computing deboard event completion. The nominal and actual travel times are computed for each terminating trip placed in the vehicle deboard list and if required, ESTLOG is invoked to record trip completion in the Completed Trip Log file. Deboard time is computed taking into account deboard parameters and minimum deboard event time constraints introduced by handicapped passenger deboarding.

6.2.77.5 PDL

See Appendix A.

6.2.77.6 Decision Tables and Algorithms

Handicapped Passenger Deboard Adjustments. Variables VDBTIM (total deboard door time) and VDBHMM (earliest end of deboard) are set to zero. When an ordinary trip (THCFL(TRPIP)=0) is encountered, the variable VDBTIM is incremented by STDBA*TPASS (TRPID) the time per passenger deboard times the number of passengers in the trip. When a handicapped trip is encountered VDBTIM is incremented by the handicapped passenger deboard time SHCDBA (TPASS=1 for all handicapped trips), and also if VDBHMM=0, it is set to SHCDBB + SHCDBA, the minimum time needed to deboard the handicapped passenger (release time and door time). If VDBHMM≠0 indicating that a handicapped passenger has already deboarded, VDBHMM is incremented by SHCDBA, giving the additional time needed to deboard the additional handicapped passenger. (All handicapped passenger release times begin at the start of deboard.) The maximum of the handicapped passenger deboard time VDBHMM and the total passenger door time VDBTIM is used as the actual deboard time, since ordinary passengers can deboard during the handicapped passenger release time. (No passengers can board until all deboards of handicapped and non-handicapped passengers are completed.) The deboard time, rather than the deboard passenger count, is now returned to the calling program.

When a handicapped passenger deboards, the handicapped passenger vehicle occupancy, as well as the total vehicle occupancy, is decremented by one. Finally, in computing the number of seats emptied by deboarding passengers, any handicapped passengers remaining on the vehicle must be added to the number of deboarding passengers in comparing with the number of prior standing passengers (all handicapped passengers are included in this count) to see if any vehicle seats were left empty and if so, how many.

Deboard time = Max $((m*SHCDBA)+SHCDBB, m*SHCDBA+y*STDBA)$, (for $m > 0$)

minimum = SMNDBT - deboard time

maximum = SMXDBT - deboard time

m = # of deboarding handicapped passengers

y = # of deboarding ordinary passengers

6.2.78 ESDIVF - Diverge Functions

6.2.78.1 Identification

- o ESDIVF - Select and Order Downstream Station Links by Occupancy
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.78.2 Argument Dictionary

None

6.2.78.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
SLN	KMSL	I*2	ordered list of downstream station links

6.2.78.4 Description

ESDIVF provides the ability to bypass automatic next entity selection within the station model and implement a defined alternate selection procedure. These alternate procedures correspond to diverge functions, used for developing a list of downstream next entity possibilities based upon a specific selection rule. The use of these procedures is specified via a diverge function ID coded by the user for a particular station link. Specification of diverge function ID 1 or 2 results in ESDIVF developing a list of downstream next entity possibilities, regardless of type or vehicle status, ordered by total vehicle occupancy or berthing occupancy, respectively. The ordering of possibilities is performed by invoking ESDIVO. Specification of any other diverge function ID results in the user interface routine EUDIVF being invoked to perform a user implemented selection procedure.

6.2.78.5 PDL

See Appendix A.

6.2.78.6 Decision Tables and Algorithms

None

6.2.79 ESDIVO - Diverge Function Ordering

6.2.79.1 Identification

- o ESDIVO - Order a List of Station Links by Occupancy
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.79.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
IND	-	I*4	ordering indicator 1 = total occupancy 2 = berthing occupancy
SLN	KMSL	I*2	downstream station link list

6.2.79.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
SLN	KMSL	I*4	downstream station link list
TEMP	-	I*4	sort variable

6.2.79.4 Description

ESDIVO sorts a list of station by either total link occupancy or berthing occupancy. The sorted list is arranged in increasing order.

6.2.79.5 PDL

See Appendix A.

6.2.79.6 Decision Tables and Algorithms

None

6.2.80 ESEVA - Empty Vehicle Assignment

6.2.80.1 Identification

- o ESEVA - DESM Empty Vehicle Assignment
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.80.2 Argument Dictionary

None

6.2.80.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
FND	-	L*1	assignment option availability indicator
MX	-	R*4	temporary variable for maximum station value for the option being processed
PRTY	-	I*4	index of the selected option being processed
RSTN	-	R*4	generated random number for cumulative distribution
RTE	-	I*4	temporary variable for station circuitous empty route assignment
SLOCC2	-	I*4	station link occupancy doubled
SRATIO	-	R*4	ratio of anticipated need to empty availability for a station
STN	-	I*4	current station location for vehicle
STN1	-	I*4	temporary variable for station ID to where vehicle is to be dispersed.

VARIABLE	DIM	TYPE	DESCRIPTION
STN2		I*4	temporary variable for station ID

6.2.80.4 Description

ESEVA is invoked during next station selection (ESNSTN) to provide for empty vehicle assignment in demand responsive operation. The next station for a vehicle is determined based on a user-specified list of priorities of where empties are to be distributed in the network. Each entry in the option list is processed in the order specified until an available option is found. If ESEVA fails to find a suitable assignment, the empty vehicle is routed to its current station and an information message is displayed on the system output device.

The processing performed for each of the six possible empty vehicle assignment options is as follows:

- a. Local Storage - A vehicle may be assigned to local storage if the storage occupancy is below capacity.
- b. Regional Storage - A vehicle may be launched to a regional center if the storage occupancy of the regional station center assigned to the present station is below capacity.
- c. Distribute According to Anticipated Need Without Considering Current Availability of Empties - If a list of receiving stations for empty vehicles dispersed from the present station exists, a random number is generated. This number is compared to the entries in a cumulative distribution table corresponding to each of the receiving stations. The station selected is that station, whose cumulative distribution function comes the closest to without exceeding the generated random number

If a list of receiving stations for empty vehicles does not exist for the present station, the vehicle is assigned to the next downstream station for empty dispersal.

- d. Distribute According to Anticipated Need While Considering Current Availability of Empties- For each station listed as a receiving station, a ratio is calculated of the assigned anticipated need value to the current number of empties available in the station storage area. Both the anticipated need and storage link occupancy (if not zero) are doubled to differentiate storage link occupancy of 1 from zero. The station selected

for empty vehicle assignment is the station with the maximum ratio ($>=1$), which reflects the station with the current greatest need.

- e. Circulate on the Guideway on a Predetermined Route - If a circuitous route is assigned to the current station the vehicle is assigned to the route and routed to the next station in the circuitous route list. The station destination is determined by searching the particular circulation route (list of station IDs) for the current station ID, which is followed by the identification of the next station along the route. If the current station is not found or is the last station listed along the route, the station assignment becomes the first station identified in the circulation route.
- f. Circulate to Next Best Station - Each station in the system is queried to determine the number of outstanding (unsatisfied) trip requests. The empty vehicle is then routed to the station with the maximum number of unsatisfied requests.

Vehicles assigned for empty dispersal based upon the first four options discussed above, are identified as having to divert to storage upon entry to the assigned station. However, for the latter two cases, vehicles are routed to the docking area to accommodate waiting trips. If no trips are boarded, the vehicle is again assigned to a station according to the empty vehicle option specifications.

6.2.80.5 PDL

See Appendix A.

6.2.80.6 Decision Tables and Algorithms

(See the discussion of the Circuitous Empty Route List in the description of EGCNXT.)

6.2.80a ESEVB - Empty Vehicle Bumping

6.2.80a.1 Identification

- o ESEVB - Empty Vehicle Bumping
- o DOT/TSC - July 1, 1981
- o PARAFOR

6.2.80a.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
VEH	-	I*4	Vehicle ID of bumped vehicle
STN	-	I*4	Station ID of bumped vehicle storage

6.2.80a.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
FND	-	L*1	Bumping option availability indicator
I	-	I*4	Loop counter
MX	-	R*4	Maximum station value
PRTY	-	I*4	Index of option being processed
RSEED	-	R*4	Not used
RSTN	-	R*4	Generated random number for cumulative distribution
RTE	-	I*4	Station circuitous empty route assignment
SLOCC2	-	I*4	Storage link occupancy doubled
SRAT10	-	R*4	Ratio of anticipated need to empty availability for station
STN1	-	I*4	Temporary station ID to where vehicle is to be bumped
STN2	-	I*4	Temporary station ID

6.2.80a.4 Description

ESEVB is involved during station link testing of an input ramp to choose a destination for a stored vehicle "bumped" from the storage link as another vehicle enters the station if vehicle bumping has been enabled during demand responsive service. The next station for a "bumped" vehicle is determined based upon the user-specified list of priorities of where empties are to be distributed in the network. Each entry in the option list is processed in the order specified until an available option is found. If ESEVB fails to find a suitable destination, the "bumped" vehicle is routed to its current station and an information message is displayed on the system output device.

The processing performed is identical to that described in Section 6.2.80.4, ESEVA Description, paragraphs a. through j. with the exception of a., local storage, which is bypassed.

6.2.80a.5 PDL

None.

6.2.80a.6 Decision Tables and Algorithms

None.

6.2.81 ESFAIL - Station Failure/Recovery

6.2.81.1 Identification

- o ESFAIL - Station Failure/Recovery Processing
- o IBM/PSD - May 1, 1978
- o PARAFOR

6.2.81.2 Argument Dictionary

None

6.2.81.3 Local Variable Dictionary

None

6.2.81.4 Description

ESFAIL is invoked by EAASYN in response to the occurrence of an asynchronous station failure event. Upon entry to the routine, a determination is made as to whether a failure or recovery event has been entered. If a failure occurrence is indicated, appropriate status items associated with individual station links are set based upon the input data associated with the failure condition. This processing results in the setting of station link entry or exit flags to preclude vehicle entry or exit while the failure is effect. Total station failure results in entry to the input ramp and in exit from the output ramp being blocked simultaneously without affecting movement internal to the station itself. Specification of a failure recovery results in the resetting of station link entry or exit status indicators and the scheduling of a system service transaction to invoke appropriate dequeue (prompt) action for the restart of any vehicles preempted by the failure. In the case of entry recovery, the recovered link is specified as the prompt entity to cause dequeuing of upstream waiting vehicles. Exit recovery results in scheduling of a dequeue for the recovered link. Total station recovery results in the scheduling of two prompt transactions to cause dequeuing to occur on both the guideway link leading to station entry and the station output ramp.

6.2.81.5 PDL

See Appendix A.

6.2.81.6 Decision Tables and Algorithms

None.

6.2.82 ESLDLY - Launch Delay Computation

6.2.82.1 Identification

- o ESLDLY - Launch Delay Computation
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.82.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TIME	-	I*4	(OUTPUT) - vehicle launch delay time

6.2.82.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ACOST	-	R*4	alternate path cost
INFINY	-	I*4	largest integer constant
PCOST	-	R*4	primary path cost

6.2.82.4 Description

ESLDLY is invoked during vehicle launch processing to determine the delay time associated with merge scheduling and to perform final vehicle path selection. If deterministic type dispatch is required, ESMDLY is invoked to schedule the merges on the primary and alternate traversal paths to the vehicles next destination. This process results in a delay time being determined for each path such that congestion free guideway traversal can be accomplished. If merge scheduling is not possible for a particular path, the process results in elimination of the path as a possible route for the vehicle. If merge scheduling is not required, the delay time associated with the primary and alternate paths are set zero and processing proceeds immediately with final path selection.

The path selection process begins by computing the cost of traversing the primary and alternate paths to the vehicles next

destination. If no possible path exists for the vehicle, as determined during merge scheduling, the vehicle launch retry indicator is set and a launch delay time corresponding to the time period encompassed by the merge scheduling table is returned to the calling routine. Otherwise, the cost of each remaining path is computed as a weighted sum of original path cost, computed by ESPATH, and merge scheduling delay. The minimum cost path is then assigned to the vehicle and the merge scheduling delay associated with the path is returned as the required launch delay time. Once path selection is complete, the merge reservation table is updated to reflect the planned arrival of the vehicle at each merge on the selected path.

6.2.82.5 PDL

See Appendix A.

6.2.82.6 Decision Tables and Algorithms

- a. Original Path Cost - See ESPATH
- b. Merge Scheduling Delay - See ESMDLY

6.2.83 ESLEAV - Station Link Exit

6.2.83.1 Identification

- o ESLEAV - Perform Station Link Exit Processing
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.83.2 Argument Dictionary

None

6.2.83.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
COUNT	-	I*4	number of vehicles assigned to next platoon
FOUND	-	L*1	logical variable for indicating board/deboard events
LINK	-	I*4	guideway link bypassing station
LMK	-	I*4	link ID of link on tow path
S	-	I*4	station ID of station on tow path
SL	-	I*4	station link ID
SLN	-	I*4	downstream station link ID
SLP	-	I*4	previous station link ID
STN	-	I*4	station ID
TIM	-	I*4	delta time for scheduling failed vehicle restart
TIME	-	R*4	current simulation time in seconds
TRP	-	I*4	trip ID on vehicle
VEH	-	I*4	vehicle ID
XTN	-	I*4	transaction ID

6.2.83.4 Description

Station link exit processing is invoked by station model event control after the next entity for the vehicle has been determined. Exit processing involves performing housekeeping tasks associated with the vehicle or train departure from a station link. The vehicle occupancy count for the station link is decremented to reflect departure of the vehicle and any entrained followers. Exit of a single vehicle from the station output ramp results in the vehicle train length being set to zero as required for correct interface to the guideway link model. If the link being exited is a dock, as determined by the presence of board/deboard events, and the vehicle departing is the last on the link, berthing occupancy is reset to zero in order to free serial berths for future arriving vehicles. If platooning is required at the dock, and all berths have been cleared, platoon processing for upstream waiting vehicles is performed. The formation of platoons involves the sequential search of upstream links to determine if vehicles waiting for berth assignment are present. If vehicles are queued at the exit of an upstream link, they are assigned the dock link as their next entity and are indicated as eligible for dock entry. The number of vehicles assigned to the platoon in this process is limited to the number of serial berthing positions available at the dock. Movement of these vehicles to the unoccupied dock is initiated by the prompt scheduling described below.

Since the departing vehicle/train may have inhibited other vehicles from downstream movement, a system prompt transaction is scheduled for attempted dequeue of vehicles present on immediate upstream links. If any vehicles are present in the current station and queued awaiting event completion by the exiting vehicle, another prompt is scheduled for the current station link to allow waiting vehicles to continue current link traversal and processing.

On the other hand, if the departing vehicle has been followed by a vehicle awaiting a launch event, hence not ready to proceed, then the following vehicle is scheduled immediately to proceed with launch event processing. When a vehicle leaves the station, a prompt is issued at the entrance to the station to dequeue vehicles which may have been prevented from entering the station by total station capacity or station gate flag restraints.

If an unclear tow path exists, ESLEAV tests whether the tow path will become clear if this vehicle is leaving a station link on the tow path but not entering another guideway or station link on the tow path. If the path does clear, the restart is scheduled at the time calculated for the tow vehicle to reach and restart the failed vehicle.

6.2.82.5 PDL

See Appendix A.

6.2.82.6 Decision Tables and Algorithms

None.

6.2.84 ESLMDL - Station Link Model

6.2.84.1 Identification

- o ESLMDL - Station Link Model
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.84.2 Argument Dictionary

None.

6.2.84.3 Local Variable Dictionary

None.

6.2.84.4 Description

ESLMDL provides sequencing control for the station link model and model specific event processing on individual station links. These processing includes:

1. Headway zone traversal
2. Link travel zone traversal
3. Deboard processing
4. Board processing
5. Storage event processing
6. Launch processing

All station links are configured to contain some or all of the above events in some specified sequence. The Station Link Model simulates the movement and actions of a vehicle within a station link by advancing it through a sequence of events. The processing required to process a vehicle for a station link event consists of the following: event initiation processing, next event determination, and event completion processing.

Each micro event within the station model requires these basic processes. For example, event initiation processing for a deboard event determines which trips will be deboarded and the amount of time required for deboarding process. Event completion physically removes trips from the vehicles and performs follow-up tasks associated with deboarding such as processing transfer trips. The next event processing determines the next scheduled activity for the vehicle based on the sequence of events for a particular station link.

ESLMDL is invoked by the system architecture and in turn invokes the required modeling component as follows:

ESMDLA - event processing completion

ESMDLN - next event determination

ESMDLB - event processing initiation

In the general case, execution of these components in the order specified above represents the procedural sequence for moving vehicles through each specified event on a particular station link. Exceptions to this sequence occur under the following conditions:

1. Initial vehicle entry into the station.
2. Vehicle removal from the active fleet
3. Completion of all events for a particular station link
4. Next event unavailability
5. Special event modeling, such as ripple berth advancement.

Upon initial vehicle entry to the station model a determination is made as to whether completion of a scheduled station event has occurred. If no prior event is found for the vehicle, event completion processing is bypassed and initial entry processing is performed. If the vehicle is currently located in the station and is entering a new station link, its location within the station is updated and next event determination is performed. If the vehicle is just entering the station from the guideway, event completion processing is also bypassed and the following is performed to accomplish vehicle transition from the guideway network:

1. The current modeling entity for the vehicle and any entrained followers is set equal to the station.
2. On-line station bypass conditions are established. If the station is on-line and not the next stop for the vehicle, a status indicator is set for bypass of board/deboard events and link traversal at bypass velocity.

3. Entrained vehicles are removed from the station arrival list if the station corresponds to the reserving station for the train. Unentrained vehicle removal from the arrival list is accomplished prior to station entry by the Guideway Link Model.
4. Single vehicles are assigned a train length of one to provide correct interface for Station Model processing.
5. Next entity determination is performed to begin station processing for the vehicle.

For demand responsive service, vehicle removal from the active fleet occurs after completion processing for the station deboard event. In this case, next entity determination and event initiation processing is bypassed and control is returned directly to the system architecture Station Model Control.

If next event determination indicates the completion of all required events on a particular station link, event initiation is bypassed and control is returned to the system architecture. Station Model Control then determines the next required entity for the vehicle and appropriately enters it into either the Guideway or Station Model for initial event processing. Next event determination can also result in an indication that the vehicle cannot perform its required launch event due to the presence of an immediately preceding in process vehicle. In this case, event initiation is also bypassed and the vehicle enters a nonqueued wait state until availability of the launch event occurs.

Typically, vehicle scheduling on the FEL occurs as the result of event initiation processing. If this scheduling does not occur, responsibility for vehicle processing passes to the system architecture. An exception to this rule occurs in the case of event preemption as described above which is handled internal to the Station Model itself.

6.2.84.5 PDL

See Appendix A.

6.2.84.6 Decision Tables and Algorithms

None.

6.2.85 ESLWTQ - Station Link Queuing

6.2.85.1 Identification

- o ESLWTQ - Enter a Vehicle into a Station Link Queue
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.85.2 Argument Dictionary

None.

6.2.85.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
FOUND	-	L*1	flag indicating failure data found
SL	-	I*4	current station link number
STN	-	I*4	current station number
VEH	-	I*4	vehicle ID
VEHN	-	I*4	temporary vehicle ID
XTN	-	I*4	transaction ID

6.2.85.4 Description

ESLWTQ is invoked by Station Model Control (EASCTL) to queue vehicles at exit of their present station link when forward movement is not possible.

This condition may arise due to failures, congestion or entry restrictions (e.g., merge priority, berth assignment) placed on the next entity for the vehicle. Movement of the queued vehicle is only restarted as the result of a prompt request, scheduled in response to other simulation activity. ESLWTQ assigns the vehicle at the head of the station link exit queue a queued status to reflect its inability to enter its next station or guideway link or perform its next station link event.

If the lead vehicle has entered a queue and is directly behind a train led by a degraded vehicle, it may be necessary to form a push coupling train. If the failure has been detected and the recovery method is push coupling, then the coupling is effected by invoking routine EAFTRN, the new minimum path table is activated, and the restart is scheduled for the user input delta time.

6.2.85.5 PDL

See Appendix A.

6.2.85.6 Decision Tables and Algorithms

None.

6.2.86 ESMDLA - Station Link Event Completion Processing

6.2.86.1 Identification

- o ESMDLA - Perform Event Termination Activity for a Specific Station Processing Event
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.86.2 Argument Dictionary

None.

6.2.86.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DIRPLA	-	I*4	Directional platform ID
DIRSTN	-	I*4	Directional station ID
LOADF	-	R*4	Vehicle load factor
NPASS	-	I*2	Number boarding/deboarding passengers
NPC	-	I*4	Number split parties passengers completed for trip
NPSEAT	-	I*4	Number seated passengers on vehicle
NTC	-	I*4	Number of split parties completed for trip
NTRPS	-	I*2	Number boarding/deboarding trips
PWLKT2	-	I*4	Passenger walk time
PWSTN2	-	I*4	Reboard station index
SL	-	I*2	Current station link for vehicle
STN	-	I*2	Current station for vehicle
STN1	-	I*4	Temporary station ID
TACT	-	I*4	Sum split party nominal travel time
TID	-	I*4	Dequeued trip ID

TIDNXT	-	I*4	Next trip ID in trip queue chain
TID1	-	I*4	Temporary trip ID
TIME	-	I*4	Trip initial arrival time
TNOM	-	I*4	Sum split party actual travel time
TRP	-	I*4	Temporary trip ID
VEH	-	I*4	Vehicle ID
VEH1	-	I*4	Temporary vehicle ID
XTID	-	I*4	Transaction ID

6.2.86.4 Description

ESMDLA is invoked by the station Link Model (ESLMDL) to perform processing associated with completion of a station link event. Event completion occurs at the conclusion of the time delay associated with event performance. The completion processing performed is based upon event type as follows:

Headway Zone Traversal

After processing the headway zone traversal event, headway zone occupancy is reset to allow subsequent vehicle entry. A prompt for the upstream link is scheduled to resume movement of any vehicle waiting at the exit of a prior link for entry onto the current station link.

Travel Zone Traversal

No follow-up processing is required for a vehicle subsequent to its station link travel event. Processing continues with Next Station Link Event Determination.

Deboard Processing

Each trip which deboarded each vehicle in the train is removed from that vehicle's list of deboarding trips. The passenger count is decremented by the number of patrons associated with the deboarding trips. As each trip is removed from the deboarding list, trip completion processing is performed based upon whether the trip is at an intermediate or final destination as described below. In the case of degraded vehicles trips deboarding the vehicle are processed as if an intermediate destination has been reached.

1. Intermediate Destination - Trips deboarding at an intermediate destination must transfer at the station and reenter the station boarding queue for trip resumption. This reentry process is delayed by a walk time which is used to account for transfer related activity performed in the station or in transferring between stations. After the walk time delay is completed, the trips reenter the boarding queue for servicing as a new station demand processed by EATORG. If the transferring trip is required to walk to a new station prior to the reboarding process, the trip's intermediate destination is changed to the designated walk transfer station.

2. Final Destination -- Each trip is removed from the vehicle deboard list and either returned to the available transaction list or placed in the station deboard queue, if it is a subgroup of a non-terminating split trip. As each trip subgroup is removed from the vehicle deboard list a determination is made as to whether or not split trip completion has occurred. If the trip is the last subgroup to complete at the station, trip completion activity related to subgroup coalescing is performed. This involves computing travel time statistics for the coalesced trip, removing each trip subgroup from the station deboard queue and returning each trip transaction to the available transaction list. If the subgroup is not the last to complete for the split trip it is queued in the station deboard list awaiting arrival of the other subgroup associated with the split trip.

An exception to the above occurs in the case of a trip subgroup completing at an alternate station assigned as a result of entry blockage at its originally desired destination. The redirected trip is considered complete upon removal from the vehicle deboarding list. If the trip is the last subgroup to complete travel, trip completion activity for the other subgroups awaiting arrival of the trip at the original destination is performed. Otherwise the subgroup relationship between the other waiting split trips and the alternately assigned subgroup is terminated.

If a demand responsive vehicle becomes empty and has no reservations pending and active fleet size management requires a smaller vehicle fleet, then it is removed from service and no longer available within the active fleet at the conclusion of the deboard event.

Board Processing

Subsequent to the boarding event, each trip which boarded each vehicle in the train is added to the vehicle's trip list and removed from the vehicle's boarding list. The passenger count is incremented by the trip size and the boarding queue occupancy for the station is decremented to reflect passenger boarding.

Once all required trips have been boarded, the next station for the vehicle is determined by invoking ESNSTN.

Storage Processing

No processing is done in the Station Link Model after a vehicle has been stored since it comes under the control of Empty Vehicle Assignment.

Launch Processing

If reservations are enabled for demand responsive service, vehicle arrival recording is performed prior to vehicle station dispatch by invoking ESVALS. Trip demand to dispatch statistics are updated for arriving and transferring passengers boarding at the station.

Vehicle Retardation

Vehicle retardation at station exit for quasi-synchronous control or local merge priority is implemented as a pseudo event within the Station Modeling process. Vehicles which must retard are scheduled for a retardation delay corresponding to the headway interval specified for the guideway link immediately downstream of the station. Upon completion of the required delay, the modeling event indicator for the vehicle is reset to the last event in the station to force Next Event Determination (ESMDLN) to recognize station completion and retry the station exit process.

6.2.86.5 PDL

See Appendix A.

6.2.86.6 Decision Tables and Algorithms

None.

6.2.87 ESMDLB - Station Link Event Initiation Processing

6.2.87.1 Identification

- o ESMDLB - Perform Initial Processing and Scheduling for Station Event
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.87.2 Argument Dictionary

None.

6.2.87.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ATERM	-	L*1	Flag indicating vehicle to terminate for AFSM
DIRSTN	-	I*4	Directional station ID
DSTN	-	I*4	Destination station ID
EVNT	-	I*4	Next event for vehicle
FIRST	-	I*4	First entity in boarding list
FOUND	-	L*1	Flag indicating search object found
LSTTP	-	I*4	Last trip in boarding queue
MNEED	-	I*4	Maximum vehicle need on route
NBERTH	-	I*4	Number of berths to move in ripple berth advancement
NEED	-	I*4	Number of vehicles needed in a route
NEXT	-	I*4	Next entity in boarding list
OSTN	-	I*2	Origin station ID
R	-	I*4	Route ID - loop counter
RANK	KMS	I*2	Nominal travel time ranking vector
RNEED	-	I*4	Route having maximum vehicle need
RTE	-	I*4	Route ID
RTE2	-	I*4	Route ID of push vehicle

SL	-	I*4	Current station link ID
SLD	-	I*4	Station link dock ID
SR	-	I*2	Rank of station in nominal travel time vector
STBOV2	-	R*4	Maximum deboard/board time overlap
STN	-	I*4	Current Station ID
TIM	-	I*4	Event scheduling time C.U.
TIME	-	R*4	Current simulation clock in seconds
TIM3	-	I*4	Vehicle deboard time
TMAX	-	I*4	Maximum board time
TPID	-	I*4	Trip ID in boarding queue
TPNXT	-	I*4	Next trip ID in boarding queue
TKNID	-	I*4	Transaction ID
ULT	-	L*1	Ultimate destination indicator
V	-	I*4	Vehicle ID
VEH	-	I*4	Vehicle ID
VEHL	-	I*4	Lead vehicle ID
VEH1	-	I*4	Next vehicle in train
VEH2	-	I*4	Push vehicle ID

6.2.87.4 Description

ESMDLB is invoked by the Station Model (ESLMDL) to perform the tasks associated with determining the specific activity required for a vehicle in performing a specified link event. Once this activity has been determined, the time associated with the event completion or retry is computed, and the vehicle is scheduled for event completion. This completion can correspond to a zero time activity in the case of deboard/board event bypass for vehicles traversing an on-line station which is not a scheduled stop, as determined upon initial entry to the routine.

However, if the vehicle is degraded, deboard processing is not bypassed since all onboard trips must be removed from the vehicle and terminated at the station so the vehicle can be removed from the active fleet. Boarding event bypass also occurs for vehicles scheduled to be removed from fleet as the result of a Scheduled Service Active Fleet Size Management request, or as a failure response.

The specific processing performed for non-bypassed events includes:

Headway Zone Traversal

The headway event on a station link serves to maintain the proper spacing interval between consecutively entering vehicles onto the link. This headway traversal event is used to model safe stopping distance maintenance between vehicles following one another on a given station link.

Vehicles about to enter a headway zone, block the headway zone for further vehicle entry until the required headway distance has been traversed by the entering vehicle. The travel time through the headway zone is given by a function of the form $ax+b$, where 'x' is train length, (an individual vehicle has assumed train length of one), and 'a' and 'b' are user specified constants. The time for headway zone traversal is adjusted to reflect degraded vehicle operations within the station by applying the degradation or penalty factors associated with the particular station link and vehicle. The computed headway zone travel time is recorded for each vehicle such that any subsequent travel event time on the station link can be adjusted to account for the time already travelled in the headway zone.

Travel Zone Traversal

Prior to scheduling a vehicle for station link traversal, the time required for the vehicle to complete travel of the main body of the link is computed. This time is computed as the difference between the time of headway zone traversal already completed for the vehicle and the specified time associated with traversing the travel portion of the station link. This travel time is then adjusted to account for station link degradation by applying a penalty factor which reflects an increase in travel time associated with the link degradation failure.

The initial travel time selected for traversing a particular station link differs depending on whether the station is designated as offline and whether vehicle station bypass is required. In the offline station case, link travel is selected as the assigned link traversal time based upon a constant station traversal speed. Link traversal time is assigned in the same manner for online stations where the station is an assigned

stop for the vehicle. However, if the vehicle is scheduled for station bypass and forward travel on the station link is unimpeded, (vehicle is first on the link or is following a vehicle traversing the link at bypass velocity) the travel time on the link is adjusted to reflect an assigned station bypass velocity.

Deboard Processing

Deboard processing is performed to determine the time required for passenger deboarding and to schedule the vehicle for deboard event completion. This involves performing the following functions.

1. **Ripple Berth Advancement** - For timeout/group demand responsive service with ripple berth advancement, deboard processing can not begin until the vehicle reaches the most downstream unoccupied unload berth. If the vehicle has one or more empty unload berths in front of it, it is scheduled to advance to the most downstream of these and will then re-enter Station Model Control to do the deboard.
2. Prior to the deboard processing, scheduled service vehicles are tested to determine if they must deboard all passengers to enable an active fleet size change. If a train has been marked to terminate, it will deboard all of its passengers when it reaches the station stop closest to its maintenance barn, unless the vehicles are needed by another route, in which case it deboards all passengers at the first station stop encountered. If the train has not been marked to terminate, then it will deboard all passengers at the closest station stop to the barn only if it is still traveling at the wrong train consist.
3. **Static Detrainment** - Automatic vehicle detrainment is always performed in demand responsive service prior to deboarding. Entrained vehicle groups in scheduled service remain intact during the deboard process since the train size assigned to a specific route always remains fixed until fleet redefinition is performed in response to an active fleet size management request. Detrainment processing, if required, is invoked whenever the lead vehicle of a train enters the deboard process. Each following vehicle of the entrained group is detrained and scheduled for individual deboard processing as defined below.
4. **Deboard List Creation** - As each vehicle or entrained vehicle group enters the deboard process, a list of trips deboarding at the current station is created by invoking ESDBL.
5. **Deboard Event Scheduling-Demand Responsive** - The deboarding time for each vehicle in the train is returned by routine ESDBL. The deboard event time is the maximum deboarding time used by any single vehicle of the train. For each vehicle in the train, the deboard time used is subtracted from the total time bounds yielding the minimum and maximum time for board:

$STBDMX (VEH) = SMXDBT - TIM \quad (\geq 0)$
 $STBDMN (VEH) = SMNDBT - TIM \quad (\geq 0)$
 $SPBDMX (VEH) = STBDMX (VEH)/STBA \text{ or } VCAP \text{ if } STBA = 0$

6. Deboard Event Scheduling-Scheduled Service - A loop is performed through all vehicles of a train and the time of deboard is subtracted from the time limits for each vehicle, STBDMX (VEH) and STBDMN (VEH), as was done for demand responsive vehicles. The maximum time for deboard (=TIM) is used to determine the deboard event time. The longest time available to board is the maximum value of STBDMX (VEH). The shortest time available to board is calculated by subtracting TIM from SMXDBT. The maximum overlap value (STBOV2) is determined by taking the difference between the longest and shortest times available for boarding. In addition, the overlap value for each vehicle (STBDOV (VEH)) is calculated by subtracting the shortest time available (SMXDBT - TIM) from the time available to board each vehicle. STBDMX (VEH) and STBDMN (VEH) are set to STBDOV (VEH), if they are less, since vehicles keep their doors open at least as long as the longest deboard time. STBOV2 is copied to STBOVX (VEH) for each vehicle in order to pass the value to the board event for these vehicles, and the value of SPBDMX (VEH) is calculated in the same manner as demand responsive service. The times calculated during deboard are used in determining the board time available for each vehicle on the train.

Board Processing

The following processing is performed for each vehicle or entrained group requiring boarding event initiation.

1. **Ripple Berth Advancement** - For timeout/group demand responsive service with ripple berth advancement, board processing can not begin until the vehicle reaches the load berth. If the vehicle has one or more empty berths in front of it, it is scheduled to advance to the most downstream of these. If the vehicle can not move up and has not reached the load berth it must queue until preceding vehicles have moved.
2. **Boarding Trip Identification** - The boarding process begins with creating a list of waiting trips which can board the vehicle or entrained group by invoking routine ESDL. For timeout/group demand responsive service, the call to ESDL is preceded by a search to assign the oldest ready and compatible vehicle request to the vehicle if it does not already have a request assigned. If there are no compatible ready requests, the vehicle simply remains at the load berth until an assignment is generated.
3. **Boarding Time Determination and Event Scheduling** - The boarding time for the train is computed by a single call to routine ESDL and is used to schedule event completion. However, in scheduled service operation, vehicles may be required to spend more time in the boarding event than required solely for passenger boarding. The situation arises when the vehicle immediately preceding the vehicle or train in the boarding event requires a longer time for event completion. In this case the time for scheduling event completion is extended to scheduled completion time of the preceding vehicle. This time represents the earliest time of completion possible due to serialization in the berthing area. This delay in boarding time completion permits later arriving trips to board the vehicle during the extended boarding interval. Scheduled Service vehicles undergoing board event completion are identified by inserting each Scheduled Service vehicle in the process vehicle boarding list assigned to the station as boarding is initiated.

Storage Event Processing

Vehicles arriving in storage are detained if necessary and marked with a status of queued in storage. If the lead vehicle is degraded, the lead vehicle is removed from service and all relevant statistics are updated. For scheduled service, an attempt is made to launch a replacement train by invoking routine EANTRN and, if applicable, a second call to EANTRN is made to re-launch the push train on its original route. For timeout/group demand responsive service the push vehicle is re-launched and sent to the closest station.

For the case of the lead vehicle not being degraded, if demand responsive service, the vehicle search process is restarted for passengers waiting in boarding queues. The stations are ranked by increasing nominal travel times from the storage station and then tested in the ranked order by invoking routine ESTABQ for all

previously unserviceable trips. For scheduled service, routine EANTRN is invoked to form and launch trains on route during active fleet size management and then routine EASTOR is invoked to assign any excess vehicles to other routes or remove them from service.

Launch Processing

Launch event processing is performed prior to the dispatching of any vehicle from a network station. The launch event must be specified to occur within each station such that all existing vehicles from the docking area and station storage are subject to launch event processing. This processing determines the earliest possible time at which the vehicle can be dispatched based on requirements for static (in station) vehicle entrainment, schedule adherence, and guideway merge scheduling.

Upon entry to the launch event, static entrainment processing, if required, is performed by invoking ESNTRN to group waiting vehicles (precluded from independent launch event initiation) for simultaneous launch processing and subsequent station departure. As the result of attempting entrainment, the launch event can be rescheduled to accommodate the availability (predicted launch event initiation) of a vehicle in process in the event immediately preceding launch for possible entrainment. If the vehicle is not rescheduled for an entrainment retry and scheduled operation is being simulated, the time delay associated with schedule adherence is determined by invoking ESSDLY. If a schedule delay is required, the vehicle is scheduled on the FEL for completion of the required delay and launch event processing is terminated. Otherwise, path determination processing is performed by invoking ESPATH. Depending upon the type of path selection required, either a primary or a primary and alternate path for a vehicle is determined. Final path assignment to the vehicle depends upon individual path cost and launch delay associated with traversing the individual paths. Once path determination has been completed, ESLDLY is invoked to select a path for the vehicle and compute the delay time associated with merge scheduling on that path. The vehicle is then scheduled for completion of the required launch delay time.

Vehicle Retardation

No initiation processing is performed for this pseudo event. It is included for recognition during event initiation processing to satisfy event process sequencing as implemented by the Station Model, ESLMDL. The actual initiation of this event is performed during event completion for the previous activity associated with a vehicle attempting to exit an off-line station.

On-line Station Bypass

The vehicle is scheduled for zero time event completion corresponding to the bypass of board/deboard event in an on-line station not scheduled as the next stop for a vehicle.

Upon completion of event initiation for the vehicle, the scheduled exit time from the station is updated to reflect the time of event completion if the vehicle is completing the last event in the station. This exit time is used during local exit merge resolution and quasi-synchronous control processing for determining station exit merge arrival time.

6.2.87.5 PDL

See Appendix A.

6.2.87.6 Decision Tables and Algorithms

a. Headway zone Travel Time

The headway zone travel time is given by:

$$H_T = H_A \times T_V + H_B$$

$$H_T' = (H_T \times L_p) / V_p$$

where:

H_T = unadjusted headway traversal time

H_T' = adjusted headway time

A = fractional time/vehicle in C.U.

T_V = Train length

H_B = constant headway time in C.U.

L_p = station link degradation factor ≤ 1

V_p = degraded vehicle penalty factor as fraction of normal operating capacity ≤ 1

b. Link Traversal Time

The time to complete link traversal is given by:

$$T_T = (L_T - H_T) \times L_p / V_p$$

where:

L_T = total station link traversal time and H_T , L_p , V_p as defined above

6.2.88 ESMDLN - Station Link Next Event Determination

6.2.88.1 Identification

- o ESMDLN - Determine Next Event for a Vehicle on a Particular Station Link
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.88.2 Argument Dictionary

None.

6.2.88.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DIRPLA	-	I*4	Directional platform ID
DIRSTN	-	I*4	Directional station ID
EVFND	-	L*1	Flag indicating event found
EXDSTN	-	I*4	Expected directional station ID
SL	-	I*4	Station link ID
STN	-	I*4	Station ID
VEH	-	I*4	Vehicle ID
VQTEMP	-	I*4	Temporary vehicle queue reason
XTN	-	I*4	Transaction ID

6.2.88.4 Description

ESMDLN is invoked by the Station Model (ESLMDL) to determine the next processing event for a vehicle traversing a station link. The next event for the vehicle is assigned based upon current processing status. If an event retry is not being attempted next event determination is made by selecting the first or a subsequent event from the event list associated with the station link. If the vehicle is initially entering the link, the first event is assigned to the vehicle and occupancy of the link is updated to reflect vehicle or train entry. If the next event for the vehicle is board or deboard, the berthing occupancy of the assigned station link is also updated to ensure correct sequencing of the vehicle in the first available

berthing position. For timeout/group demand responsive service, inventory control parameters are updated as a vehicle enters a dock link, provided the vehicle is not degraded. If the vehicle has entered a dock which is a side channel, then inventory management is scheduled for both directions of the physical station. Vehicles attempting a launch retry are assigned the launch event as the next to be performed providing event availability exists. If the vehicle is not first on the link, it must wait for a prior vehicle to complete its launch processing before its launch try can be attempted. This condition results in the vehicle being assigned awaiting launch status which is used by the Station Model to bypass immediate event initiation.

If the next event is boarding, ESMDLN checks if the vehicle is marked to wait in station as a failure response. If so, the vehicle is queued at the dock unless the dock is part of a tow path at an on-line station in which case the vehicle proceeds with the board event in order to clear the tow path.

If selection of the next event indicates the list of assigned events has been exhausted, a link completion indicator is set for use by the model architecture. In the case of output ramp completion this indication may be bypassed if quasi-synchronous control is required. Prior to indicating link completion, EGQMRG is invoked to determine exit merge availability for the vehicle. If the vehicle can merge immediately, link completion is indicated. Otherwise, control is returned to the Station Model with no further processing required for the vehicle.

6.2.88.5 PDL

See Appendix A.

6.2.88.6 Decision Tables and Algorithms

None.

6.2.89 ESMDLY - Merge Scheduling

6.2.89.1 Identification

- o ESMDLY - Perform Merge Scheduling for Deterministic and Quasi-Deterministic Dispatch
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.89.2 Argument Dictionary

None.

6.2.89.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
LINK	-	I*4	Next guideway link on path
LINKT	-	I*4	Accumulated path travel time
LINKX	-	I*4	First guideway link on alternate path
NUM	-	I*2	Column number in merge scheduling table corresponding to current clock time
PTR	-	I*4	Computed column number corresponding to expected merge arrival time
RETRY	-	L*4	Logic variable for indicating merge scheduling retry required
STATN	-	L*4	Logic variable for destination station encountered
TENTRY	-	I*4	Predicted merge occupancy

6.2.89.4 Description

ESMDLY is invoked during launch delay determination (ESLDLY) to determine merge availability on the possible traversal paths for the vehicle to its designated next destination. This processing is performed for both deterministic and quasi-deterministic dispatch to preplan downstream merge conflict resolution.

The preplanning process is based upon use of a merge reservation table which contains the predicted occupancy of each network merge over a finite scheduling interval. This interval is given by the maximum traversal time between any given origin/destination pair in the network. The scheduling interval encompassed by the table is segmented into discrete timing intervals which define a time window for merge reservation. In deterministic dispatch, these discrete timing intervals are fixed and correspond to the fixed headway or block time employed in guideway link traversal in order that individual slot reservation can be made. In quasi-deterministic dispatch, the discrete interval can be varied based upon user option. The scheduling window definition in combination with a threshold limit or merge occupancy define the accuracy which can be achieved in performing reservations and ensuring congestion-free vehicle traversal. Deterministic dispatch requires that the occupancy threshold for any merge be limited to a value of one which corresponds to the capacity of the merge during any fixed headway or slot interval. The threshold limit for quasi-deterministic dispatch can be user-specified to reflect the merge occupancy that can be accommodated for the scheduling window.

The merge scheduling table is used in the determination process as a sequence of rotating lists with a time base definition corresponding to the current launch try time. The scheduling interval is established upon entry based upon the time base required for the current launch try, as shown in Figure 6-8. Since the scheduling table is established, the merges to be traversed on both the primary and alternate path are defined along with the estimated time of arrival at each merge. This involves tracing each path until the destination station for the vehicle is encountered. The primary path trace is accomplished by selecting consecutive entries from the minimum path table until a zero entry indicating station occurrence is encountered. As each entry is selected, a determination is made as to whether a merge exists at exit of the selected link. If a merge exists, its ID is recorded along with the accumulated guideway link travel to the merge. Alternate path tracing is accomplished in a similar manner until the first link on the alternate path possibility for the vehicle is encountered. At this point path tracing continues by selecting consecutive entries from the alternate path list until either the vehicle destination is encountered or the end of the alternate path definition is encountered. If the latter condition occurs, path tracing is resumed with the minimum path table. The total accumulated traversal time on the vehicles alternate path is saved for possible use in arrival list recording by ESVALS.

Upon completion of the merge identification process, merge reservation is attempted for the primary and alternate paths of the vehicle. Beginning with the time interval associated with traversal to station exit, the reservation table is checked for each merge to determine if an open reservation exists at the anticipated merge passage time of the vehicle. In the case of deterministic dispatch, the number of consecutive

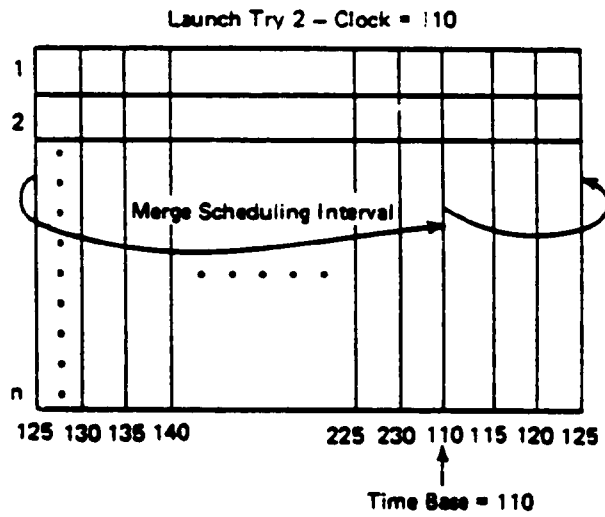
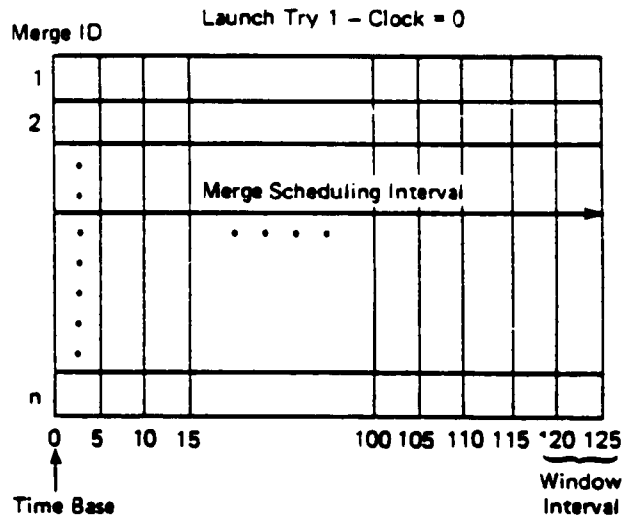


FIGURE 6-8. RESERVATION TABLE TIME BASE UPDATING

intervals required to accommodate train size must be available for successful merge reservation. Quasi-deterministic dispatch only requires that the train be accommodated within the interval associated with predicted merge arrival. If any merge on the path being processed is found to be at its threshold limit, the vehicle must undergo a scheduling delay of one window interval and merge reservation is attempted beginning with the next window interval in the table. This process is repeated until successful merge reservation is accomplished or the scheduling interval encompassed by the table is exceeded. If merge reservations cannot be made for a particular path, the vehicle path indicator is zeroed to eliminate it as a possibility for the vehicle. The absence of the path indicator is recognized by ESLDLY as an unsuccessful merge scheduling attempt requiring vehicle retry at the start of the next scheduling interval. An example of the merge reservation process is shown in Figure 6-9.

6.2.89.5 PDL

See Appendix A.

6.2.89.6 Decision Tables and Algorithms.

Alternate Path Definition - See EGALT.

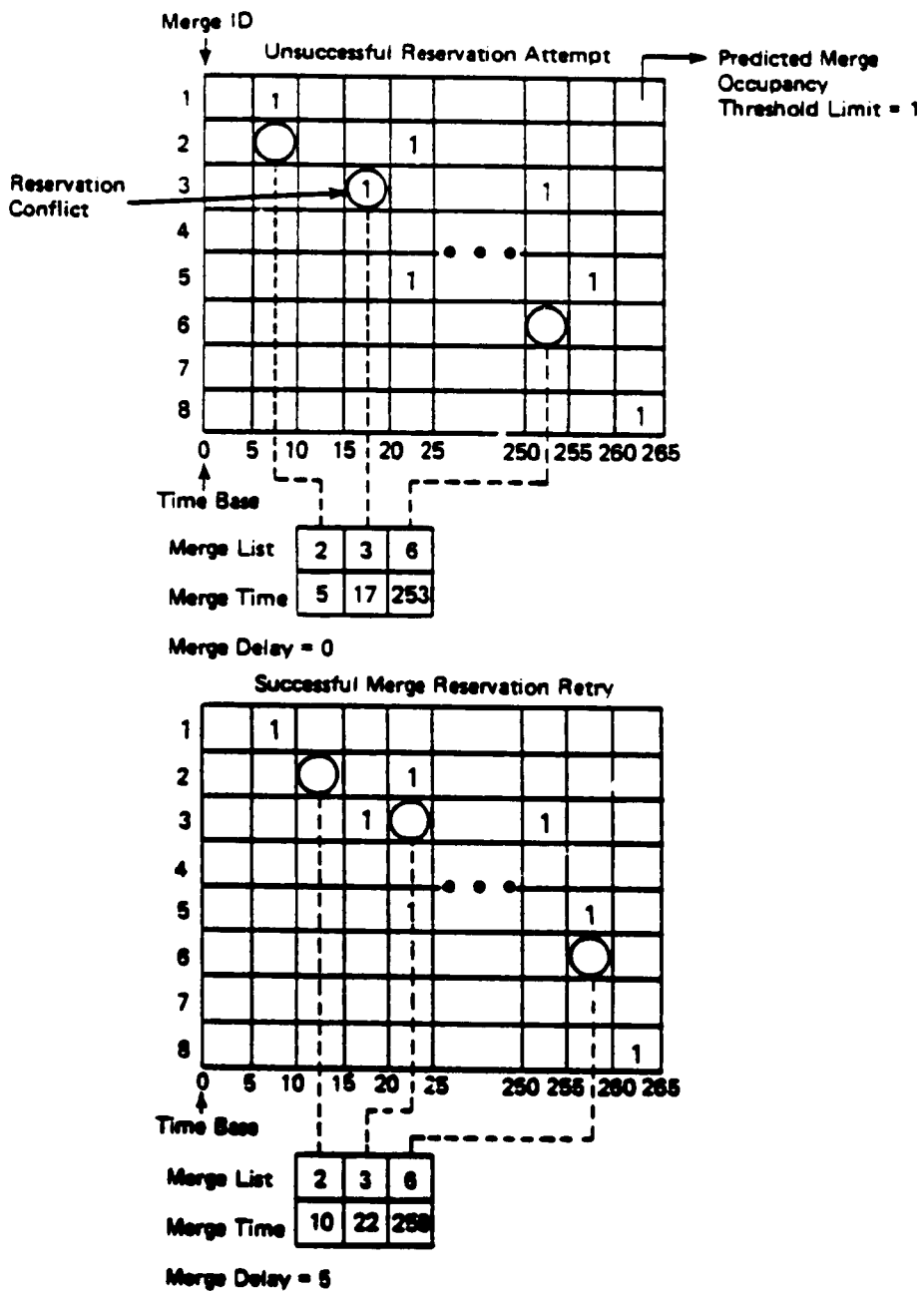


FIGURE 6-9. MERGE RESERVATION PROCESS

6.2.90 ESNEXT - Station Model Next Entity Determination

6.2.90.1 Identification

- o ESNEXT - Determine Next Entity for a Vehicle Currently in a Network Station
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.90.2 Argument Dictionary

None.

6.2.90.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DIRSTN	-	I*4	Directional station ID
DSTN	-	I*4	destination station ID
EVNT	-	I*4	Event ID
FOUND	-	L*1	Search object found indicator
IND	-	L*1	Flag for indicating type of occupancy order required (T=berthing, F=total)
LNK	-	I*4	Link ID
MIN	-	I*4	Minimum station link occupancy
NUM	-	I*4	Number of possible next links
NUMLOW	-	I*4	Number of platforms with low inventory
NUMPLT	-	I*4	Number of eligible platforms
PLELIG	4	L*1	Flag indicating platform eligible
PLLOWI	4	L*1	Flag indicating platform has low inventory
PLTFRM	-	I*4	Platform ID
SL	-	I*4	Current station link ID
SLN	-	I*4	Next station link ID
SLELIG	KMSL	L*1	Flag indicating station link eligible
STATUS	2	I*2	Empty/occupied status for local merge resolution

STN	-	I*4	Current station ID
TEMP	-	I*4	Temporary variable for switching variables
TRN	-	I*4	Vehicle request transaction ID being tested
TRNID	-	I*4	Vehicle request transaction ID using eligible platform
TYPE	-	I*4	Station link type
VEH	-	I*4	Vehicle ID
VEH1	-	I*4	Next vehicle in train
VLAST	-	I*4	Vehicle ID of first vehicle in train

6.2.90.4 Description

ESNEXT is invoked by Station Link Modeling Control (EASCTL) to determine the next entity for the vehicle currently within a network station. The next entity for the vehicle is selected based upon station configuration, and vehicle processing requirements and status. An interface for implementing user supplied station link selection algorithms is provided. Upon entry, a check is performed to ensure that the vehicle can leave its current station link. If the vehicle is not positioned at the exit of a link, its status is set as awaiting a prior vehicle's event completion or movement on the current link and next entity determination processing is terminated. Similarly, if the exit of the current station link is failed, further movement of the vehicle is precluded and processing is terminated. Upon return of control to the architecture, the vehicle is queued on its current station link.

If the vehicle can exit its current station link, next entity determination processing is performed. If the vehicle is currently located on the station output ramp, the guideway link immediately downstream of station exit is assigned as the next entity for the vehicle. If local exit merge priority resolution is required, and a competing vehicle exists on the guideway, the 2x2 local merge is examined to determine if the vehicle is next to merge. If this determination results in a competing vehicle being given priority, the next entity for the vehicle is reset to zero and immediate queuing occurs upon returns to the architecture. Development of the list of ordered downstream links for vehicles requiring further station processing is performed automatically or via diverge functions coded and implemented by the user. Before, the list of possibilities is developed, entrained vehicles at exit of the input ramp are detrained at this point, if required. Detrainment occurs if vehicles in the train are destined for both docking and storage of the station.

In automatic selection the list of downstream possibilities is developed based upon current location of the vehicle, station link connectivity, and vehicle processing requirements as summarized in Table 6-1. Vehicles designated as a member of a platoon awaiting dock entry, are assigned only one next entity possibility corresponding to the dock to which the platoon is to move. This dock is selected prior to the next entity process as the result of all berths being vacated on an immediately upstream dock link as performed by ESLEAV. Once developed, the list of possibilities is ordered by current occupancy and priority.

An exception to the ordering of the list of possibilities by current occupancy occurs for dock links. Dock links are ordered by pseudo-occupancy as developed by the berthing algorithms. However, for the timeout/group demand responsive case, a complex dock selection algorithm is used to model stations which serve two guideway directions and contain both turnaround and side channel dock links. This algorithm results in the choice of a single dock link as the next station entity so no further ordering of possibilities is required. The algorithm is defined in Section 6.2.90.6.

In the case of non-automatic selection, a diverge function ID can be assigned to the exit of any station link which serves as the entry to a station model diverge. These ID's cause the automatic selection procedure described above to be bypassed and either an alternate selection procedure or a user coded module for developing the list of downstream link possibilities being invoked via EUDIVF. The alternate selection procedures supported within the next entity determination process are provided via two default diverge functions invoked via a call to ESDIVF. These functions, if selected, will develop a list of downstream possibilities based strictly upon station connectivity without regard to vehicle processing requirements and order the list by link or berthing occupancy (diverge function 1 and 2, respectively). Currently in the DESM, these two diverge functions are not allowed to be selected by the user. Any other diverge ID specification is considered to be user implemented and executable via EUDIVF.

Once the list of possible downstream links has been developed, control is returned to the architecture for link availability testing.

TABLE 6-1. AUTOMATIC STATION LINK SELECTION

<u>Vehicle Location</u>	<u>Processing Required (Destination)</u>	<u>Next Link</u>
Input Ramp	Storage	Input to Store* Input queue or dock
	Dock	Input queue or dock
Input Queue	Dock	Dock
Dock	Storage	Dock to store
	Guideway	Output queue or Output ramp
	Dock	Dock
Output Queue	Guideway	Output Ramp
Output Ramp	Guideway	Guideway Link at station exit
Storage	Dock	Store to dock
	Guideway	Output queue or Output ramp
Store to Input	Dock	Input queue or dock
Store to Output	Guideway	Output queue or Output ramp
Dock	Storage	Dock to store
	Guideway	Output queue or Output ramp
Dock to Store	Storage	Storage
Input to Store	Storage	Storage

*Vehicles destined for storage do not queue if entry is precluded. Vehicle is routed to dock instead.

6.2.90.5 PDL

See Appendix A.

6.2.90.6 Decision Tables and Algorithms

Timeout/Group Demand Responsive Dock Link Selection

First, docks serving the direction the vehicle is entering from and also having the rear berth unoccupied are marked as eligible for selection. If there is an outstanding side channel request and a side channel is eligible, that dock link is selected. Otherwise, eligible turnaround channels are searched. A count of the number of directional platforms having eligible channels and the number of directional platforms having channels with low inventory (less than 2 vehicle occupancy) is accumulated. If no directional platforms have eligible turnaround channels, then an eligible side channel is selected and a side channel request is issued for the opposite side channel. If two directional platforms have eligible turnaround channels, then low inventory conditions are checked. If neither or both platforms have low inventory channels then the platform having the oldest ready vehicle request is selected. If neither has a ready vehicle request pending, then the channel with the least inventory is selected. If only one platform has a low inventory channel, then that platform is selected. At this point, if a platform but not a channel has been selected, then the turnaround channel of a B platform is selected, while the turnaround channel with the lowest inventory of an A platform is selected. If no eligible channels are found, then the vehicle queues on its current link until prompted by a vehicle vacating the rear berth of a dock link.

6.2.91 ESNSTN - Next Station Determination

6.2.91.1 Identification

- o ESNSTN - Next Station Determination
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.91.2 Argument Dictionary

None.

6.2.91.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
FND	-	L*1	Flag indicating search object found
MDIST	-	I*4	Minimum nominal travel time
NSTN	-	I*4	Station ID having minimum nominal travel time
PTR	-	I*4	Pointer to route list table
RTE	-	I*4	Route ID
STM	-	I*4	Current station ID
STNL	-	I*4	Station link ID
STNI	-	I*4	Next station on path to ultimate destination
TRN	-	I*4	Temporary transaction ID
TRNID	-	I*4	Transaction ID of vehicle request
TRPID	-	I*4	Trip ID in vehicle

6.2.91.4 Description

ESNSTN is invoked by ESMDLA after the completion of the board event to determine the next station for the vehicle or entrained group based upon service policy in effect as follows:

1. Demand Responsive -- If the vehicle is empty, empty vehicle assignment (ESEVA) is invoked to determine the next destination for the vehicle based on a user-specified list of priorities. The next destination for occupied single party vehicles is assigned as the destination station for the onboard trip. In multiparty service, the vehicle's next destination is determined by comparing the station sequence along route to its ultimate destination, to each destination for onboard trips. The first station which matches a trip destination is assigned as the next stop for the vehicle.
2. Scheduled Service -- The next station in the route list for the vehicle is assigned as its next destination. In demand stop operation, this station can be bypassed when it is reached if no passenger demand for service in the station exists.

3. Timeout/Group Demand Responsive Service -- The next station is determined by the destination of the vehicle request assigned to the vehicle, either by passenger or by an inventory request. At this time, the previous station of the vehicle is set to the current station and the vehicle request transaction is freed if a corresponding maximum passenger wait transaction has been freed. Otherwise, the transaction is marked complete and will be freed when the maximum passenger wait transaction is freed.

In the case of a degraded vehicle, ordinary next station processing is bypassed. Instead, for any demand responsive service, the vehicle is routed to the closest maintenance barn (could be the current station if dock to storage link exists). For scheduled service, the degraded vehicle is sent to the designated maintenance barn for its route.

6.2.91.5 PDL

See Appendix A.

6.2.91.6 Decision Tables and Algorithms

None.

6.2.92 ESNTRN - Static Vehicle Entrainment

6.2.92.1 Identification

- o ESNTRN - Entrain Vehicles in Station Prior to Launch
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.92.2 Argument Dictionary

None.

6.2.92.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DONE	-	L*1	Logical variable to indicate entrainment complete
FOLL	-	L*1	Logical variable to indicate existence of a following vehicle
SL	-	I*4	Current station link
STW	-	I*4	Current station
VEH1	-	I*4	Last vehicle of current train
VEHN	-	I*4	Next vehicle to entrain

6.2.92.4 Description

In demand responsive service, static vehicle entrainment can be performed to group vehicles for simultaneous launch processing and station departure provided that neither the lead nor following vehicles is degraded. If static entrainment is required vehicles queued awaiting start of the launch event are entrained to the current vehicle for simultaneous launch. Entrainment of these vehicles continues until an entrained group of maximum size is created, all waiting vehicles are processed or destination compatibility is violated. If no incompatibility is found during the entrainment of waiting vehicles and the vehicle undergoing launch is not a member of a maximum size train, the launch event can be delayed (scheduled for retry) to accommodate following vehicles in

process in the event immediately preceding launch on the current station link for entrainment.

Following vehicles are considered providing a launch retry is not currently being performed on the launching vehicles. Vehicle launch is delayed if destination compatibility requirements, as previously stated, are satisfied between the following and current vehicle, and arrival of the following vehicle at launch can be predicted within a fixed delta time of the current clock. The launching vehicle is scheduled for launch retry at a time corresponding to the current clock plus the fixed delta time used in determining the following vehicle's launch availability.

If the following vehicle arrives for launch processing within the retry interval, it will be entrained upon reentry of the delayed vehicle to the launch event.

6.2.92.5 PDL

See Appendix A.

6.2.92.6 Decision Tables and Algorithms

None.

6.2.93 ESPATH - Station Model Path Selection

6.2.93.1 Identification

- o ESPATH - DESM Station Model Path Selection
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.93.2. Argument Dictionary

None.

6.2.93.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
INFINY	-	I*4	Large initial variable used for minimum cost comparison
LNK	-	I*4	Link ID used in tracing vehicle paths
MCOST	-	I*4	Minimum alternate path cost
PCOST	-	R*4	Temporary variable for path cost
PTR	-	I*4	Temporary variable used as an index into the alternate path table
PTRSAV	-	I*4	Pointers to starting entry point for an alternate path
VEH	-	I*4	Temporary variable for vehicle ID

6.2.93.4 Description

ESPATH provides the ability to select an a priori route for a vehicle being dispatched from a network station. The path selection process involves the setting of a primary route for the vehicle (as provided in the current minimum path table) and, if required, concatenating an alternate path which reflects the "lowest cost" alternative for routing of the vehicle to its next destination.

Upon entry, the current primary path is assigned to the single vehicle or the first vehicle of an entrained group. If a priori algorithmic path selection is enabled a determination is made as to whether an alternate path exists for the vehicle from the first diverge point on the minimum path. This involves tracing the path to the vehicle's next station in the minimum path or successor link table (PLSLT) until a common diverge value, if any, is encountered (a value 1000). A common diverge indicates the existence of an alternate path definition of one or more alternates. The link number leading to the diverge is saved for use in alternate path evaluation. ESPATH then performs the cost computation and comparison for the primary and alternate paths available from the current station to the vehicle's destination.

Primary path evaluation involves tracing the minimum path and accumulating the cost for using the route based upon a user selected criteria as follows:

- a. link nominal travel time,
- b. link length,
- c. utilization or
- d. a weighted combination of (a) and (c)

If a failed link is encountered, the path is excluded from further consideration by adding an artificially high failure factor to the cost of the route. The primary path cost up to that point of common diverge is saved for use in alternate path evaluation. At the completion of the primary path cost evaluation, the cost of the path is recorded for use during final path selection performed as part of the launch determination process.

If an alternate path(s) exists for the vehicle, alternate path evaluation from the starting point of the alternate route definition is performed. This involves tracing each defined alternate and accumulating its cost as described above, until the last entry for the particular path is reached or the destination is encountered. Paths with failed links are excluded in the same manner as described for the primary path. If the end of the alternate path is reached prior to finding the vehicle's destination, cost accumulation continues with the primary path from the most recently processed link to the destination.

Upon the completion of cost accumulation for each defined alternate, the lowest computed cost alternate path is selected for possible use by the vehicle. The cost of the selected alternate is saved after correcting it by the cost of the primary path up to the diverge point.

Preliminary path selection, as described above, is repeated for each member of an entrained group. However, if a following member has the same destination as that of the lead, it then assumes the same alternate path assignment.

6.2.93.5 PDL

See Appendix A.

6.2.93.6 Decision Tables and Algorithms

See the corresponding sections for primary path and alternate route selection processing in the descriptions of EGPRMY and EGALT, respectively.

6.2.94 ESSDLY - Schedule Delay Determination

6.2.94.1 Identification

- o ESSDLY - Compute Schedule Service Launch Time
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.94.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TIME	-	I*4	(OUTPUT) Required schedule delay time

6.2.94.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DEV	-	I*4	Schedule deviation C.U.
ENTRY	-	I*4	Vehicle position in Scheduled Service route list
NLT	-	I*4	Next to launch time on route - current vehicle
RTE	-	I*4	Route ID assigned to vehicle
STN	-	I*4	Current station

6.2.94.4 Description

Routine ESSDLY is invoked during the launch time determination process to determine the earliest dispatch time for scheduled service vehicles. The desired launch time is determined by one of the algorithms listed below.

- o **Fixed Schedule** -- The desired launch time of each vehicle is determined by an absolute fixed time schedule which demands each scheduled launch time to be one route headway after the previous scheduled launch time.

- o Midway Spacing -- The desired launch time is based on the same fixed schedule as above; however, the desired launch time is halfway between the fixed schedule time and one route headway time after the time the last vehicle on the route actually launched from the station.
- o Fixed Interval -- The desired launch time is one route headway time after the previous vehicle on the route was actually launched from this station.
- o Midpoint Interval -- The desired launch time is halfway between the current time and one route headway after the previous vehicle on the route was actually launched.
- o Immediate Launch -- The desired launch time is the current time.

In all of the methods, if the desired launch time has passed the vehicle is behind schedule and eligible for immediate launch. If the desired launch time computation requires the vehicle to be delayed, the launch event for the vehicle is scheduled for retry at the end of the required delay time. Upon re-entry to the launch event, schedule delay processing is bypassed.

Prior to exiting the schedule delay determination process, the time of the last vehicle dispatched is updated to reflect the actual dispatch time of the current vehicle and the next scheduled (desired) dispatch time for the route is updated as required by the chosen algorithm. For fixed schedule dispatching, statistics are maintained concerning schedule adherence. For the other algorithms, statistics concerning interdispatch times are maintained. The vehicle route position pointer is updated to the next station.

6.2.94.5 PDL

See Appendix A.

6.2.94.6 Decision Tables and Algorithms

Table 6-2 summarizes the five route spacing algorithms selectable for scheduled service modeling.

TABLE 6-2. SCHEDULED SERVICE ROUTE SPACING ALGORITHMS

Route Spacing Algorithms Route Spacing Parameters	FIXED SCHEDULE	MIDPOINT SCHEDULE	FIXED INTERVAL	MIDPOINT INTERVAL	IMMEDIATE LAUNCH
Desired Launch Time	Set by PMSLV	Average of PMSLV and PLSLV + Route Headway	Set by PMSLV	Set by PMSLV	Immediate
Actual Launch Time	Immediate or PMSLV	Immediate or desired	Immediate or PMSLV	Immediate or midway between now and desired	Immediate
Reset Last Launch Time PLSLV	Actual Launch	Actual Launch	Actual Launch	Actual Launch	Actual Launch
Reset Next Launch Time PMSLV	Next Launch + Route Headway	Next Launch + Route Headway	Last Launch + Route Headway	Last Launch + Route Headway	Last Launch + Route Headway
Statistics Updated	Schedule Deviation	Interdispatch Time	Interdispatch Time	Interdispatch Time	Interdispatch Time

6.2.95 ESTABQ - Perform Trip/Vehicle Assignment at Trip Arrival

6.2.95.1 Identification

- o ESTABQ - Trip/Vehicle Assignment
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.95.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
FOUND	-	L*1	available vehicle found indicator
TRPID	-	I*4	arriving trip to be served
STN	-	I*2	station location of arriving trip

6.2.95.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ATIME	-	I*4	delta time between ETA of vehicle at station and the current clock time
HCCNT	-	I*4	count of handicapped passengers
MIN	-	I*4	minimum nominal travel time
RTE	-	I*4	temporary variable for route ID used in identifying the routes associated with a route group
S	-	I*2	station location of arriving trip
SEAT	-	I*4	number of passengers assuming seats
TRIP1	-	I*2	temporary variable used for identifying a current trip while processing a trip queue chain
TRNID	-	I*4	vehicle request transaction ID
TRP	-	I*4	temporary trip ID
TRTE	-	I*4	arriving trip route assignment
VEH	-	I*4	temporary variable for the vehicle ID being queried for availability while processing a membership queue

VARIABLE	DIM	TYPE	DESCRIPTION
VEH1	-	I*4	ID being queried for availability while processing each member of a train
XTN	-	I*4	event scheduling transaction ID
XTN2	-	I*4	event scheduling transaction ID

6.2.95.4 Description

ESTABQ attempts to perform trip/vehicle assignment upon the arrival of a trip, at a network station. In demand responsive service, processing is performed to ensure timely serving of the trip request. This involves the requesting or reservation of a vehicle according to a user-specified prioritized list of options. Each listed option is processed in the order specified until the list is exhausted or a possible vehicle is found as follows:

- a. A non-circuitous empty vehicle about to arrive at the station (is on the station's vehicle arrival list). If an arrival list exists at the station then each vehicle, not assigned to a circuitous route, is queried until an available vehicle, if any, is found. Since the arrival list is time ordered, the earliest arriving vehicle is the first vehicle processed. Even though the vehicle may be the first to arrive, it must have an arrival time within an acceptable user-specified time interval of the current clock. In demand responsive single party, the vehicle must either be empty or unreserved to be a possibility for servicing the trip. In addition, no prior waiting unreserved trips must be present in the station, otherwise the vehicle is reserved for the prior trip and another acceptable vehicle must be found.

This particular case can arise if a trip arrived before any available vehicles existed in the station's vehicle arrival list. In all cases, if a prior trip without a given reservation exists in the trip queue, it is given priority to reserve space on the available vehicle. `ESVRES` is invoked to establish the vehicle reservation. However, if no such prior trip is found, indication is made that an available vehicle has been found for the arriving trip.

In demand responsive multi-party service, a vehicle with reservations may still be considered for serving of the arriving trip. As described above, unserved previously arriving trips are given priority in the vehicle selection

process. ESTCHK is invoked for each unreserved trip to determine if the trip is compatible with the trips already on board or reserved on the vehicle. For each compatible trip ESVRES is called to make a reservation for the trip on the available vehicle. After each trip in the trip queue is scanned, ESTCHK is invoked to determine if the arriving trip can also be accommodated and is compatible with existing trips associated with the vehicle. If compatibility is found, indication is made that an available vehicle has been found.

- b. An upstream vehicle in the station, due to arrive at a docking area. The upstream station links to be searched for station the existence of a vehicle are specified by the user. Each vehicle identified as a member or entrained follower as on each appropriate station link is queried until an available vehicle, if any, is found. The upstream possibility must not be bypassing the station to be considered. Thus a vehicle passing through a non-scheduled on-line station cannot be considered for trip servicing. A vehicle about to stop at the station dock is checked for availability depending on the existence of reservations and the service policy in effect. A demand responsive single-party vehicle with a reservation is immediately excluded as an available vehicle. The remainder of the processing for handling prior trips in both single and multi-party service is identical to that described in (a) above.
- c. An empty vehicle in local storage. If vehicles exist in station storage, then each vehicle is queried until an available one, if any, is found. As an available vehicle is found, indication is made and a prompt is scheduled to ensure that the vehicle begins forward movement in the station from storage to the docking area.
- d. An empty vehicle in a regional storage center assigned to the station. If a regional center is assigned to the each vehicle in station storage of that regional center is queried until an available one, if any, is found. If an available vehicle is found, indication is made, the next stop of the vehicle is set to the arrival station and a prompt is scheduled to ensure that the vehicle begins movement toward that station.
- e. An empty vehicle circulating on the guideway (assigned to a circuitous route), about to arrive at the station.

If an arrival list exists at the station, then each vehicle is queried until an available circuitous empty vehicle, if any, is found. A vehicle with a reservation, operating in demand responsive single party service is excluded from consideration. The remainder of the processing required for handling prior trips, in both single and multi-party service is identical to that described in (a) above.

- f. An earliest arriving vehicle at the station (or the station's vehicle arrival list) regardless of its empty or full status. The processing for this option is identical to that of (a) above except that each vehicle identified on the station vehicle arrival list is considered for availability. No distinction is made between empty vehicles assigned to a circuitous route and vehicles in revenue service.

If no available vehicle possibility is found in examining all of the selection options, an outstanding service request is entered for the station. The existence of this unserved request may later cause the diversion, if permitted, of an available passing vehicle into the station for servicing of the trip. If an available vehicle is found, however, and if reservations are required then space for the arriving trip is reserved on the vehicle by invoking ESVRES.

- g. All station storages are searched to find a vehicle available to serve a new trip request. If more than one empty vehicle is available, the closest one to the trip request (in terms of nominal travel time) will be dispatched to service the request.

In scheduled service operation, an attempt is also made to minimize the time a trip must wait for service. As a first priority, an arriving trip is placed on a currently boarding vehicle, if possible. Thus, each vehicle currently boarding at the dock is queried until an available vehicle, if any, is found. A vehicle assigned the same route as the arriving trip is immediately considered compatible. If a route group is assigned to the trip, then each route associated with the particular group is scanned to determine if an available boarding vehicle associated with any route in the group exists. If route compatibility is found space availability is then checked along with maximum passenger boarding constraints, as specified by the user. Note that handicapped passengers are not allowed to be boarded on currently boarding vehicles.

If the trip can be accommodated on a currently boarding vehicle, it is removed from the station trip queue and added to the vehicle boarding list. In addition, the number of passengers associated with and currently boarding the vehicle is incremented by the number of passengers in the arriving trip.

In timeout/group demand responsive service, passenger travel requests are grouped by origin-destination pairs and a vehicle request is not issued until either a minimum travel group size is reached or a maximum passenger wait time has elapsed. As a trip arrives at the boarding queue, an attempt is made to match it to an existing travel group. If a ready request group exists which has not been assigned a vehicle and has fewer passengers than the vehicle capacity, the trip is simply added to the group. If the trip is handicapped, then there must be fewer handicapped passengers in the travel group than the vehicle handicapped capacity.

If no ready request groups can be joined, then pending request groups are checked in a similar fashion. If the trip can join a pending vehicle request group it is added to the group and a check is performed to determine if the minimum group size to request a vehicle has been reached. If so, routine ESVREQ is invoked to attempt to find a vehicle to service the travel request. If no pending request group can be joined, then the trip forms a new travel request group and also schedules an event after a time delay of the maximum passenger wait time to request a vehicle if the group hasn't become ready by that time.

As trips are assigned to vehicles, various statistics are collected by station, vehicle and/or route as appropriate for the service policy or option satisfied.

6.2.95.5 PDL

See Appendix A.

6.2.95.6 Decision Tables and Algorithms

None.

6.2.96 ESTCHK - Station Model Multi-Party Trip Compatibility Check

6.2.96.1 Identification

- o ESTCHK - Station Model Multi-Party Trip Compatibility Check
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.96.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TRIP	-	I*4	station trip being processed for compatibility
VEH	-	I*4	vehicle on which trip being checked wishes to board or make a reservation

6.2.96.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
NUMD	-	I*4	computed number of passengers to deboard vehicle at reserving station
NUMHCD	-	I*4	number of handicapped passengers deboarding
STN		I*4	station location of trip being checked for compatibility
STN1		I*4	temporary station ID variable used in tracing the route of a trip or the vehicle
TRP		I*4	temporary trip ID variable used in scanning chains of trips or trip queues

6.2.96.4 Description

ESTCHK provides trip compatibility checking upon trip arrival at a network station (for reservation purposes) or during the selection process for determining which trips can board a demand responsive

multi-party vehicle. A trip is considered compatible if vehicle space is or will be available for the trip and if the trip destination is compatible with the existing vehicle trips or those reserved or about to board the vehicle, whichever is appropriate. The single trip, desiring a reservation or boarding, along with the vehicle for which the trip is being checked are input to the routine.

Upon entry, the number of passengers scheduled to deboard at the reserving station is computed in order to determine if space exists for the arriving trip. The first criteria for compatibility is met if the sum of the number of passengers onboard the vehicle, the number of passengers reserved on the vehicle and the number of passengers associated with the trip minus the number deboarding is less than or equal to the capacity of the vehicle. This check is also performed for handicapped passengers. If vehicle space is available then the processing to determine trip destination compatibility is performed. The trips currently associated with the vehicle are used as a basis for determining compatibility. In the case of single stop multi-party demand responsive service, the trip destination must match the vehicle destination.

If the vehicle is on the guideway, trip compatibility with any existing onboard trip, not about to deboard in the station, along with any of the trips previously reserved in the station is performed. Compatibility exists if any of the following conditions are satisfied:

- a. destination of the trip matches that of one of the boarding trips
- b. destination of the trip is a bypassed station enroute to the ultimate destination of the vehicle determined by its onboard trips
- c. the most remote destination among the trips already on the vehicle is a bypassed station along the route that would be used in servicing the trip.

Similarly, trip compatibility can exist if any of the following conditions are satisfied in comparison against reserving trips.

- a. the destination of the trip matches that of one of the reserving trips,
- b. the trip destination is a bypassed station enroute to the destination of at least one of the reserving trips
- c. the destination of every reserving trip is on the route implied by the destination of the trip.

If the vehicle is in the process of boarding at the station destination compatibility is determined based upon onboard trips as well as with any of the trips currently boarding the vehicle. Compatibility checking with the onboard trips is identical as that described above for those trips already on the vehicle. Trip compatibility also exists if one of the conditions is satisfied in comparison with the boarding trips:

- a. destination of the trip matches that of one of the boarding trips
- b. destination of the trip is a bypassed station enroute to the ultimate destination of the vehicle determined by the onboard and boarding trips
- c. the most remote destination among the trips already on or boarding the vehicle is a bypassed station along the route implied by the destination of the trip. In this case, the ultimate destination of the vehicle is modified to the destination of the trip since route expansion is required. In addition, if no trips exist for any of the various categories, such as onboard, reserving or boarding trips, compatibility for that particular process is assumed.

6.2.96.5 PDL

See Appendix A.

6.2.96.6 Decision Tables and Algorithms

None.

6.2.97 EATEST - Station Link Entry Testing

6.2.97.1 Identification

- o EATEST - Station Link Entry Testing
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.97.2 Argument Dictionary

None.

6.2.97.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ARRINV	-	I*4	Number of unassigned vehicles in channels
DSTN	-	I*4	Destination station ID
FOUND	-	L*1	Indicates board or deboard event on station link
FOUND2	-	L*1	Indicates available unassigned vehicle found
LEN	-	I*4	Train length
LINK	-	I*4	Guideway link ID at exit of station
NBERTH	-	I*4	Number of berths in arrival channels
OSTN	-	I*2	Origin station ID
SL	-	I*4	Station link ID
SLELIG	KMSL	L*1	Link has vehicle eligible to be dispatched
SLHI	-	I*4	Highest numbered eligible station link
STN	-	I*4	Station ID
TRNID	-	I*4	Vehicle request transaction ID
VEH	-	I*4	Vehicle ID
XTN	-	I*4	Event scheduling transaction ID

6.2.97.4 Description

ESTEST is invoked by the architecture to determine whether downstream vehicle station movement can be performed. A list of possible downstream station links is examined to determine whether at least one link is available for vehicle entry. This list is created as part of next entity determination process based upon current vehicle location (ESNEXT for vehicles located in stations, EGNEXT for vehicles located on guideway).

Testing of successive downstream link possibilities is performed until either a satisfactory link has been found or the list is exhausted. The entry testing performed for each link includes:

- a. Availability -- Station configuration within the DESM can be varied from a basic station definition by specification of link availability on a station by station basis. Thus a link specified in the list of downstream possibilities may not be available for consideration within the current station of the vehicle and therefore, not eligible for vehicle entry.
- b. Entry Failure -- If entry to the station link being considered is blocked due to failure, vehicle entry is precluded.
- c. Headway -- Any prior vehicle entering the link being considered must complete headway traversal, if required, before further vehicle entry is allowed.
- d. Occupancy -- Sufficient capacity must exist on the link prior to allowing vehicle entry. Additionally, in the case of docking links, serial berth availability is checked as well as platooning requirements. Platooning at dock entry requires that the vehicle desiring dock entry be a member of the current platooned group assigned for entry to the particular docking area or is the first vehicle to enter the link.

In addition, if a vehicle is to enter the input ramp, total station capacity and the station input gate flag are tested.

Additional testing is performed for vehicles entering the input ramp from a guideway link. If vehicle bumping has been enabled, the station storage link is searched for a vehicle available to be bumped. If one is found, a prompt is issued to ensure the vehicle leaves storage and routine ESEVB is invoked to choose a destination. For timeout/group demand responsive service, the station overfull protection algorithm is executed. The number of berths in arrival channels is counted and the number of available but unassigned vehicles in those berths and upstream of those berths is calculated. Also, dock links with vehicles at the load berth but

unassigned are marked as eligible to dispatch a vehicle, if necessary. If adequate space does not exist at the docks (number of berths less arrival inventory less than user input adequate space parameter), it is necessary to dispatch a vehicle from an eligible dock link, if one exists. The oldest pending (but not ready) request that is compatible to an eligible channel is changed to an inventory request, thereby requesting a vehicle. If no pending and compatible vehicle request exists, then the vehicle in the side channel is dispatched if eligible. Otherwise, the vehicle in the highest numbered turnaround channel is dispatched. In either case, this vehicle is routed to the closest downstream station.

The identification of a satisfactory link is returned to the architecture in order that link exit processing and the station modeling can be invoked for the continuation of vehicle movement from the current link to its next selected link. If no satisfactory downstream link is found, the next entity for the vehicle, if located within the station, is set to zero, and upon return to the architecture the vehicle is queued at the exit of its current station link. If however, the vehicle is currently on the guideway and entry onto the input ramp of the station is not possible, then rerouting of the vehicle to an alternate station is attempted. This can only occur at the entrance to an off-line station when non-synchronous control is in effect. In this case, ESTEST returns the guideway link bypassing the station's next vehicle entity. This is recognized by the architecture which initiates processing for attempting vehicle traversal to an alternate station.

6.2.97.5 PDL

See Appendix A.

6.2.97.6 Decision Tables and Algorithms

The capacity algorithm used as part of the link availability testing is given by:

$$L_c \text{ must be } \geq L_o + T_L$$

where:

L_c = station link capacity

L_o = station link occupancy (i.e., number of vehicles on station link), if the link being checked is not a dock or berthing area, or otherwise

L_o = berthing occupancy, and

T_L = train length or number of vehicles in the train
(=1 for single vehicle).

Berthing occupancy is defined as the link capacity minus the number of available upstream berths if a board or deboard event is being processed.

6.2.98 ESTLOG - Completed Trip Log Processing

6.2.98.1 Identification

- o ESTLOG - DESM Station Model, Write Completed Trip Log Entry
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.98.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TRPID		I*4	ID of completed trip

6.2.98.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ACTT		R*4	temporary variable for trip actual travel time (sec)
ART		R*4	temporary variable for trip arrival time (sec)
CLK		R*4	trip termination time (sec)
DDT		R*4	trip dispatch time from origin station (sec)
DIST		R*4	travel distance (meters)
NOMT		R*4	nominal travel time (sec)
TXFT			total trip transfer time (sec)

6.2.98.4 Description

ESTLOG writes a record summarizing the trip on the Completed Trip Log, if completed trip logging is required. It is initiated at the time a trip is selected for deboarding.

6.2.98.6 Decision Tables and Algorithms

None.

6.2.98.5 PDL

See Appendix A.

6.2.99 ESVALS - Vehicle Arrival List Recording

6.2.99.1 Identification

- o ESVALS - Insert a Vehicle in a Downstream Station Arrival List in Time Order
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.99.2 Argument Dictionary

None.

6.2.99.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DONE		L*1	path tracing completed indicator
ENTRY		I*4	alternate path table entry being processed
ETA		R*4	computational variable for estimated time of arrival
FIRST		I*4	temporary vehicle ID used for comparison
LINK		I*4	temporary variable for current link being processed
LINK1		I*4	ID of first link along a priori alternate path assignment
NEXT		I*4	temporary vehicle ID used for comparison - ID following 'FIRST' in arrival list
STN		I*4	current vehicle station location
VEH		I*4	temporary variable for vehicle being processed.

6.2.99.4 Description

ESVALS determines the estimated time of arrival (ETA) of the vehicle at its next downstream or reserving station prior to vehicle launch. The vehicle's projected arrival is recorded in the station's vehicle arrival list in time order. This list is used in assigning vehicles in the servicing of arriving trips at the station in demand responsive service.

Upon entry, the reserving station for the vehicle is set to its next network destination. The ETA to that destination is determined by the path assigned to the vehicle as follows:

- a. If real-time path selection is enabled or the vehicle is assigned to traverse a minimum path route, the time of arrival at the next station is derived from the estimated time of arrival table. This table provides the travel time from any station to any other along the minimum path. This time is adjusted by the time required to complete travel in the current station (Equation (1)).
- b. If an a priori alternate route has been assigned to the vehicle and the path along this route was previously traced during the merge scheduling process, then the ETA is assigned the travel time along the alternate path as computed by ESMDLY.

If the travel time along the alternate path has not been determined, then the ETA is computed by tracing this path and accumulating the travel times for each link, between the current station and the vehicle's destination. (Equation (2)). The assigned minimum path or successor link table is used to identify each successive link along the path to the destination until a common diverge beginning the alternate path definition is found (represented by a table entry >1000). Tracing of the path and accumulation of the travel times continues by processing each link identified in the particular alternate route list until a '0' or '-1' is found, indicating the end of that alternate path. At this time the minimum path is again used to continue the trace process. If the destination station is encountered while using the alternate path table, tracing of the path is complete. Similarly, ETA processing is complete when a '0' value indicating station arrival is encountered while tracing the path in the minimum path table.

Once the ETA for the vehicle has been determined, the vehicle ID is placed on the downstream station arrival list in time order. If the arrival list is currently empty, the vehicle is placed first in the list. Otherwise, its ETA is compared to the ETA of the other vehicles in the list until the proper position is found. The vehicle is then enqueued into the arrival list.

The above processing is repeated for each vehicle of an entrained group.

6.2.99.5 PDL

See Appendix A.

6.2.99.6 Decision Tables and Algorithms

Estimated Travel Times

The estimated travel time to the next station computed for the vehicle if real time path selection is enabled or if an a priori alternate has not been assigned is given by:

$$1. \quad \text{ETA} = C_T + E_T + T_{XY}$$

where:

C_T = current clock time (cu),

E_T = station link travel time from launch to exit ramp merge with guideway (cu) and

T_{XY} = travel time along the guideway from the end of the exit ramp of station X to the entrance of station Y (assuming the minimum path) (cu)

The estimated travel time computed for a vehicle with an alternate route assignment is given by:

$$2. \quad \text{ETA} = C_T + E_T + EL_T$$

where:

L_T = link travel time of a link along the minimum and alternate paths taken enroute to the vehicle's destination (cu) and

C_T and E_T as described above.

See EGALT for discussion of the relationship between minimum path table common diverge values and alternate routes.

ARRIVAL LIST PROCESSING

See EGVALS

6.2.99a ESVREQ - Process Ready Vehicle Request

6.2.99a.1 Identification

- o ESVREQ - Process Ready Vehicle Request
- o GM TSC - July 1, 1981
- o PARAFOR

6.2.99a.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
OSTN	-	I*2	Origin station ID of vehicle request
TRNID	-	I*4	Transaction ID of vehicle request

6.2.99a.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DSTN	-	I*4	Destination station ID
FOUND	-	L*1	Flag indicating vehicle available in proper channel
MAXINV	-	I*4	Maximum inventory for any proper channel
ODIRPL	-	I*4	Origin station directional platform ID
ODIRST	-	I*4	Origin station directional station ID
SL	-	I*4	Station link loop counter
SLDOCK	-	I*4	Station link ID chosen as proper channel with maximum inventory
TRN	-	I*4	Temporary transaction ID
VEH	-	I*4	Vehicle ID of most downstream vehicle on station link
XTN	-	I*4	Transaction ID used to schedule inventory management

6.2.99a.4 Description

Routine **ESVREQ** performs processing to attempt to match a vehicle to satisfy a ready vehicle request during timeout/group demand responsive

service. The ready request is first removed from the pending request queue and placed in the ready request queue. Inventory control parameters at the origin station directional platform and directional station are updated to reflect the new ready vehicle request. A search is conducted to find an unassigned vehicle at the load berth of a channel which dispatches in the correct direction. In addition, if passengers will board, the channel must be part of the correct directional platform. The eligible vehicle in the channel with the highest inventory is selected. Routine ESAREQ is then invoked to assign the request to the selected vehicle and a prompt is issued to begin the board event. Inventory management is then scheduled for the selected origin directional station or for the origin directional station for a turnaround channel if no vehicle was found to satisfy the ready request.

6.2.99a.5 PDL

None.

6.2.99a.6 Decision Tables and Algorithms

None.

6.2.100 ESVRES - Perform Trip Reservation

6.2.100.1 Identification

- o ESVRES - Reserve Space on a Vehicle for a Trip
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.2.100.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TRP		I*4	trip requesting a reservation
VEH		I*4	vehicle to be reserved

6.2.100.3 Local Variable Dictionary

None.

6.2.100.4 Description

ESVRES is invoked to reserve space for a trip on a specified vehicle. This is accomplished by the following:

- o adding the trip ID to the reserved trip list associated with the vehicle,
- o assigning the vehicle ID on which the trip will be reserved to the trip
- o incrementing the number of reserved passengers associated with the vehicle by the number of trip passengers.
- o if the trip is handicapped, incrementing the number of reserved handicapped passengers associated with the vehicle by the number of trip passengers.

If the vehicle is operating in demand responsive multi-service and no prior passengers or reservations have been assigned to the vehicle, the ultimate destination of the vehicle becomes the destination of the reserving trip.

6.2.100.5 PDL

See Appendix A.

6.2.100.6 Decision Tables and Algorithms

None.

6.2.101 EUDIVF - User Diverge Function Interface Routine

6.2.101.1 Identification

- o EUDIVF - User Diverge Function Interface Routine
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.101.2 Argument Dictionary

None.

6.2.101.3 Local Variable Dictionary

None.

6.2.101.4 Description

EUDIVF is provided to facilitate the recognition of a user-defined station model diverge function(s). The routine currently consists only of an error message indicating the non-existence of such a function. The message can be replaced at the time a user-supplied diverge function becomes available.

6.2.101.5 PDL

See Appendix A.

6.2.101.6 Decision Tables and Algorithms

None.

6.2.102 EUEVA - User Empty Vehicle Assignment Interface Routine

6.2.102.1 Identification

- o EUEVA - User-Specified Empty Vehicle Management Algorithm
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.102.2 Argument Dictionary

None.

6.2.102.3 Local Variable Dictionary

None.

6.2.102.4 Description

EUEVA is provided to facilitate the recognition of a user-defined empty vehicle management algorithm. The routine currently consists only of an error message indicating the non-existence of such a function. The message can be replaced at the time a user-supplied algorithm for empty vehicle management becomes available.

6.2.102.5 PDL

See Appendix A.

6.2.102.6 Decision Tables and Algorithms

None.

6.2.102a EUEVB - User Empty Vehicle Bumping Interface Routine

6.2.102a.1 Identification

- o EUEVB - User Empty Vehicle Bumping Interface Routine
- o GM TSC - July 1, 1981
- o PARAFOR

6.2.102a.2 Argument Dictionary

None.

6.2.102a.3 Local Variable Dictionary

None.

6.2.102a.4 Description

EUEVB is provided to facilitate the recognition of a user-defined empty vehicle bumping algorithm. The routine currently consists only of an error message indicating the non-existence of such a function. The message can be replaced at the time a user-supplied algorithm for empty vehicle bumping becomes available.

6.2.102a.5 PDL

None.

6.2.102a.6 Decision Tables and Algorithms

None.

6.2.103 EUPCMP - User Periodic Computation Event Interface Routine

6.2.103.1 Identification

- o EUPCMP - User-Specified Periodic Computation Processing
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.103.2 Argument Dictionary

None .

6.2.103.3 Local Variable Dictionary

None .

6.2.103.4 Description

EUPCMP is provided to facilitate the recognition of user-defined periodic computation processing. The routine currently consists only of an error message indicating the non-existence of such a function. The message can be replaced at the time a user-supplied process becomes available.

6.2.103.5 PDL

See Appendix A.

6.2.103.6 Decision Tables and Algorithms

None .

6.2.104 EZINT - Correct Time Integrals and Record System Totals

6.2.104.1 Identification

- o EZINT - Correct Time Integrals for Obtaining Time Averages
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.104.2 Argument Dictionary

None.

6.2.104.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
AINT	-	I*4	divisor for computing time averages
QDV	-	R*4	queue delay

6.2.104.4 Description

EZINT is invoked during periodic sampling to correct time integral accumulations recorded over the sampling to reflect current status on modeling entities and compute system summary statistics. Time integrals are maintained for all statistical data reflecting averages over the sampling interval. Averages are obtained by:

1. Initializing the time integral for a particular statistic at the start of the sampling interval to the negative of the product of the current time (clock) value and the current occupancy of the element to which the integral is associated. This value is reinitialized after each periodic sample is taken.
2. Whenever the occupancy decreases during the sampling intervals, the integral is increased by the product of the current time value and the size of the occupancy decreases.
3. Whenever the occupancy increases during the sampling interval, the integral is decreased by the product of the current time and size of increase.

At the end of the interval, the integral value is corrected by increasing it by the product of the current time value and the current occupancy. At this point, the integral contains the desired value. The average value of the statistic can then be obtained by dividing the computed time integral by the periodic sampling interval. This computation is performed as required during output processing and statistical reporting during simulation execution. The specific sample statistics reflecting time average accumulation are listed in the table of raw statistics presented in Appendix C.

As EZINT cycles through each set of statistics, each sample item which contains an historical minimum is checked to determine if it contains its initialized value. This value is a large positive number assigned to the statistic by EZZERO prior to start of data accumulation. If the statistic contains its initialization value, it is reset to zero in order that an artificial minimum not be recorded in the sampling file. Once all corrections have been applied to individual sampling statistics, system level accumulation of individual guideway, station, station link, trip and vehicle statistics is performed. This involves the appropriate summation of individual statistics and recording of minimum and maximum values selected on an entity group basis.

6.2.104.5 PDL

See Appendix A.

6.2.104.6 Decision Tables and Algorithms

None.

6.2.105 EZHDR - Sampling Header Record Processing

6.2.105.1 Identification

- o EZHDR - Write Raw Statistics Header Record
- o IBM/FSD - May 1, 1978

6.2.105.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
NFOLL		I*4	number of follower records
NTYPE		I*4	type of follower records

6.2.105.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
HEADRE		R*4	character string, 'HEADER', to write in records
MBYTES		I*2	number of bytes in header
MFOLL		I*2	number of follower records
MTYPE		I*2	type of follower records

6.2.105.4 Description

EZHDR is invoked during periodic sampling to write a header record to the raw statistics file that indicates the number and type of records to follow. The header record contains the keyword 'HEADER', its own length, the clock time, number and type of follower records. This header is used by the Output Processor to control recognition, storage allocation and processing for sampling data contained in the follower records without regard to sample file ordering.

6.2.105.5 PDL

See Appendix A.

6.2.105.6 Decision Tables and Algorithms

None.

6.2.106 EZZERO - Initialize Periodic Sampling Statistics

6.2.106.1 Identification

- o EZZERO - Reset Statistics at the Beginning of a Periodic Sampling Interval
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.106.2 Argument Dictionary

None.

6.2.106.3 Local Variable Dictionary

None.

6.2.106.4 Description

Once the statistical data has been recorded in the sampling file, all statistics are reset as required by invoking EZZERO to begin data collection for the next sampling period. This process involves initialization of all time integrals as described in EZINT, the zeroing of all status related statistics associated with simulation activity over a periodic sampling interval, and resetting of historical minimums and maximums to the current status of the their associated modeling entity. Those historical minimums associated only with completion activity in the simulation are assigned an artificially high value to ensure minimum recording will occur as required. If no recording occurs, these minimums are corrected by EZINT prior to sample data recording. The initialization value for each sampling statistic is presented in summary form in the table contained in Appendix C.

6.2.106.5 PDL

See Appendix A.

6.2.106.6 Decision Tables and Algorithms

None.

6.2.107 FREE - Return a Transaction to an Available List

6.2.107.1 Identification

- o FREE - Return an Entity to an Available List Macro
- o IBM/FSD- May 1, 1978
- o PL/I

6.2.107.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TYPE	-	C*1	entity type to be freed (x,v,t)
INDEX	-	C	variable name of entity to be returned to the list of available transactions.
XLIST	-	C	head to the available list (optional; default xavail/vavail/tavail)

6.2.107.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
OUT	-	C	generated FORTRAN code
M	-	C	hold margin pointer
LIST	-	C	'AVAIL' default string
TYPE1	-	C	null string

6.2.107.4 Description

The purpose of FREE is to return an entity to the available list of corresponding entities. This macro generates code which when executed checks to see if the entity is in a chain and if so, generates an error message. Otherwise, it changes the entity's chain word to point to the current top of the available transactions list and changes the list head to point to it.

6.2.107.5 PDL

See Appendix A.

6.2.107.6 Decision Tables and Algorithms

None.

6.2.108 GET - Obtain an Available Transaction

6.2.108.1 Identification

- o GET - Obtain an Entity from the Available List Macro
- o IBM/FSD - May 1, 1978
- o PL/I

6.2.108.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TYPE	-	C*1	entity type to be requested (x,v,t)
INDEX	-	C	variable name of entity to be gotten from the list of available transactions
XLIST	-	C	head to the available list (optional; default xavail/vavail/ tavail)

6.2.108.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
OUT	-	C	generated FORTRAN code
M	-	C	hold margin pointer
LIST	-	C	'AVAIL' default string
TYPE1	-	C	null string

6.2.108.4 Description

The purpose of GET is to obtain an entity from the available list. This macro generates code which when executed checks to see if there are any more entities in the available list. If none are available, an error message is generated. Otherwise, the list head is changed to the value of the current list head, the chainword of this now-old top entity is set to zero, and its ID returned.

6.2.108.5 PDL

See Appendix A.

6.2.108.6 Decision Tables and Algorithms

None .

6.2.109 MULTICK - Test Transaction for Current Enqueue

6.2.109.1 Identification

- o MULTICK - Test if Currently Enqueued Macro
- o IBM/FSD - May 1, 1978
- o PL/I

6.2.109.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TYPE	-	C*1	entity type to be requested (x,v,t)
INDEX	-	C	variable name of entity
XCHAIN	-	C	entity chain word
NAME		C	text designation of chain to appear in error message (optional)

6.2.109.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
OUT	-	C	constructed FORTRAN code

6.2.109.4 Description

This macro is used by other macros to ensure that an entity is queued in only one list at a time. It generates code which when executed tests the chainword of the entity. If it is non-zero, an error message is generated.

6.2.109.5 PDL

See Appendix A.

6.2.109.6 Decision Tables and Algorithms

None.

6.2.110 NQUE - Enqueue a Transaction

6.2.110.1 Identification

- o NQUE - Place an Entity into a Queue Macro
- o IBM/FSD- May 1, 1978
- o PL/I

6.2.110.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TYPE	-	C*1	entity type to be enqueued (x,v,t)
HEAD	-	C	queue list head/tail word must be fully qualified
INDEX	-	C	entity ID of entity to be enqueued
LIFO	-	I*4	type of enqueue required: number greater than zero = LIFO null = FIFO
YCHAIN	-	C	transaction chain word (optional; default QUECH/FELCH).

6.2.110.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
OUT	-	C	constructed FORTRAN code
M	-	C	hold margin pointer
XCHAIN	-	C	'QUECH' default string

6.2.110.4 Description

The purpose of NQUE is to place an entity into a queue. This macro generates code which first uses MULTICK to determine if the entity is already enqueued. If not, it then proceeds to alter chain-

words and the queue head in order to LIFO/FIFO enqueue the entity.

6.2.110.5 PDL

See Appendix A.

6.2.110.6 Decision Tables and Algorithms

None.

6.2.111 PROMPT - Schedule a Transaction Dequeue Event

6.2.111.1 Identification

- o PROMPT - Schedule a Transaction Dequeue from a Guideway Link or Station
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.111.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TYPE	-	C	entity type for dequeue (G=guideway, S=Station)
NUM	-	C	entity number for dequeue
XSL	-	C	station link number for dequeue
XFLAG	-	C	type of dequeue required (0=upstream, 1=specified entity)

6.2.111.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
FLAG	-	C	type of dequeue
GOTO	-	C	system event ID for dequeue process (5)
LCLA	-	C	computational variable
LCLB	-	C	computational variable
LCLC	-	C	computational variable
M	-	C	margin indicator
OUT	-	C	generated FORTRAN text

VARIABLE	DIM	TYPE	DESCRIPTION
SL	-	C	station link ID
TYPE1	-	C	dequeue type

6.2.111.4 Description

The PROMPT macro provides a standardized programming interface for generating the code associated with dequeuing transactions from guideway or station modeling entities. The macro generates the code for examining required queue lists for the existence of queued transactions and scheduling a dequeue event on the Future Events List if queued transactions exist. In general, the queues examined for waiting vehicles are those immediately upstream of the specified entity as follows:

- o Station - guideway link immediately upstream of the station
- o Station Link - upstream links in the station, guideway link upstream of station in the case of the input ramp
- o Guideway Link - the upstream guideway links or a combination of an upstream station and a guideway link

The macro also provides the ability to perform a dequeue event on the specified entity without considering upstream connectivity. In this case, only the specified entity is examined for the existence of queued transactions.

6.2.111.5 PDL

None.

6.2.111.6 Decision Tables and Algorithms

None.

6.2.112 QCHECK - Check Transaction Availability in a Queue

6.2.112.1 Identification

- o QCHECK - Check for Existence of a Transaction in a Queue List
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.2.112.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TYPE	-	C*1	transaction type to check (x-system, v-vehicle, t-trip)
HEAD	-	C	Queue list pointer (fully qualified)
INDEX	-	C	(INPUT) Transaction ID (OUTPUT) transaction ID if available in queue, 0 if not available
YCHAIN	-	C	Transaction Chain word (Optional: default standard chain word QUECH/FELCH)

6.2.112.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
LOC1	-	C	current transaction in queue
LOC2	-	C	next transaction in queue
LOC3	-	C	transaction ID being checked, if found 0, otherwise
M	-	C	margin indicator
OUT	-	C	generated FORTRAN text

6.2.112.4 Description

OCHECK provides a standardized programming procedure for generating a Fortran code sequence to examine a modeling queue list for the existence of a specific transaction. The code generated by the macro checks the indicated queue for the existence of the specified transaction. If it is found, the transaction ID is returned. Otherwise, a zero transaction ID is returned.

6.2.112.5 PDL

None .

6.2.112.6 Decision Tables and Algorithms

None.

6.2.113 XGDIPF4 - Read Full Word GDIP Data

See subsection 6.1.38, XGDIPF4.

6.2.114 XGDIPH4 - Read Half-Word GDIP Data

See subsection 6.1.39, XGDIPH4.

6.2.115 XGDIPX4 - Read Single Word GDIP Data

See subsection 6.1.40, XGDIPX4.

6.2.116 XNDBOR - Generalized Data Input Processing

See subsection 6.1.41, XNDBOR.

6.2.117 XPSEUDO - I/O Interrupt Routine

See subsection 6.1.42, XPSEUDO.

6.2.118 XTRACBK - Interrupt Handler

See subsection 6.1.43, XTRACBK.

6.2.119 XTRCBKP - Save Area Trace Formatting

See subsection 6.1.44, XTRCBKP.

6.3 OUTPUT PROCESSOR

This section contains the subprogram descriptions for the DESM Output Processor.

6.3.0 AADATE - Provide Date in Character Format

6.3.0.1 Identification

- o AADATE - Provide Date in Character Format
- o GM TSC - July 1, 1981
- o PARAFOR

6.3.0.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
MMDDYY	14	L*1	Month - day - year in character format

6.3.0.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
COLON	-	L*1	The character ":"
DAY	-	I*2	Day of the month
DIGITS	-	L*1	The digits "0" through "9"
HR	-	I*2	Hour
MIN	-	I*2	Minute
MO	-	I*2	Month
SEC	-	I*2	Second
SLASH	-	L*1	The character "/"
SPACE	-	L*1	The character " " (space)
YR	-	I*2	Year

6.3.0.4 Description

This routine provides the date and time in character format. First, routine DAYTIM is called to obtain the date and time in numeric fields from the system. Then these numeric fields are converted to character format and stored in array MMDDYY.

6.3.0.5 PDL

None.

6.3.0.6 Decision Tables and Algorithms

The format of the date and time stored in array MMDDYY is as follows:

MM/DD/YY HH:MM

6.3.1 CKFOLLOW - Validate Follower Record

6.3.1.1 Identification

- o CKFOLLOW - Validate Follower Record
- o IBM/FSD - May 1, 1978
- o PL/I

6.3.1.2 Argument Dictionary

None.

6.3.1.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
OUT	-	C	Generated Fortran code sequence

6.3.1.4 Description

CKFOLLOW generates a code sequence to verify that a specified record is an expected follower. This verification is performed by examining the first eight bytes of the record for the identifier 'FOLLOWER'. If this ID is not found, an error indicating incorrect sequencing of the raw statistics file is issued and Output Processor termination occurs.

6.3.1.5 PDL

None.

6.3.1.6 Decision Tables and Algorithms

None.

6.3.2 DAYTIM - Obtain Date and Time

See subsection 6.1.1, DAYTIM.

6.3.3 DEBUG - Auxiliary Output Macro

See subsection 6.1.2, DEBUG.

6.3.3a DERROR - Error Termination Routine

6.3.3a.1 Identification

- o DERROR - Error Termination Routine
- o GM TSC - July 1, 1981
- o PARAFOR

6.3.3a.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
RETCO	-	I*4	Error return code

6.3.3a.3 Local Variable Dictionary

None.

6.3.3a.4 Description

Routine DERROR terminates the output processor with the proper condition code following an error generated during station-to-station measures display processing.

6.3.3a.5 PDL

Contained in code segment.

6.3.3a.6 Decision Tables and Algorithms

None.

6.3.4 DTIMEL - Read System Clock

See subsection 6.1.3, DTIMEL.

6.3.5 EABIN - Bin Area Reallocation

6.3.5.1 Identification

- o EABIN - Perform Bin Reallocation when Initial Allocation Exceeded
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.5.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
NB	-	I*4	Bin number to be checked
IP	-	I*4	Required bin size in words

6.3.5.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
K	-	I*4	Pointer to area beyond bin
L	-	I*4	Current location of bin
I1	-	I*4	Pointer to start of old bin
I2	-	I*4	Pointer to start of new bin
KL	-	I*4	Pointer to blank area beyond expanded bin
LMX	-	I*4	Number of bin positions to be moved
IDIF	-	I*4	Left over bin area
IREAL	-	I*4	Required bin size plus 4 for bin header
ISIZE	-	I*4	Current size of bin

6.3.5.4 Description

EABIN is invoked by ESTORE during the data acquisition process to expand a bin data storage area when its limits have been exceeded. This may require shifting data within the bin storage area to acquire additional space or expanding the limits of the bin to encompass more available entries as follows:

1. If sufficient space has been allocated, no changes are made.
2. If sufficient space has not been allocated, but a pseudo-bin (unused area) immediately follows the bin being allocated, then:
 - a. If the total space of the two bins (allocated plus adjacent unused area) is within four positions of the requirements, the total space is allocated to the bin and the pseudo-bin is eliminated.
 - b. If the total space exceeds the requirement by at least four positions, then the excess space over and above the required space is made into a pseudo-bin.
3. If the bin cannot remain where it is, since insufficient empty space exists, then an attempt to find a new location with sufficient space is made. First, the empty area at the end of the entire bin storage area is checked.
 - a. If the end of the area is large enough, the old bin contents are copied into it, the previous bin location is set to a pseudo-bin, and the array bin location pointer is updated.
 - b. If the end of the bin storage area has insufficient space, all bins are moved up by eliminating pseudo-bins to compress unused space. Test 3(a) is then repeated. If it fails, step (4) is tried.
4. The amount of area covered by the bin itself plus the space available back of the bin storage is checked. If this is below the required space, processing terminates. Otherwise:
 - a. If the bin being allocated and the free area are adjacent, the bin is simply enlarged by using part of the free area space in the bin storage area.
 - b. If the two areas are not adjacent, then all bins between the current one and the free area are moved downward to provide the necessary space.

6.3.5.5 PDL

See Appendix A.

6.3.5.6 Decision Tables and Algorithms

None.

6.3.6 EBNCHK - Check Bin Allocation

6.3.6.1 Identification

- o EBNCHK - Check Current Allocation of Required Data Accumulation Bin
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.6.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
NB	-	I*4	Bin Number to be checked
IP	-	I*4	Required bin size in words

6.3.6.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
K	-	I*4	Pointer to area beyond bin
L	-	I*4	Current location of bin
I1	-	I*4	Pointer to start of old bin
I2	-	I*4	Pointer to start of new bin
KL	-	I*4	Pointer to blank area beyond expanded bin
LMX	-	I*4	Number of bin positions to be moved
IDIF	-	I*4	Left over bin area
IREAL	-	I*4	Required bin size plus 4 for bin header
ISIZE	-	I*4	Current size of bin

6.3.6.4 Description

EBNCHK is invoked during the request filing process to ensure that the bin assigned for data accumulation to satisfy a particular request is allocated large enough. The following processing is performed:

1. If the bin has enough space allocated already, then:
 - a. If the allocation is four positions or more than required, the extra space is made into a pseudo-bin (available space in bin storage area).
 - b. If the allocation is within four positions of required, no changes are made.
2. If more space is needed, then the currently allocated bin area is changed to a pseudo-bin and an attempt is made to relocate the bin as:
 - a. If the end of the bin storage area has enough unused space, the bin is placed there.
 - b. If the back of the bin storage area does not have enough space, then:
 - (1) All bins are moved towards the top of the bin area by eliminating any pseudo-bins that may be interspersed.
 - (2) Test 2(a) above is repeated. If it fails this time, no additional space is available and processing terminates.

6.3.6.5 PDL

See Appendix A.

6.3.6.6 Decision Tables and Algorithms

None.

6.3.7 EDBIN - Bin Allocation

6.3.7.1 Identification

- o EDBIN - Perform Initial Allocation of Bin Area
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.7.2 Argument Dictionary

None.

6.3.7.3. Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
KX	-	I*4	pointer to end of current bin
KNN	-	I*4	pointer to beginning of remaining bin area
LONG	-	I*4	5 = initial length of bin
KN	-	I*4	displacement of bin being allocated

6.3.7.4 Description

EDBIN defines an initial number of bins in the bin area each having header information initialized to indicate the bin is currently empty and available for use. Any space remaining in the storage area after definition is complete, is allocated to one large bin area. The allocation process involves cycling through the bin storage area and establishing each five locations in the area as a bin with the following characteristics defined:

1. Total number of words allocated to bin (=5)
2. Bin number
3. Starting index of bin data
4. Ending index of bin data
5. Identification mnemonic = 0

Any remaining space in the bin storage area is defined as a large bin which serves as the basis for dynamic bin storage area allocation during data acquisition and manipulation processing.

6.3.7.5 PDL

See Appendix A.

6.3.7.6 Decision Tables and Algorithms

None.

6.3.8 EDUMBIN - Formatted Dump of Bin Area

6.3.8.1 Identification

- o EDUMBIN - Formatted Dump of Bin Area
- o IBM/FSD - July 1, 1977
- o PARAFOR

6.3.8.2 Argument Dictionary

None.

6.3.8.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
J	-	I*4	Pointer to start of data in bin header
I1	-	I*4	Total words allocated to bin
I2	-	I*4	Logical bin number
I3	-	I*4	Pointer to start of data
I4	-	I*4	Pointer to end of data
I5	-	I*4	Length of data in bin
JNN	-	I*4	JN-5
JTOT	-	I*4	Number of free words
MTOT	-	I*4	Number of items in bins
NTOT	-	I*4	Total number of words allocated to bins

6.3.8.4 Description

EDUMBIN produces a formatted dump of the bin storage area as an aid to debugging under auxiliary output control. The routine is invoked by EOUTPT prior to initiating data acquisition if the appropriate auxiliary output flag is set.

6.3.8.5 PDL

See Appendix A.

6.3.8.6 Decision Tables and Algorithms

None.

6.3.9 EGRAPH - Time Series Plotting

6.3.9.1 Identification

- o EGRAPH - Format and Display a Time Series Plot of Sampled Items
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.9.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
XO	-	R*4	Starting X value
DELTA	-	R*4	X increment
NDELTA	-	I*4	# points to be plotted
NY	-	I*4	# bins to be plotted
Y1	-	R*4	Bin #1
Y2	-	R*4	Bin #2
Y3	-	R*4	Bin #3
Y4	-	R*4	Bin #4
BOTTOM	-	R*4	Lower limit on Y values
TOP	-	R*4	Upper limit on Y values
SYMBOL	4	R*4	Plotting symbols

6.3.9.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
PLOT	101	R*4	Every 10th line to be plotted
GRID	101	R*4	1st thru 9th lines to be plotted

VARIABLE	DIM	TYPE	DESCRIPTION
GRDMK	11	R*4	Y scale values for titling
TITLE	21	R*4	Unused
LABEL	21	I*4	Unused
LABAB	21	I*4	Unused
Y1	-	R*4	Value of first graph plotted
Y2	-	R*4	Unused
Y3	-	R*4	Unused
Y4	-	R*4	Unused
BLANK	-	R*4	6 blanks " "
DASH	-	R*4	6 dashes " - - - - - "
CROSS	-	R*4	6 dots ". "
J	-	I*4	Pointer to value to be plotted
X	-	R*4	Sample number
SYM	-	R*4	Symbol(1)
LAB1	-	I*4	Label(1)
LAB2	-	I*4	Label(2)
LINE	-	I*4	Count of lines printed
NOUT	-	I*4	SYSOUT unit number
YMIN	-	R*4	Minimum of all bins to be plotted
YMAX	-	R*4	Maximum of all bins to be plotted
LIMIT	-	I*4	Number of points to be plotted
SAVE1	-	R*4	Hold previous value of grid/plot
SAVE2	-	R*4	Unused

VARIABLE	DIM	TYPE	DESCRIPTION
SAVE3	-	R*4	Unused
SAVE4	-	R*4	Unused
NLABEL	-	I*4	Unused
ORDSCL	-	R*4	Ordinal scale used to compute location of symbol
SAMSCL	-	R*4	Unused

6.3.9.4 Description

EGRAPH sets up grid lines to be displayed, computes scaling factors, scales data points and produces desired hardcopy output for sampled items in a bin. Off scale points are plotted with a special symbol at the appropriate limit of the output plot.

6.3.9.5 PDL

See Appendix A.

6.3.9.6 Decision Tables and Algorithms

None.

6.3.10 EHEADER - Read a Header Record

6.3.10.1 Identification

- o EHEADER - Read Next Header Record
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.10.2 Argument Dictionary

None.

6.3.10.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
HEADRE	-	R*8	'HEADER '

6.3.10.4 Description

EHEADER is invoked by EZRED to read the next record from the Raw Statistics File. If expected header is not found, it issues a warning message to indicate incorrect formatting of the file. The routine is invoked after a previously defined sequence of follower records has been processed for the preceding header record contained in the file.

6.3.10.5 PDL

See Appendix A.

6.3.10.6 Decision Tables and Algorithms

None.

6.3.11 EHIST - Format and Display Histogram

6.3.11.1 Identification

- o EHIST - Format and Display Histogram of Sampled Data Item
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.11.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
B	-	R*4	Location of bin containing data
C	-	R*4	Location of work bin
DLT	-	R*4	Class interval width
NB	-	R*4	Bin number to be processed

6.3.11.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TYPE	10	I*2	2 character designation of SL type
K	-	I*4	Start of work bin
IA	-	I*4	Histogram slot number to be incremented
IC	-	I*4	Count in a particular histogram slot
I1	-	I*4	Pointer to start of data in data bin
I2	-	I*4	Pointer to end of data in data bin
I3	-	I*4	Start of work bin
I4	-	I*4	Last position in work bin
JA	-	I*4	Number of markers to be associated with a particular histogram slot

VARIABLE	DIM	TYPE	DESCRIPTION
MK	-	I*4	The character 'X' used to print histogram
AMP	-	R*4	Number of markers per count
AMX	-	R*4	Amplitude per interval boundary
DLX	-	R*4	Class interval width
DOT	-	R*4	The character '.'
SUM	-	R*4	Sum of values in data bin
ANNN	-	R*4	Number of samples
AVAR	-	R*4	Variance of values in data bin
GRID	101	R*4	101 *'s
NDIS	-	I*4	Displacement to 16 character title for selected statistic
XMAX	-	R*4	Largest count in a histogram slot
AMEAN	-	R*4	Mean of values in data bin
SUMSQ	-	R4	Sum of squares of values in data bin
COMMON HISTO	See	EZHIST	

6.3.11.4 Description

EHIST cycles through the sampled data items stored in a specified bin accumulating the sum, sum squared of each sampled item along with a frequency of occurrence within a given class of intervals. The mean and variance of the data is computed and the desired histogram is formatted and displayed on the system output device.

6.3.11.5 PDL

See Appendix A.

6.3.11.6 Decision Tables and Algorithms

a. Mean

The mean value of the data is given by:

$$\mu = \sum_{i=1}^n x/n$$

where:

x = individual sampled values of the selected statistic

n = number of sampled items

b. Variance

The variance of the data is given by:

$$\sigma^2 = \sum_{i=1}^n x^2/n - \mu^2$$

where:

x, n and μ are defined as above.

6.3.12 ELIST - Format and Display Time Series List or Statistical Summary

6.3.12.1 Identification

- o ELIST - List or Produce Statistical Summary of Sampled Values for a Selected Statistic
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.12.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
B	-	R*4	Pointer to beginning of data in bin to be processed
IP	-	I*4	The index for listing bin elements (IP=1 list every element)
NB	-	I*4	Bin number to be processed

6.3.12.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TYPE	10	I*2	2 character designation of SL type
AN	-	R*4	Number of samples including 0's
BX	-	R*4	First data item in bin
I1	-	I*4	Pointer to start of data in bin
I2	-	I*4	Pointer to end of data in bin
ANO	-	R*4	Number of samples excluding 0's
IBX	-	I*4	First data item in bin
AMIN	-	R*4	Minimum including 0's
NDIS	-	I*4	Pointer to 16 character title

VARIABLE	DIM	TYPE	DESCRIPTION
AMEAN	-	R*4	Mean including 0's
INDEX	-	I*4	Station/Trip link number
STDEV	-	R*4	Standard deviation including 0's
AMEANO	-	R*4	Mean excluding 0's
STDEVO	-	R*4	Standard deviation excluding 0's

6.3.12.4 Description

ELIST displays the individual sample items for a selected statistic, listing every Kth element, or performs the computations necessary for producing a statistical summary of the data. If a statistical summary has been requested, the following items are computed and displayed for all sampled values including and excluding zero values:

1. Number of samples
2. Sum of values
3. Mean per sample
4. Standard Deviation from the mean
5. Minimum value
6. Time of minimum (seconds)
7. Maximum value
8. Time of maximum (seconds).

6.3.12.5 PDL

See Appendix A.

6.3.12.6 Decision Tables and Algorithms

a. Mean

The mean value of the data is given by:

$$\mu = \frac{\sum_{i=1}^n x/n}{n}$$

where:

x = individual sampled values of the selected statistic

n = number of sampled items

b. Standard Deviation

The standard deviation of the data is given by:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n x^2/n}{n} - \mu^2}$$

where:

x, n and μ are defined as above.

6.3.13 EMNMX - Compute Minimum and Maximum Values of Sampled Items

6.3.13.1 Identification

- o EMNMX - Compute Minimum and Maximum Values
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.13.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
B	21	R*4	Specified bin
C	21	R*4	Bin for storing min, max and range
IP	-	I*4	Word in Bin C for storing computed values

6.3.13.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
IC	-	I*4	C(1)
I1	-	I*4	Start of data
I2	-	I*4	End of data
BMAX	-	R*4	Maximum value found
BMIN	-	R*4	Minimum value found
RANGE	-	R*4	BMAX-BMIN

6.3.13.4 Description

EMNMX cycles through a specified bin, determines the minimum, maximum data values and computes the range given by the difference and stores in some specified bin location.

6.3.13.5 PDL

See Appendix A.

6.3.13.6 Decision Tables and Algorithms

None.

6.3.14 EODATA - Output Processor Data Initialization

6.3.14.1 Identification

- o EODATA - Initialize Values of Fixed Data Areas
- o IBM/FSD - May 1, 1978
- o FORTRAN

6.3.14.2 Argument Dictionary

None.

6.3.14.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
MATTAB	120	I*2	Major category match table
MAINTA	120	I*4	Major category EBCDIC mnemonic table
MCOTAB	120	I*2	Major category range definition table - each entry contains the number of columns required in MSUTAB for subcategory mnemonics
MSUTAB	15X35	I*4	Subcategory EBCDIC mnemonics allocated as specified in MCOTAB
TITLES	2100	I*4	Statistic output labels allocated 4/statistic (16 character titles)
MSUTYP	15X35	I*2	Subcategory type table

6.3.14.4 Description

EODATA provides standard FORTRAN block data initialization of the tables used by the Output Processor.

6.3.14.5 PDL

None. Block data programs are not executable.

6.3.14.6 Decision Tables and Algorithms

Table usage is described in Section 2.

6.3.15 EOERR - Error Message Processing

6.3.15.1 Identification

- o EOERR - Error Message Processing
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.15.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
MSGNO	-	I*4	Error message number
MSG	2	L*1	Message text
MSEVER	-	I*4	Message severity code 1 - Information 2 - Warning 3 - Serious

6.3.15.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
MCLOCK	-	I*4	Clock time (not used)
MSGTYP	-	L*1	Alphabetic message type designator
NUM	-	I*4	Number of characters in message text
PGM	3	I*4	Processor identifier
SCLN	-	L*1	End of text symbol (;)
TYPE	3	L*1	Message type symbols (I,W,S)

6.3.15.4 Description

Upon entry, EOERR determines whether or not error processing was invoked while attempting to terminate OP processing due to a prior error condition. If this situation exists, a message indicating the condition

is issued and immediate termination of the OP occurs. Otherwise, the required error message is formatted and displayed on the system output device.

After message display, a determination is made as to whether any specified output limits have been violated or whether system termination is required based on message severity. If the number of times a given message is issued, total number of messages issued or the number of messages by severity code exceed SYSGEN limits, termination of the OP occurs. This termination activity attempts graceful system shutdown by invoking the DESM traceback facility (TRACBK). Similar processing is performed following the issuance of a severe error message. If error processing results in OP termination for any reason, a system condition code 16 is returned to the user.

6.3.15.5 PDL

See Appendix A.

6.3.15.6 Decision Tables and Algorithms

None.

6.3.16 EOFLAG - Auxiliary Output Request Processing

6.3.16.1 Identification

- o EOFLAG - Auxiliary Output Request Processing
- o IBM/FSD - May 1, 1978
- o PARFOR

6.3.16.2 Argument Dictionary

None.

6.3.16.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
FINI	-	L*1	Indicates last card found
TEMP	18	I*4	Flag numbers from current card
AEOF	-	L*1	End of file on asynchronous input file
L	-	I*4	Lower bound of range of flags to be set true
U	-	I*4	Upper bound of range of flags to be set true

6.3.16.4 Description

This routine sets flags associated with selecting auxiliary output generated using the DEBUG macro. This routine first resets all flags and reads cards containing the flag numbers to be set until a zero field is found. The requests can define individual flag numbers or a contiguous set of flags, e.g. 47-54. The format and contents of the request card are defined in subsection 4.1.1 of the DESM User's Manual under the description of the FLAG asynchronous input command. The requested flags remain set until another FLAG asynchronous input command is read.

6.1.16.5 PDL

See Appendix A.

6.1.16.6 Decision Tables and Algorithms

None.

6.3.17 EOINDX - Update Index File

6.3.17.1 Identification

- o EOINDX - Decode Parm Field and Update Index
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.17.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
COUNT	-	I*2	Number of characters contained in PARM field string
STRING	2	I*1	PARM field character string

6.3.17.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
BLANK	-	L*1	EBCDIC blank constant
CHAR	-	I*2	Character variable for comparison against COMMA
COMMA	-	I*2	EBCDIC COMMA constant
DAY	-	I*2	Day of month
HOUR	-	I*2	Hour of day
PTR	-	I*2	Index into STRING array
YEAR	-	I*2	Year

6.3.17.4 Description

EOINDX is called during initialization to decode PARM field information and write the initial entries reflecting OP execution to the Run Index file. The routine is invoked via an indirect call to entry point EQUPTX in routine EONTIX which initially saved the

address of the PARM information, after receiving control from the Operating System. EOUPTX retrieves the address of the PARM field data and its length, and passes them as calling arguments to EAINDX.

Upon entry, EOINDX decodes the PARM field information containing input and output file member names and saves the data for use in updating the Run Index file. The current data and time is then obtained by invoking DAYTIM. The initial entries containing data and time, processor ID, and load module name are then written to the Run Index.

During system termination the routine is invoked via entry point EOWTIX to update the Run Index file with entries reflecting output files created during MP execution. A summary of the Run Index information is also displayed on the system output device.

6.3.17.5 PDL

See Appendix A.

6.3.17.6 Decision Tables and Algorithms

None.

6.3.18 EONTIX - Save PARM Field Information

6.3.18.1 Identification

- o EONTIX - Save PARM Field Information
- o IBM/FSD - May 1, 1978
- o ASSEMBLER

6.3.18.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
AR61	-	I*2	(OUTPUT) - Length of PARM Field in bytes
AR62	-	L*1	(OUTPUT) - PARM Field Character String

6.3.18.3 Local Variable Dictionary

None.

6.3.18.4 Description

EONTIX receives control from the Operating System and saves the address of the PARM field information coded on the EXEC JCL statement. The address is passed to the routine via standard linkage in general purpose register 1. Control is then passed to the OP main routine EOUTPT to begin the simulation process.

Entry point EOUPTX is invoked by EOZNIT during initialization to retrieve the saved PARM field information for decoding and use in updating the Run Index file via EOINDX. Return from the entry point to EOZNIT is accomplished via non-standard linkage in order that a direct control transfer can occur between EOINDX and EOZNIT after PARM field decoding.

6.3.18.5 PDL

See Appendix A.

6.3.18.6 Decision Tables and Algorithms

None.

6.3.19 EOPRPT - Performance Summary Report Processing

6.3.19.1 Identification

- o EOPRPT - DESM Performance Summary Report Routine
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.19.2 Argument Dictionary

None.

6.3.19.3 Local Variable Dictionary

None.

6.3.19.4 Description

EOPRPT generates the report of the computed performance summary measures. This report, the Performance Summary Report, is also referred to as the pre-formatted Standard Report #1. The previously computed system, guideway, station and route related statistics are formatted and listed as either resource utilization, performance or level of service measures.

6.3.19.5 PDL

See Appendix A.

6.3.19.6 Decision Tables and Algorithms

None.

6.3.20 EOPSUM - Performance Summary Computation Processing

6.3.20.1 Identification

- o EOPSUM - DESM Performance Summary Data Computational Routine
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.20.2 Argument Dictionary

None.

6.3.20.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
CONV1	-	R*4	Conversion factor to hours of non-time integral data
CONV2	-	R*4	Conversion factor to hours of time integral data
DIST	-	R*4	Total vehicle distance travelled
INTRVL	-	I*4	Request interval
SECPCU	-	R*4	Seconds per clock unit
TOTLCU	-	R*4	total clock units over request interval

6.3.20.4 Description

EOPSUM computes the required performance summary measures from the sums, maximums and minimums of individual system level raw statistics accumulated over the user-specified read (data acquisition) interval. The derived statistics are categorized into resource utilization, performance and level of service measures. As each measure is computed it is added to an array which is immediately referenced if the Performance Summary File is to be written. In addition, this array of derived measures is the source data used as EOPRPT is invoked to write the Performance Summary Report.

6.3.20.5 PDL

See Appendix A.

6.3.20.6 Decision Tables and Algorithms

The derivations of the measures for the Performance Summary Report are provided in Table C-3 of Appendix C. The raw statistics referenced are summarized in Table C-2, preceding the derivations.

6.3.21 EOSRPT - System Summary Report Processing

6.3.21.1 Identification

- o EOSRPT - DESM System Summary Statistics Report Routine
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.21.2 Argument Dictionary

None.

6.3.21.3 Local Variable Dictionary

None.

6.3.21.4 Description

EOSRPT generates the report of the computed system summary statistical measures. This report, the System Summary Statistics Report, is also referred to as the pre-formatted Standard Report #2. The previously computed system-wide, guideway, station and route measures are each appropriately identified and formatted according to the specified requirements.

6.3.21.5 PDL

See Appendix A.

6.3.21.6 Decision Tables and Algorithms

A table summary of the derived system summary statistics is contained in Table C-4 of Appendix C. Associated with each measure is an indication as to whether a total or average (in addition to a maximum and minimum) value is to be derived and reported. Based on the requirements, the total or average value will be placed under the proper report heading.

6.3.22 EOSSUM - System Summary Statistics Computation Processing

6.3.22.1 Identification

- o EOSSUM - DESM System Summary Statistics Computational Routine
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.22.2 Argument Dictionary

None.

6.3.22.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
CUPSEC	-	R*4	Clock units per second conversion factor
DPML	LMAX (preprocessor) (variable)	R*4	Array equivalenced to portion of DPM array with guideway link data
DPMR	RMAX (preprocessor) (variable)	R*4	Array equivalenced to portion of DPM array with route data
SEPCU	-	R*4	Seconds per clock unit conversion factor
TOTLCU	-	R*4	Total number of clock units over request interval

6.3.22.4 Description

EOSSUM computes the required system summary statistics from the sums, maximums and minimums of individual sample items accumulated over the user-specified read (data acquisition) interval. The derived statistics reflect both system-wide and individual guideway, station and scheduled route measures of performance. As the total or average, maximum and minimum value is computed for each measure, it is added to a data area which is subsequently referenced by EOSRPT as it is invoked to generate the pre-formatted Standard Report #2.

6.3.22.5 PDL

See Appendix A.

6.3.22.6 Decision Tables and Algorithms

A table summary of the derived system summary statistics is contained in Table C-4 of Appendix C. Associated with each measure is an indication as to whether a total or average is to be derived and whether the maximum and minimum are to be historical or status values.

Following this table, in Table C-5, is a summary of the derivations of the measures for the System Summary Statistics Report or Standard Report #2. The raw statistics referenced are summarized in Table C-2, preceding the derivations.

6.3.23 EOUTPT - Output Processor Control

6.3.23.1 Identification

- o EOUTPT - Output Processor Main Routine
- o IBM/FSD- May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.3.23.2 Argument Dictionary

None.

6.3.23.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
IN	-	I*4	Numeric card type (FLAG, REQU, READ....)
AEND	-	L*4	End-of-file request cards
FND	-	L*4	Indicator card type found
TEMP	-	L*4	First character of name of data item (ignored)
MC	-	I*4	Major Category numeric code (2=SYST, 3=STN, 4=STNL, 5=LINK, 6=RTE)
XX	-	R*4	Date accumulation interval in seconds
BIN	-	I*4	Bin containing data
SUB	-	I*4	Name of data item = subcategory requested on card
FORM	-	I*4	Output format requested on card (LIST, SUMM, PLOT, HIST, PERF)
IDHI	-	I*4	High limit variable entity range

VARIABLE	DIM	TYPE	DESCRIPTION
IDLO	-	I*4	Low limit variable entity range
MAIN	-	I*4	Main category requested on card
ATERM	-	I*4	Output to terminal indicator (unused)
NAME	-	I*4	Type of card (FLAG, REQU, READ,...)
IFORM	-	I*4	Numeric form requested
MAJ	-	I*4	Major entity number
BOTTOM	-	R*4	Low limit on plot scale
TOP	-	R*4	High limit on plot scale
SCALE1	-	R*4	Input limit on plotting scale
SCALE2	-	R*4	Input limit on plotting scale
CWIDTH	-	I*4	Input histogram class interval
ICAT	-	I*4	Loop counter; stn-stn measure request names
MSRRD	-	L*1	Flag indicating trip file has been read
DBGNAM	4	I*4	Stn-stn debug name table
SSPMSR	-	L*1	Flag indicating stn-stn measures requested
WIDTHS	400	R*4	Histogram class interval
IFORMS	400	I*2	Numeric form requested
NN	-	I*4	Temporary-plot time span
IDBG	-	I*4	Loop counter: stn-stn debug name table
NANEL	5	I*4	User request names
CATNAM	13	I*4	Stn-stn measure request names
NCATGR	-	I*4	Number of stn-stn measure requests(13)
SYMBOL	5	I*4	Plot symbols
IB	-	I*4	Bin pointer
FORMS	2,12	I*4	Format names translation table

COLS	-	I*4	Number of print columns across page
DEBUG	4	L*1	Debug flags
ETIME	-	R*4	End time for triplog processing
FILE	-	L*1	Output measures to J-files
KEYTMP	-	I*4	Associated variable for TMPFIL
LINES	-	I*4	Number of lines on a printed page
MSPACE	-	I*4	Dimension of space array
MSTN	-	I*4	Maximum number of stations
REPORT	-	L*1	Report flag (T to generate report)
REQST	MMEAS	L*1	Request measure computation
RETC	-	I*4	Return code
SSPFIL	-	I*4	Unit for J-file (SSP) output
STIME	-	R*4	Start time for triplog processing
TMPFIL	-	I*4	Temporary DA file
SPACE	-	I*4	Empty space
SYSERR	-	I*4	Unit for error messages
SYSRPT	-	I*4	Unit for printed reports
TRPLOG	-	I*4	Unit with trip logs

6.3.23.4 Description

EOUTPT provides the mechanism for recognizing user output requests and invoking service components required to satisfy those requests. Control is passed to EOUTPT from an auxiliary entry point defined for saving PARM field information (EONTIX) required for later index file updating. Upon entry, EOUTPT performs initialization of the bin storage areas by invoking EOZKIT. The basic control loop for recognizing user output requests is then started. The basic loop consists of the following processing which is performed until the last user request is satisfied:

1. Read user service request and classify it as to whether it specifies required data to be collected or the acquisition and display of data.
2. If the request is for data, verify index ranges, establish index defaults and determine the number of requests which must be filed for data acquisition. The requests are then recorded by invoking EZREQU. If a performance summary or standard report request is entered, the data accumulation arrays used in computing required measures are initialized prior to the filing process. Each entity specified in a data request requires a separate bin storage area for data

acquisition. Thus, a range of entities specified on one data request causes the automatic generation of multiple internal data requests as does a request for performance summary or standard report output.

3. If the request is for data acquisition (READ Command), reading of the raw statistics file is performed by invoking EZREAD which controls data accumulation within the allocated bin areas. Once completed, the appropriate data manipulation and display is performed for each filed request by invoking one of the following processing routines:
 - o EZHIST - Histogram
 - o EZLIST - Time Series List or Summary
 - o EZPLOT - Time Series Plot
 - o EOPSUM - Performance Summary
 - o EOSSUM - DPM Summary
 - o ETSSPM - Station-to-Station Measures

Once data display has been completed, the control loop is recycled to begin processing of the next user specified group of service requests.

6.3.23.5 PDL

See Appendix A.

6.3.23.6 Decision Tables and Algorithms

None.

6.3.24 EOZNIT - Output Processor Initialization

6.3.24.1 Identification

- o EOZNIT - Initialization of the Output Processor
- o ISM/FSD - May 1, 1978
- o PARAFOR

6.3.24.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
NAREA	-	I*4	Total size in words of bin storage area
NBINS	-	I*4	Number of bins required
NREQO	-	I*4	Maximum number of requests
NLINE	-	I*4	Number of lines/page for output formatting

6.3.24.3 Local Variable Dictionary

None.

6.3.24.4 Description

Initialization is performed to establish initial conditions for the output processing of a Raw Statistics file. Upon entry, interface to the interrupt handler is established by invoking entry point SPIEL in the traceback routine. Message limits are then established and the auxiliary output indicators are zeroed. Initial bin area allocation is performed by invoking EDBIN and default parameters for raw statistics processing are established from header data in the Raw Statistics file by invoking EZREAD.

These data are acquired from the file by filing a system service request and invoking the data acquisition process in a manner analogous to processing of user service commands.

6.3.24.5 PDL

See Appendix A.

6.3.24.6 Decision Tables and Algorithms

None.

6.3.25 EREAD02 - Process Type 2 System Statistics

6.3.25.1 Identification

- o EREAD02 - Process System Wide Statistics
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.25.2 Argument Dictionary

None.

6.3.25.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
CUSAM	-	I*4	Clock units 1 second
END	-	L*1	Logic variable for END FILE
FSYST	15	I*4	Fullword summary statistics buffer
HSYST	21	I*2	Halfword summary statistics buffer
FSYSTS	46	I*4	Fullword station statistics buffer
HSYSTS	76	I*2	Halfword station statistics buffer
RSYSTG	3	R*4	Real guideway statistics buffer
FSYSTG	8	I*4	Fullword guideway statistics buffer
HSYSTG	19	I*2	Halfword guideway statistics buffer
FSYSTR	11	I*4	Fullword route statistics buffer
HSYSTR	8	I*2	Halfword route statistics buffer
VAL	-	R*4	Retrieved or computed sample value
ISUB	-	I*4	Subcategory number for statistic
IREQ	-	I*4	Request number for statistic

6.3.25.4 Description

EREAD02 reads system summary statistics containing non-indexed overall system data written to the Raw Statistics file during each sampling interval. Each sampled item is read into a predefined buffer from which the data can be retrieved and processed. The required items are retrieved from the buffer by cycling through the chained request table entries indicating which statistics are required.

The processing performed for each item is based upon the relative position of the requested data, as indicated by its subcategory index, within the input records. All time related values are converted from internal clock units to seconds and time averages are derived by dividing the sampled item by the length of the sampling interval in C.U. If performance summary or standard report outputs have been requested the sum, min and max of each system statistic is automatically derived and saved for output purposes at the conclusion of data acquisition. As each individual sampled item is retrieved it is stored in an assigned bin for the data by invoking ESTORE.

6.3.25.5 PDL

See Appendix A.

6.3.25.6 Decision Tables and Algorithms

See Appendix C for formatting of System Summary Raw Statistic Records.

6.3.26 EREAD03 - Process Type 3 Station Statistics

6.3.26.1 Identification

- o EREAD03 - Process Individual Station Statistics
- o IBM/FDS - May 1, 1973
- o PARAFOR

6.3.26.2 Argument Dictionary

None.

6.3.26.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
CUSAM	-	I*4	Clock units/second
END	-	L*1	Logic variable for END FILE
FSTAT1	KMS*22	I*4	Fullword station statistics buffer
HSTAT1	KMS*46	I*2	Halfword station statistics buffer
IREQ	-	I*4	Request number for statistics
ISUB	-	I*4	Subcategory number for statistics
RSTAT1	KMS*2	R*4	Real station statistics buffer
STN	-	I*4	Station number index
VAL	-	R*4	Retrieved or computed sample value

6.3.26.4 Description

EREAD03 reads statistics related to individual station performance written to the Raw Statistics file during each sampling interval. Each sampled item, by type, is read into a predefined buffer from which the data can be retrieved and processed. Each required item is identified by the request table entries associated with station related input

requests. Sample items are retrieved by cycling through the request entries and retrieving the data required to satisfy the entry as identified by its subcategory index. If the retrieved value is a time related measure, it is converted from interval clock units to seconds and time average data is converted by dividing the sampling interval in C.U. As each requested item is retrieved or derived from the input data, it is stored in an assigned bin by invoking ESTORE. If standard report 2 is required as an output, individual station summary statistics are computed and stored in special arrays for formatting and output upon the conclusion of data acquisition.

6.3.26.5 PDL

See Appendix A.

6.3.26.6 Decision Tables and Algorithms

See Appendix C for formatting of individual Station Raw Statistics Records.

6.3.27 EREAD04 - Process Type 4 Station Link Statistics

6.3.27.1 Identification

- o EREAD04 - Process Individual Station Link Statistics
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.27.2 Argument Dictionary

None.

6.3.27.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
END	-	L*1	Logic Variable for END FILE
FSTAT1	KMS* KMSL*6	I*4	Fullword station link statistics buffer
HSTAT1	KMS* KMSL*8	I*2	Halfword station link statistics buffer
IREQ	-	I*4	Request number for statistic
ISUB	-	I*4	Subcategory number for statistic
ISUB1	-	I*4	Computational variable for record displacement
ISUB2	-	I*4	Computational variable for record displacement
RSTAT1	KMS* KMSL*4	R*4	Real station link statistics buffer
SL	-	I*4	Station link ID for statistic
VAL	-	R*4	Retrieved or computed statistic
XSTAT1	KMS* KMSL*2	I*2	Halfword station link statistics

6.3.27.4 Description

EREAD04 reads and processes individual station link statistics recorded by the M.P. for each defined station at each periodic sampling point in a simulation run. The required statistics are retrieved by cycling through chained request table entries requiring station link output data. As each request is processed, the subcategory indicator in the request table along with the station link number is used to retrieve the required data from the input buffer and perform appropriate processing. All time related items are converted to seconds. Each retrieved or computed statistic is then stored in an assigned bin area by invoking ESTORE. If Standard Report 2 output is required, associated statistics are computed and stored in separate data accumulation arrays for use in processing upon the conclusion of data acquisition.

6.3.27.5 PDL

See Appendix A.

6.3.27.6 Decision Tables and Algorithms

See Appendix C for formatting of individual Station Link sampling records.

6.3.28 EREAD05 - Process Type 5 Guideway Statistics

6.3.28.1 Identification

- o EREAD03 - Process Individual Guideway Statistics
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.28.2 Argument Dictionary

None

6.3.28.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
CUSAM	-	I*4	Clock units/second
END	-	L*1	Local variable for END FILE
FSTAT1	KML*8	I*4	Fullword guideway link statistics buffer
HSTAT1	KML*20	I*2	Halfword guideway link statistics buffer
IREQ	-	I*4	Request number for statistic
ISUB	-	I*4	Subcategory number for statistic
LNK	-	I*4	LINK number
RSTAT1	KML	R*4	Real guideway link statistics buffer
VAL	-	R*4	Retrieved or computed sample value

6.3.28.4 Description

EREAD05 reads statistics related to individual guideway performance written to the Raw Statistics file during each sampling interval. Each sampled item, by type, is read into a predefined buffer from which the data can be retrieved and processed. Each required item is identified by the request table entries associated with guideway related input

requests. Sample items are retrieved by cycling through the request entries and retrieving the data required to satisfy the entry as identified by its subcategory index. If the retrieved value is a time related measure, it is converted from interval clock units to seconds and time average data is converted by dividing by the sampling interval in C.U. As each requested item is retrieved or derived from the input data, it is stored in an assigned bin by invoking ESTORE. If standard report 2 is required as an output, individual guideway summary statistics are computed and stored in special arrays for formatting and output upon the conclusion of data acquisition.

6.3.28.5 PDL

See Appendix A.

6.3.28.6 Decision Tables and Algorithms

See Appendix C for formatting of individual Guideway Raw Statistics Records.

6.3.29 EREAD06 - Process Type 6 Route Statistics

6.3.29.1 Identification

- o EREAD06 - Process Individual Route and Route Group Statistics
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.29.2 Argument Dictionary

None.

6.3.29.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
CUSAM	-	I*4	Clock units/second
END	-	L*1	Logic Variable for End File
FSTAT1	KMR*25	I*4	Fullword route statistics buffer
HRL	KMR* KML*2	I*2	Halfword route by link statistics buffer
HSTAT1	KMR*18	I*2	Halfword route statistics buffer
IREQ	-	I*4	Request number for statistic
ISUB	-	I*4	Subcategory number for statistic
RTE	-	I*4	Route number for statistic
RTELNK	KMR* KML*2	I*4	Route by link fullword statistics buffer
VAL	-	R*4	Retrieved or computed sample value

6.3.29.4 Description

EREAD06 reads statistics related to individual scheduled service route performance written to the sampling file by the MP during each

sampling interval. This record type is omitted from the Raw Statistics file of a non-Scheduled Service simulation run. Each sampled item, by type, is read into a predefined buffer from which required data items are retrieved and processed. Each required item is identified by the request table entries associated with route related output requests. Sample items are retrieved by cycling through the request entries and retrieving the data required to satisfy the entry as identified by its subcategory index. If the retrieved value is a time related measure, it is converted from internal clock units to seconds. Time average data is converted by dividing by the sampling interval in C.U. and distance related statistics are converted to KM. As each requested item is retrieved or derived from the input data, it is stored in an assigned bin by invoking ESTORE. If standard report 2 is required as an output, individual station summary statistics are computed and stored in special arrays for formatting and output upon the conclusion of data acquisition.

6.3.29.5 PDL

See Appendix A.

6.3.29.6 Decision Tables and Algorithms

See Appendix C for formatting of route related sampling records.

6.3.30 EREQTLU - Record/Request Correlation

6.3.30.1 Identification

- o EREQTLU - Record/Request Correlation
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.20.2 Argument Dictionary

None.

6.3.30.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
L	-	I*4	First available space in bin
NOW	-	I*4	Used to loop through request table chain
SUB	-	I*4	Subcategory 4 character abbreviations
IDNO	-	I*4	Subcategory from request table
IREQ	-	I*4	Request Number (Index to ZREQUE)
ISUB	-	I*4	Subcategory
MAIN	-	I*4	Main category 4 character abbreviation
NEXT	-	I*4	Used to Loop thru request table chain
NREQ	-	I*4	Request number
IFLAG	-	I*4	Unused
IMAIN	-	I*4	Main category number

6.3.30.4 Description - EREQTLU is invoked each time a record of a particular type is encountered in the Raw Statistics File. The Record/Request Correlation process involves cycling through each request table entry. Each time a request requiring the particular record type is encountered, it is chained to the previous request requiring the record type and the major and subcategory indices are converted to numerical values.

6.3.30.5 PDL

See Appendix A.

6.3.30.6 Decision Tables and Algorithms

None.

6.3.31 EREQU - Store an Output Processor Request

6.3.31.1 Identification

- o EREQU - Store a Data Request
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.31.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
FORM	-	I*4	Form of requested output
MAIN	-	I*4	Main category mnemonic
SUB	-	I*4	Subcategory mnemonic
IDNOA	-	I*4	Low index
IDNOB	-	I*4	High index
BINA	-	I*4	Beginning bin number
SIZE	-	I*4	Dummy argument = 0

6.3.31.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
IA	40	I*4	First 40 words of bin area
FORMS	10	I*4	'TSER' 'STAT' & 8 0'S
FMSIZE	10	I*4	100, 11, & 8 0'S
L	-	I*4	Pointer to bin to be processed
LD	-	I*4	Pointer to start of data in bin
BIN	-	I*4	Bin to be used to store data

VARIABLE	DIM	TYPE	DESCRIPTION
IDNO	-	I*4	Station/trip link number
IFORM	-	I*4	Number of form selected
ISIZE	-	I*4	Size of form selected

6.3.31.4 Description

Request processing is invoked by Output Processor Control (EOUTPT) for filing a data request in the request table. Requests are accumulated until a read command is encountered which causes initiation of the data acquisition process.

Request filing begins with creating an entry in the request table by initializing the following data associated with the request:

1. Assignment bin number (next available unused)
2. Initial bin size-assigned based on type of data display required. Initial bin size allocation is made to accommodate data acquisition and manipulation requirements. This allocation serves only as an initial size estimate of the bin area which may be expanded as required during data acquisition.
3. Main category of data (input mnemonic).
4. Subcategory of data (input mnemonic).

In addition, the required bin space allocation to accommodate the acquisition of data is performed (PDL segments EBNCHK and ESHIFT) and three other entries in the request table are initialized:

1. Next available position in the bin.
2. Number of entries remaining in the bin.
3. Request chain pointer (=0).

Requests in the table are only erased after servicing (data acquisition, manipulation, and display).

6.3.31.5 PDL

See Appendix A.

6.3.31.6 Decision Tables and Algorithms

None.

6.3.32 ESETUP - Initialize Data Tables

6.3.32.1 Identification

- o ESETUP - Initialize Data Tables
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.32.2 Argument Dictionary

None.

6.3.32.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
INFINY	-	R*8	Largest I*4 number possible
L	-	I*4	First available space in bin

6.3.32.4 Description

ESETUP reinitializes the match table which is used in establishing record request correlation to a specified initial state as defined and described within the EODATA block data routine. The requested form of each entry in the request table is validated and optionally the tables used by the Output Processor are displayed. ESETUP is invoked during acquisition of the Raw Statistics file header information by EZREAD during initialization.

6.3.32.5 PDL

See Appendix A.

6.3.32.6 Decision Tables and Algorithms

None.

6.3.33 ESHIFT - Shift Data in Bin Area

6.3.33.1 Identification

- o ESHIFT - Reallocate Bin Storage Assignments
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.33.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
J	-	I*4	Starting position for bin
L	-	I*4	Current location of bin
K	-	I*4	End position in bin to be moved

6.3.33.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DIFF	-	L*4	Indicates current position & new position are different
I1	-	I*4	L
I2	-	I*4	K-1

6.3.33.4 Description

ESHIFT is invoked to cycle through a given bin, relocate contents in a new area, and zero old bin entries.

6.3.33.5 PDL

See Appendix A.

6.3.33.6 Decision Tables and Algorithms

None.

6.3.34 ESKIPFO - Skip a Follower Record

6.3.34.1 Identification

- o ESKIPFO - Skip a Follower Record
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.34.2 Argument Dictionary

None.

6.3.34.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
FOLLOW	-	R*8	'FOLLOWER'

6.3.34.4 Description

ESKIPFO is invoked by EZREAD to skip an expected follower record not required for data acquisition. If the expected follower is not found, it issues a warning message. Otherwise, the record is read and control is returned to the calling routine.

6.3.34.5 PDL

See Appendix A.

6.3.34.6 Decision Tables and Algorithms

None.

6.3.35 ESTORE - Store Sampled Data Item in Bin

6.3.35.1 Identification

- o ESTORE - Store Data in Assigned Bin
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.35.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
L	-	I*4	Request number for which data applies
ITEM	-	R*4	Sample data to be stored

6.3.35.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
LL	-	I*4	Bin number

6.3.35.4 Description

ESTORE is invoked by the individual read routines to store a sampled data item in its assigned bin area. The bin area number is retrieved from the request entry corresponding to the sampled data item. The data is stored in the next available position in the assigned bin. If during the store process a bin becomes full, it is automatically reallocated to contain more space (by invoking EABIN). Thus, the file reading process does not require the user to "second guess" how much of each type of data actually resides in the Raw Statistics File. Once storage of a data item has been performed, the bin space pointers contained in the request table entry are updated to reflect bin usage.

6.3.35.5 PDL

See Appendix A.

6.3.35.6 Decision Tables and Algorithms

None.

6.3.35a ETACUM - Accumulate the Raw Statistics

6.3.35a.1 Identification

- o ETACUM - Accumulate the Raw Statistics
- o GM TSC - July 1, 1981
- o PARAFOR

6.3.35a.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ISTN	-	I*4	Origin station index
ITEM	-	R*4	An item to be accumulated
JSTN	-	I*4	Destination station index
MEASUR	MSTN,MSTN,5	R*4	Accumulated statistics
MSTN	-	I*4	Number of stations

6.3.35a.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
COUNT	-	I*4	Constant subscript for MEASUR (2)
MAX	-	I*4	Constant subscript for MEASUR (4)
MIN	-	I*4	Constant subscript for MEASUR (5)
SQUARE	-	I*4	Constant subscript for MEASUR (3)
TOTAL	-	I*4	Constant subscript for MEASUR (1)

6.3.35a.4 Description

Routine ETACUM accumulates the raw statistics for a single station pair. It totals the number of items, their total, their square, and finds their minimum and maximum.

6.3.35a.5 PDL

Contained in code segment.

6.3.35a.6 Decision Tables and Algorithms

None.

6-523a

6.3.35b ETCAPT - Capture the Raw Data

6.3.35b.1 Identification

- o ETCAPT - Capture the Raw Data
- o GM TSC - July 1, 1981
- o PARAFOR

6.3.35b.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DEBUG	-	L*1	Raw debug flag
ETIME	-	R*4	End time for data collection
IMEAS	-	I*4	Index of measure to collect
MEASUR	MSTN,MSTN,5	R*4	Accumulated statistics
MSTN	-	I*4	Maximum number of stations
NSTN	-	I*4	Number of stations
RETCO	-	I*4	Return code
STIME	-	R*4	Start time for data collection
SYSERR	-	I*4	Unit for error messages
TRPLOG	-	I*4	Input unit containing the trip log

6.3.35b.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ACTTIM	-	R*4	Actual travel time
DSPTIM	-	R*4	Dispatch time
ISTAT	-	I*4	Statistics index
ISTN	-	I*4	Origin station index
JSTN	-	I*4	Destination station index
NOMTIM	-	R*4	Nominal travel time

VARIABLE	DIM	TYPE	DESCRIPTION
NOPASS	-	I*4	Number of passengers in a trip
NOXFERS	-	I*4	Number of transfers
ORGTIM	-	R*4	Trip origination time
TRMTIM	-	R*4	Trip termination time
TRVDST	-	R*4	Travel distance
XFRTIM	-	R*4	Total time spent transferring

6.3.35b.4 Description

This routine reads the trip log file to capture the numbers required to compute the requested statistics. Only trip log file records with a termination time between the starting and ending data collection times (inclusive) are used in the calculations. Routine ETACUM is called to accumulate the measures. The actual number of stations is observed from the trip log data and one additional row and column of the station-by-station arrays are set aside for totals.

6.3.35b.5 PDL

Contained in cod segment.

6.3.35b.6 Decision Tables and Algorithms

None.

6.3.35c ETCOMP - Compute the Measure Requested

6.3.35c.1 Identification

- o ETCOMP - Compute the Measure Requested
- o GM TSC - July 1, 1981
- o PARAFOR

6.3.35c.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
COLS	-	I*4	Number of print columns on a page
DEBUG	4	L*2	Debug flags
ETIME	-	R*4	End time for trip log data
FILE	-	L*1	Flag that enables J-file output
IMEAS	-	I*4	Measure type to calculate
KEYS	MMEAS	I*4	Keys for measures in TMPFIL
KEYTMP	-	I*4	Associated variable with TMPFIL
LINES	-	I*4	Number of lines on a print page
MEASUR	MSTN,MSTN,5	R*4	The measures
MSTN	-	I*4	Maximum number of stations
NSTN	-	I*4	Number of stations
REPORT	-	L*1	Flag that enables report output
RETCOD	-	I*4	Return code
STIME	-	R*4	Start time for trip log data
SYSERR	-	I*4	Unit for error messages
SYSVRT	-	I*4	Unit for reports
TMPFIL	-	I*4	Temporary DA unit
TRPLOG	-	I*4	Unit containing the trip log

6.3.35c.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
FMTS	-	I*4	Symbolic subscript for DEBUG (1)
ISTN	-	I*4	Origin station index
MMEAS	-	I*4	Number of measures (7)
MSRSAY	MMEAS	I*4	Statistic to save (by measure)
RAW	-	I*4	Symbolic subscript for DEBUG (3)
SUMS	-	I*4	Symbolic subscript for DEBUG (4)

6.3.35c.4 Description

This routine processes a trip log, computes one of the measures, and outputs the results as requested. Routine ETCAPT is invoked to capture the raw data from the trip log. Then, routine ETTOTL is invoked to calculate row and column totals and routine ETSTAT is invoked to calculate statistics. If the report is requested, routine ETRPTS is invoked to generate it. Finally, if J-file generation is request, the statistics are saved in a temporary file for the merge.

6.3.35c.5 PDL

Contained in code segment.

6.3.35c.6 Decision Tables and Algorithms

None.

6.3.35d ETMEAS - Compute Raw Data for Measure

6.3.35d.1 Identification

- o ETMEAS - Compute Raw Data for Measure
- o GM TSC - July 1, 1981
- o PARAFOR

6.3.35d.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ACTTIM	-	R*4	Actual travel time
DEBUG	-	L*1	Raw debug flag
DSPTIM	-	R*4	Dispatch time
IMEAS	-	I*4	Measure type to calculate
NOPASS	-	I*4	Number of passengers in a trip
NOXFRS	-	I*4	Number of transfers
ORGTIM	-	R*4	Trip origination time
SYSERR	-	I*4	Unit for error messages
TRMTIM	-	R*4	Trip termination time
TRVDST	-	R*4	Travel distance
XFRTIM	-	R*4	Total time spent transferring

6.3.35d.3 Local Variable Dictionary

None.

6.3.35d.4 Description

ETMEAS is a function which returns the desired statistic from the raw data.

6.3.35d.5 PDL

Contained in code segment.

6.3.35d.6 Decision Tables and Algorithms

Calculated derivations:

Initial wait time = Trip termination time -
Trip origination time -
Actual travel time

Total travel time = Trip termination time -
Trip origination time

Passenger travel speed = Travel distance /
Actual travel time

(Note: Passenger travel speed set to zero if actual travel time is zero).

6.3.35e ETMERG - Convert Statistics to UTPS Format

6.3.35e.1 Identification

- o ETMERG - Convert Statistics to UTPS Format
- o GM TSC - July 1, 1981
- o PARAFOR

6.3.35e.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
BUFFER	NSTN	R*4	Buffer for a record of measures
IBUFF	NSTN	I*4	Integer value of buffer array
KEYS	MMEAS	I*4	Keys for TMPFIL
KEYTMP	-	I*4	Associated variable for TMPFIL
MMEAS	-	I*4	Number of valid measure numbers
NSTN	-	I*4	Number of stations
REQST	MMEAS	L*1	Flags to request particular measures
SSPFIL	-	I*4	Unit to contain J-files
SYSERR	-	I*4	Unit for error messages
TMPFIL	-	I*4	Temporary DA unit containing measures

6.3.35e.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
COMMNT	20	I*4	Comment record (mostly EBCDIC)
FMT	-	I*2	Format and table fields
IMEAS	-	I*4	Measure type to calculate
ISTN	-	I*2	Origin station index
JSTN	-	I*2	Destination station index
NSTN1	-	I*4	Number of stations plus 1

6.3.35e.4 Description

This routine converts the statistics to the UTPS (J-file) format and produces the J-file output. First, the header record comment is generated. Then, the saved statistics are read from the temporary file, reordered into UTPS order, reformatted as required, and written to the output device.

6.3.35e.5 PDL

Contained in code segment.

6.3.35e.6 Decision Tables and Algorithms

None

6.3.35f ETNMBR - Convert a Number to EBCDIC

6.3.35f.1 Identification

- o ETNMBR - Convert a Number to EBCDIC
- o GM TSC - July 1, 1981
- o PARAFOR

6.3.35f.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
BINARY	-	I*4	Binary number to be converted
SYSERR	-	I*4	Unit for error messages

6.3.35f.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
GIN	-	I*4	Workspace word
CHAR	4	L*1	Array for EBCDIC
DIGIT	-	I*4	Temporary digit in question
I	-	I*4	Index of characters
NUMBER	9	L*1	Constant with EBCDIC digits
SPACE	-	L*1	EBCDIC blank character
WORD	-	I*4	EBCDIC result (equivalenced with CHAR(1))
ZERO	-	L*1	EBCDIC zero digit
ZEROS	-	L*1	Flag indicating zero suppression

6.3.35f.4 Description

This function converts a binary number to its EBCDIC equivalent and returns the EBCDIC word. The result is I*4 but actually contains four characters. It can also be used as an I*2 routine to yield the low order two digits.

6.3.35f.5 PDL

Contained in code segment.

6.3.35f.6 Decision Tables and Algorithms

None.

6.3.35g ETRPTS - Generate the Station-to-Station Measures Report

6.3.35g.1 Identification

- o ETRPTS - Generate the Station-to-Station Measures Report
- o GM TSC - July 1, 1981
- o PARAFOR

6.3.35g.1 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
COLS	-	I*4	Number of print columns on page
DEBUG	-	L*1	Prints debug flag
IMEAS	-	I*4	Measure to be output
LINES	-	I*4	Number of lines of page
MEASUR	MSTN,MSTN,5	R*4	Measures to be output
MSTN	-	I*4	Maximum number of stations
NSTN	-	I*4	Number of stations
RETCO	-	I*4	Return code
SYSERR	-	I*4	Unit for error messages
SYSVRT	-	I*4	Unit for report printing

6.3.35g.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
BINARY	-	I*4	Temporary binary word
BLANK	-	L*1	The character blank
BLANKS	-	I*4	A 4-character string of blanks
COLHDR	-	I*4	Number of lines for column header
COLUMN	MCOLS	I*4	Column headers (station ID's)
COLWTH	-	I*4	Number of print positions for a column

DATE	4	I*4	EBCDIC date
DIGIT	10	L*1	The characters for the digits 0 to 9
DTAFMT	32	L*1	Data record format
FSTCOL	-	I*4	First station column on this page
FSTROW	-	I*4	First station row on this page
HDRFMT	159	L*1	Format for titles
I	-	I*4	Temporary index for output
ISTN	-	I*4	Origin station index
JSTN	-	I*4	Destination station index
LNGRP	-	I*4	Number of lines for each station row
MCOLS	-	I*4	Maximum number of data columns per page
NOCOLS	-	I*4	Number of station rows on a page
NOROWS	-	I*4	Number of data columns on a page
PAGENO	-	I*4	Current page number
PAGES	-	I*4	Number of pages total
ROWHDR	-	I*4	Number of print positions for row header
ROWID	-	I*4	EBCDIC station number for this row
STATS	9,5	L*1	Statistics titles format
TITLES	27,MNEAS	L*1	Titles for the report by measure

6.3.35g.4 Description

This routine prints the station-to-station measures report as requested and splits the report across pages if required.

6.3.35g.5 PDL

Contained in code segment.

6.3.35g.6 Decision Tables and Algorithms

None.

6.3.35h ETSSPM - Station-to-Station Measures Control

6.3.35h.1 Identification

- o ETSSPM - Station-to-Station Measures Control
- o GM TSC - July 1, 1981
- o PARAFOR

6.3.35h.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
COLS	-	I*4	Number of print columns across page
DEBUG	4	L*1	Debug flags
ETIME	-	R*4	End time for trip log processing
FILE	-	L*1	Output measures to J-files
KEYTMP	-	I*4	Associated variable for TMPFIL
LINES	-	I*4	Number of lines on a printed page
MSPACE	-	I*4	Dimension of space array
MSTN	-	I*4	Maximum number of stations
REPORT	-	L*1	Report flag
REQST	MMEAS	L*1	Request measure computation
RETCO	-	I*4	Error return code
SPACE	MSPACE	I*4	Empty space
SSPFIL	-	I*4	Unit for J-file output
STIME	-	R*4	Start time for trip log processing
SYSERR	-	I*4	Unit for error messages
SYSPT	-	I*4	Unit for printed reports
TMFIL	-	I*4	Temporary DA file
TRPLOG	-	I*4	Unit with trip logs

6.3.35h.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DKEYS	-	I*4	Constant subscript for debug (2)
IMEAS	-	I*4	Index for measures
KEYS	MMEAS	I*4	Beginning of each measures output
MMEAS	-	I*4	Number of measures (7)
NSTN		I*4	Number of stations

6.3.35h.4 Description

This is the main subprogram to control generation of the station-to-station measures report and output file. After initializing the temporary file, the routine invokes routine ETCOMP to compute each requested measure. If the J-file output has been requested, routine ETMERG is invoked to merge the matrices for the temporary file and generate the J-files in UTPS format.

6.3.35h.5 PDL

Contained in code segment.

6.3.35h.6 Decision Tables and Algorithms

None.

6.3.351 ETSTAT - Compute Statistics

6.3.351.1 Identification

- o ETSTAT - Compute Statistics
- o GM TSC - July 1, 1981
- o PARAFOR

6.3.351.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
MEASUR	MSTN,MSTN,5	R*4	Accumulated statistics
MSTN	-	I*4	Maximum number of stations
NSTN	-	I*4	Number of stations

6.3.351.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
AVERAG	-	R*4	Average value for the station pair
COUNT	-	I*4	Symbolic subscript for MEASUR (2)
ISTN	-	I*4	Origin station index
JSTN	-	I*4	Destination station index
MEAN	-	I*4	Symbolic subscript for MEASUR (2)
SD	-	I*4	Symbolic subscript for MEASUR (3)
SQUARE	-	I*4	Symbolic subscript for MEASUR (3)
TOTAL	-	I*4	Symbolic subscript for MEASUR (1)
VARIAN	-	R*4	Variance for the station pair

6.3.351.4 Description

This routine computes the mean and standard deviation for each station-to-station pair in the measures array.

6.3.351.5 PDL

Contained in code segment.

6.3.351.6 Decision Tables and Algorithms

None.

6.3.35j ETSTID - Convert Station ID to EBCDIC

6.3.35j.1 Identification

- o ETSTID - Convert Station ID to EBCDIC
- o GM TSC - July 1, 1981
- o PARAFOR

6.3.35j.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
JSTN	-	I*4	Destination station index
NSTN	-	I*4	Number of stations
SYSERR	-	I*4	Unit for error messages

6.3.35j.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
ALL	-	I*4	The character string "ALL" (right justified)

6.3.35j.4 Description

This function converts a station ID to EBCDIC for display. If the station number is equivalent to the total number of stations then the character string "ALL" is returned rather than the station number in EBCDIC.

6.3.35j.5 PDL

Contained in code segment.

6.3.35j.6 Decision Tables and Algorithms

None.

6.3.35k ETTOTL - Total the Raw Statistics

6.3.35k.1 Identification

- o ETTOTL - Total the Raw Statistics
- o GM TSC - July 1, 1981
- o PARAFOR

6.3.35k.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
MEASUR	MSTN,MSTN,5	R*4	Accumulated statistics
MSTN	-	I*4	Maximum number of stations
NSTN	-	I*4	Number of stations

6.3.35k.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
COUNT	-	I*4	Symbolic subscript for MEASUR (2)
ISTAT	-	I*4	Statistic index
ISTN	-	I*4	Origin station index
JSTN	-	I*4	Destination station index
MAX	-	I*4	Symbolic subscript for MEASUR (4)
MIN	-	I*4	Symbolic subscript for MEASUR (5)
SQUARE	-	I*4	Symbolic subscript for MEASUR (3)
TOTAL	-	I*4	Symbolic subscript for MEASUR (1)

6.3.35k.4 Description

This routine sums (or computes maximum and minimum of) the accumulated statistics by rows over all stations and by columns over all stations.

6.3.35k.5 PDL

Contained in code segment.

6.3.35k.6 Decision Tables and Algorithms

None.

6.3.36 EZHIST - Histogram Output Control

6.3.36.1 Identification

- o EZHIST - Histogram Output Control
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.36.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
IQA	-	I*4	Bin number
ZA	-	R*4	Class interval width

6.3.36.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
DMY	5	R*4	Used like bin in call to EMMX; first 2 positions unused; 3=min; 4=max; 5=range
COMMON OUTPUT			
NBIN	10	I*4	Histogram data
PAR	7	R*4	PAR(1)=class interval width
IPAR	7	I*4	Unused
COMMON HISTO			
MIN	-	I*4	Unused
MAX	-	I*4	Unused
AMAX	-	I*4	Largest value in histogram
AMIN	-	I*4	Smallest value in histogram
ASLOT	-	I*4	Number of slots in histogram

6.3.36.4 Description

EZHIST is invoked by EOUTPT to produce a histogram of sampled items contained in a bin storage area. EMNMX is invoked to select the minimum and maximum values of the sampled items to be output. A work bin to be used in accumulating class interval frequencies is then allocated by invoking EBNCHK. The actual formatting and display of the histogram is performed by EHIST.

6.3.36.5 PDL

See Appendix A.

6.3.36.6 Decision Tables and Algorithms

None.

6.3.37 EZLIST - List Output Control

6.3.37.1 Identification

- o EZLIST - List Output Control
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.37.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
IQA	-	I*4	Bin number
IZA	-	I*4	K (KTH element listing indicator)

6.3.37.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
COMMON OUTPUT			
NBIN	10	I*4	NBIN(1)=Number of bin to be listed
PAR	7	R*4	Unused
IPAR	7	I*4	IPAR(1)=K (KTH element listing indicator)

6.3.37.4 Description

EZLIST is invoked by EOUPT to produce either a listing of sampled data items or a statistical summary. Upon entry, the routine establishes the default list option to cause output of each sampled item and ELIST is invoked to format and display the data as required.

6.3.37.5 PDL

See Appendix A.

6.3.37.6 Decision Tables and Algorithms

None.

6.3.38 EZPLOT - Plot Output Control

6.3.38.1 Identification

- o EZPLOT - Plot Output Control
- o IBM/FSD - May 1, 1978 (modified GM TSC - July 1, 1981)
- o PARAFOR

6.3.38.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
X0	-	R*4	Starting X value
DELTA X	-	R*4	X increment
NDELTA	-	I*4	# points to be plotted
NY	-	I*4	# bins to be plotted
N1	-	I*4	Bin #1
N2	-	I*4	Bin #2
N3	-	I*4	Bin #3
N4	-	I*4	Bin #4
BOTTOM	-	R*4	Lower limit on Y values
TOP	-	R*4	Upper limit on Y values
SYMBOL	5	R*4	Plotting symbols

6.3.38.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TYPE	17	I*4	4 character designation of SL types
J1	-	I*4	# of first bin to be plotted
J2	-	I*4	Unused
J3	-	I*4	Unused

VARIABLE	DIM	TYPE	DESCRIPTION
J4	-	I*4	Unused
TOP	-	R*4	Largest value to be plotted
NDIS	-	I*4	Pointer to 16 character title
INDEX	-	I*4	Station/trip link number
BOTTOM	-	I*4	Smallest value to be plotted
NDELTA	-	I*4	Number of values to be plotted

6.3.38.4 Description

EZPLOT is invoked by EOUTPT to provide a time series plot of sampled data items. The actual data accumulation, scaling, and formatting is performed by invoking EGRAPH. It formats the required output by manipulating the contents of a bin and outputting the desired results. Format processing includes establishing necessary grids and titles, and establishing a scaling factor applied to data for accommodating the image size on the output medium (page size).

6.3.38.5 PDL

See Appendix A.

6.3.38.6 Decision Tables and Algorithms

None.

6.3.39 EZREAD - Data Acquisition

6.3.39.1 Identification

- o EZREAD - Read Raw Statistics
- o IBM/FSD - May 1, 1978
- o PARAFOR

6.3.39.2 Argument Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
TAPE	-	I*4	FORTTRAN unit number for Raw Statistics Files
START	-	I*4	Begin time of acquisition interval
STOP	-	I*4	Stop time of acquisition interval

6.3.39.3 Local Variable Dictionary

VARIABLE	DIM	TYPE	DESCRIPTION
JUNK	-	I*4	Dummy argument used to call header

6.3.39.4 Description

EZREAD reads the Raw Statistics File to acquire data items from samples within a start/stop interval as required to service previous data requests. The data acquisition process is initiated by EOUTPT in response to a Read Command. The data acquisition process is partitioned into three functions:

1. Obtain initial, critical data from the tape and perform various other initializations.
2. Skip to the beginning of the request interval.
3. Read groups of records from the tape, ascertain whether a group has requested data within it, obtain the requested data, and store it into the appropriate bin.

Since data acquisition serves to obtain system default parameters during initialization, a check is made to determine if this is the initial read of the Raw Statistics File. If it is the initial read, the default parameters are read from the initial file header. The time units specified for the simulation experiment acquired during this processing are used in subsequent data acquisition processing as described below. Entry into the data acquisition process for satisfying data requests begins with data table initialization performed by invoking ESETUP and establishing request/record correlation via CREQTLU, conversion of the request interval to simulator clock units and repositioning of the Raw Statistics File at its beginning. The Raw Statistics File is read, processing each header record by calling EHEADER and skipping successive records via ESKIPFO until the file is positioned to the start of the read (acquisition) interval. Basically, in this process, record groups are read and their followers are skipped until one is found whose time is not less than the request interval start time.

Once the file is positioned to the beginning of the read interval, subsequent records are read and one of three actions is taken based upon the initial setting of the major category indicator and summarized below:

1. 0 -- Meaning that records of type 0 are not needed, the follower records are skipped, and the next header is read.
2. 1 -- Meaning that following records of 1 are needed processing for acquiring and storing data is performed.
3. -1 -- Meaning that the records might be needed, but whether or not they are has yet to be determined. At this time, the program must determine if they are or are not needed by invoking the data matching function previously described. The result of determining whether this record type is required results in changing the major category indicator to 0, indicating the first request requiring data from the record type.

Actual data acquisition from required record types is performed by I/O processing by invoking read routines based on individual record type for the major data category as indicated in the record group header. This processing iterates upon each of the follower records in turn and then upon each of the requests in the request table associated with the particular record type (as defined by the chain beginning with the major category indicator).

If the main category is one that requires no entity index number (e.g., as for system as opposed to guideway link, which does), then only one follower record exists and it contains a single set of data items.

However, if the main category can have an associated entity number (e.g., a particular guideway link number for the link category), the each follower contains several replications of data items, one each for several entity indices. In this case, I/O processing (besides iterating on the followers and request lines) must also iterate upon the number of data item replications in a particular follower record.

For each iteration, the required appropriate read routine for the specific record type is called to store the data for processing as contained in a follower record. Each required data item is located within the record, retrieved, and stored in the appropriate bin area. In general, data position information is determined from the major category and sub-category indices contained in each request table entry as the result of performing the data mapping function.

6.3.39.5 PDL

See Appendix A.

6.3.39.6 Decision Tables and Algorithms

None.

6.3.40 XPSEUDO - I/O Intercept Routine

See subsection 6.1.42, XPSEUDO.

6.3. 41 XTRACBK - Interrupt Handler

See subsection 6.1.43, XTRACBK.

6.3.42 XTRCBKP - Save Area Trace Formatting

See subsection 6.1.44, XTRCBKP.

APPENDIX A
REPORT OF NEW TECHNOLOGY

The Discrete Event Simulation Model (DESM) provides the capability to model the operations of a guideway transit system operating over a network composed of guideway links and stations within a given time domain. The DESM provides, for the first time, a tool for the transit analyst to evaluate the performance of a wide range of configurations and operational strategies of guideway transit systems to the detail of individual vehicles and passengers.

110 copies

A-1/A-2