LOAN COPY

EPORT NO. DOT-TSC-OST-73-16 A, IVB

ANALYSIS OF DUAL MODE SYSTEMS IN AN URBAN AREA

Volume IVB: Program Documentation of the Transportation Economic Analysis Model (continued)

Peter Benjamin et al.

DECEMBER 1973

FINAL REPORT

Document is available to the public through the
National Technical Information Service,
Springfield, Virginia 22151

Prepared for

DEPARTMENT OF TRANSPORTATION

OFFICE OF THE SECRETARY

Office of Systems Engineering

Washington DC 20590

Technical Report Documentation Page

1. Report No. DOT-TSC-OST-73-16A, IVB	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle ANALYSIS OF DUAL MODE SYSTEMS IN AN URBAN AREA VOLUME IVB (CONTINUED)		5. Report Date April 1973 Revised December 1973
		6. Performing Organization Code
7. Author(s) Peter Benjamin et al*		8. Performing Organization Report No. DOT-TSC-OST-73-16A, IVB
9. Performing Organization Name and Address Department of Transportation Transportation Systems Center Kendall Square Cambridge MA 02142		10. Work Unit No. (TRAIS) OS418/R4533
		11. Contract or Grant No.
12. Sponsoring Agency Name and Address Department of Transportation Office of the Secretary Office of Systems Engineering Washington DC 20590		13. Type of Report and Period Covered Final Report August 1971-August 1972
		14. Sponsoring Agency Code

15. Supplementary Notes

*J. Barber, R. Favout, D. Goeddel, C. Heaton, R. Kangas
G. Paules, E. Roberts, L. Vance

16. Abstract

Various forms of Dual Mode transportation were analyzed in order to assess the economic viability of the Dual Mode concept. A Dual Mode vehicle is one which operates under manual control on a street network for some portion of its trip, and operates under automatic control on an exclusive guideway for some other portion. Specially designed new small Dual Mode vehicles, modifications of existing automobiles, and pallet systems, all operating in conjunction with Dual Mode buses, were examined. The study was conducted in a Boston 1990 scenario, in which an extensive Dual Mode system providing service for the entire urban region was presumed to exist. This study was not intended to be a proposal for Dual Mode in Boston. The following conclusions are considered to be generally applicable to other large urban areas as well: (a) Dual Mode systems appear to be sufficiently attractive to warrant further technological development; (b) for urban-wide applications, a Dual Mode system which includes both buses and personal vehicles is more effective than one consisting of either fleet of vehicles alone; (c) a Dual Mode transportation system benefits from the use of various Dual Mode concepts throughout its development.

An effective first step might be to install a limited network Dual Mode minibus system, with capacity for ultimate growth to a longer guideway network with personal vehicles and buses.

17. Key Words Urban Transportation Systems Dual Mode Systems		18. Distribution Statement DOCUMENT IS AVAILABLE TO THE PUBLIC THROUGH THE NATIONAL TECHNICAL INFORMATION SERVICE, SPRINGFIELD, VIRGINIA 22151.	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 248	22. Price

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

5. Highway-Transit Cost Program

As described earlier, this program performs all cost and impact calculations for highways and transit systems. In the Dual Mode analysis, this program was used to quantify all capital and operating costs and impacts of the null alternative (the 1990 transportation plan).

A Capital Costs

Figure 5-1 identifies all capital costs computed within this program. For highways, the program quantifies capital costs for way land acquisition and way construction. Cost data for freeways and surface arterials are generated and identified with respect to three land cost areas and four types of construction. These costs are computed using as input units of land and construction and corresponding unit cost data.

Transit capital costs are calculated and reported with respect to individual transit systems (a maximum of four) being analyzed. Costs for way, vehicles, terminals, storage and maintenance facilities are quantified by this program. Land acquisition costs are computed using as input units of land area required and land unit cost data. Construction costs are calculated based upon the number of facilities required and the type of facility construction.

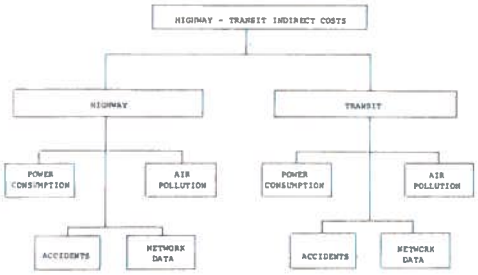


FIGURE 5-3 Indirect Costs Computed by Highway-Transit Cost Program

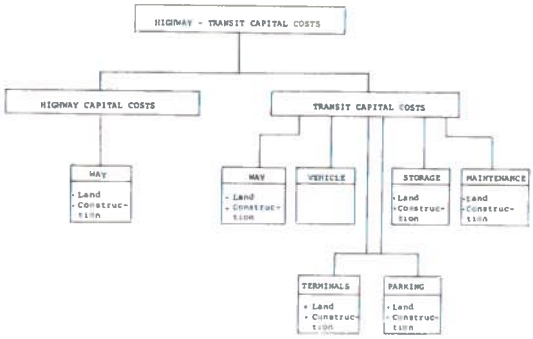


FIGURE 5-1 Capital Costs Computed by the Highway-Transit Cost Program

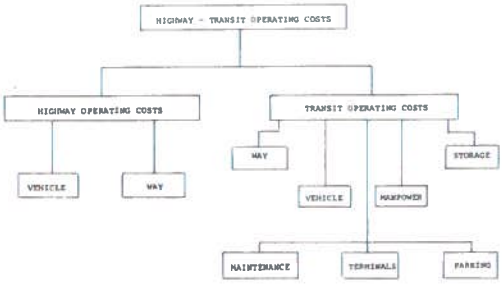


FIGURE 5-2 Operating Costs Computed by Highway-Transit Cost Program

Each of the highway and transit capital costs identified above are annualized by the program. Annual costs are determined utilizing capital recovery factors computed for each highway and transit capital element.

B. Operating Costs

All operating costs computed by this program are identified in Figure 5-2.

For freeways and surface arterials, the program calculates way operating costs for power, manpower, and maintenance as a function of the annual vehicle miles travelled. Auto and truck vehicle operating costs are also computed based upon the amount of vehicle travel.

For transit systems, annual operating costs are computed for:

- Way
- Vehicles
- Storage facilities
- Maintenance facilities
- Terminals
- Parking facilities

Operating costs for way power are computed based upon the annual transit VMT; way manpower and maintenance costs are determined based upon the number of route (or track) miles that exist for each transit system. Vehicle operating costs and transit manpower costs are calculated based upon the annual vehicle miles traveled by each system. Annual operating costs for transit storage, maintenance, and parking facilities are computed on the basis of the number of vehicle spaces required. Terminal operating costs are calculated as a function of the number of terminals that exist for each transit system.

C. Impacts and Network Analysis Data

Figure 5-3 shows that in addition to network analysis data, the following impacts are quantified by this program.

- Power consumption
- Air pollution
- Accidents

For highway and transit vehicle systems, network analysis data, i.e., person-miles traveled, vehicle miles traveled, average speeds, average trip lengths and trip times, etc. are computed and reported with respect to four temporal periods (peak, off-peak, daily and annual).

The program also quantifies power consumption and air pollution levels for highway and transit vehicle systems. This data is generated as a function of the vehicle miles traveled by each of these systems. Data is stratified with respect to four temporal periods and for air pollution with respect to five pollutant types.

Highway and transit accidents are calculated for a number of accident types - vehicle accidents, property damage accidents, personal fatalities and injuries. Using as input accident rate data, the program computes the number of accidents as a function of the amount of system travel (VMT and person trips).

5.1 Program Structure

The Highway-Transit Cost Program is comprised of a main control program, NULCASE, and twenty-two subroutines.

The program was structured such that the highway capital cost, the transit capital cost, and the operating-indirect cost sections of the program are independent of each other. Because of this type of program structure, each or any combination of these three cost sections can be independently executed. The user controls the execution of each of these three sections by means of three input parameters.

Another feature of the program is the ability to control the printing of all intermediate output data tables. Through input parameters, the program can be controlled by the user to print only summary data output tables.

Figure 5-4 describes the overall structure of the Highway-Transit Cost Program. A definition of the functions performed by each of the subroutines of this program is presented in Table 5-1.

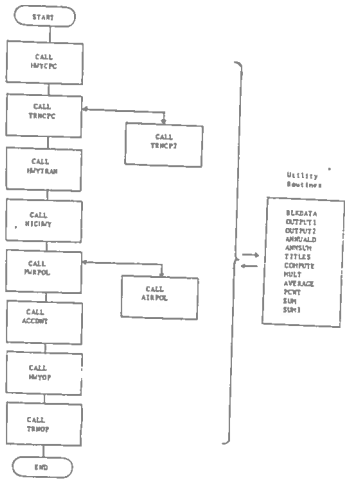


FIGURE 5-6 Highway-Transit Cost Program

Program/Subroutine	Function
MULCASE	Control program - calls all computational subroutines
HWYCPC	Computes all highway capital costs
TRNCPC	Computes all transit capital costs for way, vehicles and storage facilities
TRNCP2	Computes all transit capital costs for maintenance, terminals and parking facilities
BLKDATA	Stores highway and transit capital cost data names for output tables
OUTPUT1	Prints highway and transit capital cost output data
OUTPUT2	Prints highway and transit capital cost output data
ANNUALD	Computes and prints highway and transit annual cost data
ANNSUM	Computes and prints highway and transit annual cost summary data
TITLES	Prints titles of highway and transit capital cost output tables
COMPUTE	Computes highway and transit capital costs, average cost per unit, and cost percentages thru ugh the use of the MULT,SUM,AVERAGE, and PCNT subroutines

TABLE 5-1 Highway-Transit Cost Program

Program/Subroutine	Function
MULT	Multiplies a three-dimensional unit and a three-dimensional unit cost array to produce a cost product array
AVERAGE	Computes an average cost per unit for each highway and transit capital cost element
PCNT	Computes percentages of highway and transit capital costs
SUM	Sums all row, column and depth elements of an array that is dimensioned (5,6,12)
HWYTRAN	Computes all network data for the various transit systems
HIGHWY	Computes all highway vehicle network data
PWRPOL	Computes power consumption data for the highway and transit vehicle systems
AIRPOL	Computes highway and transit air pollution data
ACCDNT	Calculates highway and transit accident data
HWYOP	Computes all highway operating costs
TRNOP	Computes all transit operating costs
SUM3	Sums all row, column, and depth elements of an array that is dimensioned (5,8,5).

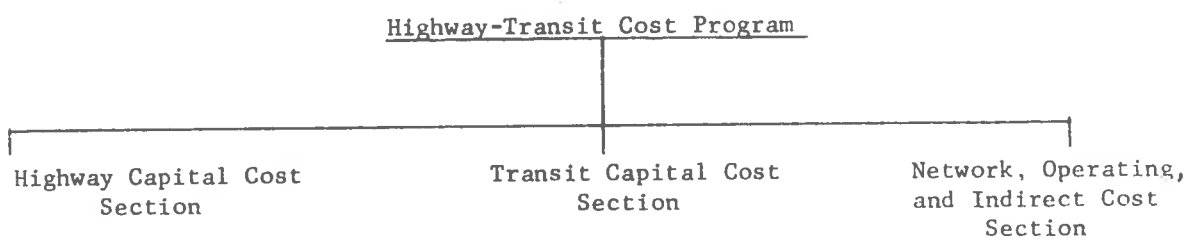
TABLE 5-1 Highway-Transit Cost Program (continued)

5.1.1 Program NULCASE

NULCASE is the control program of the Highway-Transit Cost Program. Its primary function is to call in sequence the following computational subroutines:

- o HWYCPC - Computes all capital costs related to highways and new highway construction.
- o TRNCPC - Computes transit capital costs for way, vehicles and storage facilities. TRNCPC, in turn, calls TRNCP2 for the calculation of transit maintenance, terminals, and parking facility capital costs.
- o HWYTRAN- Computes all transit network analysis data.
- o HIGHWY - Calculates all highway network analysis data.
- o PWRPOL - Computes power consumption and air pollution emission data for the highway and transit vehicle systems.
- o ACCDNT - Calculates highway and transit accident data.
- o HWYOP - Computes all highway operating costs
- o TRNOP - Computes all transit operating costs.

The Highway-Transit Cost program is comprised of three major computational sections as illustrated below:



Each of these sections is capable of being executed independent of each other; the user controls the execution of each of these sections by means of the NULLH, NULLX, and NULOTH variables. Whenever a section is not to be executed, the input data for that section is required not to appear in the input data deck.

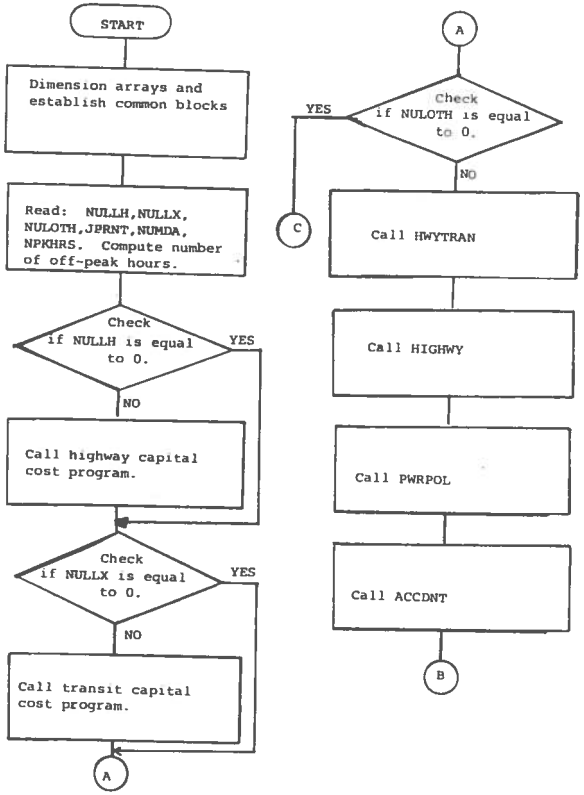


FIGURE 5-5 General Flow Chart - Fortran IV Program NULCASE

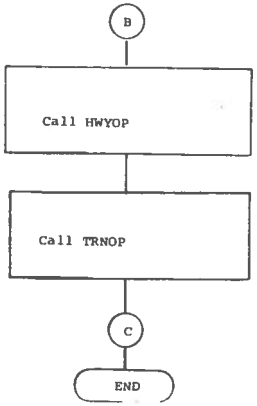


FIGURE 5-5 General Flow Chart - Fortran IV Program NULCASE (continued)

5.1.2 Subroutine HWYCPC

Subroutine HWYCPC computes all capital and annual costs associated with new highway construction. Costs for way land acquisition and way construction are stratified by types of land area, by construction type for both freeways and surface arterials.

Initially the subroutine reads alpha numeric title data as defined on input card format types 3 through 8. This data, stored within the ANNHD, SYS, NAME, and TITLE arrays, are the names of various land types, construction types, etc. for the output data tables. Life and interest data for way land and way construction are read by HWYCPC and are defined respectively, by the variables LIF and INT.

The first data calculation performed by HWYCPC is the right-of-way capital cost for land acquisition. Subroutine COMPUTE performs this calculation utilizing a unit and a unit-cost array passed to it by HWYCPC.

In this case, the unit array (HROWM) contains the amount of land required for right-of-way while the HROWUD array contains the land unit cost data. The product array ITEMS, generated by subroutine COMPUTE, contains the land acquisition capital cost data.

ITEMS(KI,KJ,KM) = HROWM(KI,KJ,KM) * HROWUD(KI,KJ,KM)

(land capital cost - $)=(land units - miles) (land unit cost - $/mile)

where: KI-land area type
KJ- construction type
KM-freeways or surface arterials

Besides computing the ITEMS array, subroutine COMPUTE calls subroutine SUM, PCNT and AVERAGE for the summuation of the unit and cost data arrays, for the calculation unit and cost percentage data, and average cost per unit data. Once this data is computed, control is returned back to HWYCPC where repeated

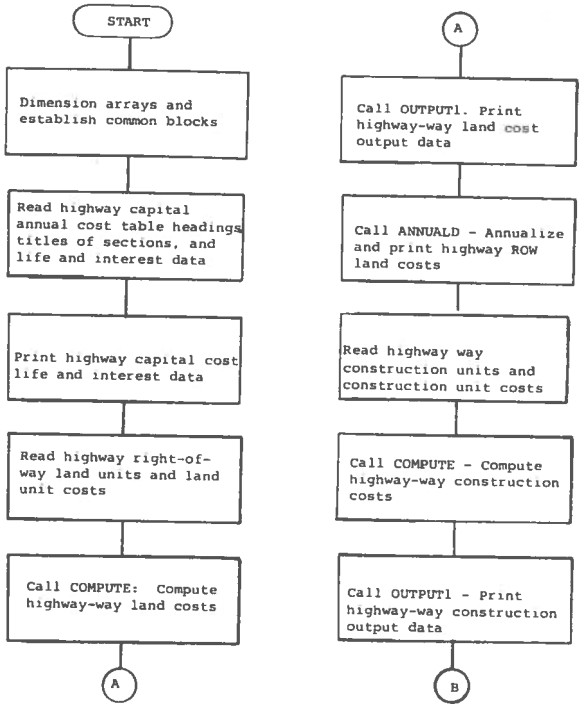


FIGURE 5-6 General Flow Chart - Subroutine HWYCPC

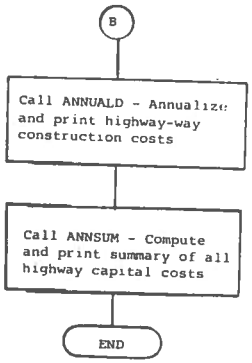


FIGURE 5-6 General Flow Chart - Subroutine HWYCPC

calls are then made to the OUTPUT1 subroutine for the printing of the land acquisition capital cost data. This data is only printed provided the user input variable JPRNT is not equal to an integer 1. HWYCPC then calls subroutine ANNUALD which computes and prints way land acquisition annual cost data.

At this point, HWYCPC begins the capital cost calculations for way construction. Unit and unit cost data are read by this subroutine and stored, respectively, within the HCONM and HCONUD arrays. Subroutine COMPUTE is then called to compute the way construction capital cost data as follows:

ITEMS(KI,KJ,KM)= HCONM(KI,KJ,KM)* HCONUD(KI,KJ,KM)

(way construction capital cost -$) = (way construction units-miles) X (construction unit cost-$/mile)

where: KI-land area type
KJ-construction type
KM-freeways, surgace arterials

Unit and cost summation data, percentage data, and average cost per unit data are also computed by subroutine COMPUTE through the use of the SUM, PCNT, and AVERAGE routines, respectively.

Upon receiving control from COMPUTE, subroutine HWYCPC then calls subroutine OUTPUT1 for the printing of way construction unit and capital cost data. ANNUALD is then called for the calculation and the printing of way construction annual cost data.

An output table summarizing all annual costs for way land acquisition and way construction is then printed by subroutine ANNSUM. This subroutine is called by HWYCPC.

At this point control is then returned to the main control program NULCASE for the calling of the next subroutine TRNCPC.

5.1.3 Subroutine TRNCPC

The primary function of this routine and subroutine TRNCP2 is to calculate and print all capital costs of transit systems. This routine calculates all costs associated with transit way land acquisition and construction, vehicle purchase, and transit storage facilities. TRNCP2, which is called by this routine, calculates all costs of transit terminal, maintenance, and parking facilities.

Initially, subroutine TRNCPC reads all life and interest data for each transit capital cost area identified in Table 5-2 . This data is stored in the LIF and INT variable arrays. Alphanumeric title data defined within card format types 14 through 18 are then read. This data is later used within the program in the printing of the capital cost output data tables.

The capital cost of land acquisition for new way construction is first computed by this routine. Subroutine COMPUTE is called to perform this data calculation utilizing unit and unit cost data arrays input by the user. The capital cost of way land acquisition is computed and stored within an array, named ITEMS, as follows:

```
ITEMS(KI,KJ,KM) = XROWM(KI,KJ,KM) * XROWUD(KI,KJ,KM)
```

(Way land capital cost - $) = (land units - miles) (land unit cost - $/mile)

where: KI - land area type
KJ - construction type
KM - transit system

Summations of total miles and costs of land acquisition, percentage data, and average cost per unit data are also computed by COMPUTE through the use of the SUM, PCNT, and AVERAGE subroutines. Once this data has been computed, control is then returned to TRNCPC for the printing of the data. Subroutine OUTPUT1, called by TRNCPC, performs this task.

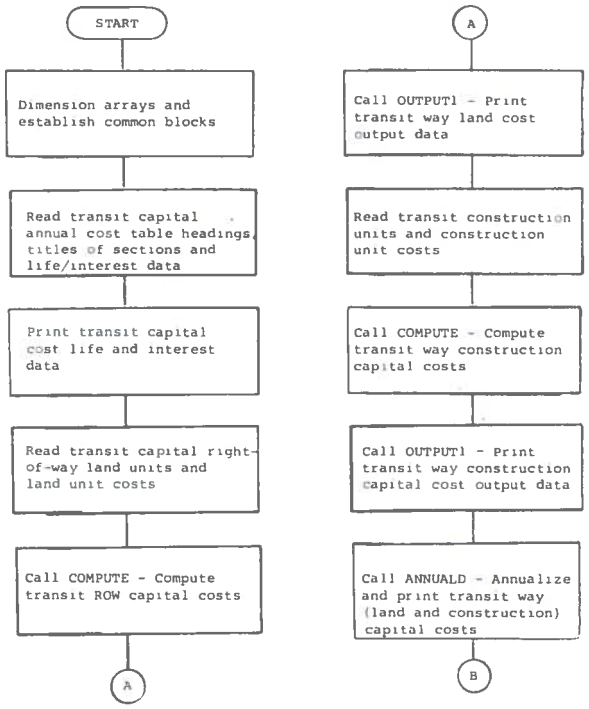


FIGURE 5-7 General Flow Chart - Subroutine TRNCPC

FIGURE 5-7 General Flow Chart - Subroutine TRNCPC (continued)

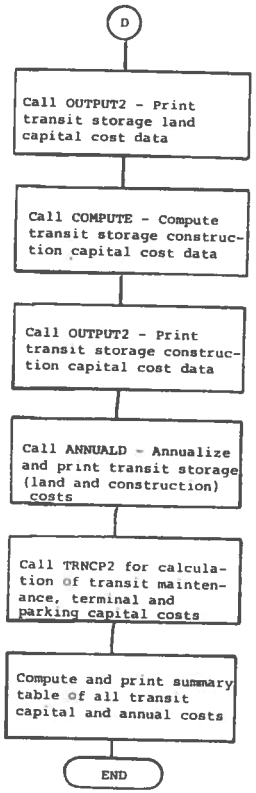


FIGURE5-7 General Flow Chart - Subroutine TRNCPC (continued)

IZ	Life/Interest Data	IZ	Life/Interest Data
1	Transit 1 Way - Land	26	Transit 3 Maint. Construction
2	Transit 1 Way - Construction	27	Transit 4 Maint. Land
3	Transit 2 Way - Land	28	Transit 4 Maint. Construction
4	Transit 2 Way - Construction	29	Transit 1 Terminal Land
5	Transit 3 Way - Land	30	Transit 1 Terminal Construction
6	Transit 3 Way - Construction	31	Transit 2 Terminal Land
7	Transit 4 Way - Land	32	Transit 2 Terminal Construction
8	Transit 4 Way - Construction	33	Transit 3 Terminal Land
9	Transit 1 Vehicles	34	Transit 3 Terminal Construction
10	Transit 2 Vehicles	35	Transit 4 Terminal Land
11	Transit 3 Vehicles	36	Transit 4 Terminal Construction
12	Transit 4 Vehicles	37	Transit 1 Parking Land
13	Transit 1 Storage Land	38	Transit 1 Parking Construction
14	Transit 1 Storage Construction	39	Transit 2 Parking Land
15	Transit 2 Storage Land	40	Transit 2 Parking Construction
16	Transit 2 Storage Construction	41	Transit 3 Parking Land
17	Transit 3 Storage Land	42	Transit 3 Parking Construction
18	Transit 3 Storage Construction	43	Transit 4 Parking Land
19	Transit 4 Storage Land	44	Transit 4 Parking Construction
20	Transit 4 Storage Construction	45	Not Used
21	Transit 1 Maint. Land	46	Not Used
22	Transit 1 Maint. Construction	47	Not Used
23	Transit 2 Maint. Land	48	Not Used
24	Transit 2 Maint. Construction	49	Not Used
25	Transit 3 Maint. Land	50	Not Used

TABLE 5-2 Transit Capital Cost Life/Interest Data

Way construction capital costs are then computed for each transit system. TRNCPC reads way construction unit and unit cost data and then calls subroutine COMPUTE to perform the cost calculation as follows:

```
ITEMS(KI,KJ,KM) = XCONM(KI,KJ,KM) * XCONUD(KI,KJ,KM)
(Way construction capital cost - $) = (construction units - miles)
                                      (construction unit cost - $/mile)
```

where: KI - land area type
KJ - construction type
KM - transit system

Upon calculation of this data along with percentage and cost per unit data, subroutine COMPUTE returns control to TRNCPC for the printing of the data. All data output is handled by repeated calls to the OUTPUT1 subroutine. Once this data is printed subroutine ANNUALD is then called for the calculation and printing of all way (land and construction) annual costs.

Capital costs associated with the purchase of new transit vehicles are then computed for each transit system (KM). Subroutine COMPUTE performs these cost calculations by multiplying the number of vehicles purchased (input by the user within the NXVEH data array) by average cost per vehicle (XVEUD). The number of vehicles, average cost per vehicle, and total capital cost data are then output for each transit system by the OUTPUT2 subroutine. Annual costs for transit vehicles are then computed and printed by a call to subroutine ANNUALD.

Capital costs for the land acquisition and the construction of transit storage facilities are then computed at this point in the routine. These costs are computed based upon the number of storage vehicle spaces required (input by user in the XSTVSP data array). The amount of land area required for storage is computed with respect to various land areas (KI) and with respect to individual transit systems (KM) as follows:

```
XSLSQF(KI,KJ,KM) = XSTVSP(KI,KJ,KM) * XSLSQF(KI,KJ,KM)
(storage land area - sq.ft.) = (storage vehicle spaces) (vehicle space area -
                                sq.ft./vehicle space)
```

XSLACR(KI,KJ,KM) = XSLSQF(KI,KJ,KM)* ACRSQF
(storage land area - acres) = (storage land area - sq. ft.) (acres/sq. ft.)

Once the land area is computed, subroutine COMPUTE is called for the calculation of land capital costs, unit and cost percentage data, and average cost per unit data. The cost of land acquisition is computed by multipling the land area required by appropriate land unit cost data (input by the user):

ITEMS(KI,KJ,KM) = XSLACR(KI,KJ,KM) * XSTLUD(KI,KJ,KM)

(storage land cost - $) = (storage land area - acres) (land unit cost - $/acre)

Subroutine OUTPUT2 is then called for the printing of this data.

Likewise, the capital costs incurred by the construction of new storage facilities are computed, by subroutine COMPUTE as follows:

ITEMS(KI,KJ,KM) = XSTVSP(KI,KJ,KM) * XSTCUD(KI,KJ,KM)

(storage construction cost-$) = (storage vehicle spaces)(construction unit cost- $/vehicle space)

Having computed this data, COMPUTE returns control to TRNCPC which in turn, calls subroutine OUTPUT2 for the output of the storage construction capital cost data. The ANNUALD subroutine is then called for the annualization of all storage land and storage construction capital costs.

At this point, subroutine TRNCP2 is called to calculate capital and annual costs associated with transit terminal, maintenance, and parking facilities. Whenever control is returned to TRNCPC from TRNCP2 an output table summarizing all capital and annual costs (computed in the TRNCPC and TRNCP2 routines) is printed for each transit system.

This completes all processing performed by TRNCPC, control is then returned to the main program NULCASE for the calling of the next subroutine HWYTRAN.

5.1.4 Subroutine TRNCP2

The primary function of this routine is to compute and to output all land acquisition and construction capital costs for transit maintenance, terminal, and parking facilities. These cost calculations are performed as illustrated in Figure 5-8.

Initially, the routine computes all capital costs associated with transit maintenance facilities. The number of maintenance vehicle spaces required, land unit costs, and construction unit cost data are read by TRNCP2 and stored, respectively, within arrays XMAVSP, XMALUD, and XMACUD. The routine then proceeds to calculate the total area required for maintenance facilities in units of square feet (XMASQF) and acres (XMAACR). Land capital costs are then computed by subroutine COMPUTE by multiplying the total acres required by the land unit cost data:

ITEMS (KI, KJ, KM) = XMAACR (KI, KJ, KM) * XMALUD (KI, KJ, KM)

(Land capital cost - $) = (Land area - acres) (Land unit cost - $/acre).

Subroutine Output 2 is then called for the output of all land acquisition capital costs for transit maintenance facilities.

Transit maintenance construction capital costs are computed by multiplying the maintenance facility area (in square feet) by the construction unit cost data.

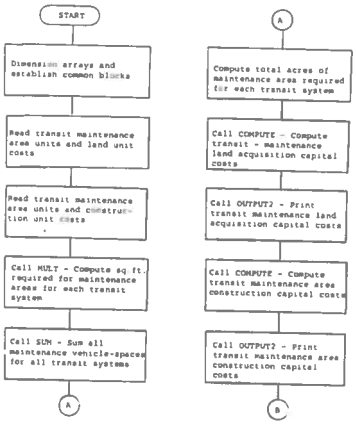


FIGURE 5-8 General Flow Chart - Subroutine TRNCP2

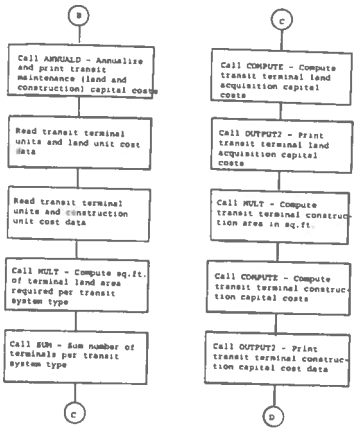


FIGURE 5-8 General Flow Chart - Subroutine TRNCP2 (continued)

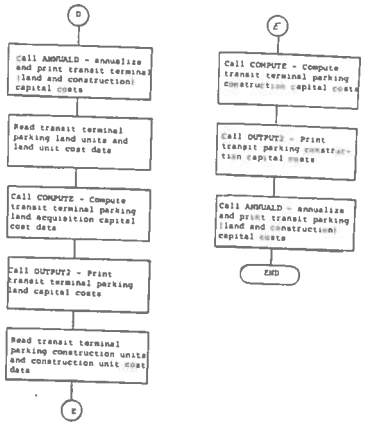


FIGURE 5-8 General Flow Chart - Subroutine TRNCP2 (continued)

ITEMS (KI, KJ, KM) = XMASAF (KI, KJ, KM) * XMACUD (KI, KJ, KM)

(Construction capital cost - $) = (Maintenance area - sq. ft.)

(Construction unit cost - $/sq. ft.)

Once this data is computed, subroutine OUTPUT2 is again called for the output of all construction capital cost data. TRNCP2 then calls subroutine ANNUALD for the annualization of all land and construction capital costs for transit maintenance facilities. An output table containing these annualized costs is also printed at this time by ANNUALD.

At this point, subroutine TRNCP2 begins the data calculations of transit terminal land and construction capital costs. The number of transit terminals, land unit costs, and terminal construction unit costs are read as input data by this routine. This data is stored, respectively, within variable arrays NXTER, XTLUD, and XTCUD. The total land area required for transit terminals is computed by multiplying the number of terminals (NXTER) by the average area per terminal (XTERLA). This produces the total terminal land area required in square feet (XTLSQF) which, in turn, is converted to acres (XTLACR). The terminal land acquisition capital cost is then computed (by subroutine COMPUTE) as follows:

ITEMS (KI, KJ, KM) = XTLACR (KI, KJ, KM) * XTLUD (KI, KJ, KM)

(Land capital cost - $) = (Land area - acres) (Land unit cost - $/acre)

where: KI - land area type

KJ - terminal type

KM - transit system

Total unit and cost data, unit and cost percentage data, and average cost per unit data, also computed by subroutine COMPUTE, are then output by OUTPUT2.

Transit terminal construction capital costs are then computed in much the same manner. Subroutine COMPUTE performs this data calculation by multiplying the required terminal construction area by the appropriate terminal construction unit cost data:

ITEMS (KI, KJ, KM) = XTCSQF (KI, KJ, KM) * XTCUD (KI, KJ, KM)

(Construction capital cost - $) = (Construction area - sq. ft.) (Construction unit cost - $/sq. ft.)

Once this data is computed, subroutine OUTPUT2 is called again for the printing of all output data. Subroutine ANNUALD is then called in order to annualize all terminal land and construction capital costs. An annual cost table summarizing these costs is then printed by ANNUALD for each transit system.

Capital costs associated with the purchase of land and for the construction of transit parking facilities are computed at thispoint in subroutine TRNCP2. Capital costs are computed (by subroutine COMPUTE) by multiplying the number of parking vehicle spaces required by the land unit cost data:

ITEMS (KI, KJ, KM) = XTPKVS (KI, KJ, KM) * XTPLUD (KI, KJ, KM)

(Land capital cost - $) = (Parking vehicle spaces) (Land unit cost - $/vehicle space)

Subroutine OUTPUT2 is then called to output this data.

Capital costs for the construction of parking facilities are computed by multiplying the number of vehicle spaces required by the average construction unit cost data. This calculation is performed within subroutine COMPUTE.

ITEMS (KI, KJ, KM) = XTPKVS (KI, KJ, KM) * XTPCUD (KI, KJ, KM)

(Construction capital cost - $) = (Parking vehicle spaces) (Construction unit cost - $/Vehicle spaces)

Having computed this capital cost data, along with unit and cost percentage data and average cost per unit data, control is returned to TRNCP2 for the calling of subroutine OUTPUT2. Once all the data has been printed, subroutine ANNUALD is called for the annualization of all parking land and construction capital costs. Annual costs are computed and output for each transit system by this routine.

At this point, TRNCP2 returns program control to calling subroutine TRNCPC.

5.1.5 Subroutine ANNUALD

Subroutine ANNUALD acts as a utility routine to the highway and transit capital cost subroutines (HWYCPC, TRNCPC, TRNCP2) of the program. Its primary function is to annualize the capital costs of pre-defined highway and transit capital elements. An annual cost table is also printed for each capital cost element annualized. In order to perform the annual cost calculations, data, representing the total number of units, the total capital cost, the interest rate and the life of each capital cost element, is passed to ANNUALD from the calling subroutine. This information is passed by means of arguments defined within the call statement.

ANNUALD was designed to annualize capital costs in two ways. The first method does not consider the interest cost of borrowing the money and, thus, is computed by dividing the capital cost of the element by its life expectancy.

$$AC(NX) = TCOST(NX)/LIF(NZ)$$

The second method of annualization considers the interest cost. In this case, the annual cost is computed by multiplying the total capital cost by an appropriate capital recovery factor.

$$ACI(NX) = TCOST(NX) * CRF$$

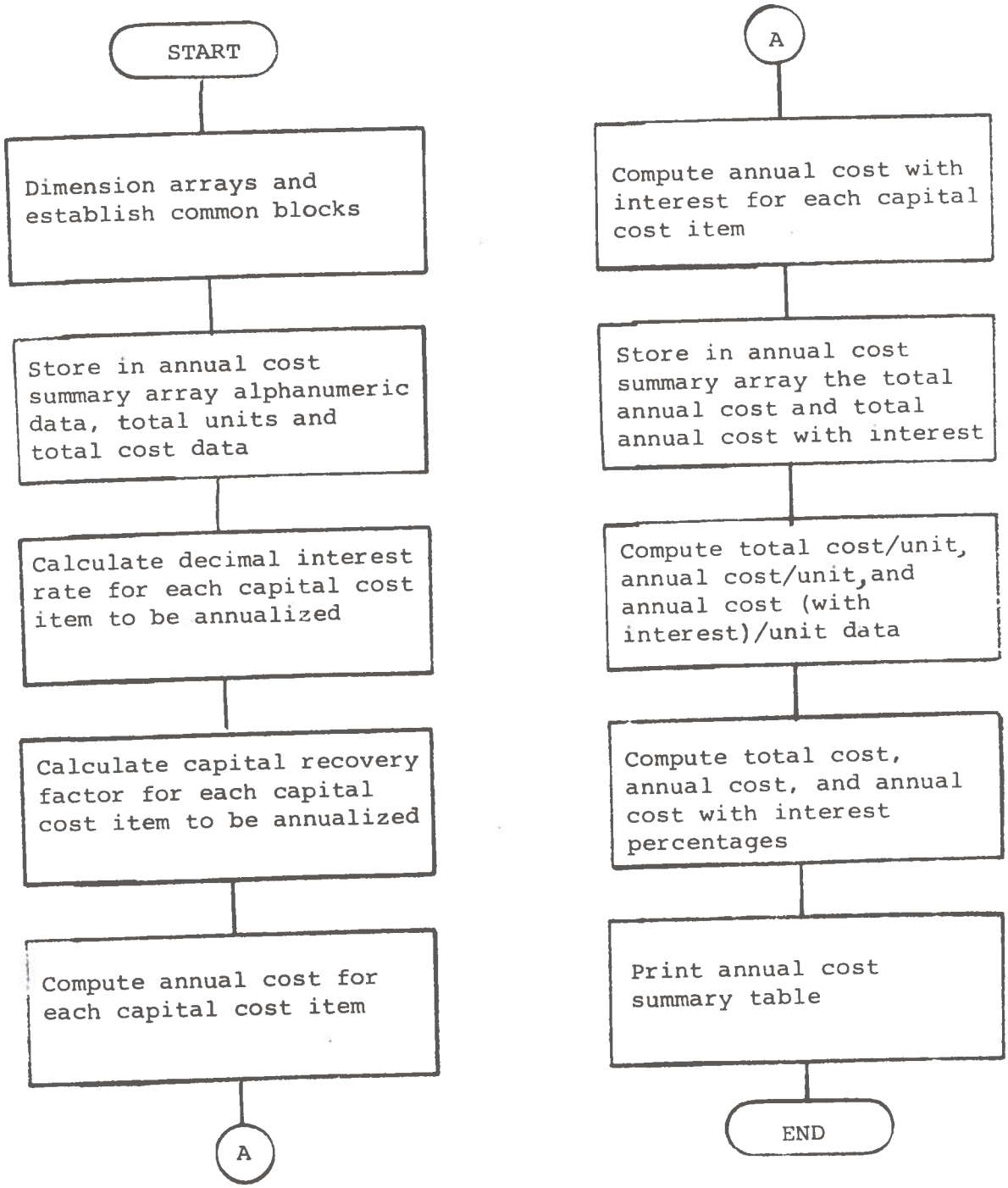


FIGURE 5-9 General Flow Chart - Subroutine ANNUALD

For each capital cost element being annualized, subroutine ANNUALD computes an appropriate capital recovery factor according to the following equation:

```
CRF = (XINT * (1.0 + XINT) ** LIF(NZ))/((1.0 + XINT) ** LIF (NZ)-1.0)
 where:  XINT is the interest rate (decimal %)
         LIF(NZ) is the life of capital cost element (years)
         NZ identifies the capital cost element.
```

The routine then computes the total cost per unit, and the annual costs per unit for each capital element being annualized. Appropriate cost totals are computed, as are the percentages of costs that each subsystem component represents of the total capital element cost.

Once the data described above is computed and printed, ANNUALD returns program control to the calling subroutine.

5.1.6 Subroutine BLOCKDATA

This routine stores alphanumeric data name titles for the highway and transit capital cost output tables.

5.1.7 Subroutine PCNT

The function of this routine is to compute percentage ratios that each element of an expanded, summed three dimensional array is of the total sum of elements in a given row, column, or depth direction of the array. The summed three-dimensional array from which the percentage ratios are to be computed is passed to PCNT from the calling subroutine as one of the arguments (defined as the UNIT array) of the CALL statement. The three arrays (ROWPCT, COLPCT, DPTPCT), containing the percentage ratios computed by PCNT in the row, column and depth directions, respectively, are returned to the calling routine also as arguments of the CALL statement. This routine is typically used within the Highway-Transit Cost program to calculate percentages that subsystem units and costs are of total system units and costs.

The following example illustrates the percentage data calculations that are performed by PCNT. In this example, an unexpanded 3x3x1 UNIT (I,J,M,) array having nine a_{ij} elements is considered. The row and column summations of these elements are also shown in the J+1 and I+1 locations of the array.

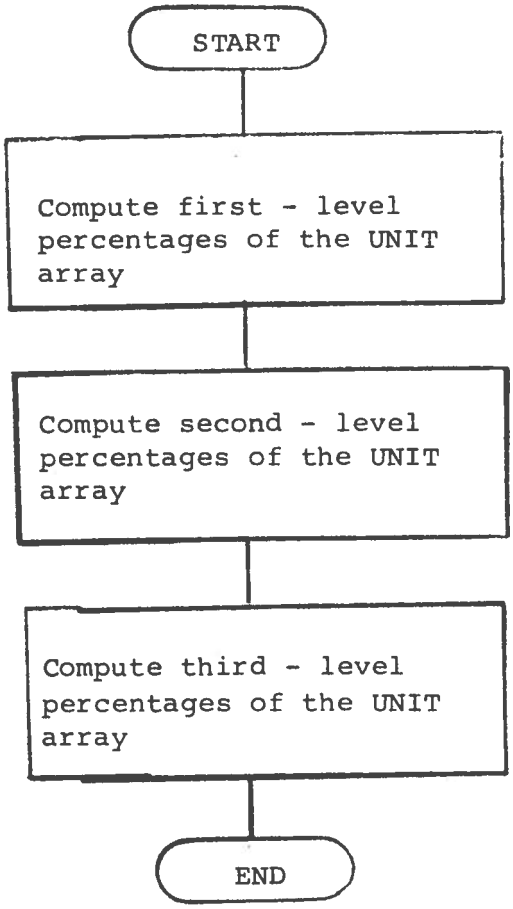


FIGURE5-10 General Flow Chart - Subroutine PCNT

UNIT (I+1,J+1,M)

	a_{11}	a_{12}	a_{13}	Σa_{1j}
	a_{21}	a_{22}	a_{23}	Σa_{2j}
	a_{31}	a_{32}	a_{33}	Σa_{3j}
	Σa_{i1}	Σa_{i2}	Σa_{i3}	Σa_{ij}

j →

i ↓

where i = 1,3

j = 1,3

In this case, row (ROWPCT) and column (COLPCT) data arrays would be computed as shown below. The routine would also compute a depth percentage data array (DPTPCT), however since in this case M=1 all the percentage data computed in this direction would be equal to 100%.

ROWPCT (I+1,J+1,M)

	$a_{11}/\Sigma a_{1j}$	$a_{12}/\Sigma a_{1j}$	$a_{13}/\Sigma a_{1j}$	$\Sigma a_{1j}/\Sigma a_{1j}$
	$a_{21}/\Sigma a_{2j}$	$a_{22}/\Sigma a_{2j}$	$a_{23}/\Sigma a_{2j}$	$\Sigma a_{2j}/\Sigma a_{2j}$
	a_{31}/Σ_{3j}	$a_{32}/\Sigma a_{3j}$	$a_{33}/\Sigma a_{3j}$	$\Sigma a_{3j}/\Sigma a_{3j}$
	$\Sigma a_{i1}/\Sigma a_{ij}$	$\Sigma a_{i2}/\Sigma a_{ij}$	$\Sigma a_{i3}/\Sigma a_{ij}$	$\Sigma a_{ij}/\Sigma a_{ij}$

j →

i ↓

where i = 1,3

j = 1,3

COLPCT (I+1,J+1,M)

$j \rightarrow$

$i \downarrow$

$$
\begin{array}{cccc}
a_{11}/\Sigma a_{i1} & a_{12}/\Sigma a_{i2} & a_{13}/\Sigma a_{i3} & \Sigma a_{1j}/\Sigma a_{ij} \\
a_{21}/\Sigma a_{i1} & a_{22}/\Sigma a_{i2} & a_{23}/\Sigma a_{i3} & \Sigma a_{2j}/\Sigma a_{ij} \\
a_{31}/\Sigma a_{i1} & a_{32}/\Sigma a_{i2} & a_{33}/\Sigma a_{i3} & \Sigma a_{3j}/\Sigma a_{ij} \\
\Sigma a_{i1}/\Sigma a_{i1} & \Sigma a_{i2}/\Sigma a_{i2} & \Sigma a_{i3}/\Sigma a_{i3} & \Sigma a_{ij}/\Sigma a_{ij}
\end{array}
$$

where $i = 1,3$

$j = 1,3$

Once the ROWPCT, COLPCT and DPTPCT data arrays are computed, PCNT returns control to the calling subroutine.

5.1.8 Subroutine COMPUTE

This subroutine was designed to perform all the cost data calculations of the highway and transit capital cost routines. COMPUTE operates on a units array (UNITS) and a unit cost array (ITMUNT) producing the following data:

- total units data by summing the UNITS (I, J, M) array. The SUM subroutine performs this summation.
- cost data by multiplying the units array (UNITS) and the unit cost array (ITMUNT). Subroutine MULT performs this array multiplication producing the cost array (ITEMS).
- total cost data by summing the cost array (ITEMS). Again, the SUM subroutine is used to perform the summation.
- cost percentage data representing the ratio of indivudual subsystem costs to the total cost data. Cost percentage data is computed by subroutine PCNT and is stored within arrays ROWPC2, COLPC2, and DPTPC2.

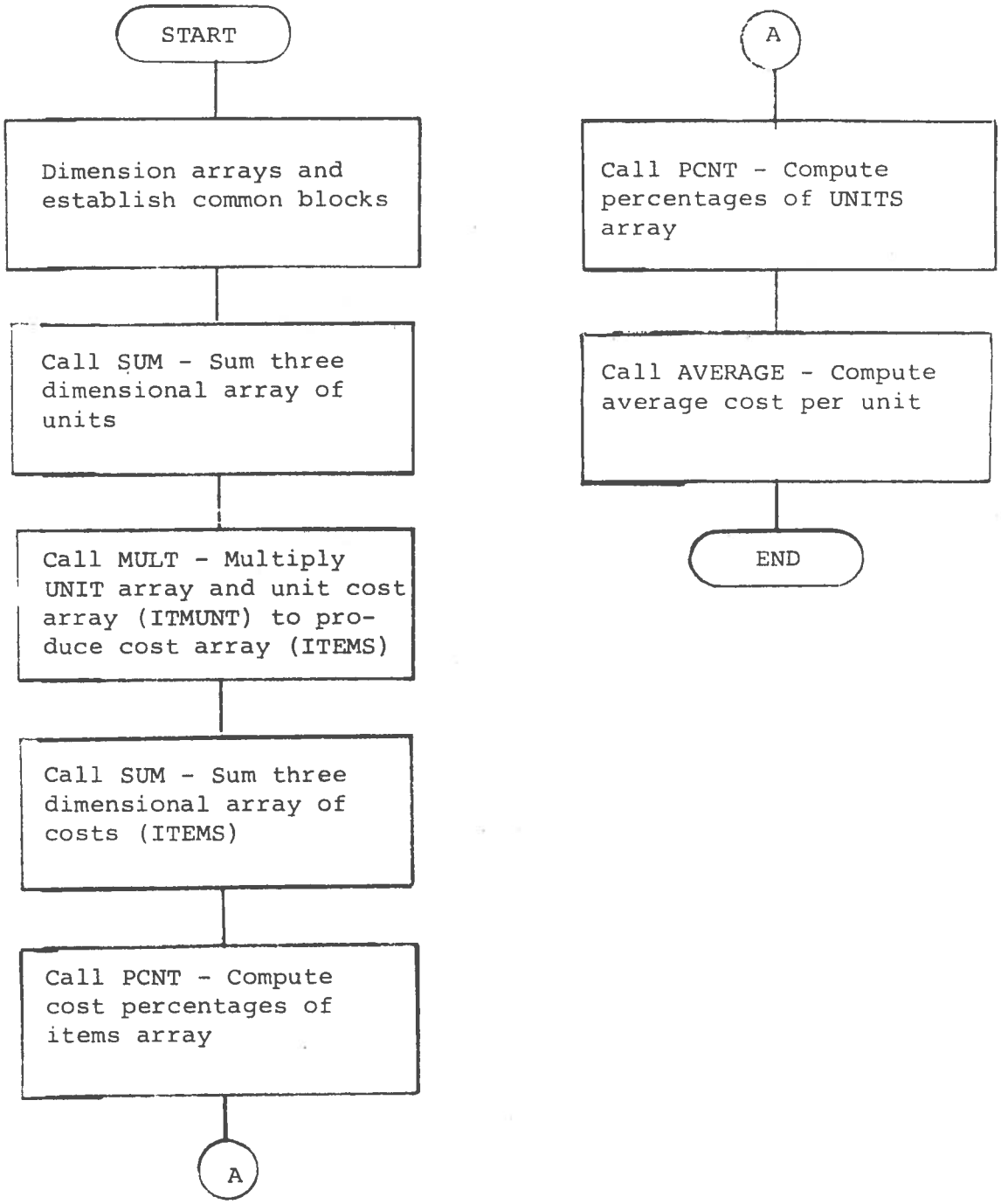


FIGURE 5-11 General Flow Chart - Subroutine COMPUTE

- unit percentage data reprensenting the ratio of the subsystem units to the total number of units. This percentage data is also computed by PCNT and is stored within arrays ROWPC1, COLPC1, and DPTPC1.

- average cost per unit data. This data is computed by subroutine AVERAGE and is stored within the I+1, J+1, and M+1 location of the ITMUNT array.

Once this data is computed, subroutine COMPUTE returns program control to the calling subroutine.

5.1.9 Subroutine MULT

This routine performs on element by element multiplication of two three-dimensional arrays - UNIT(I,J,M,) and ITMUNT(I,J,M) to produce a product array ITEMS(I,J,M):

```
ITEMS(I,J,M) = UNIT(I,J,M) * ITMUNT(I,J,M)
```

The I,J,M indices, the UNIT, and ITMUNT arrays are passed to this routine as arguments within the CALL statement. Similarily, the product array ITEMS is returned to the calling subroutine as the sixth argument of the CALL statement.

This routine is typically used throughout the Highway-Transit Cost program in the multiplication of unit and unit-cost data arrays.

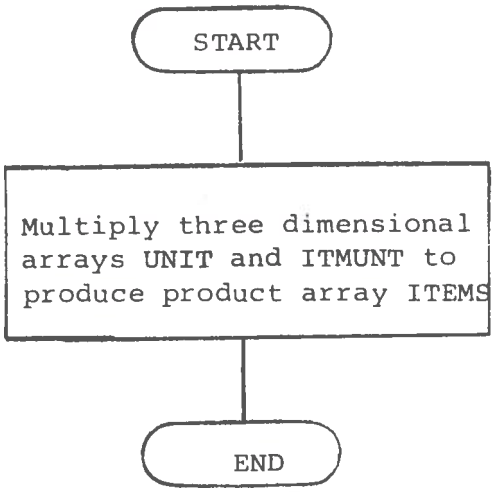


FIGURE 5-12 General Flow Chart - Subroutine MULT

5.1.10 Subroutine AVERAGE

Subroutine AVERAGE performs an element-by-element division of the I+1, J+1, M+1 elements within two three-dimensional arrays - ITEMS (I+1, J+1, M+1) and UNIT (I+1, J+1, M+1) producing a third array ITMUNT (I+1, J+1, M+1) containing the averaged data. The ITEMS and the UNIT arrays that are to be operated upon by this routine are passed from the calling subroutine as arguments within the CALL statement. The ITMUNT array containing the averaged data is returned to the calling subroutine as the sixth argument within the CALL statement.

This routine is typically used throughout the program to compute average cost/unit data. This is done by dividing a cost array (ITEMS) by a units array (UNITS).

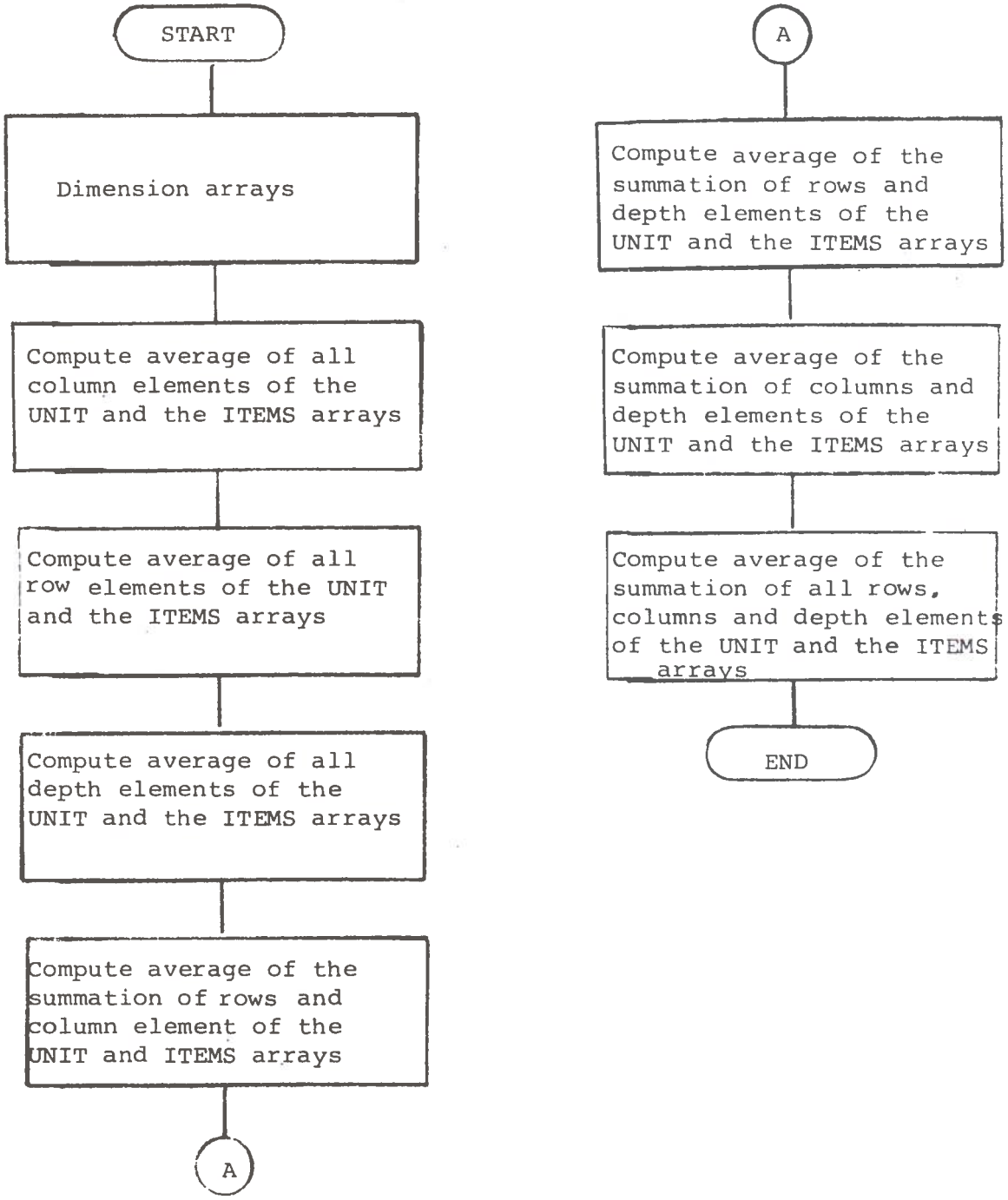


FIGURE 5-13 General Flow Chart - Subroutine AVERAGE

5.1.11 Subroutine HWYTRAN

The primary function of this routine is to compute and output all transit network data for the various transit systems under analysis. Network analysis data (i.e., VMT, VHT, average trip lengths, average trip time, etc.) are generated by this routine for individual transit systems with respect to four temporal periods (peak, off-peak, daily and annual). Figure 5-14 illustrates the sequence of functions performed by this routine.

Transit system characteristics and transit demand data comprise the input data to HWYTRAN. Transit characteristic data includes such variables as vehicles capacity and load factors, average speed and average delays, Transit demand data are input in units of person-miles traveled/day and person trips. A further description of the input data is provided within paragraph 5.2 Input Data (Card Format Type 31 through 41).

Initially, HWYTRAN computes for each transit system (IXV) the average vehicle occupancy (XVHOCC) during the peak and the off-peak temporal periods. This data is computed by multiplying the transit vehicle capacity by the appropriate (peak and off-peak) vehicle load factors:

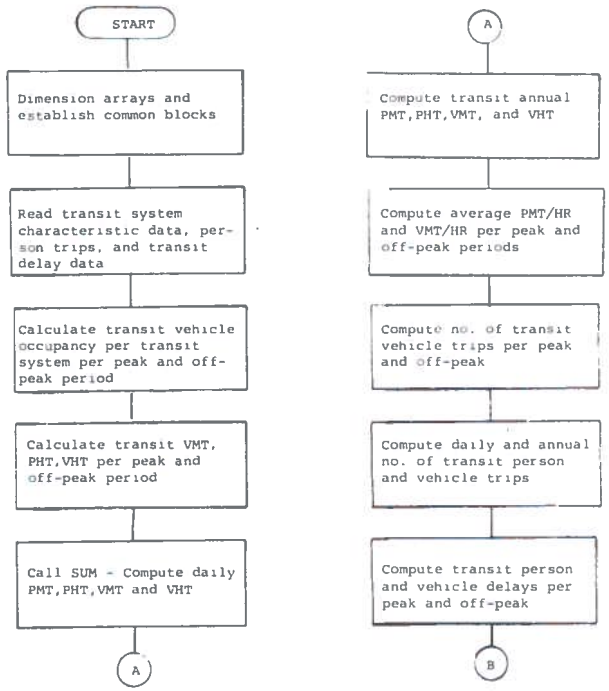


FIGURE 5-14 General Flow Chart - Subroutine HWYTRAN

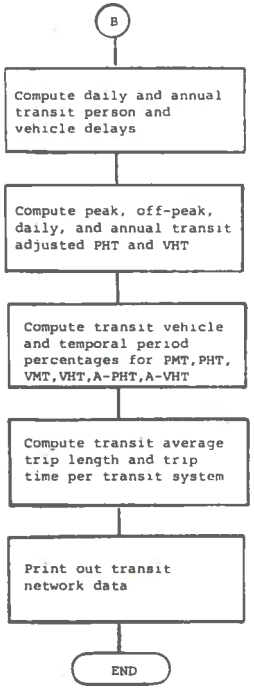


FIGURE 5-14 General Flow Chart - Subroutine HWYTRAN (continued)

XVHOCC(IXV,$\frac{1}{2}$) = XSYSCH(IXV,1) * XSYSCH(IXV,$\frac{3}{4}$)

Vehicle occupancy (peak & off-peak) = Vehicle capacity X Vehicle load factor (peak & off-peak)

The person-miles traveled during the peak temporal period is then computed for each transit system by applying a "peak percentage factor" to the transit system's daily PMT.

XNDATA(IXV,1,1) = XSYSCH(IXV,5) * XPKPCT

(PMT - peak) = PMT/Day X (peak % factor)

The off-peak person-miles traveled is then the daily PMT less the peak period PMT.

XNDATA(IXV,2,1) = XSYSCH(IXV,5) - XNDATA(IXV, 1,1)

(PMT - off-peak) = PMT/Day - (PMT - peak)

The 'peak percentage factor' (XPKPCT) and the daily person-miles traveled data are input by the user.

At this point, HWYTRAN computes, for each transit system (IXV) and for the peak and off-peak temporal periods (ITP = 1,2), transit vehicle-miles traveled (VMT), person-hours traveled (PHT), and vehicle-hours traveled (VHT) data according to the following equations:

XNDATA(IXV,ITP,2) = XNDATA(IXV,ITP,1) / XVHOCC(IXV,ITP)

VMT = PMT / (ave. number of persons/vehicle)

XNDATA(IXV,ITP,3) = XNDATA(IXV,ITP,1) / XSYSCH(IXV,2)

PHT = VMT / (ave. speed - mph)

XNDATA(IXV,ITP,4) = XNDATA(IXV,ITP,2) / XSYSCH(IXV,2)

VHT = VMT / (ave. speed - mph)

Daily and annual PMT, PHT, VMT and VHT data are then computed by first summing (by subroutine SUM) the XNDATA array for the daily amounts and then multiplying by the variable NUMDA - the number of days/year.

Average PMT/hour and VMT/hour data are computed for each transit system for the peak and off-peak temporal periods as follows:

<u>Peak</u>: XNDPHR(IXV,1,$\frac{1}{2}$) = XNDATA(IXV,1,$\frac{1}{2}$) / NPKHRS

PMT/hr (or VMT/hr) = Peak - PMT (or VMT) / (no. of peak hours/day)

<u>Off-peak</u>: XNDPHR(IXV,2,$\frac{1}{2}$) = XNDATA(IXV,2,$\frac{1}{2}$) / NOPHRS

PMT/hr (or VMT/hr) = off-peak PMT (or VMT)/ (no. of off-peak hour/day)

Subroutine HWYTRAN then calculates the number of transit vehicle trips occurring during the peak and off-peak temporal periods. Transit person trip and vehicle occupancy data are used to perform this data calculation as follows:

XVTRIP(ITP,IXV) = XPTRIP(ITP,IXV) / XVHOCC(IXV,ITP)

Vehicle trips = Person trips / (Persons/vehicle)

Daily person and vehicle trips are also computed by summing the respective peak and off-peak trips; annual trips are computed by multiplying the daily trips by the number of days/year defined in variable NUMDA.

Transit person delays (occurring within vehicle and outside the vehicle)and transit vehicle delays are then calculated for each transit system for the peak and off-peak temporal periods (ITP = 1,2) as follows:

XPVDLY(ITP,IXV,1) = XPTRIP(IPT,IXV) * XDELAY(ITP,IXV,1) / 60.

(person delay outside vehicle-hrs) = (person trips) X (ave. delay-min/person trip) / 60.

XPVDLY(ITP,IXV,2) = XVTRIP(ITP,IXV) * XDELAY(ITP,IXV,2) / 60.

(vehicle delay-hrs) = (vehicle trips) X (ave. delay-min/veh. trip) / 60.

XPVDLY(ITP,IXV,3) = (XPTRIP(ITP,IXV) * XDELAY(ITP,IXV,2) / XVHOCC(IXV,ITP)) / 60.

(person delay within vehicle-hrs) = (person trips) X (ave. delay-min/veh. trip) / (vehicle occupancy)) / 60.

These delays are then utilized by HWYTRAN to adjust the transit person-hours traveled and vehicle-hours traveled to account for all delays experienced. Adjusted PHT data is computed by adding the total person-hours of delay to the total PHT; similarly the adjusted transit VHT data is calculated by the addition of the total vehicle hours of delay to the VHT data. Once this data is computed, average transit trip length and trip times are then calculated for transit person trips and vehicle trips as follows:

(Average Trip Length-miles/trip) = PMT(or VMT) / person trips (or vehicle trips)

(Average Trip Time-hours/trip) = Adjusted PHT(or VHT) / person trips (or vehicle trips)

At this point, all data computed above is output by HWYTRAN. The user input variable IPRNT controls the printing of several of the output data tables. Upon completion of all data output, HWYTRAN returns program control to NULCASE for the calling of the next subroutine HIGHWY.

5.1.12 Subroutine HIGHWY

HIGHWY computes and outputs network analysis data related to highway operations. The flow of autos, trucks and transit vehicles are considered in these data calculations. Data is generated with respect to freeways and surface arterials and with respect to four temporal periods (peak, off-peak, daily, and annual). The sequence of functions performed by HIGHWY are described in the following figure.

The subroutine reads all input data as defined within Card Format Type 42 through 45 of paragraph 5.2. Highway network characteristic data (i.e., number of lane miles, lane capacity), auto characteristic data (i.e., vehicle capacity and load factors), and auto demand data (i.e., PMT/Day, person trips) comprise the input data to this routine.

Initially, the subroutine computes the average auto vehicle occupancy utilizing the input vehicle capacity and vehicle load factor data:

AUTOCH(11) = AUTOCH(1) * AUTOCH(2)

(Occupancy - person/vehicle) = (Vehicle capacity - persons/vehicle) X (Vehicle load factor)

HIGHWY then calculates the auto person miles traveled on freeways and surface arterials during the peak and off-peak

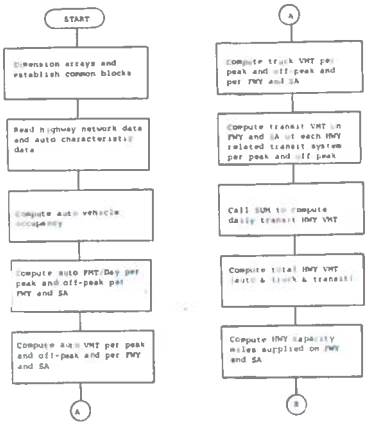


FIGURE1-14 General Flow Chart - Subroutine HIGHWY

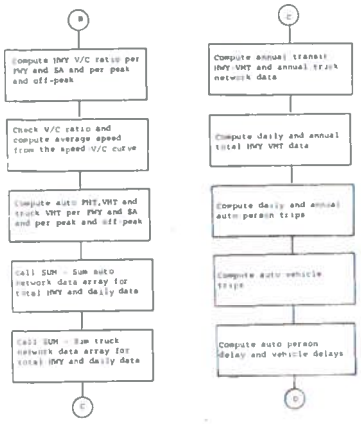


FIGURE1-15 General Flow Chart - Subroutine HIGHWY (continued)

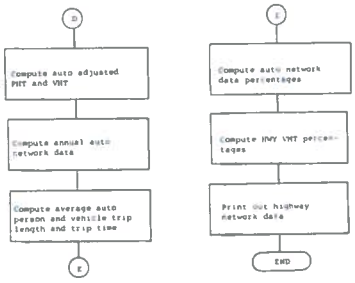


FIGURE1-16 General Flow Chart - Subroutine HIGHWY (continued)

temporal periods. Peak and off-peak auto PMT are computed by applying a temporal splitter to the auto daily PMT. The peak period temporal splitter is input by the user as variable AUTOCH(7); whereas the off-peak period temporal splitter is equal to 1.0 - AUTOCH(7). Auto vehicle-miles travelled on freeways and surface arterials (MH = 1,2) during the peak and off-peak temporal periods (ITP = 1,2) are then calculated as follows:

AUTOND(MH,ITP,2) = AUTOND(MH,ITP,1) / AUTOCH(11)

AUTO VMT = AUTO PMT / (persons/vehicle)

HIGHWY then computes the vehicle miles travelled by trucks on freeways and surface arterials during the peak and off-peak temporal periods. This data is computed as a function of the auto VMT as follows:

TRCKND(MH,ITP,1) = (AUTOND(MH,ITP,2) / AUTOCH(8)) - AUTOND(MH,ITP,2)

Truck VMT = (AUTO VMT / (% Auto VMT in area)) - AUTO VMT

The '% AUTO VMT in area' factor in the above equation should represent the amount of auto travel in the area under analysis. It is input by the user within the auto characteristics data card.

After computing the amount of vehicle travel (VMT) by transit systems operating on freeways and surface arterials, the routine then calculates the total VMT on freeways and surface arterials during the peak and off-peak temporal periods. This is accomplished by summing the appropriate data from the auto, truck and transit VMT data arrays.

At this point, data calculations are performed to determine the level of congestion (V/C ratio) and the average speed on freeways and surface arterials during the peak and off-peak temporal periods. The freeway and surface arterial capacity miles supplied during the peak and off-peak temporal periods are computed according to the following equations:

<u>Freeways</u>: HCMS(1,ITP) = (HNCHAR(1) * HLNDIR(1,ITP)) * HNCHAR(3)

<u>Surface Arterials</u>: HCMS(2,ITP) = (HNCHAR(2) * HLNDIR(2,ITP)) * HNCHAR(4)

Capacity miles supplied = (Lane miles) (lane mile directional factor-decimal %) (lane mile capacity-vehicles/hour/lane mile)

The lane mile directional factors in the above equations allow the user to adjust the existing lane miles of freeways

and surface arterials in the area under analysis to account for directional flow of vehicles during the peak and off-peak temporal periods of the day.

Volume/capacity ratios equal to the total hourly VMT divided by the capacity miles supplied, are then computed for freeways and surface arterials and for the peak and off-peak periods as follows:

<u>PEAK</u>: HVCRAT(MH,1) = (THVMT(MH,1) / NPKHRS/HCMS(MH,1))

<u>OFF-PEAK</u>: HVCRAT(MH,2) = (THVMT(MH,2) / NOPHRS) / HCMS(MH,2))

The computed V/C ratios are then used in the following equations to arrive at an average speed for the peak and off-peak temporal periods on freeways and surface arterials:

<u>Freeways</u>:

$$\text{Ave. Speed (mph)} = 54.0 - 10.0\ (V/C) \qquad 0.0 \leq V/C < .6$$

$$\text{Ave. Speed (mph)} = (810.0\ (1.0 - V/C))^{1/2} + 30.0 \qquad .6 \leq V/C \leq 1.0$$

<u>Surface Arterials</u>:

$$\text{Ave. Speed (mph)} = (180.0\ (1.0 - V/C))^{1/2} + 12.0 \qquad 0.0 \leq V/C \leq 1.0$$

Refer to Figure 3-40 of paragraph 3.4.3 for the average speed - V/C relation curve.

Knowing the average peak and off-peak period speed on the freeways and surface arterials, auto PHT, auto VHT, and truck VHT are then calculated according to the following equations:

AUTOND(MH,ITP,3) = AUTOND(MH,ITP,1) / HAVSPD(MH,ITP)

Auto PHT = Auto PMT/Average speed (mph)

AUTOND(MH,ITP,4) = AUTOND(MH,ITP,2) / HAVSPD(MH,ITP)

Auto VHT = Auto VMT/Average speed (mph)

TRCKND(MH,ITP,2) = TRCKND(MH,ITP,1) / HAVSPD(MH,ITP)

Truck VHT = Truck VMT/Average speed (mph)

Daily and annual auto and truck network data are then calculated by first summing the AUTOND and TRCKND data arrays by means of the SUM subroutine, and then multiplying the daily data by the variable NUMDA - number of days/year.

The number of auto vehicle trips, auto trip delays, and adjusted auto PHT and VHT data are computed next by this routine. Auto vehicle trips are computed by dividing the number of person trips by the average auto vehicle occupancy. Passenger and vehicle delays are computed by multiplying average delays per trip by the number of person and vehicle trips. Adjusted PHT and VHT data are then computed by adding, respectively, the total person and vehicle hours of delay to the auto PHT and VHT data.

The average trip length and trip time for auto person and vehicle trips are then calculated according to the following equations:

Average trip length (miles/trip) = PMT(orVMT)/Person trips (or vehicle trips)

Average trip time (hours/trip) = adjusted PHT (or VHT)/ Person trips (or vehicle trips)

At this point, HIGHWY begins outputting the highway network analysis data computed above. The printing of certain output data tables is controlled by the user input variable IPRNT. Upon completion of all data output, program control is returned to program NULCASE for the calling of the next subroutine PWRPOL.

5.1.13 Subroutine PWRPOL

The primary function of this subroutine is to determine the amount of power consumed by autos, trucks, and transit systems. Besides performing the power consumption data calculations, the routine also prints all output data. The following figure illustrates the sequence of functions that are performed by this routine.

Within this subroutine, the amount of power consumed by autos, trucks, and the various transit systems is calculated in units of KWHR or pounds.

For electric vehicles the following equation is utilized in the calculation of the power consumed:

$$\text{POWER(KWHR)} = \left\{\left\{(W_V \times C_T) + (K_1\ C_A \times A \times V^2)\right\} \times \left[\text{VMT} \times K_2\right]\right.$$

For internal combustion engine vehicles the power consumption is computed utilizing the following equation:

$$\text{POWER(LBS)} = \left\{\left\{(W_V \times C_T) + (K_1 \times A \times V^2)\right\} \times \left[\text{VMT} \times K_3\right]\right.$$

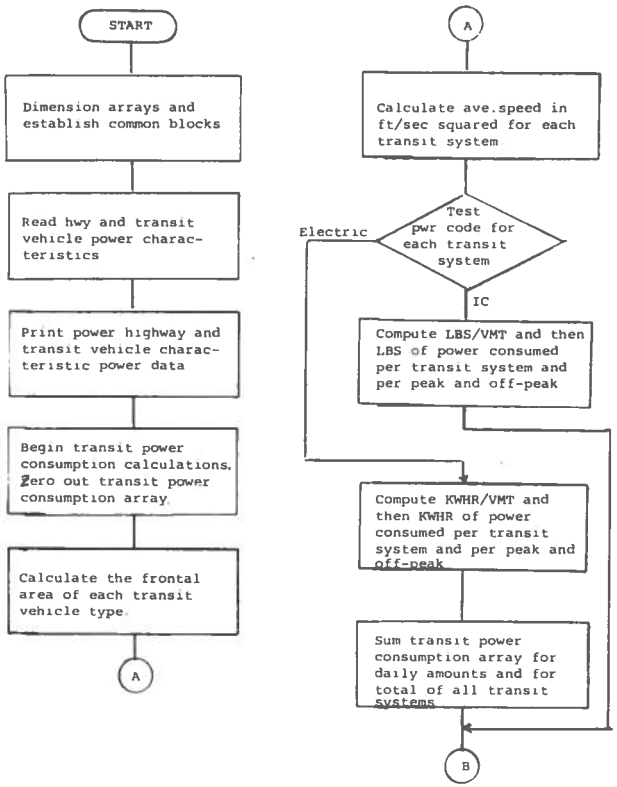


FIGURE 5-16 General Flow Chart - Subroutine PWRPOL

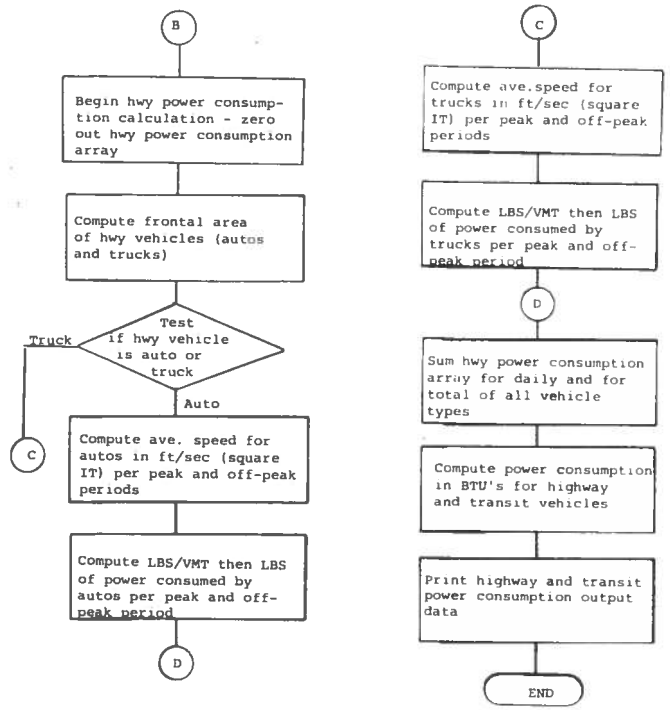


FIGURE 5-16 General Flow Chart - Subroutine PWRPOL (continued)

where in the above equations:

W_V = vehicle weight (lbs)

C_T = tire friction coefficient (unitless)

CA = aerodynamic friction coefficient (unitless)

A = frontal area of vehicle (ft^2)

V = velocity of vehicle (ft/sec)

VMT = vehicle miles travelled

K_1 = air density (lbm/ft^3) / (2x gravitational constant) ($lbm\ ft/lb_f\ sec^2$)

K_2 = changes VMT to feet, ft-lb_f to kw hrs, and accounts for power transmission and consumption efficiencies and drive train efficiency, = 0.00311

K_3 = changes VMT to feet, ft-lb_f to hp hr, and accounts for vehicle drive train efficiency, = 0.00167

Initially, PWRPOL reads the user card data input to this routine. This data is totally comprised of vehicle characteristic data (weight, height, width, type of propulsion, etc.) for each of the transit and highway vehicle systems. Once this data is read in, PWRPOL prints all input data before beginning the transit power consumption data calculations.

Once the input data is printed, PWRPOL begins the transit power consumption data calculations by first computing the vehicle frontal area for each transit system (IXV). A multiplication of the vehicle's height and width produces this data within array A(IXV).

The average speed (in ft/sec) during the peak and off-peak temporal periods is computed and then squared according to the following equation:

```
E = (((XNDATA(IXV,ITP,2) / XNDATA(IXV,ITP,4)) * 1.47) ** 2)
```

$$\left[\text{Speed (ft/sec)}\right]^2 = \left[\frac{\text{transit VMT}}{\text{transit VHT}} \times 1.47\right]^2$$

Then for each transit system, the input power code is tested to determine the type of propulsion used by the vehicle. Depending upon the type of propulsion (electric or IC), energy consumption rates (KWHR/VMT or LBS/VMT) and the total energy consumed (KWHR or LBS) are computed for each transit system with respect to the peak and off-peak temporal periods. The equations utilized in these data calculations are presented below:

<u>Electric</u>

```
C = ((XPCVCH(IXV,1) * XPCVCH(IXV,2)) + (.00134 * XPCVCH
(KWHR/VMT)          (IXV,3) * A(IXV) * E)) * .00311

XPWRC(IXV,ITP,1) = XNDATA(IXV,ITP,2) * C

(KWHR)           = (VMT) x (KWHR/VMT)
```

<u>IC</u>

```
C =((XPCVCH(IXV,1) * XPCVCH(IXV,2)) + (.00134 * XPCVCH
(LBS/VMT)           (IXV,3) * A(IXV) * E)) * .00167

XPWRC(IXV,ITP,2) = XNDATA(IXV,ITP,2) * C
    (LBS)        = (VMT) x (LBS/VMT)
```

Once the peak and off-peak transit energy consumption data are computed, the SUM subroutine is called to sum the XPWRC data array. This produces the daily power consumption and the total power consumption of all transit systems.

At this point, PWRPOL begins the energy consumption data calculations for autos and trucks. Each of these vehicles are assumed to be powered by internal combustion engines. After having computed the vehicle frontal area for each of these vehicles, the routine then calculates the average vehicular speed (in ft/sec) squared according to the following equations:

<u>Autos</u>

```
E = (((AUTOND(3,ITP,2) / AUTOND(3,ITP,4)) * 1.47) ** 2)
```

<u>Trucks</u>

```
E = (((TRCKND(3,ITP,1) / TRCKND(3,ITP,2)) * 1.47) ** 2)
```

A fuel consumption rate (lbs/VMT) and the total power consumed during the peak and off-peak temporal periods are then computed as follows:

```
C = ((HPCVCH(NHV,1) * HCPVCH(NHV,2)) + (.00134 * HPCVCH
     (NHV,3) * A(NHV) * E)) * .00167
```

<u>Autos</u>

```
HPWRC(NHV,ITP,1) = AUTOND(3,ITP,2) * C
```

<u>Trucks</u>

```
HPWRC(NHV,ITP,1) = TRCKND(3,ITP,1) * C
```

Subroutine SUM is then called to sum HPWRC for the daily energy consumption and for the total amount of power consumed by all highway vehicles.

At this point, PWRPOL prints all auto, truck and transit energy consumption data computed above. Once the printout of this data has been completed, subroutine AIRPOL is then called by PWRPOL.

5.1.14 Subroutine AIRPOL

The primary function of this routine is to calculate and to output data representing the amount of air pollution emitted by autos, trucks, and the various transit systems. These data calculations are performed in the sequence outlined in Figure 5-17 .

User input data to this routine is comprised of pollution rates and vehicle propulsion codes. Initially, all input data is read by the routine and then printed prior to any data calculations. For autos, trucks and transit vehicles, the routine computes the amount of air pollution emitted as a function of the amount of vehicle travel; the type of pollutants emitted is based upon the vehicle's propulsion system.

Transit air pollution data calculations are first performed by this routine. For each transit system (IXV), AIRPOL checks the input vehicle propulsion code to determine the type of propulsion (IC - gas, IC - diesel, or electric) used by each system. Based upon the setting of this code,

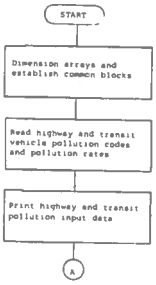


FIGURE5-17 General Flow Chart - Subroutine AIRPOL

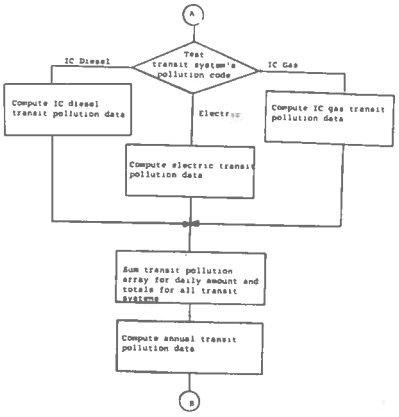


FIGURE5-17 General Flow Chart - Subroutine AIRPOL (Continued)

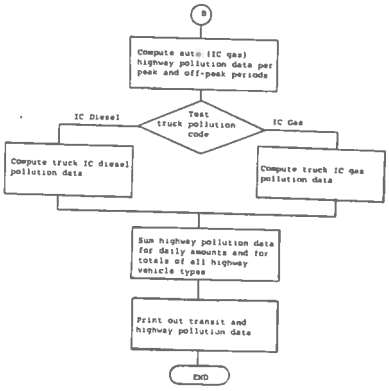


Figure 5-17 General Flow Chart - Subroutine AIRPOL (continued)

the amount and the type of pollutants emitted by each transit system are computed utilizing one of the following equations:

IC GAS:

XPOL(IXV,JTX,ITP) = POLRAT(1,JTX) * XNDATA(IXV,ITP,2)
(Air Pollution-LBS) = (Pollution Rate-lbs/VMT) x (transit VMT)

where: JTX = 1 - CO
= 2 - HC
= 3 - NOX

ITP = 1 peak period
= 2 off-peak period

IC Diesel:

XPOL(IXV,JTX,ITP) = POLRAT(2,JTX) * XNDATA(IXV,ITP,2)
(Air Pollution-LBS) = (Pollution Rate-LBS/VMT) x (transit VMT)

where: JTX = 1 - CO
= 2 - HC
= 3 - NOX

ITP = 1 peak period
= 2 off-peak period

Electric:

XPOL(IXV,JTX,ITP) = POLRAT(3,JTX) * XPWRC(IXV,ITP,1)
(Air Pollution-LBS) = (Pollution Rate-LBS/KWHR) x (transit power consumed-KHWR)

where: JTX = 1 - Particulates
= 2 - SO_2
= 3 - NOX

ITP = 1 - peak period
= 2 - off-peak period

The transit air pollution data array (XPOL) is then summed by the SUM subroutine for the daily air pollution emission levels (peak + off-peak) of each transit system and for the total emission of all transit systems. The amount of pollutants emitted annually by the various transit systems is then computed by multiplying the daily emission levels by the factor NUMDA - number of days/year.

The air pollution emission by autos and trucks are then computed by AIRPOL in the same manner as above. For autos (all of which are assumed to be powered by gas internal combustion engines), the amount of pollutants emitted during the peak and off-peak temporal periods is computed according to the following equation:

<u>IC GAS</u>

```
HWYPOL(1,JTX,ITP) = POLRAT(1,JTX) * AUTOND(3,ITP,2)
(Auto air pollution-LBS) = (Air pollution rate-LBS/VMT) *
                                (Auto VMT)
```

Similarly, the air pollution emission from trucks during the peak and off-peak temporal periods is computed according to one of the following equations:

IC GAS:

```
HWYPOL(2,JTX,ITP) = POLRAT(1,JTX) * TRCKND(3,ITP,1)
(Truck air pollution-LBS) = (Pollution rate-LBS/VMT) x
                            (Truck VMT)
```

IC DIESEL:

```
HWYPOL(2,JTX,ITP) = POLRAT(2,JTX) * TRCKND(3,ITP,1)
(Truck air pollution-LBS) = (Pollution rate-LBS/VMT) x
                            (Truck VMT)
```

The propulsion code input for trucks identifies which of the equations listed above is used.

Once this data calculation has been completed, AIRPOL then proceeds to calculate the daily and annual amounts of pollutants emitted by autos and trucks.

This completes all data calculations performed by this routine. All computed data is then printed before control is returned to the calling subroutine PWRPOL.

5.1.15 Subroutine ACCDNT

As illustrated in the following flow diagram (Figure 5-18), subroutine ACCDNT calculates and outputs accident data for autos and for transit systems.

Initially, the subroutine reads accident rate input data for autos and for each transit system being considered. The format of this card input data is defined in paragraph 5.2 as card format type 54 and 55. ACCDNT also utilizes person trip and VMT data in its accident data calculations. This data is received from the highway and transit network subroutines (HIGHWY and HWYTRAN) within common block /DG/. Once the card input data is read, ACCDNT prints this data before performing any accident data calculations.

Auto accidents, for a number of accident types, are computed by this routine with respect to their occurrence on freeways and surface arterials and with respect to four temporal periods (peak, off-peak, daily and annual). These accident data calculations are performed for the peak and off-peak temporal periods as shown in the following equations:

```
VEHACC (IMX, IAX, ITPX) = (AUTOND (IMX, ITPX, 2) * HACRAT
                                (IMX,IAX,ITPX)) / 100000000.
```

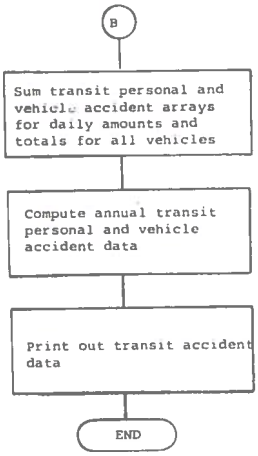


Figure 5-18 General Flow Chart - Subroutine ACCDNT

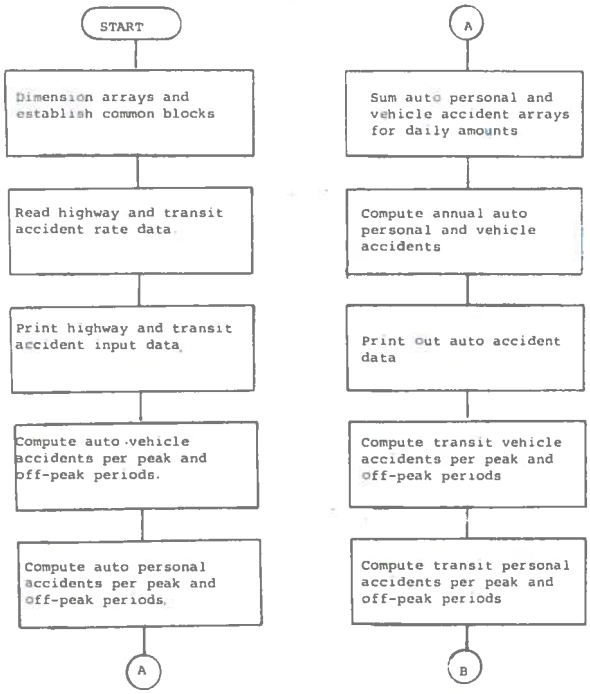


Figure 5-18 General Flow Chart - Subroutine ACCDNT

$$\text{(Vehicle Accidents)} = \text{(Auto VMT)} \left(\text{Accident rate} - \frac{\#\ \text{Accidents}}{100\ \text{million VMT}}\right) / 100000000.$$

where: IMX = 1 - Freeways

= 2 - Surface Arterials

IAX = 1 - fatal accidents

2 - injury accidents

3 - property damage accidents

PERACC (IMX, IBX, ITP) = (AUTOND (IMX, ITP, 2) * HACRAT (IMX, IAX,ITP)

/100000000.

where: IMX = 1 - Freeways IBX = 1 - fatalities

= 2 - Surface Arterials 2 - injuries

Daily and annual auto accident data are then calculated by this routine. Daily auto accidents are computed by summing (by means of the SUM subroutine) the VEHACC and the PERACC data arrays. Annual auto accidents are computed by multiplying the daily accident data by the factor NUMDA. All auto accident data is then printed at this time in order that PERACC and VEHACC data arrays may be utilized to store transit accident data.

Similar to the auto accident data calculations above, transit accidents are quantified with respect to four accident types (traffic accidents, passenger accidents, passenger fatalities and non-passenger fatalities) and with respect

to four temporal periods (peak, off-peak, daily, and annual). Peak period and off-peak period accident data calculations are performed as follows:

VEHACC (IXV,1,ITP) = (XNDATA (IXV,ITP,2) * (XACRAT (IXV,1)) /1000000.

(Traffic Accidents) = (Transit VMT) (Accident Rate - # $\frac{\text{accidents}}{\text{million VMT}}$) /1000000.

VEHACC(IXV, ITP) = (XTRIP(ITP,IXV) * XACRAT(IXV,2)) /1000000.

(Passenger accidents) = (Transit person trips)(accident rate - $\frac{\#\text{accidents}}{\text{million person trips}}$) /1000000.

where: IBX = 1 passenger fatalities

= 2 non-passenger fatalities

Daily transit accidents are calculated by summing the peak period plus off-peak period accidents. Annual accidents are computed by multiplying the transit daily accidents by the factor NUMDA. Having computed this data, ACCDNT then prints all transit accident data.

This completes all processing performed by this routine. Program control is returned to the main program NULCASE for the calling of the next subroutine HWYOP.

5.1.16 Subroutine HWYOP

Subroutine HWYOP calculates and outputs all operating costs associated with highways and highway vehicle systems. Figure 5-19 further describes the sequence of functions performed by this routine.

Initially, HWYOP reads all user input data. This data is primarily comprised of unit operating costs for the maintenance and operation of ways and vehicles. This subroutine also utilizes auto and truck VMT data which it receives from the highway network (HIGHWY) subroutine by means of common block /DG/. All user input data is printed by HWYOP prior to any cost calculations.

The routine then proceeds to calculate for freeways (MHX = 1) and for surface arterials (MHX = 2) the total number of route miles and the total number of manyears required for the area under analysis. The number of freeway and surface arterial route miles is computed by dividing the total number of freeway (or surface arterial) lane miles by the average number of lanes.

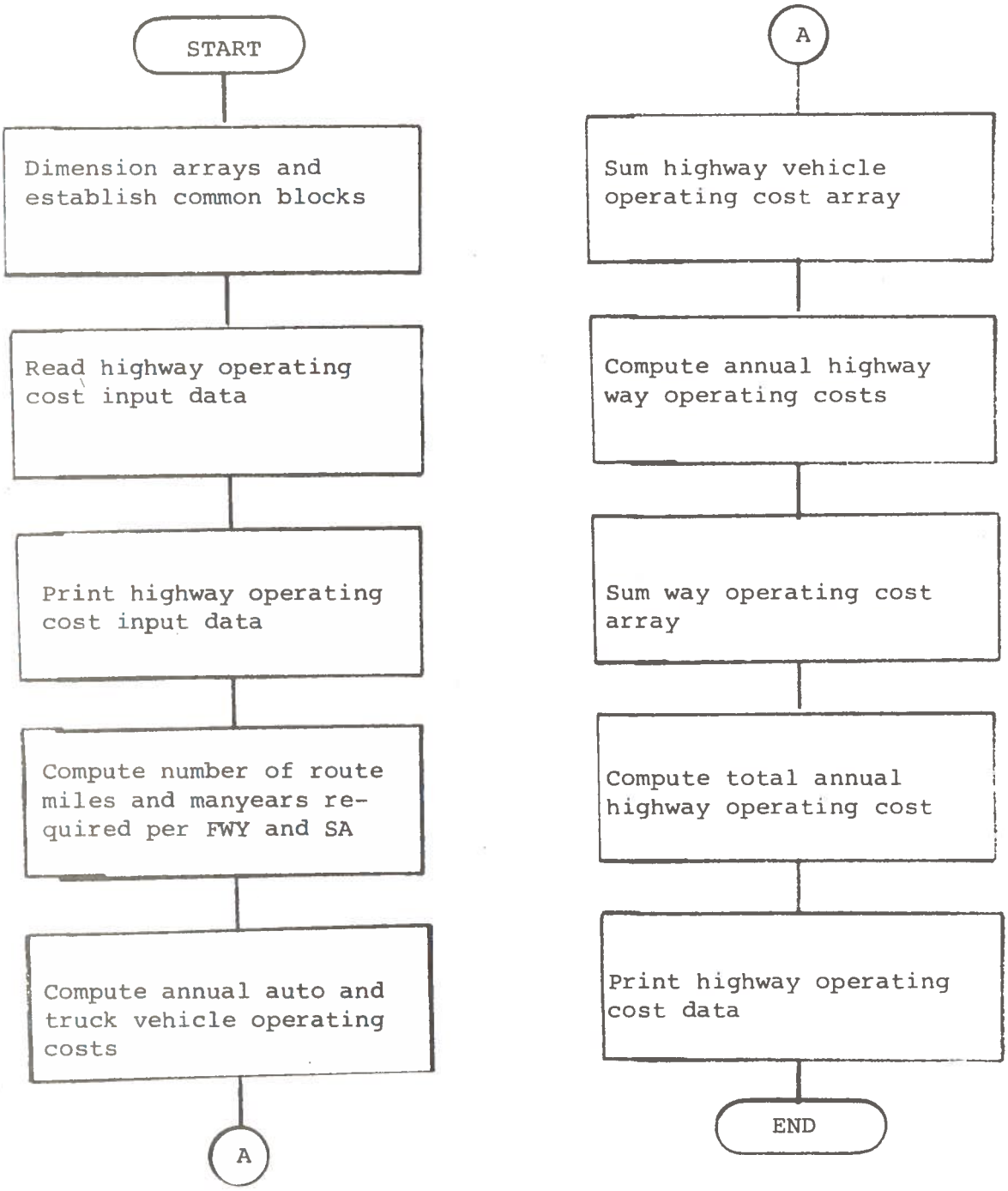


Figure 5-19 General Flow Chart - Subroutine HWYOP

HOPDAT(MHX,3) = HNCHAR(MHX) / HOPDAT(MHX,1)

(No. of route miles) = (No. of lane miles) / average no. of lanes

The number of manyears required for way manpower operating costs is computed based upon the number of freeway and surface arterial route miles as follows:

HOPDAT(MHX,4) = HOPDAT(MHX,3) * HOPDAT(MHX,2)

(No. of manyears) = (No. of route miles) x (No. of manyears/ route mile)

Annual vehicle operating costs are then computed for autos and trucks based upon the amount of vehicle travel (VMT). These costs for vehicle depreciation, maintenance, fuel, insurance, parking, etc. are computed according to the following equations:

<u>Autos</u>: HVOPCT(1,ICTX) = HVOPCS(1,ICTX) * AUTOND(3,4,2)

<u>Trucks</u>: HVOPCT(2,ICTX) = HVOPCS(2,ICTX) * TRCKND(3,4,1)

(Annual operating cost-$) = (Unit operating cost-$/VMT) x (Annual VMT)

where: ICTX = 1 - depreciation
= 2 - maintenance
= 3 - fuel
= 4 - insurance
= 5 - parking
= 6 - other

Annual operating costs for way power, manpower, and maintenance are then determined by this routine for freeways and surface arterials. These costs are computed according to the following three equations:

<u>Power</u>: HWYOCT(MHX,1) = HWYOPC(MHX,1) * HOPDAT(MHX,3)

(Annual operating cost-$) = (Power unit cost - $/route mile) x (route miles)

<u>Manpower</u>: HWYOCT(MHX,2) = HWYOPC(MHX,2) * HOPDAT(MHX,4)

(Annual operating - $) = (Manpower unit cost-$/MY) x (manyears)

<u>Maintenance</u>: HWYOCT(MHX,3) = HWYOPC(MHX,3) * HOPDAT(MHX,3)

(Annual operating cost-$) = (Maintenance unit cost-$/route mile) x (route miles)

HWYOP then computes the total highway annual operating cost by summing the way and vehicle operating costs. The routine then prints all computed operating costs before program control is returned to NULCASE for the calling of the next subroutine TRNOP.

5.1.17 Subroutine TRNOP

TRNOP is the transit operating cost routine of the Highway-Transit Cost program. Its function is to calculate and to output operating costs for a maximum of four transit systems. These cost calculations are performed as outlined in Figure 5-20 .

Initially, this routine reads all transit operating cost input data as defined in variable arrays XOPCU and XOPICD. This data, comprised of transit operating cost units and operating unit-cost data, is printed by TRNOP before any cost calculations are performed. Transit VMT data is also utilized by this routine; this data is received from HWYTRAN through common block /DG/.

For each transit system being considered, this subroutine calculates annual operating costs for way maintenance and operation, manpower, vehicles, and for the maintenance of terminal, storage and parking facilities. Way power operating costs are computed as a function of the vehicle miles travelled by the individual transit systems. Way

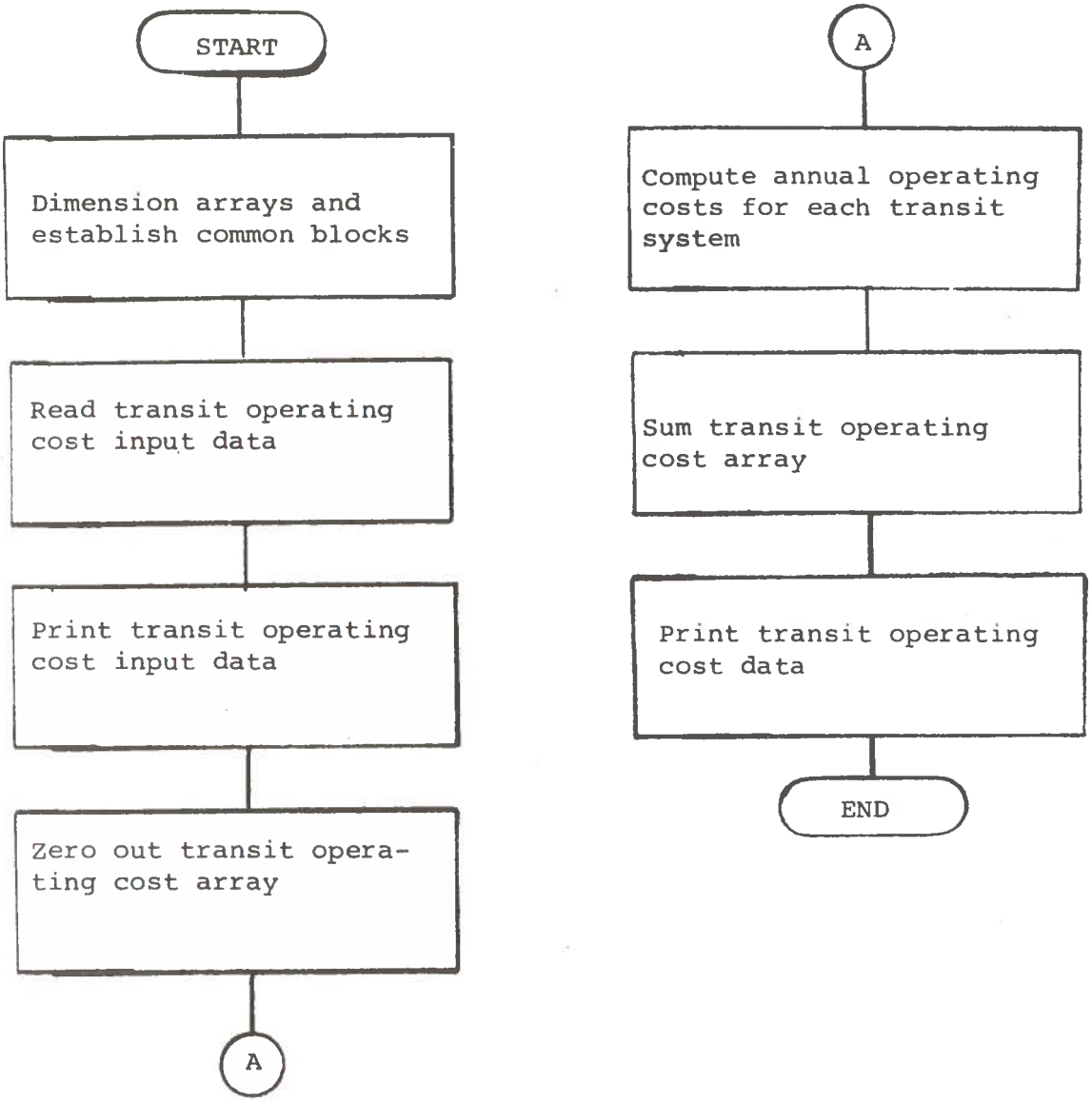


FIGURE 5-20 General Flow Chart - Subroutine TRNOP

manpower and maintenance costs are calculated based upon the number of route or track miles of each transit system.

<u>Way Power</u>: XOPCD(IXV,1,1) = XOPICD(IXV,1,1) * XNDATA(IXV,4,2)
(Annual operating cost-$) = (Unit cost rate-$/VMT) x (Annual VMT)

<u>Way Manpower</u>: XOPCD(IXV,1,2) = XOPICD(IXV,1,2) * XOPCU(IXV,1,1)

<u>Way Maintenance</u>: XOPCD(IXV,1,3) = XOPICD(IXV,1,3) * XOPCU(IXV,1,1)
(Annual operating cost-$) = (Unit cost rate-$/route mile/yr) x (Route miles)

Transit vehicle operating costs and transit manpower costs are also computed as a function of the vehicle miles travelled by each system. This data is computed according to the following equations:

<u>Vehicles</u>: XOPCD(IXV,2,1) = XOPICD(IXV,2,1) * XNDATA(IXV,4,2)

<u>Manpower</u>: XOPCD(IXV,3,1) = XOPICD(IXV,3,1) * XNDATA(IXV,4,2)
(Annual operating cost-$) = (Unit cost rate-$/VMT) x (Annual VMT)

For transit storage, maintenance, and parking facilities annual operating costs are computed based upon the number of vehicle spaces that are required. Terminal operating costs are computed in accordance with the number of terminals that exist for each transit system. These cost

calculations are shown in the following equations:

<u>Storage</u>: XOPCD(IXV,4,1) = XOPICD(IXV,4,1) * XOPCU(IXV,2,1)

<u>Maintenance</u>: XOPCD(IXV,5,1) = XOPICD(IXV,5,1) * XOPCU (IXV,3,1)

<u>Parking</u>: XOPCD(IXV,7,1) = XOPICD(IXV,7,1) * XOPCU(IXV,5,1)

(Annual operating cost-$) = (Unit cost rate-$/vehicle space/yr) x (No. of vehicle spaces)

<u>Terminals</u>: XOPCD(IXV,6,1) = XOPICD(IXV,6,1) * XOPCU(IXV,4,1)

(Annual operating cost-$) = (Unit cost rate-$/terminal/yr) x (No. of terminals)

Once these calculations are performed, subroutine SUM3 is then called to sum the transit operating cost array - XOPCD. This summation produces total operating costs for each transit system and for all transit systems.

At this point, all transit operating cost data is output and program control is returned to the main program NULCASE.

5.1.18 Subroutine SUM

This routine is identical to the SUM subroutine within the New System Cost program. Refer to paragraph 3.2.11 for a description of the data calculations performed by this routine.

5.1.19 Subroutine SUM3

This routine is identical to the SUM3 subroutine within the New System Cost program. A description of the data calculations performed by this routine can be found in paragraph 3.4.12.

5.2 Definition of Input Data

The sixty-one card formats, identified in Figures 5-22 through 5-82, define the total set of input data to the Highway-Transit Cost Program. For each card format type, the input variables, the variable names, the units of input, the field lengths, and field formats are specified.

The input data deck structure is illustrated in Figure 5-21. Preceeding the input data cards are the control cards required to load and execute the program. The control cards are separated from the data cards by an End-of-Record (7/8/9) card. An End-of-File (6/7/8/9) card terminates the entire job input. Paragraph 6.5 describes in detail the control cards that are required to load and execute this program.

Within this program, the input data cards are read by the individual subroutines. The following identifies these subroutines along with their respective card input data:

Subroutine	Input Data
NULCASE	Card Format Types 1-2
HWYCPC	Card Format Types 3-11
TRNCPC	Card Format Types 12-23
TRNCP2	Card Format Types 24-30
HWYTRAN	Card Format Types 31-41
HIGHWY	Card Format Types 42-45

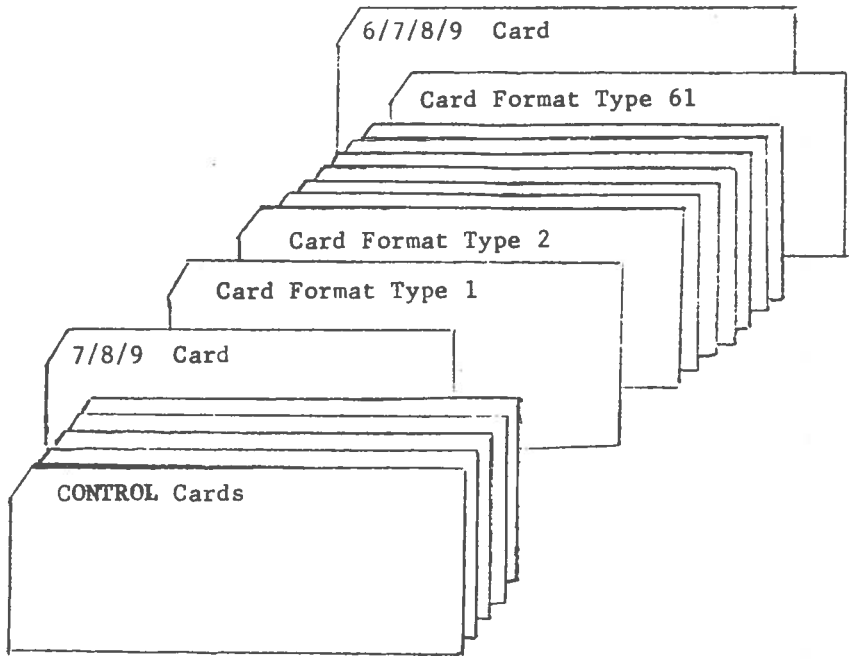


FIGURE 5-21 Highway-Transit Cost Program Input Deck Structure

Subroutine	Input Data
PWRPOL	Card Format Types 46-48
AIRPOL	Card Format Types 49-53
ACCDNT	Card Format Types 54-55
HWYOP	Card Format Types 56-58
TRNOP	Card Format Types 59-61

A typical input data deck, including control cards, is defined in Appendix F.

Card Columns	Format	Variable Name	Definition	Units
1-10	I10	NULLH	Variable to control the execution of the highway capital cost program	---
11-20	I10	NULLX	Variable to control the execution of the transit capital cost program	---
21-30	I10	NULOTH	Variable to control the execution of the highway/transit operating and indirect cost program	---
31-40	I10	JPRNT	Variable to suppress the printing of detailed output tables in highway and transit capital cost program	---
41-80	---	---	Not used	---

Number of cards: 1

Comments:
NULLH = 1 Execute highway capital cost
= 0 Do not execute highway capital cost
NULLX = 1 Execute transit capital cost
= 0 Do not execute transit capital cost
NULOTH = 1 Execute highway/transit operating & indirect
= 0 Do not execute highway/transit operating & indirect
JPRNT = 1 Do not print
= 0 Print

FIGURE 5-2 CARD FORMAT TYPE 1

Card Columns	Format	Variable Name	Definition	Units
1-10	F10.1	NPKHRS	Number of peak hours in a day	Hours
11-20	F10.0	NUMDA	Number of days/year	Days
21-80	---	---	Not used	---

Number of cards: 1

Comments: (24 - NPKHRS) = Number of off-peak hours/day

FIGURE 5-3 CARD FORMAT TYPE 2

Card Columns	Format	Variable Name	Definition	Units
1-80	8A10	ANNHD(KM,KL)	Alphanumeric title names of highway annual capital cost summary tables	---

Number of cards: 3 - Highway right of way
- Highway construction
- Highway annual summary table

FIGURE 5-4 CARD FORMAT TYPE 3

Card Columns	Format	Variable Name	Definition	Units
1-50	5A10	SYS(IALP)	Alphanumeric title name of program	---
51-80	---	---	Not used	---

Number of cards: 1

Comments: Title = Highway Capital Cost Program

FIGURE - CARD FORMAT TYPE 4

Card Columns	Format	Variable Name	Definition	Units
1-50	5A10	NAME(IS,IALP)	Alphanumeric name of various sections within the highway capital cost program	---
51-80	---	---	Not used	---

Number of cards: 2 - Highway right-of-way capital costs
- Highway construction capital costs

Comments: IS = 2

FIGURE 5-16 CARD FORMAT TYPE 5

Card Columns	Format	Variable Name	Definition	Units
1-5	I5	IX	Number of computational sections within the highway capital cost program	---
6-80	---	---	Not used	---

Number of cards: 1

Comments: This index is used in reading the sets of alphanumeric data following this data card.

FIGURE 5-17 CARD FORMAT TYPE 6

Card Columns	Format	Variable Name	Definition	Units
1-5	I5	IV(IX)	Index for the number of land types within computational section IX	---
6-10	I5	JV(IX)	Index for the number of construction types within computational section IX	---
11-15	I5	MV(IX)	Index for the number of highway modes within computational section IX	---
16-80	---	---	Not used	---

Number of cards: 1 card for each computational section (IX) of the highway capital cost program

Comments: IV ≤ 3
JV ≤ 4
MV ≤ 3

FIGURE 5-18 CARD FORMAT TYPE 7

Card Columns	Format	Variable Name	Definition	Units
1-50	5A10	TITLE(IX,IALP,INN)	Alphanumeric title names for the various land types, construction types, and highway modes within each computational section (IX) of the highway capital cost program	---
51-80	---	---	Not used	---

Number of cards: For each computational section (IX), (IV + JV + MV + 3) - one card for each IV,JV,MV plus three cards as summary title cards for all IV, all JV, and all MV.

FIGURE 5-19 CARD FORMAT TYPE 8

Card Columns	Format	Variable Name	Definition	Units
1-10	I10	LIF(IZ)	Life of highway capital cost item (IZ)	Years
11-20	I10	INT(IZ)	Interest rate for highway capital cost item (IZ)	%
21-80	---	---	Not used	---

Number of cards: 4 - IZ = 1 Freeways - ROW
= 2 Surface arterials - ROW
= 3 Freeway - construction
= 4 Surface arterials - construction

FIGURE [illegible] CARD FORMAT TYPE 9

Card Columns	Format	Variable Name	Definition	Units
1-40	4F10.2	HCONM(KI,KJ,KM)	Highway construction miles per land type KI, per construction type KJ, per highway mode KM	Miles
41-80	4F10.2	HCONUD(KI,KJ,KM)	Highway construction unit costs per land type KI, per construction type KJ, per highway mode KM	$/miles

Number of cards: (KI x KM) - 1 card for each land type KI and for each highway mode KM

Comments: Each card contains the construction miles and the construction unit costs for each construction type KJ
KI ≤ 3
KJ ≤ 4
KM ≤ 2

FIGURE [illegible] CARD FORMAT TYPE 11

Card Columns	Format	Variable Name	Definition	Units
1-40	4F10.2	HROWM(KI,KJ,KM)	Highway right-of-way miles per land type KI, per construction type KJ, per highway mode KM	Miles
41-80	4F10.2	HROWUD(KI,KJ,KM)	Highway right-of-way unit costs per land type KI, per construction type KJ, per highway mode KM	$/mile

Number of cards: (KI x KM) - 1 card for each land type KI and for each highway mode KM

Comments: Each card contains the ROW miles and ROW unit cost for each construction type KJ
KI ≤ 3
KJ ≤ 4
KM ≤ 2

FIGURE [illegible] CARD FORMAT TYPE 10

Card Columns	Format	Variable Name	Definition	Units
1-10	F10.2	SQFACR	Number of square feet/acre	Sq.ft/acre
11-80	---	---	Not used	---

Number of cards: 1

FIGURE [illegible] CARD FORMAT TYPE 12

Card Columns	Format	Variable Name	Definition	Units
1-10	I10	LIF(IZ)	Life of transit capital cost item (IZ)	Years
11-20	I10	INT(IZ)	Interest rate for transit capital item (IZ)	%
21-80	---	---	Not used	---

Number of cards: 50

Comments: Refer to Table 5-2 for detailed list of transit capital cost items.

FIGURE 5-34 CARD FORMAT TYPE 13

Card Columns	Format	Variable Name	Definition	Units
1-50	5A10	SYS(IALP)	Alphanumeric title name of program	---
51-80	---	---	Not used	---

Number of cards: 1

Comments: Title: Transit Capital Cost Program

FIGURE 5-35 CARD FORMAT TYPE 14

Card Columns	Format	Variable Name	Definition	Units
1-50	5A10	NAME(IS,IALP)	Alphanumeric name of various sections within the transit capital cost program	---
51-80	---	---	Not used	---

Number of cards: 11

IS = 1 Transit right-of-way
= 2 Transit way construction
= 3 Transit vehicle
= 4 Transit storage land acquisition
= 5 Transit storage construction
= 6 Transit maintenance land acquisition
= 7 Transit maintenance construction
= 8 Transit terminal land acquisition
= 9 Transit terminal construction
=10 Transit parking land acquisition
=11 Transit parking construction

FIGURE 5-36 CARD FORMAT TYPE 15

Card Columns	Format	Variable Name	Definition	Units
1-5	I5	IX	Number of computational sections within transit capital cost program	---
6-80	---	---	Not used	---

Number of cards: 1

Comments: This index is used to establish the number of sets of alphanumeric data to be read following this card

FIGURE 5-37 CARD FORMAT TYPE 16

Card Columns	Format	Variable Name	Definition	Units
1-5	I5	IV(IX)	Index for the number of land types within computational section IX	---
6-10	I5	JV(IX)	Index for the number of construction types (terminal types or parking facility types) within computational section IX	---
11-15	I5	MV(IX)	Index for the number of transit systems within computational section IX	---
16-80	---	---	Not used	

Number of cards: 1 card for each computational section IX

Comments: IV ≤ 3
JV ≤ 4
MV ≤ 4

FIGURE 5-39 CARD FORMAT TYPE 17

Card Columns	Format	Variable Name	Definition	Units
1-40	4F10.2	XROWM(KI,KJ,KM)	Transit right-of-way miles for each construction type KJ	Miles
41-80	4F10.2	XROWUD(KI,KJ,KM)	Transit right-of-way unit cost for each construction type KJ	$/mile

Number of cards: (KI x KM) - 1 card for each land type KI and for each transit system KM

Comments: KI ≤ 3
KJ ≤ 4
KM ≤ 4

FIGURE 5-40 CARD FORMAT TYPE 19

Card Columns	Format	Variable Name	Definition	Units
1-50	5A10	TITLE(IX,IALP,INN)	Alphanumeric title names for the various land types, construction types, (terminal types or parking facility types) and transit system types within each computational section IX	---
51-80	---	---	Not used	---

Number of cards: For each computational section IX, (IV + JV + MV + 3) - 1 card for each IV,JV,MV plus three cards as summary title cards for all IV, all JV, and all MV.

FIGURE 5-41 CARD FORMAT TYPE 18

Card Columns	Format	Variable Name	Definition	Units
1-40	4F10.2	XCONM(KI,KJ,KM)	Transit construction miles per construction type KJ	Miles
41-80	4F10.2	XCONUD(KI,KJ,KM)	Transit construction unit cost per construction type KJ	$/mile

Number of cards: (KI x KM) - 1 card for each land type KI and for each transit system type KM

Comments: KI ≤ 3
KJ ≤ 4
KM ≤ 4

FIGURE 5-41 CARD FORMAT TYPE 20

Card Columns	Format	Variable Name	Definition	Units
1-10	F10.2	NXVEH(KI,KJ,KM)	Number of transit vehicles	Vehicles
11-20	F10.2	XVEUD(KI,KJ,KM)	Transit vehicle unit cost	$/vehicle
21-80	---	---	Not used	---

Number of cards: 1 card for each transit system KM

Comments: KI and KJ are dummy index variables - normally equal to 1

FIGURE 5-12 CARD FORMAT TYPE 21

Card Columns	Format	Variable Name	Definition	Units
1-10	F10.2	XSTVSA(1,KJ,KM)	Transit storage vehicle space area	Sq.ft/veh.-space
11-20	F10.2	XSTCUD(1,KJ,KM)	Transit storage construction unit cost	$/acre
21-80	---	---	Not used	---

Number of cards: 1 card for each transit system KM

Comments: KJ is a dummy index variable normally equal to 1
KM ≤ 4

FIGURE 5-14 CARD FORMAT TYPE 23

Card Columns	Format	Variable Name	Definition	Units
1-30	3F10.2	XSTVSP(KI,KJ,KM)	Transit storage vehicle spaces per land type KI	Vehicle-spaces
31-60	3F10.2	XSTLUD(KI,KJ,KM)	Transit storage land unit cost per land type KI	$/acre
61-80	---	---	Not used	---

Number of cards: 1 card for each transit system type KM

Comments: KJ is a dummy variable - normally equal to 1.
KI ≤ 3
KM ≤ 4

FIGURE 5-13 CARD FORMAT TYPE 22

Card Columns	Format	Variable Name	Definition	Units
1-30	3F10.2	XMAVSP	Transit maintenance area vehicle spaces required per land type KI	Vehicle-space
31-60	3F10.2	XMALUD(KI,KJ,KM)	Transit maintenance area land unit cost per land type KI	$/acre
61-80	---	---	Not used	---

Number of cards: 1 card for each transit system type KM

Comments: KJ is a dummy variable - normally equal to 1.
KI ≤ 3
KM ≤ 4

FIGURE 5-15 CARD FORMAT TYPE 24

Card Columns	Format	Variable Name	Definition	Units
1-40	4F10.2	XMAVSA(1,KJ,KM)	Transit maintenance vehicle-space area per transit system KM	Sq.ft/veh.-space
41-80	4F10.2	XMACUD(1,KJ,KM)	Transit maintenance area construction unit cost	$/sq.ft

Number of cards: 1

Comments: KJ is a dummy variable - normally equal to 1.
KM ≤ 4

FIGURE 5-46 CARD FORMAT TYPE 25

Card Columns	Format	Variable Name	Definition	Units
1-40	4F10.2	XTERLA(1,KJ,KM)	Transit terminal land area per terminal type KJ	Sq.ft/terminal
41-80	---	---	Not used	---

Number of cards: 1 card for each transit system type KM

FIGURE 5-46 CARD FORMAT TYPE 27

Card Columns	Format	Variable Name	Definition	Units
1-40	4F10.2	NXTER(KI,KJ,KM)	Number of transit terminals per terminal type KJ	Terminals
41-80	4F10.2	XTLUD(KI,KJ,KM)	Transit terminal land unit costs per terminal type KJ	$/acre

Number of cards: (KI x KM) - 1 card for each land type KI, and for each transit system type KM.

Comments: KI ≤ 3
KJ ≤ 4
KM ≤ 4

FIGURE 5-47 CARD FORMAT TYPE 26

Card Columns	Format	Variable Name	Definition	Units
1-40	4F10.2	XTERCA(1,KJ,KM)	Transit terminal construction area required for terminal type KJ	Sq.ft/terminal
41-80	4F10.2	XTCUD(1,KJ,KM)	Transit terminal construction unit cost per terminal type KJ	$/sq.ft

Number of cards: 1 card per transit system type KM

Comments: KJ ≤ 4
KM ≤ 4

FIGURE 5-49 CARD FORMAT TYPE 28

Card Columns	Format	Variable Name	Definition	Units
1-40	4F10.2	XTPKVS(KI,KJ,KM)	Transit terminal parking vehicle spaces per parking facility type KJ	Vehicle-spaces
41-80	4F10.2	XTPLUD(KI,KJ,KM)	Transit terminal parking land unit cost per parking facility type KJ	$/veh.-space

Number of cards: (KI x KM) = 1 card per land type KI and per transit system type KM

Comments: KI ≤ 3
KJ ≤ 4
KM ≤ 4

FIGURE 5-50 CARD FORMAT TYPE 29

Card Columns	Format	Variable Name	Definition	Units
1-10	I10	NH	Number of highway related transit systems	---
11-20	I10	NX	Number of transit related transit systems	---
21-30	I10	IPRNT	Variable to suppress printing of various output tables in highway and transit network programs	---
31-80	---	---	Not used	---

Number of cards: 1

Comments: NH - Highway related transit system are those systems that have a VMT that can be assigned to highways
NX - Non-highway related transit systems
NH + NX = Total number of transit systems (max.=4)
IPRNT = 1 Print output tables
= 0 Do not print output tables

FIGURE 5-52 CARD FORMAT TYPE 31

Card Columns	Format	Variable Name	Definition	Units
1-40	4F10.2	XTPCUD(1,KJ,KM)	Transit parking construction unit cost per parking facility type KJ	$/veh.-space
41-80	---	---	Not used	---

Number of cards: 1 card per transit system type KM

FIGURE 5-51 CARD FORMAT TYPE 30

Card Columns	Format	Variable Name	Definition	Units
1-10	F10.2	XSYSCH(1,1)	Vehicle capacity - transit system type 1	Persons/veh
11-20	F10.2	XSYSCH(1,2)	Vehicle capacity - transit system type 2	Persons/veh
21-30	F10.2	XSYSCH(1,3)	Vehicle capacity - transit system type 3	Persons/veh
31-40	F10.2	XSYSCH(1,4)	Vehicle capacity - transit system type 4	Persons/veh
41-80	---	---	Not used	---

Number of cards: 1

FIGURE 5-53 CARD FORMAT TYPE 32

Card Columns	Format	Variable Name	Definition	Units
1-10	4F10.2	XSYSCH(2,IXV)	Average speed for each transit system (IXV)	Miles/hr
41-80	---	---	Not used	---

Number of cards: 1

Comments: IXV ≤ 4

FIGURE 5-54 CARD FORMAT TYPE 33

Card Columns	Format	Variable Name	Definition	Units
1-40	4F10.2	XSYSCH(3,IXV)	Vehicle load factor during peak temporal period for each transit system IXV	Decimal %
41-80	---	---	Not used	---

Number of cards: 1.

Comments: IXV ≤ 4

FIGURE 5-55 CARD FORMAT TYPE 34

Card Columns	Format	Variable Name	Definition	Units
1-40	4F10.2	XSYSCH(4,IXV)	Vehicle load factor during off-peak temporal period for each transit system (IXV)	Decimal %
41-80	---	---	Not used	---

Number of cards: 1.

Comments: IXV ≤ 4

FIGURE 5-56 CARD FORMAT TYPE 35

Card Columns	Format	Variable Name	Definition	Units
1-40	4F10.2	XSYSCH(5,IXV)	Person miles travelled per day for each transit system (IXV)	PMT/day
41-80	---	---	Not used	---

Number of cards: 1.

Comments: IXV ≤ 4

FIGURE 5-57 CARD FORMAT TYPE 36

Card Columns	Format	Variable Name	Definition	Units
1-10	F10.2	XPKPCT	Percent PMT/day for transit during peak temporal period of day	Decimal %
11-80	---	---	Not used	---

Number of cards: 1.

FIGURE 5-58 CARD FORMAT TYPE 37

Card Columns	Format	Variable Name	Definition	Units
1-40	4F10.2	XPTRIP(1,IXV)	Transit person trips during peak temporal period of day per transit system (IXV)	Person trips
41-80	---	---	Not used	---

Number of cards: 1.

Comments: IXV ≤ 4

FIGURE 5-59 CARD FORMAT TYPE 38

Card Columns	Format	Variable Name	Definition	Units
1-40	4F10.2	XPTRIP(2,IXV)	Transit person trips during off-peak temporal period of day per transit system (IXV)	Person trips
41-80	---	---	Not used	---

Number of cards: 1

Comments: IXV ≤ 4

FIGURE 5-60 CARD FORMAT TYPE 39

Card Columns	Format	Variable Name	Definition	Units
1-10	F10.2	XDELAY(1,IXV,1)	Average transit person delay outside vehicle during peak temporal period of day	Minutes/person trips
11-20	F10.2	XDELAY(2,IXV,1)	Average transit person delay outside vehicle during off-peak temporal period of day	Minutes/person trips
21-80	---	---	Same as above for transit system (IXV) = 2,3,4	Minutes/person trips

Number of cards: 1

FIGURE 5-61 CARD FORMAT TYPE 40

Card Columns	Format	Variable Name	Definition	Units
1-10	F10.2	XDELAY(1,IXV,2)	Average transit vehicle delay per peak temporal period of day	Minutes/vehicle trips
11-20	F10.2	XDELAY(2,IXV,2)	Average transit vehicle delay per off-peak temporal period of day	Minutes/vehicle trips
21-80	---	---	Same as above for transit system (IXV) = 2,3,4	Minutes/vehicle trips

Number of cards: 1

FIGURE 5-62 FORMAT TYPE 41

Card Columns	Format	Variable Name	Definition	Units
1-40	4F10.2	XVMTPC(NH)	Percent of transit VMT that is to be assigned to freeways per transit system (NH)	Decimal %
41-80	---	---	Not used	---

Number of cards: 1

Comments: NH $\leq$ 4
(1.0 - XVMTPC) = % of transit VMT that will be assigned to surface arterials

FIGURE 5-64 CARD FORMAT TYPE 43

Card Columns	Format	Variable Name	Definition	Units
1-10	F10.3	HLNDIR(1,1)	Percentage factor to adjust FWY lane miles during peak temporal period	Decimal %
11-20	F10.3	HLNDIR(1,2)	Percentage factor to adjust FWY lane miles during off-peak temporal period	Decimal %
21-30	F10.3	HLNDIR(2,1)	Percentage factor to adjust surface arterial lane miles during peak temporal period	Decimal %
31-40	F10.3	HLNDIR(2,2)	Percentage factor to adjust surface arterial lane miles during off-peak temporal period	Decimal %
41-80	---	---	Not used	---

Number of cards: 1

FIGURE 5-63 CARD FORMAT TYPE 42

Card Columns	Format	Variable Name	Definition	Units
1-10	F10.0	AUTOCH(1)	Auto vehicle capacity	Persons/vehicle
11-15	F5.2	AUTOCH(2)	Auto vehicle load factor	Decimal %
16-25	F10.0	AUTOCH(3)	Auto person trips during peak temporal period of day	Person trip
26-35	F10.0	AUTOCH(4)	Auto person trips during off-peak temporal period of day	Person trips
36-45	F10.0	AUTOCH(5)	Auto person miles travelled/day on freeways	PMT/day
46-55	F10.0	AUTOCH(6)	Auto person miles travelled/day on surface arterials	PMT/day
56-60	F5.2	AUTOCH(7)	Percent auto PMT/day during peak temporal period	Decimal %
61-65	F5.2	AUTOCH(8)	Percent auto VMT in city	Decimal %
66-69	F4.0	AUTOCH(9)	Person delay outside auto	Minutes/person trip
70-73	F4.0	AUTOCH(10)	Auto vehicle delay	Minutes/vehicle trip
74-80	---	---	Not used	

Number of cards: 1

Comments: Auto person trips (peak and off-peak) are required to be non-zero.
Truck VMT = [(auto VMT/AUTOCH(5)) - auto VMT]

FIGURE 5-65 CARD FORMAT TYPE 44

Card Columns	Format	Variable Name	Definition	Units
1-10	F10.2	HNCHAR(1)	Lane miles of freeway	Lane miles
11-20	F10.2	HNCHAR(2)	Lane miles of surface arterials	Lane miles
21-30	F10.2	HNCHAR(3)	Average freeway capacity	Veh/hr/lane mi.
31-40	F10.2	HNCHAR(4)	Average surface arterial capacity	Veh/hr/lane mi.
41-80	---	---	Not used	---

Number of cards: 1

FIGURE 5-66 CARD FORMAT TYPE 45

Card Columns	Format	Variable Name	Definition	Units
1-10	F10.1	XPCVCH(IXV,1)	Transit vehicle power characteristics - vehicle weight	Lbs.
11-20	F10.3	XPCVCH(IXV,2)	Vehicle tire friction coefficient	Unitless
21-30	F10.3	XPCVCH(IXV,3)	Vehicle aerodynamic friction coefficient	Unitless
31-40	F10.1	XPCVCH(IXV,4)	Vehicle height	Ft.
41-50	F10.1	XPCVCH(IXV,5)	Vehicle width	Ft.
51-60	F10.0	XPCVCH(IXV,6)	Vehicle power code	---
61-80	---	---	Not used	---

Number of cards: 1 card for each transit system IXV

Comments: XPCVCH(IXV,6) - Power code = 1. Electrical
= 0. IC Engine

FIGURE 5-68 CARD FORMAT TYPE 47

Card Columns	Format	Variable Name	Definition	Units
1-4	4X	---	Not used	---
5-8	I4	IXV	Number of transit systems for power consumption claculations	---
9-12	I4	NHV	Number of highway vehicles	---
13-80	---	---	Not used	

Number of cards: 1

Comments: IXV ≤ NH + NX
IXV ≤ 4
NHV ≤ 2 NHV = 1 Autos
= 2 Trucks

FIGURE 5-67 CARD FORMAT TYPE 46

Card Columns	Format	Variable Name	Definition	Units
1-10	F10.1	HPCVCH(NHV,1)	Highway vehicle power characteristics - vehicle weight	Lbs.
11-20	F10.3	HPCVCH(NHV,2)	Vehicle tire friction coefficient	Unitless
21-30	F10.3	HPCVCH(NHV,3)	Vehicle aerodynamic friction coefficient	Unitless
31-40	F10.1	HPCVCH(NHV,4)	Vehicle height	Ft.
41-50	F10.1	HPCVCH(NHV,5)	Vehicle width	Ft.
51-80	---	---	Not used	---

Number of cards: 1 card for each highway vehicle NHV

Comments: The power code for highway vehicles is assumed to be IC
NHV ≤ 2 NHV = 1 Autos
= 2 Trucks

FIGURE 5-69 CARD FORMAT TYPE 48

Card Columns	Format	Variable Name	Definition	Units
1-12	4I3	IPOLCD(IXV,1,2)	Pollution codes for each transit system (IXV)	---
13-80	---	---	Not used	---

Number of cards: 1

Comments: Pollution codes (IPOLCD) = 1 IC - Gas
= 2 IC - Diesel
= 3 Electric

IXV ≤ 4

FIGURE 5-70 CARD FORMAT TYPE 49

Card Columns	Format	Variable Name	Definition	Units
1-10	F10.6	POLRAT(1,1)	IC - Gas pollution rate - CO	Lbs/VMT
11-20	F10.6	POLRAT(1,2)	IC - Gas pollution rate - HC	Lbs/VMT
21-30	F10.6	POLRAT(1,3)	IC - Gas pollution rate - NOX	Lbs/VMT
31-80	---	---	Not used	---

Number of cards: 1

FIGURE 5-72 CARD FORMAT TYPE 51

Card Columns	Format	Variable Name	Definition	Units
1-6	2I3	IPOLCD(NHV,1,3)	Pollution codes for highway vehicles	---
7-80	---	---	Not used	---

Number of cards: 1

Comments: Pollution codes (IPOLCD) = 1 IC - Gas
= 2 IC - Diesel

NHV ≤ 2 where NHV = 1 - Autos
= 2 - Trucks

FIGURE 5-71 CARD FORMAT TYPE 50

Card Columns	Format	Variable Name	Definition	Units
1-10	F10.6	POLRAT(2,1)	IC - Diesel pollution rate - CO	Lbs/VMT
11-20	F10.6	POLRAT(2,2)	IC - Diesel pollution rate - HC	Lbs/VMT
21-30	F10.6	POLRAT(2,3)	IC - Diesel pollution rate - NOX	Lbs/VMT
31-80	---	---	Not used	---

Number of cards: 1

FIGURE 5-73 CARD FORMAT TYPE 52

Card Columns	Format	Variable Name	Definition	Units
1-10	F10.6	POLRAT(3,1)	Electric vehicle pollution rate - particulate	Lbs/KWHR
11-20	F10.6	POLRAT(3,2)	Electric vehicle pollution rate - SO_2	Lbs/KWHR
21-30	F10.6	POLRAT(3,3)	Electric vehicle pollution rate - NOX	Lbs/KWHR
31-80	---	---	Not used	---

Number of cards: 1

FIGURE 5-74 CARD FORMAT TYPE 53

Card Columns	Format	Variable Name	Definition	Units
1-8	F8.2	HACRAT(MH,1,ITP)	Accident rate of fatal accidents per MH per peak temporal period (ITP=1)	No. of fatal accidents per 100 million VMT
9-16	F8.2	HACRAT(MH,2,ITP)	Accident rate of injury accidents per MH per peak temporal period (ITP=1)	No. of injury accidents per 100 million VMT
17-24	F8.2	HACRAT(MH,3,ITP)	Accident rate of property damage accidents per MH per peak temporal period (ITP=1)	No. of property damage accidents per 100 million VMT
25-32	F8.2	HACRAT(MH,4,ITP)	Accident rate of fatalities per MH per peak temporal period (ITP=1)	No. of fatalities per 100 million VMT
33-40	F8.2	HACRAT(MH,5,ITP)	Accident rate of injuries per MH per peak temporal period (ITP=1)	No. of injuries per 100 million VMT
41-80	5F8.2	Same as above except for off-peak temporal period (ITP=2)		---

Number of cards: 1 card per MH where MH = 1 - Freeways
MH = 2 - Surface arterial

FIGURE 5-75 CARD FORMAT TYPE 54

Card Columns	Format	Variable Name	Definition	Units
1-10	F10.2	XACRAT(IXV,1)	Accident rate of transit traffic accidents per transit system IXV	No. accidents per million VMT
11-20	F10.2	XACRAT(IXV,2)	Accident rate of passenger accidents per transit system IXV	No. accidents per million person trips
21-30	F10.3	XACRAT(IXV,3)	Accident rate of passenger fatalities per transit system IXV	No. fatalities per million person trips
31-40	F10.3	XACRAT(IXV,4)	Accident rate of non-passenger fatalities per transit system IXV	No. fatalities per million person trips
41-80	---	---	Not used	---

Number of cards: 1 card for each transit system IXV where IXV ≤ 4

FIGURE 5-76 CARD FORMAT TYPE 55

Card Columns	Format	Variable Name	Definition	Units
1-5	I5	NHV	Number of highway vehicle types	---
6-15	F10.3	HOPDAT(1,1)	Average number of lanes for freeways	Lanes
16-25	F10.3	HOPDAT(2,1)	Average number of lanes for surface arterials	Lanes
26-35	F10.3	HOPDAT(1,2)	Average number of manyears/route miles for freeways	My/route miles
36-45	F10.3	HOPDAT(2,2)	Average number of manyears/route miles for surface arterials	My/route miles
46-80	---	---	Not used	---

Number of cards: 1

Comments: NHV ≤ 2

FIGURE 5-77 CARD FORMAT TYPE 56

Card Columns	Format	Variable Name	Definition	Units
1-10	F10.3	HVOPCS(NHV,1)	Highway vehicle unit operating costs - depreciation	$/VMT
11-20	F10.3	HVOPCS(NHV,2)	Highway vehicle unit operating costs - maintenance	$/VMT
21-30	F10.3	HVOPCS(NHV,3)	Highway vehicle unit operating costs - fuel	$/VMT
31-40	F10.3	HVOPCS(NHV,4)	Highway vehicle unit operating costs - insurance	$/VMT
41-50	F10.3	HVOPCS(NHV,5)	Highway vehicle unit operating costs - parking	$/VMT
51-60	F10.3	HVOPCS(NHV,6)	Highway vehicle unit operating costs - other	$/VMT
61-80	---	---	Not used	---

Number of cards: 1 card for each highway vehicle type (NHV)
NHV = 1 Autos
= 2 Trucks

FIGURE 5-78 CARD FORMAT TYPE 57

Card Columns	Format	Variable Name	Definition	Units
1-10	F10.3	HWYOPC(MH,1)	Highway way unit operating cost - power	$/route mile/yr
11-20	F10.3	HWYOPC(MH,2)	Highway way unit operating cost - manpower	$/manyear
21-30	F10.3	HWYOPC(MH,3)	Highway way unit operating cost - maintenance	$/route mile/yr
31-80	---	---	Not used	---

Number of cards: 1 card per highway mode (MH)
MH = 1 Freeways
= 2 Surface arterials

FIGURE 5-79 CARD FORMAT TYPE 58

Card Columns	Format	Variable Name	Definition	Units
1-10	F10.0	XOPCU(IXV,1,1)	Number of route or track miles	Miles
11-20	F10.0	XOPCU(IXV,2,1)	Number of storage vehicle-spaces	Vehicle-spaces
21-30	F10.0	XOPCU(IXV,3,1)	Number of maintenance vehicle-spaces	Vehicle-spaces
31-36	4F4.0	XOPCU(IXV,4,MX)	Number of terminals per terminal type (MX) where MX ≤ 4	Terminals
37-70	4F8.0	XOPCU(IXV,5,MX)	Number of parking vehicle-spaces per parking facility type (MX) where MX ≤ 4	Vehicle-spaces
71-80	---	---	Not used	---

Number of cards: 1 card per each transit system IXV (IXV ≤ 4)

FIGURE 5-60 CARD FORMAT TYPE 59

Card Columns	Format	Variable Name	Definition	Units
1-6	F6.3	XOPICD(IXV,1,1)	Transit operating unit cost data - way power	$/VMT
7-13	F7.2	XOPICD(IXV,1,2)	Transit operating unit cost data - way manpower	$/route mile/yr
14-22	F9.0	XOPICD(IXV,1,3)	Transit operating unit cost data - way maintenance	$/route mile/yr
23-29	F7.3	XOPICD(IXV,2,1)	Transit operating unit cost data - vehicle depreciation	$/VMT
30-36	F7.3	XOPICD(IXV,2,2)	Transit operating unit cost data - vehicle other	$/VMT
37-43	F7.0	XOPICD(IXV,3,1)	Transit operating unit cost data - manpower	$/VMT
44-50	F7.0	XOPICD(IXV,4,1)	Transit operating unit cost data - storage	$/veh-space/yr
51-57	F7.0	XOPICD(IXV,5,1)	Transit operating unit cost data - maintenance	$/veh-space/yr
58-80	---	---	Not used	---

Number of cards: 1 card per each transit system IXV (IXV ≤ 4)

Comments: Card Format Type 60 and 61 are interleaved in the input data stream

FIGURE 5-61 CARD FORMAT TYPE 60

Card Columns	Format	Variable Name	Definition	Units
1-28	4F7.0	XOPICD(IXV,6,KX)	Transit operating terminal unit operating costs per terminal type KX where KX ≤ 4.	$/terminal/yr
29-58	4F5.0	XOPICD(IXV,7,KX)	Transit operating parking unit unit operating costs per parking facility type KX where KX ≤ 4	$/vehicle-space/yr
59-80	---	---	Not used	---

Number of cards: 1 card per each transit system IXV (IXV ≤ 4)

Comments: Card Format Type 60 and 61 are interleaved in the input data stream

FIGURE 5-62 CARD FORMAT TYPE 61

5.3 Description of Output Data

Appendix G contains selected output data tables of the Highway-Transit Cost Program.

The data output tables of this program are printed in the following order:

Data Output	Appendix G Page Reference
Highway Capital Cost Data	G1 - G6
Transit Capital Cost Data	G7 - G28
Network Analysis Data	G29 - G39
Power Consumption Data	G40 - G41
Air Pollution Data	G42 - G44
Accident Data	G45 - G47
Highway Operating Cost Data	G48 - G49
Transit Operating Cost Data	G50 - G52

6 TEAM Model User's Guide

6.1 Introduction

This portion of the model documentation describes the procedures for the execution of the TEAM model, for the creation and manipulation of program files, and the procedures for performing program updates. Reference notes on the CDC-6600 Computer System and the Import/Export UT-200 terminal operations are also presented.

6.2 CDC 6600 SYSTEM DESCRIPTION

6.2.1 COMPUTER DESCRIPTION

CDC 6600 is one of the CDC 6000 series computers. It has one central processor which contains 24 operating registers. This offers greater efficiency since the operating registers are linked to ten individual arithematic sections (Figure 6-1) which can operate simultaneously. The central processor can hold up to 7 user jobs concurrently and executes instructions which manipulate the 24 operating registers, retrieve information from central memory, and store information back into central memory. Information is stored in central memory in words of 60 binary digits (bits) and the memory access time is 100 nanoseconds (10^{-7} seconds). The central processor shares access to the central memory with ten peripheral processors. Unlike the central processor, each peripheral processor has its own memory. Peripheral processors are used to transfer the information between the central memory and peripheral devices such as disks, magnetic tape, etc. They relieve the central processor of all input/output tasks, so that it can perform data calculations while the peripheral processors are engaged in the input/output functions. Peripheral processors handle input from an output to the peripheral devices through data channels.

6.2.2 OPERATING SYSTEM

CDC 6600 utilizes an operating system called SCOPE (Supervisory Control of Program Execution). SCOPE is an operating system which monitors and controls compilation or assembly, loading, execution and output of all the programs submitted to the computer. It controls the input/output operations for on-site or remote computer terminals and controls storage assignment tasks. In addition, it prints the DAYFILE of each job run (Figure 6-2), allows the

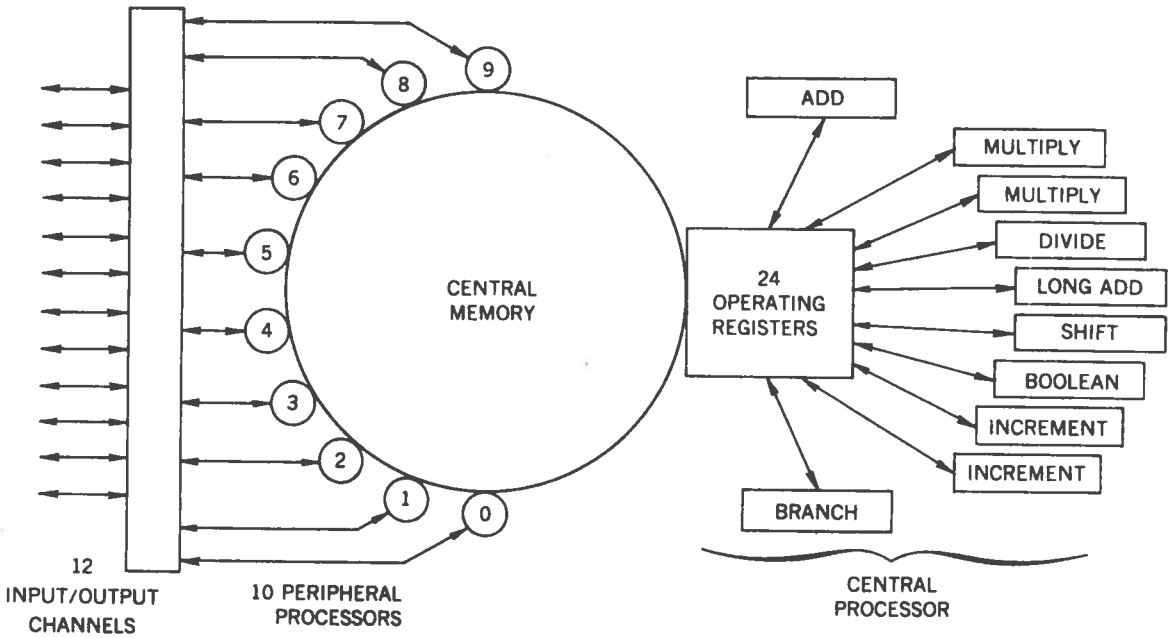


FIGURE 6-1. CDC 6600 COMPUTER SYSTEM

```
  06/15/72 +DS SCOPE 3.3          CYC N      10/27/71
16.04.44.WSCMOFJ
16.04.44.IP   000016 INPUT UNITS USED.
16.04.44.$SEQUENCE,SCM. TEST
16.04.44.
16.04.44.$CHARGE
16.04.44.
16.04.44.TST,P10,IO100,CM40000,T20.
16.04.44.
16.04.44.MAP(OFF)
16.04.44.RUN(S)
16.04.45.FILE OPENED - OUTPUT
16.04.45.FILE OPENED - LGO
16.04.46.REDUCE.
16.04.46.LGO.
16.04.49.CM   040000      CM CELLS USED.
16.04.49.CP   000000.699 CP SEC. USED.
16.04.49.IO   000003.200 IO SEC. USED.
16.04.49.SS   000001.099 SYSTEM SEC. USED.
16.04.51.EXIT
16.04.51.OP   000016 STORAGE DATA BLOCKS ON FILE  OUTPUT
16.04.51.CM   012700      CM CELLS USED.
16.04.51.CP   000001.420 CP SEC. USED.
16.04.51.IO   000004.000 IO SEC. USED.
16.04.51.SS   000001.853 SYSTEM SEC. USED.
16.04.51.AC -  END OF JOB.
```

FIGURE 6-2. DAY FILE

operator at the computer console to modify stored programs, protects information from access by unauthorized persons through the use of private permanent files and performs other miscellaneous functions. SCOPE accepts programs written in the COMPASS assembly language, or in compiler or procedure oriented languages such as: FORTRAN, FORTRAN Extended, COBOL, ALGOL, SIMSCRIPT, APT, OPTIMA, SORT/MERGE and PERT/TIME. SCOPE programs are written in COMPASS assembly language. If SCOPE were a small program that did not require much space for storage, it would be convenient to store most of this operating system in central memory. However since it does require extensive core storage, only system tables and pointer words which are continuously referenced are permanently stored in central memory. They are particularly used for communication between subroutines frequently used by the operating system and programs submitted by the users, and for intercommunication between other SCOPE routines. Most of these tables and pointers reside in low core (the beginning address in central memory); others reside in HIGH CORE (the last areas, or highest numbered addresses in central memory) as shown in Figure 6-3. Neither the low core nor the high core areas occupied by SCOPE can be overwritten by the users programs. Memory areas in ten peripheral processors are used to provide storage for the transient programs that are called by SCOPE from disc storage to perform input and output functions. Once these routines have been executed, they can be overwritten by other routines. The MONITOR routine which controls all other SCOPE routines and subroutines resides in peripheral processor zero. The MONITOR routine is linked to the other routines in the other PP's by the communication areas in central memory. Those portions of SCOPE that do not reside in central memory or the peripheral processors are stored on the system disc. The disc is an allocatable device that also holds

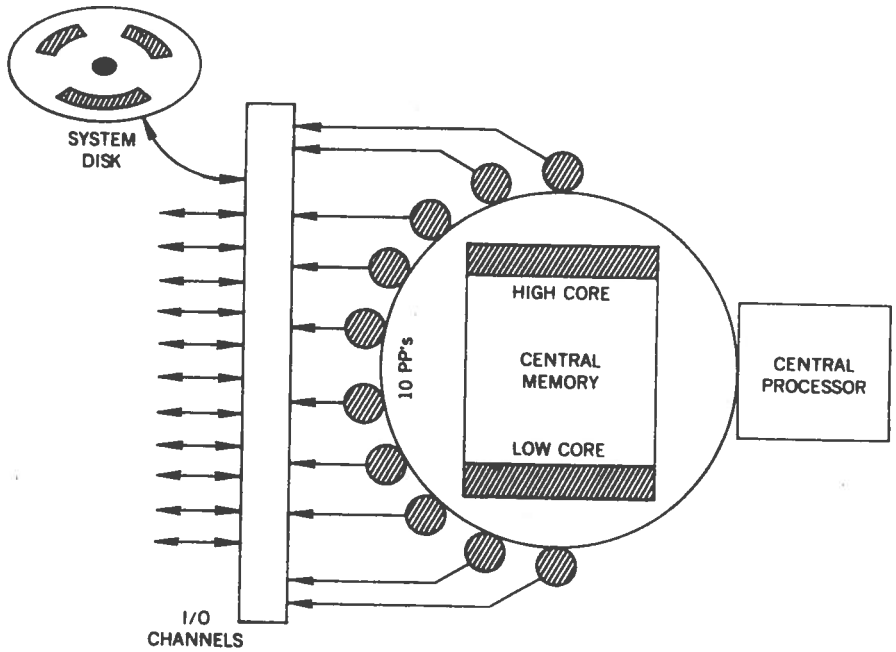


FIGURE 6-3. SCOPE RESIDENCE (SHADED AREA)

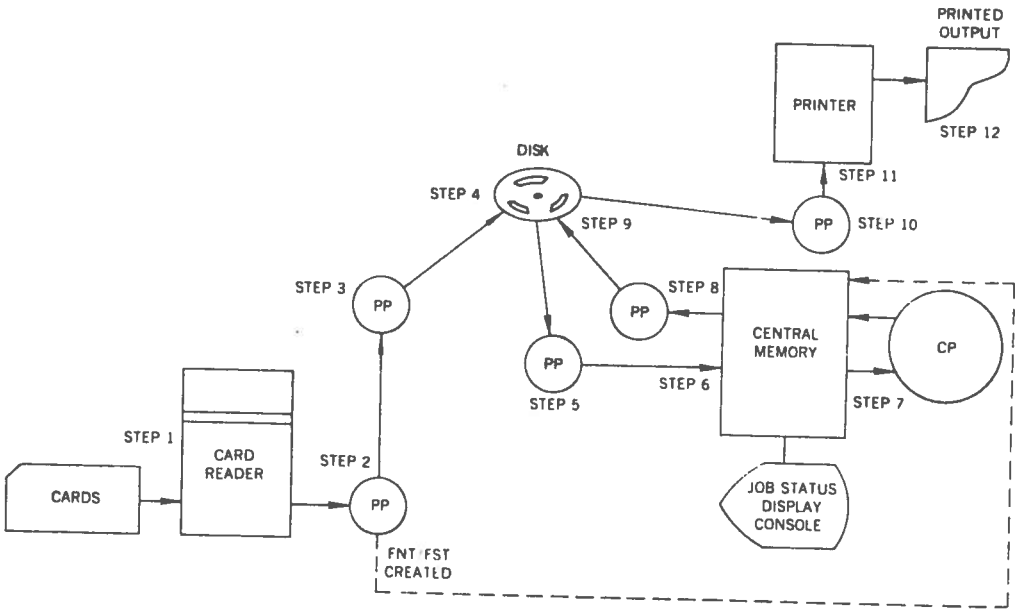


FIGURE 6-4 TYPICAL JOB FLOW

user programs and data as well as SCOPE routines.

6.2.3 JOB INPUT, EXECUTION AND OUTPUT:

Each job submitted through the computer is identified by a JOB NAME and is accepted as a unique file by the SCOPE operating system. Each job which progresses through the computer traces a path which as shown in Figure 6-4. The information on the cards is input through a PP at step 2. SCOPE gets the job name, core requirements, priority in relation to other jobs and creates a new FNT/FST entry which contains this and other information about the job.

After the FNT/FST has been initiated, SCOPE moves the job to another PP (step 3) which places it on a disk storage area (step 4) until central memory facilities are available to run the job. Each program is executed within an area (identified by a CONTROL POINT) of central memory partitioned off by SCOPE for that job alone. As soon as a control point and sufficient storage are available, SCOPE will move the job from the disk, through a PP (step 5), into central memory (step 6), assigning the job to that control point. SCOPE will then shift other jobs around within central memory so that the space will be made contiguous to accommodate any new job. This shifting process is called DYNAMIC RELOCATION. The progress of the job can be monitored on the Job Status Display, the information about the job appears next to the number of CONTROL POINT to which the job is assigned. There is an area in low core called the CONTROL POINT AREA which contains information about the job such as its name, length, starting address and other parameters. Within the control point area, there is a 16-word EXCHANGE package that enables the job to communicate with the central processor. The CONTROL POINT and CONTROL POINT AREA are shown in Figure 6-5.

Output from the job is transferred from control memory through a PP (step 8) and onto a disk (step 9). It is then considered part of the OUTPUT QUEUE, which is the logical line-up of job output awaiting access to peripheral output devices. Unless the user designates otherwise, the output is always assigned to the printer. When a printer is available, the output will go through a PP (step 10) to the printer (step 11) where it will be printed in lines of 136 characters. The last portion of the printer output will show the final contents of the dayfile. Besides showing various items pertaining to the job execution, the dayfile also contains the central processor and the peripheral processor times used by the job.

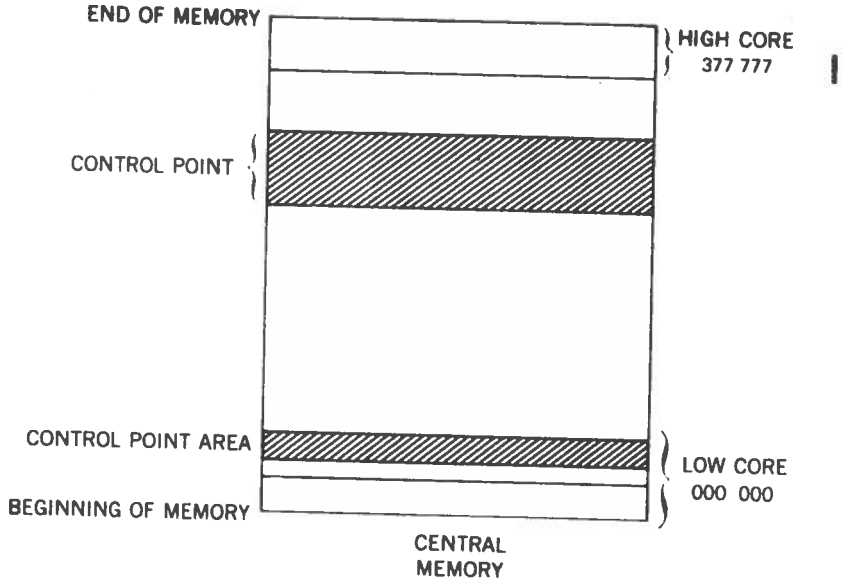


FIGURE 6-5. CONTROL POINT AND CONTROL POINT AREA

6.2.4 Core Map Generation:

If the job is compiled properly, the loader links various subprograms, subroutines etc. and creates a core map. The core map gives a programmer the information about the starting addresses and entry addresses of the various programs and system routines and the addresses of the common blocks. A sample core map is shown in Figure 6-6 with each of the identified labels defined as follows:

1. Time of day (24-hour clock) when program was loaded
2. Type of loading: Normal, Overlay, or Segment
3. Overlay or segment level numbers (blank for normal loading)
4. TYPE is CONTROL if load was requested via control card; otherwise TYPE is address to user's call to loader
5. Meaningless if TYPE is CONTROL; otherwise a 2-word loader request appears above USER---++---CALL
6. First word address of loaded user program
7. Last word address of loaded user program
8. First word address of blank common
9. Length of blank common
10. First word address of loader and loader tables. Table address may be reassigned to blank common.
11. Name and first word address of all subprograms loaded. These may be user and/or library subprograms.
12. Labeled common name and first word address of each block. Each is listed in the same line as the subprogram in which the block is first defined.
13. Entry point name and address of each subprogram
14. List of subprograms referring to entry points and addresses at which each reference is made.

```
                 (1)        (2)                (3)        (4)                (5)        (5)                (6)      (7)      (8)      (8)
CORE MAP   16.52.31.  NORMAL                          CONTROL                                          000100   011076   000000   000000
      (10) ---TIME---LOAD MODE --L1--L2-----TYPE-------------USER-------CALL-------------FWA LOAD--LWA LOAD--BLNK COMN--LENGTH--
           FWA LOADER  054010  FWA TABLES  052230
           -PROGRAM----ADDRESS-                  (11)          --LABELED--COMMON--
           GEN         000720                                (12) FP          000100
                                                                  OP          000244
                                                                  IN          000554
           DUMMY       003045
           DEBUG       003052
           SYSTEM      005154                                     SCOPE2      005154
           OUTPTC      006231
           SIOB        006322
           KODER       007651
           GETBA       011057
           --ENTRY-----ADDRESS-                                          REFERENCES
           GEN         000721  (13)
           DUMMY       003045            GEN         000763  (14)

           POINT1      003046
           POINT2      003047
           POINT3      003050
           TSCARD      003052
           SETADR      003722
           WRDEBUG     004261
           SNACE       004660
           STORE       004440
           RESTORE     004542
           RB0         004603
           RA0         004613
           RX0         004623
           TSSEBUP     004382
           OVFLAG      003713
           ZZGETPL     004620
           PCNTR       004244
           ZSQUIZW     004293
           QBNTRY      005155            GEN         000722

           SYSTEM      005301            OUTPTC      000306
                                         KODER       010663
           ---UNSATISFIED EXTERNALS-----                                 REFERENCES
      -2499     -2496     -2491     -2484     -2475     -2464     -2451     -2436
      -2419     -2400     -2379     -2356     -2331     -2304     -2275     -2244
      -2211     -2176     -2139     -2100     -2059     -2016     -1971     -1924
      -1875     -1824     -1771     -1716     -1659     -1600     -1539     -1476
      -1411     -1344     -1275     -1204     -1131     -1056     -979      -900
       -819      -736      -651      -564      -475      -384      -291      -196
        -99         0       101       204       309       416       525       636
       -749       864       981      1100      1221      1344      1469      1596
       1725      1856      1989      2124      2261      2400      2541      2684
       2829      2976      3125      3276      3429      3584      3741      3900
       4061      4224      4389      4556      4725      4896      5069      5244
       5421      5600      5781      5964      6149      6336      6525      6716
       6909      7104      7301      7500
```

FIGURE 6-6. CORE MAP

6.3 JOB PROCESSING CONTROL CARDS

The processing of a job is controlled by a set of control cards which determine the sequence of program execution. On a CDC 6600 Computer, the first record of every job consists of set control cards which control the job flow. The processing of all other records in the job file is dependent on the control cards in the first record of this job file. A sample job with data is shown in Figure 6-7. The first three control cards of any job are normally the SEQUENCE, CHARGE and JOB cards. These three cards, along with directive control cards as the compile, load and excute cards, form the first record of the job. Each record in the job is separated by an End of Record card which has a 7, 8 and 9 punched in column 1 and is identified as a (7,8,9) card. All control cards should begin in column 1 and are terminated by a period or, if a parenthesized list appears a closing parenthesis. The last card in the job is an End of File Card which has a 6,7,8, and 9 punched in column 1 and is referred to as (6,7,8,9) card.

6.3.1 SEQUENCE Card

The first card in the job is a SEQUENCE card. This has a $ sign in column 1 and the word SEQUENCE between columns 2 to 9 followed by a comma and the Job Name followed by a period. The first character of the Job Name should be alphabetic. The operating system picks up the first three characters of the Job Name and adds three more unique characters creating a unique six character job name.

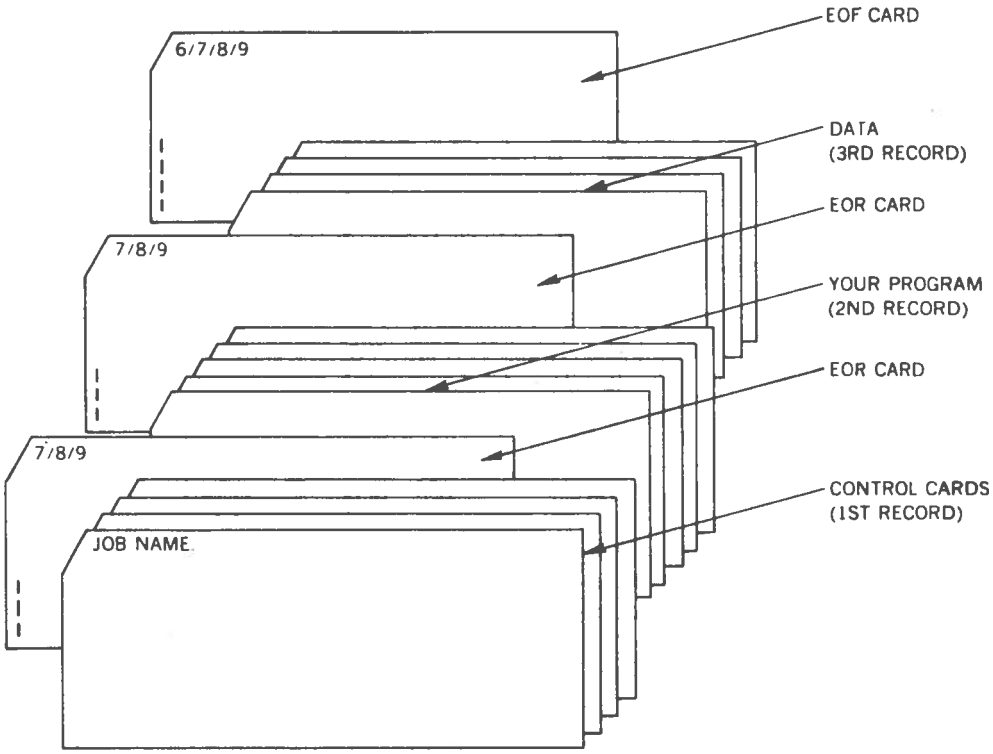


FIGURE 6-7 JOB WITH DATA

6.3.2 CHARGE Card

This is the second card in any job. First column contains the character $. Columns 2 to 7 contain the word CHARGE which is followed by a comma the charge number and a period. This card is basically there to account for the total core used by the job, the system seconds, and the I/O used by the job.

6.3.3 JOB Card

The JOB card assumes the following format and contains such system information as the job priority, central processor time limit, and the central memory core requirement.

JOB, priority, time limit, field length.

The fields are separated by commas and the last field is terminated with a period. If this information is left out, priority 1, time limit 1 minute, field length 40000_8 is assumed.

- o Job: Identifies this as a control card.
- o Priority: 2,4,6, or 10. Highest priority at which a job can be run is 10 and the lowest priority is 2.
- o Time Limit: Total <u>central processing</u> time for the job in seconds - a maximum of 5 octal digits. The octal value in hundreds is approximately the time in minutes. The time is actually rounded up to a multiple of 10_8 by the system.

  Time must suffice for the whole job, including all compilation, loading, and execution.

o Field Length

Total field length of a job, in octal, maximum of 6 octal digits. The field length (storage requirements) is rounded up to a multiple of 100_8 by the system.

If the job includes a compilation, the field length must include the compiler. The standard amount $40,000_8$ is frequently sufficient. If a listing is requested when the RUN compiler is called, the compiler prints out the amount of storage that was <u>not</u> needed for each compiled program and the program length including buffers of each routine. If the program is loaded (by execution or a NOGO card) the program length (field length) for the entire program is given. Thus, for future runs the field length should be adjusted accordingly.

6.3.4 FORTRAN COMPILE Card - RUN

The FORTRAN compiler is called by the control card:

RUN(cm,fl,bl,if,of,rf,lc,as,cs)

o cm - Compiler mode option (if omitted, assume G; if unrecognized, assume S)

- G compile and execute.
- S compile with source list, no execute.
- P compile with source list and punch deck on file PUNCHB, no execute.
- L compile with source and object list, no execute.
- M compile with source and object list, produce a punch deck on file PUNCHB, no execute.

o fl - Object program field length (octal); if omitted, it is set equal to the field length at compile time.

o bl - Object program I/O buffer lengths (octal) if omitted, assumed to be 2022_8. Minimum value is 1001_8.

- o if - File name for compiler input: if omitted, assumed to be INPUT.
- o of - File name for compiler output; if omitted assumed to be OUTPUT.
- o rf - File name on which the binary information is always written; if omitted, assumed to be LGO.
- o lc - In order to specify the line limit at any time, a new control card has been implemented (see LC paramenter on Execute or Load and Go Card). Therefore, the lc aprameter will be bypassed. It will act as a dummy, and space must be allowed for it if the parameters as or cs are specified.
- o as - If nonzero or non blank, the ASA switch causes the ASA I/O list/format interaction at execution time. It has no effect on the compilation method.
- o es - Cross-reference switch. If nonzero, a cross-reference listing is produced.

Compiler output, except in the G mode, includes a reproduction of the source program, a variable map, and indications of fatal and nonfatal errors detected during compilation. If the G mode is selected, all output is suppressed unless fatal errors are detected, in which case the output is the same as indicated for the S mode of compilation. If the L mode is selected, the output includes an octal list of the compiled instructions.

A copy of the compiled programs is always left in disk storage as a binary record on a file name either LGO or the name specified as the rf parameter in the call to the compiler. The compiled program may be called and executed repeatedly by using the name of the load-and-go file. In the output file at the end of compilation of each subprogram, the compiler indicates the amount of unused compilation space.

Examples:

(1) To compile, requesting an object program field for production smaller than for the compilation

RUN(s,fl)

where fl is the field length during execution time

(2) To get COMPILE listings with the binary output on the file LGO

RUN(s)

(3) To compile, withinput from a file called COMPILE and binary output on the file LGO.

RUN(S,,,COMPILE)

6.3.5 LOAD and EXECUTE CARDS

A binary file lfn is loaded in central memory by the following instruction.

LOAD(lfn)

If lfn is INPUT,loading begins from the current position of the file; all other files are rewound by the system prior to loading. Loading terminates when the end-of-information or an empty record is encountered. All loader directives must appear in the named file before any subprogram. These directives whether overlay, segment, or section processing are required. Overlays, segment, and relocatable binary decks may be loaded with the LOAD control card; the first record of the file lfn specifies the kind of loading operations to be performed.

If subprograms are to loaded from more than one file, more than one LOAD card is needed, but the first record of the first file always determines the kind of loading for all subsequent LOAD cards.

Once a program has been loaded, it is executed by the following request:

EXECUTE.

The combined load/execute is just a shorter version for the separate load and execute requests e.g. the request,

```
LOAD(lgn)
EXECUTE.
```

is equivalent to a single request

```
lgn
```

and accomplishes the same result as the previous request.

For Overlay operations, program execution begins in the main overlay. Subsequent overlays must be loaded by user calls from these programs or from the main overlay.

6.3.6 MISCELLANEOUS CONTROL CARDS

Limit:

```
LIMIT (n)
```

This card limits the amount of mass storage assigned to a job - n is the decimal number of word blocks to which job is limited. Each word block equals 4096 words of central memory.

Map:

This card requests cancels, full or partial map of the job in central memory.

```
MAP (p)
```

where p may be ON, OFF or PART which selects map , no map or partial map option.

NOGO:

This card completes loading and linking of all elements needed for program execution but suspends job execution.

```
NOGO.
```

Reduce:

This card reduces the field length (memory requirement) of a job to a minimum amount needed for execution.

```
REDUCE.
```

Load:

This card loads program from the file lfn into central memory. In the New System COST Program LOAD control cards are used to load the overlays in core.

```
LOAD (lfn)
```

Unload:

```
UNLOAD (lfn1, lfn2,.....lfnn)
```

Releases files and the associated devices from the job.

Rewind:

```
REWIND (lfn1, lfn2, .... lfnn)
```

This card rewinds the files named in the list.

XPAUSE:

This card is used to send message to the central site about the job. It is terminated by a period and is then followed by the message e.g.

```
XPAUSE. DIVERT OUTPUT TO CENTRAL SITE.
```

Comment:

This card is used to add comments about the job which eventually appear in the DAY FILE.

```
COMMENT.  NEW SYSTEM COST PROGRAM
```

The word COMMENT is followed by a terminator (period) and then the comment is inserted.

6.4 Creation of Program Files

This section describes the procedures for the creation of source and binary files on a system disk and magnetic tape. These procedures assume that the files are to be created from a deck of program cards.

Since the New System Cost program is comprised of four overlays, four source and four binary files are required to be created. For explanatory purposes, these files shall be named as follows:

Overlay	Program	Binary File Name	Source File Name
Overlay 0.0	DULMODE	PRMYB	PRMYS
Overlay 1.0	DMCAPC	DLMCB	DLMCS
Overlay 2.0	NTWORK	NTWKB	NTWKS
Overlay 3.0	ALLOTH	OTHRB	OTHRS

The binary and source file names for the Highway-Transit Cost program shall be NULLB and NULLS, respectively.

6.4.1 Creation of Source and Binary Files on Disk:

The computer accepts all its input in binary formats. The FORTRAN RUN compiler is used to compile the FORTRAN programs and put the binary modules in permanent files on the system disk. In addition, the source files are also created which may be used for any future corrections and modifications in the model and also to get the compiled listings without using the system deck of cards. The UPDATE technique is used to accomplish this. The input to the RUN compiler

is from a file called COMPILE created by the UPDATE program. The output from the RUN compiler is written on a file called LGO which is subsequently catalogued as a permanent file and contains the binary module of the source program. In the creation run, the UPDATE writes the source program on a file called NEWPL which is catalogued as a permanent file and contains the source program. The library listings of the binary file are obtained by using a LIBLIST card to ensure that all the routines are compiled properly. Rewinding of the files is carried out by using REWIND cards which aligns the reader head at the beginning of the file. The card image of an End of Record Card contains 7, 8 and 9 punched in column 1 while that of an End of File Card contains 6, 7, 8 and 9 punched in column 1.

6.4.2 Creation of New System Cost Program Files on Disk

The following illustrates the control cards and the deck organization for the creation of each of the source and binary files of the New System Cost program.

Overlay 0.0

```
$SEQUENCE,SCM.
$CHARGE,XXXXXX.
PRM,P6,T20.
REQUEST,NEWPL,*PF.
UPDATE(F,N)
REWIND(COMPILE)
REQUEST,LGO,*PF.
RUN(S,,,COMPILE)
REWIND (LGO)
LIBLIST (LGO,OUTPUT)
REWIND(LGO)
CATALOG(NEWPL,PRMYS,RP=999)
CATALOG(LGO,PRMYB,RP=999)
END OF RECORD CARD (7,8,9)
:
:
:
:
:
:
:
DULMODE PROGRAM DECK
:
:
:
:
:
:
:
END OF FILE CARD(6,7,8,9)
```

Overlay 1.0

```
$SEQUENCE,SCM.
$CHARGE,XXXXXX.
DMC,P6,T25,CM165000.
REQUEST,NEWPL,*PF.
UPDATE(F,N)
REWIND(COMPILE)
REQUEST,LGO,*PF.
RUN(S,,,CØMPILE)
REWIND(LGO)
LIBLIST(LGO,OUTPUT)
REWIND (LGO)
CATALOG(NEWPL,DLMCS,RP=999)
CATALOG(LGO DLMCB , RP = 999)
END OF RECORD CARD(7,8,9)
:
:
:
:
:
:
:
:
:
DMCAPC  PROGRAM DECK
:
:
:
:
:
:
:
:
:
:
END OF FILE CARD(6,7,8,9)
```

```
Overlay 2.0                          Overlay 3.0
$SEQUENCE,SCM.                       $SEQUENCE,SCM.
$CHARGE,XXXXXX.                      $CHARGE,XXXXXX.
NTW,P6,T25,CM100000.                 OTH,P6,T20,CM150000.
REQUEST,NEWPL,*PF.                   REQUEST,NEWPL,*PF.
UPDATE(F,N)                          UPDATE(F,N)
REWIND(COMPILE)                      REWIND(COMPILE)
REQUEST,LGO,*PF.                     REQUEST,LGO,*PF.
RUN(S,,,COMPILE)                     RUN(S,,,COMPILE)
REWIND(LGO)                          REWIND(LGO)
LIBLIST(LGO,OUTPUT)                  LIBLIST(LGO,OUTPUT)
REWIND(LGO)                          REWIND(LGO)
CATALOG(NEWPL,NTWKS,RP=999)          CATALOG(NEWPL,OTHRS,RP=999)
CATALOG(LGO,NTWKB,RP=999)            CATALOG(LGO,OTHRB,RP=999)
END of RECORD CARD(7,8,9)            END of RECORD CARD(7,8,9)
:                                    :
:                                    :
:                                    :
:                                    :
:                                    :
:                                    :
:                                    :
NTWORK Program Deck                  ALLOTH Program Deck
:                                    :
:                                    :
:                                    :
:                                    :
:                                    :
:                                    :
:                                    :
:                                    :
END of FILE CARD(6,7,8,9)            END of RECORD CARD(6,7,8,9)
```

6.4.3 Creation of Highway-Transit Cost Program Files on Disk

Binary (NULLB) and source (NULLS) program files of the Highway-Transit Cost program can be created on disk by use of the following control cards:

```
$SEQUENCE,SCM.
$CHARGE,XXXXXX.
NUL,P6,720,CM150000.
REQUEST,NEWPL,*PF.
UPDATE(F,N)
REWIND(COMPILE)
REQUEST,LGO,*PF.
RUN(S,,,COMPILE)
REWIND (LGO)
LIBLIST(LGO,OUTPUT)
REWIND(LGO
CATALOG(NEWPL,NULLS,RP=999)
CATALOG(LGO,NULLB,RP=999)
END of RECORD CARD (7,8,9)
:
:
:
NULCASE Program Deck
:
:
END of FILE CARD (6,7,8,9)
```

6.4.4 Creation of Binary Files on Tape from Disk

Listed below are the control cards for the copying of binary program files from a system disk to magnetic tape.

The binary modules are transfered on the tape in a specified order. Every time a file is copied, it is backspaced and compared to ensure that the copying operation was carried out properly. If the copying was done properly, a message -- GOOD COMPARE appears in the DAYFILE. If the message BAD COMPARE appears, it indicates that the user has committed some error in the control cards.

New System Cost Program

```
$SEQUENCE,SCM.
$CHARGE,XXXXXX.
JOBCOPY,CM50000,P6,T50,TP1.
REQUEST,TAPE1,HY.     (**YYYY,RING)
REWIND(TAPE1)
ATTACH(FRP,PRMYB)
REWIND(FRP)
COPYBF(FRP,TAPE1)
SKIPB,TAPE1,1,17,B.
REWIND(FRP)
COMPARE(TAPE1,FRP,1,17)
ATTACH(GRP,DLMCB)
REWIND(GRP)
COPYBF(GRP)
SKIPB,TAPE1,1,17,B.
REWIND(GRP)
COMPARE(TAPE1,GRP,1,17)
ATTACH(HRP,NTWKB)
REWIND(HRP)
COPYBF(HRP,TAPE1)
SKIPB,TAPE1,1,17,B.
REWIND(HRP)
COMPARE(TAPE1,HRP,1,17)
ATTACH(IRP,OTHRB)
REWIND(IRP)
COPYBF(IRP,TAPE1)
SKIPB,TAPE1,1,17,B
 REWIND (IRP)
COMPARE(TAPE1,IRP,1,17)
END of RECORD CARD
END of FILE CARD
```

Highway-Transit Cost Program:

```
$SEQUENCE,SCM.
$CHARGE,XXXXXXX.
JOBCOPY,CM50000,P6,T50,TP1.
REQUEST,TAPE1,HY.     (**YYYY,RING)
REWIND(TAPE1)
ATTACH(FRP,NULLB)
REWIND(FRP)
COPYBF(FRP,TAPE1)
SKIPB,TAPE1,1,17,B.
REWIND(FRP)
COMPARE(TAPE1,FRP,1,17)
```

END of RECORD CARD (7,8,9)

END of FILE CARD (6,7,8,9)

6.5 Execution of Program Files

Described within this section are the procedures for the execution of binary program files that are stored on disk and on tape.

As indicated within Chapter 4 Input Data, The New System Cost program requires input data from cards and from a disk file called TAPE 11. Guideway, acceleration/deceleration lane, interchange, and terminal link segment data comprise the disk input data to this program. A permanent file housing this data is required to be created prior to the execution of the program. Paragraph 6.5.1 below describes the procedures for the creation of this file. The required control cards for the execution of the New System Cost program from disk and from tape are defined in paragraphs 6.5.2.

The procedures and control cards for the execution of the Highway-Transit Cost program from disk and tape are defined in paragraph 6.5.3.

6.5.1 Creation of a TAPE II Permanent File

A disk permanent file containing guideway, acceleration/deceleration lane, interchange, and terminal link segment data can be created once this data has been organized and punched on cards according to the formats defined in paragraph 4.2. The permanent data file is created on disk by the program TEST 2; the listing of this program is shown in Figure 6-8. The following control cards are

required for the creation of this data file on disk:

```
$SEQUENCE,SCM.
$CHARGE,XXXXXX.
DATA,P10,T30,CM40000.
REQUEST(TAPE 11)
RUN(S)
LGO.
REWIND(TAPE11)
CATALOG(Tape 11,BOSDTA,CY=01,RP=999)
END OF RECORD CARD (7,8,9)
[                                  ]
[                                  ]
[  Program TEST 2                  ]
[                                  ]
[                                  ]
END OF RECORD CARD (7,8,9)
[                                  ]
[                                  ]
[  Tape  11 DATA                   ]
[                                  ]
[                                  ]
END OF FILE CARD (6,7,8,9)
```

```
      PROGRAM TEST2(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE11=11)
      DIMENSION ACARD(8)
      REWIND 11
  105 READ(5,100) ACARD
  100 FORMAT(8A10)
      IF(EOF,5) 102,95
   95 WRITE(11,100) ACARD
      WRITE(6,100) ACARD
      GO TO 105
  102 END FILE 11
      CALL EXIT
      END
```

Figure 6-8 Program TEST 2

6.5.2 Execution of New System Cost Program Files

In order to execute the New System Cost program, it is necessary to have four permanent binary files on a system disk or tape. These files can be catalogued under the following four file names:

OVERLAY	FILE NAME
OVERLAY 0.0	PRMYB
OVERLAY 1.0	DLMCB
OVERLAY 2.0	NTWKB
OVERLAY 3.0	OTHRB

The control cards required to execute the binary program files from disk and from tape are presented below.

DISK Execution:

```
$SEQUENCE,SCM TEST.
$CHARGE,XXXXXX.                LOAD(LGOP)
DMS,P6,T200,CM275000.          LOAD(LGO1)
REQUEST(TAPE 11)               LOAD(LGO2)
ATTACH(TAPE11,BSLINE)          LOAD(LGO3)
REWIND(TAPE11)                 EXECUTE (LGOP,LC=777777)
ATTACH(LGOP,PRMYB)             END OF RECORD CARD (7,8,9)
ATTACH(LGO 1, DLMCB)           [DATA]
ATTACH (LGO2,NEWKB)            END OF FILE CARD (6,7,8,9)
ATTACH(LGO3,OTHRB)
```

TAPE Execution:

```
$SEQUENCE,SCM TEST.
$CHARGE,XXXXXX.                COPYBF(TAPE1,OTHRB)
TAPERUN,P6,T200,CM275000,TP1.  REWIND(PRMYB, DLMCB,NTWKB,OTHRB)
REQUEST,TAPE1,HY. (**XXXX,RING)   LOAD(PRMYB)
REWIND(TAPE1)                  LOAD(DLMCB)
REQUEST(TAPE11)                LOAD(NTWKB)
ATTACH(TAPE11,BSLINE)          LOAD(OTHRB)
REWIND(TAPE11)                 EXECUTE(PRMYB,LC=777777)
COPYBF(TAPE1,PRMYB)            END OF RECORD CARD (7,8,9)
COPYBF(TAPE1,DLMCB)            [DATA]
COPYBF(TAPE1,NTWKB)            END OF FILE CARD (6,7,8,9)
```

Note: For program executions from tape, the 'XXXX' parameter in the REQUEST card denotes the tape number.

The first card is the SEQUENCE card which identifies the

user's job; this is followed by the CHARGE card on which the charge number is specified. The next card lists the job name, priority, time limit and the core requirements. If any of the parameter is not specified, its default value is assumed. The next three cards assign the base line data from a permanent file BSLINE to TAPE11 and rewind TAPE11. The permanent binary files are then made local in order that the current job may access them. If the files are stored on disk, this function is performed by the four ATTACH cards. If the files are stored on tape, the four COPYBF cards copies these files from the tape onto files that may be accessed. These cards are then followed by four LOAD cards which convert the binary files into the absolute load formats. EXECUTE does the actual loading and processes the job. The line count (LC) option is specified on the EXECUTE card so that the printed output may exceed 10,000 lines. The EXECUTE card is followed by End of Record card, after which the data cards are inserted. The last card in the job is the End of File card.

6.5.3 Execution of Highway-Transit Cost Program Files

A binary permanent file (NULLB) of the Highway-Transit Cost program can be executed from the system disk or from tape by use of the appropriate set of control cards defined below.

DISK Execution:

```
$SEQUENCE,SCM TEST.
$CHARGE, XXXXXXXX.
DMS,P6,T50,CM165000.
ATTACH (LGOP,NULLB)
LOAD (LGOP)
EXECUTE (LGOP,LC=777777)
END OF RECORD CARD (7,8,9)
[DATA]
END OF FILE CARD (6,7,8,9)
```

TAPE Execution:

```
$SEQUENCE,SCM TEST.
$CHARGE, XXXXXXXX.
DMS,P6,T50,CM165000,TP1.
REQUEST,TAPE1,HY.     (**YYYY,RING)
REWIND(TAPE1)
COPYBF(TAPE1,NULLB)
LOAD(NULLB)
EXECUTE(NULLB,LC=777777)
END OF RECORD CARD (7,8,9)
[DATA]
END OF FILE CARD (6,7,8,9)
```

6.6 Overlays

Overlay processing allows programs to be divided into independent parts which may be called and executed as needed. Each overlay consists of a single main program and any necessary subprograms.

An overlay is numbered with an ordered pair of numbers (I,J) each in the range of $0\text{-}77_8$. I denotes the primary level and J is the secondary level. An overlay with a non-zero secondary level is called a secondary overlay. The initial or main overlay which always remains in memory has levels (0,0). All primary overlays have their origin at the same point immediately following the main overlay (0,0). The loading of any primary overlay destroys any other primary overlay.

Overlays were used in the design of the New System Cost program because of its high core requirements. For simplicity, however, only primary overlays were used. The overlays in the model are designed such that each is independent of the other. The initial or main overlay always remains in core and is not destroyed by the loading of the additional overlays. Consequently, the various parameters which have to be shared by the various overlays are placed in common blocks in the main overlay.

6.6.1 Overlay Loading in the New System Cost Program

All the overlays are called from the main overlay and are loaded from separate files. Their binary files are created on the system disk or magnetic tape and are loaded in the sequence specified by the OVERLAY calls in the main overlay. The contents of the files are:

- Main Program - Overlay (0,0)
- New System Capital Cost Program - Overlay (1,0)
- New System Network Program - Overlay (2,0)
- All Other Programs - Overlay (3,0)

The main overlay (0,0) is always core resident and is not destroyed by the loading of the other overlays, namely (1,0), (2,0), and (3,0). The basic overlay structure of the program is shown in Figure 6-9. The loading address for overlays (1,0), (2,0) and (3,0) is the same because the optional parameter-cnnnnnn - is not specified on the overlay cards.

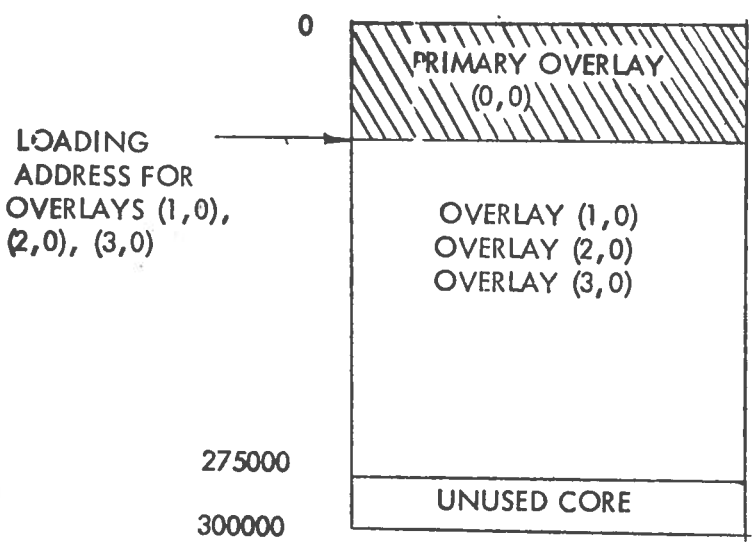


FIGURE 6-9 OVERLAY LOADING IN THE NEW SYSTEM COST PROGRAM

All card input data to the New System Cost program is read in Overlay (0,0). This is done in order to conduct parametric analysis runs by varying the input data without necessarily reading the entire input data deck.

Input variables read in the main Overlay (0,0) are placed in labeled common blocks - /RDI/ad/RDR/ - for use by the other program overlays. Variables that are computed in an overlay are shared with other overlays by placing these variables in labeled common blocks in the main overlay. Since the main overlay is always core resident these variables can always be referenced by the appropriate overlays.

6.6.2 Overlay Cards

Overlays are identified by an Overlay card. This card has the following format:

OVERLAY (fn, I_1, I_2, cnnnnnn)

where:

fn File name onto which the generated overlay is to be written.

I_1 Primary Level Number } Must be (0,0) for first overlay card and must be in octal.

I_2 Secondary Level Number

cnnnnnn optional; nnnnnn is 6-octal digits. If absent, overlay is loaded normally. If present, overlay is loaded nnnnnn words from the start of the blank common. This provides a method for changing the size of blank common at execution time.

The first overlay card must have a fn. Subsequent overlay card may omit the fn; if so, then the overlay is written on the same fn.

Each OVERLAY card must be followed by a program card. The program card for the main overlay must specify all needed file names , such as INPUT, OUTPUT, TAPE 1 etc. for all overlay levels. Files names should not appear in the program cards for other than the main overlay (0,0).

The level numbers given in the CALL OVERLAY function are decimal, while on the overlay card they are octal. Thus, the overlay card for overlay (1,10) would be OVERLAY (fn, 1, 12).

6.6.3 CALL Overlay Function

Overlay is a Fortran execution time subroutine which translates the Fortran call into a call to the loader.

Overlays are called by:

```
CALL OVERLAY (fn, I, J, P, I)
```

where:

fn	Variable name of a location which contains the name of the file that contains overlay.
I	Primary level of overlay.
J	Secondary level of overlay
P	Recall parameter. If p equals 6HRECAL, the overlay is not reloaded if it is already in memory.
I	Load parameter. Used to determine which value of

the fn will be used. I may be any value. If I is present and is non-zero, the overlay designated by fn will be loaded from the system library; otherwise, it will be loaded from the file designated by fn.

Before the execution of this call which causes the loading and execution of the overlay, the overlay must have been in absolute form and written on the file fn. When an END statement in the main program of the overlay is encountered, control returns to the statement following the CALL OVERLAY which initializes the execution of the overlay.

In the New System Cost program, the third parameter on the Overlay cards is always zero because only primary overlays were used. The fourth parameter is optional and is not specified. Consequently, all overlays are loaded immediately after the main overlay. The overlay cards within the program are:

```
OVERLAY (PRMY, 0,0)
OVERLAY (DMCP, 1,0)
OVERLAY (NTWK, 2,0)
OVERLAY (OTHR, 3,0)
```

PRMY, DMCP, NTWI, OTHR are the alphanumeric names of the files from which the overlays are loaded. These overlays are called by the following CALL statements.

```
CALL     OVERLAY (FILE2, 1, 0,0)
CALL     OVERLAY (FILE3, 2, 0, 0)
CALL     OVERLAY (FILE4, 3, 0,0)
```

The contents of FILE2, FILE3 and FILE 4 are DMCP, NTWK and OTHR which are the file names that contain the corresponding overlays. NO RECALL parameter is specified and the default value of the load parameter is assumed. In other words, the loading of these overlays is carried out from the files contained in the locations FILE2, FILE3 and FILE4.

6.7 Program Update Procedures

UPDATE is a system program used to create, manipulate and maintain library files. UPDATE determines the specific operations it is to perform from two sources. One source is the parameters on the UPDATE control card. The other source is the UPDATE input stream which contains directives and text. The following control card calls the UPDATE Program:

UPDATE (parameter list)

The control card parameters specify the mode of the update and the files to be used.

6.7.1 Update Terminology

Program Library:

A program library is a file created and maintained by the UPDATE program. It is in the form of card images with sufficient information to allow each card image to be uniquely referenced by UPDATE. The logical file name used to reference a program library is OLDPL, commonly considered synonymous with old program library.

The program library contains card images in the compressed format used by UPDATE. Card images on the program library are grouped arbitrarily into decks. A deck is defined by *DECK directive and includes all card images up to but excluding the next *DECK directive or the end of the library. An UPDATE deck can contain any number of PROGRAM statements or assembler IDENT cards, or sets of data.

Update Directives:

UPDATE directives are cards in the input stream of the UPDATE deck which contain a control character in column 1 (*) and UPDATE directive beginning in column 2, and a blank or comma immediately following the directive.

UPDATE directives are written in the form:

*Control Word

or

*Control Word parameter list

where * represents the control character. The control word is terminated by a comma or one or more blanks; any number of additional blanks may appear between the control word and parameter list. Parameters are separated from each other by commas; embedded blanks are not permitted. The control word must be defined within UPDATE as a valid directive.

Each card in the UPDATE input stream is assigned a unique card identifier consisting of two parts. The first part is the ident name. Each card is a member of only one UPDATE ident from which it obtains the ident portion of its identifier. A *IDENT is established by the introduction of a *IDENT directive or *DECK directive. The second part of the directive is the sequence number within the ident. Therefore, a card image is referenced by idname·seqnum where idname is the ident under which the card was introduced and seqnum is the sequence number within that ident. Once

the identifier is established it is retained along with the card image on the OLDPL; the UPDATE directives may reference the card image by its identifier.

Update Correction Modes:

The correction mode identifies the type of processing required to be performed by UPDATE. A correction run is defined as any run in which the first card on the input file is not a *DECK or *COMDECK card. The correction modes that UPDATE may operate under are:

FULL MODE (F-Selected)

QUICK MODE (Q-Selected)

In the FULL Mode, all the decks on the program library file are written on the compile file. *COMPILE cards are ignored.

In the QUICK Mode (or partial update mode), only decks specified on the *COMPILE card are processed. Modifications which reference cards in decks not specified on the *COMPILE file are not written on the COMPILE file.

Compile File:

The compile file parameter on the UPDATE control card determines the name of the logical file as follows:

Parameter	Meaning
Omitted	Output on file name COMPILE.
C or K	Output on file name COMPILE.
C=lfn;K=lfn	Output on logical file name given for lfn.
C=O	Compile file not to be produced.

During a file creation run or during a full update mode (F-Mode), all decks in the program library are written to the compile file. In a quick mode update (Q-Mode), only decks specified on the *COMPILE card are written to the compile file. By default, the decks are written on the COMPILE file in the order specified on the OLDPL.

In the full update, the order in which the decks are written on the COMPILE file cannot be changed.

In the quick mode, the order can be altered by using the K-file identifier on the UPDATE control card. When K is used, the decks are written to the compile file in the order of specification on *COMPILE card. If both K and C file parameters appear on the UPDATE control card, the K option will take precedence.

Compile File Sequence Field:

UPDATE will attempt to include sequence information in the compile file output. This information is compressed into the available space at the end of the card image. If the entire card identification will not fit, UPDATE will truncate the ident field until it fits, leaving the sequence number intact. When (C=PUNCH) and 8 options are selected, punched card output of the COMPILE file is obtained with ident and the seqnum punched in the compressed format in columns 73 through 80.

File Manipulation:

File manipulation directives are used to direct UPDATE.

REWIND fname

This directive causes the file indicated by fname to be rewound or repositioned to the beginning of its information.

*SKIP fname, n

*SKIP causes the UPDATE input file fname to be spaced forward over n number of records. If n is omitted or specified as zero the input file is spaced forward over one logical record.

*READ fname

Under this directive, UPDATE reads from the file fname until an end of record mark is encountered, at which point UPDATE resumes reading from the main input stream containing the *READ directive.

*WEOR n

*WEOR is used to organize the compile file for ease of input to compilers. It writes an end-of-record on the COMPILE file. Level 0 end of record is assumed if n is not specified. *WEOR directive itself is not written on the COMPILE file, *WEOR may be inserted anywhere in a deck on the program library and becomes effective only when the deck is written to COMPILE file.

6.7.2 Creation of a New Program Library

A program library is called new when the source file of a program is created for the first time and placed on the disk.

In subsequent compilation runs, the same program library is used and is updated through the UPDATE control cards and compiled to get the new binary files. UPDATE considers any run to be a creation run in which first card image in the input stream is a *DECK directive.

In a creation run, each *DECK in the input stream defines a deck to be incorporated into the library file being created. UPDATE identifies and sequentially numbers each card image with identifiers in the form of ident . seqnum.

The following directive is used in a normal creation run:

```
*DECK dname
```

This deck introduces a new deck for the library file being created. The parameter dname must not duplicate any dname previously used.

Listed below are the required control cards for the creation of a new program library (NEWPL). The source file is catalogued as PERMS; the binary file is catalogued as PERMB.

```
$SEQUENCE,SCM.
$CHARGE,XXXXXX.
NEW PGM,P6,T20,CM100000.
REQUEST,NEWPL,*PF.
UPDATE(F,N)
REQUEST,LGO,*PF.
RUNS(S,,,COMPILE)
REWIND(LGO)
```

```
LIBLIST(LGO,OUTPUT)
REWIND(LGO)
CATALOG(LGO,PERMB,RP=999)
CATALOG(NEWPL,PERMS,RP=999)
(7,8,9)
[SOURCE PROGRAM]
(6,7,8,9)
```

The main program and all the subroutines in the deck are preceded by *DECK cards. A *WEOR, which indicates - Write End of File is placed before the (6,7,8,9) card.

6.7.3 Correction of Program Library Files

The library correction process is the most common use of UPDATE. UPDATE accepts a correction set composed of UPDATE directives and data card images. The instructions provide UPDATE with the necessary information concerning the modification of the program library test stream. The old library is not disturbed. The corrections are permanent only if a new program library is created.

6.7.3.A Correction Directives

The following directives are used for correcting and updating a program library:

*IDENT:

The *IDENT introduces a correction set. All corrective operations except *PURGE must occur after *IDENT. All cards in the correction set are identified by an idnam. This identifier must not be the same as any identifier

currently in effect, such as original names or any later identifier names. The identifier holds until the next *IDENT or *PURGE card is encountered.

*PURGE:

The *PURGE card may appear like an *IDENT card. When two identifiers are separated by a period, these identifiers are purged as well as all identifiers that occur between them on the identifier list.

All cards introduced under the specified identification are physically removed from the new program library.

*DELETE:

Cards may be deactivated with the *DELETE card. When a card image is marked inactive, it remains in the library and retains its sequence. It may be referenced in the same way as an active card; however, it is not copied to the compile or source file.

```
*DELETE i1.n1
*DELETE i1,n1,i2.n2
```

where: i1,i2 = identifier names

n1,n2 = decimal sequence numbers

The first form of the directive specifies one card to be deactivated; the second, specifies and inclusive range of cards.

*RESTORE:

Specified cards are reactivated with the *RESTORE card. The first form of the statement specifies one card; the second an inclusive range of cards. Text cards may be inserted after the last card restored.

```
*RESTORE a.n
*RESTORE a.n.,b.m.
```

where: a,b = Alphanumeric identifier (a deck or a correction set name)

n,m = Decimal numbers corresponding to card sequence numbers

*INSERT:

Cards may be inserted with the *INSERT card. The parameter a.n specifies the card after which the insertion is to be made.

```
*INSERT a.n
```

where: a = Alphanumeric identifier (a deck or a correction set name)

n = Decimal number corresponding to card sequence numbers

*COMPILE:

The *COMPILE card specifies the decks to be compiled. In the first form the specific decks are mentioned while in the second form a range of decks are specified. Decks written on the COMPILE file are the ones specified by the COMPILE card.

```
*COMPILE a,b,c,....,d
*COMPILE a,d
```

where: a,b,c,d, are deck names.

In the full update mode, the *COMPILE card is ignored.

*SEQUENCE:

The *SEQUENCE directive allows selective routines to be resequenced on the new program library.

```
*SEQUENCE dname
*SEQUENCE dname 1.dname 2
*SEQUENCE dn1,dn2,dn3
```

All the active cards in the deck dname are resequenced under the identifier dname. All the inactive cards are purged. In the second form above, all decks from dname 1 to dname 2 inclusive are resequenced. One or more decks may be specified on the one *SEQUENCE directive if the deck names are separated by commas as in the third form above.

6.7.3.B <u>Partial Update Using COPYL</u>:

As described above, partial update is used to recompile selected subroutines. UPDATE(Q) writes only those decks on the COMPILE file which have to be updated. COPYL can only be used, if in a previous run, the binary modules of all the decks in the source program are placed on a binary file. COPYL replaces the old binary records with the new records. If PERMS is a source file and PERMB is a binary file, the following control cards should be used to update the binary file PERMB.

```
$SEQUENCE,SCM.
$CHARGE,XXXXXX.
PARTUPD,P6,T20,CM100000.
ATTACH(OLDPL,PERMS)
ATTACH(LOCB,PERMB)
UPDATE(Q)
REQUEST,TEMB,*PF.
REQUEST,LGO,*PF.
REWIND(COMPILE)
RUN(S,,,COMPILE)
REWIND(LOCB,LGO,TEMB)
COPYL(LOCB,LGO,TEMB)
REWIND(TEMB)
LIBLIST(TEMB,OUTPUT)
REWIND(TEMB)
PURGE(LOCB)
CATALOG(TEMB,PERMB,RP=999)
(7,8,9)
[UPDATE CORRECTION CARDS]
*COMPILE DECKNM1,DECKNM2
(6,7,8,9)
```

DECKNM1 and DECKNM2 are the deck names of the subroutines to be recompiled after the necessary corrections.

6.7.3.C <u>Partial Update Using COPYBR and CDPYBF</u>

The COPYL function cannot be used to update certain decks after recompilation if such decks had compilation errors

in the creation run. In these cases, the copy binary record (COPYBR) and the copy binary file (COPYBF) functions are used in the update procedure as a means to selectively copy binary records and files.

Assume for a given source file PERMS there exists a binary file PERMB containing eight records of which records four and five are in error. The partial update procedure shall be used to correct records four and five and the COPYBR and COPYBF functions shall be used to copy the good binary records from PERMB to the file being created. Using as input the source file (OLDPL=PERMS), the binary file (LOCB=PERMB), and the source corrections to records four and five, UPDATE creates new binary records (for records four and five) on a file called LGO. The old binary file LOCB is not changed in this process. At this point, the first COPYBR function copies the first three good records from LOCB to the temporary file TEMB. The second COPYBR function copies from LOCB the two records (4 and 5) to a dummy file. Next, the two corrected records on LGO are copied to the TEMB file, followed by a copying of the remaining three good records on LOCB to TEMB. The last function (COPYBF) copies the end-of-file record from LOCB to TEMB. This entire procedure is illustrated in Figure 6-10. The control cards for this process are listed below:

```
$SEQUENCE, SCM.
$CHARGE,XXXXXX.
UPHOLD,P6,T20,CM100000.
ATTACH(OLDPL,PERMS)
ATTACH(LOCB,PERMB)
UPDATE(Q)
REQUEST,TEMB,*PF.
REQUEST,LGO,*PF.
REWIND(COMPILE)
RUN(S,,,COMPILE)
REWIND(LOCB,LGO,TEMB)
COPYBR(LOCB,TEMB,3)
COPYBR(LOCB,DUMMY,2)
COPYBR(LGO,TEMB,2)
COPYBR(LOCB,TEMB,3)
COPYBF(LOCB,TEMB)
REWIND(TEMB)
LIBLIST(TEMB,OUTPUT)
REWIND(TEMB)
PURGE(LOCB)
CATALOG(TEMB,PERMB,RP=999)
(7,8,9)
[UPDATE CORRECTION CARDS]
*COMPILE DECKNM1
(6,7,8,9)
```

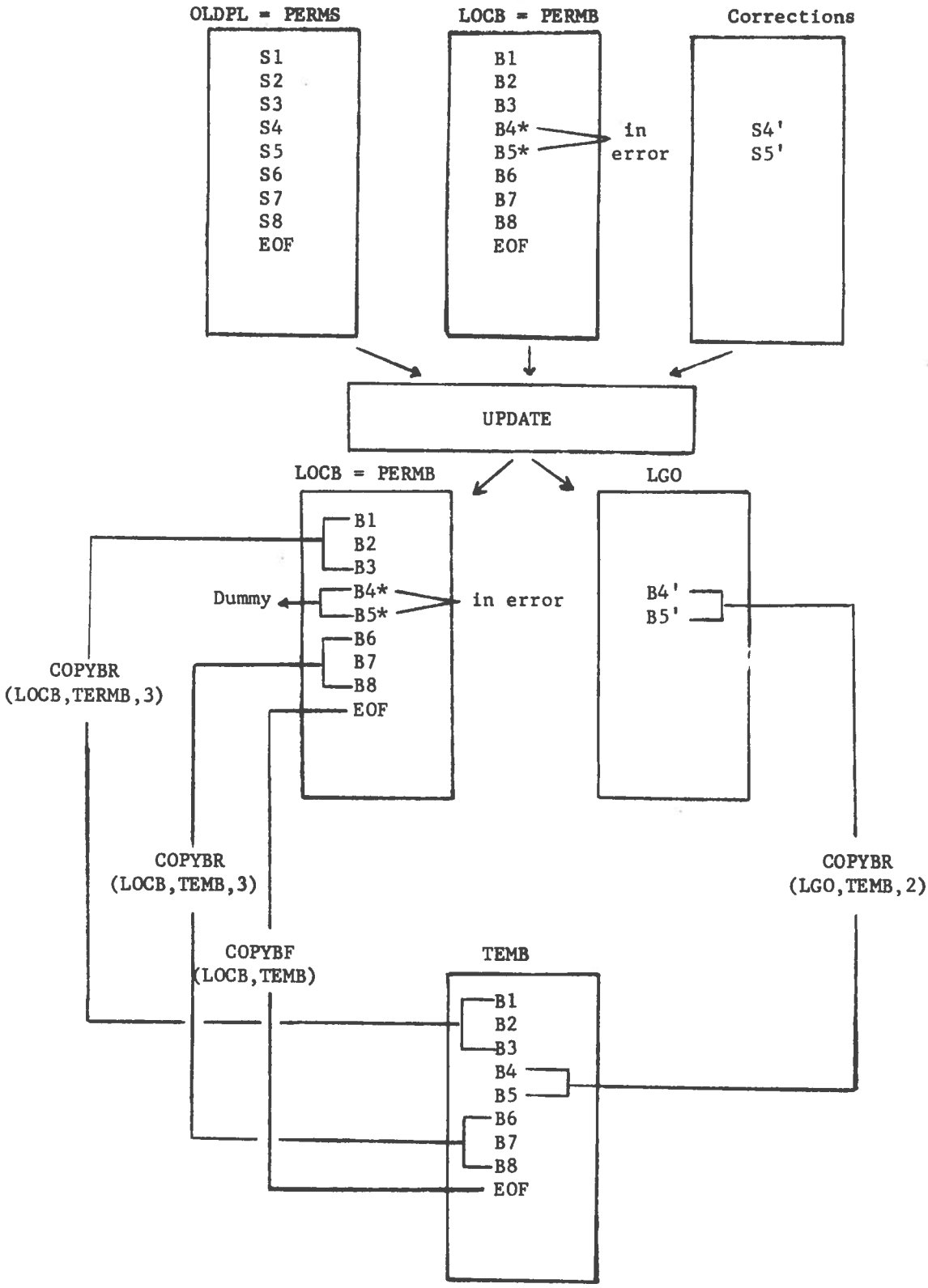


FIGURE 6-10 COPYBR-COPYBF in an Update Procedure

6.8 Manipulating Files and Records

Records are a group of adjacent data items that may be manipulated as a unit; there exists physical and logical records. A logical record is a collection of related data items that the program logic treats as a unit. A physical record is a set of adjacent data characters terminating with an end-of-record indication.

A file is a collection of related records that may be treated as a unit.

6.8.1 Permanent Files

A permanent file is a mass storage file catalogued by the system, so that its location and identification are always known to the system. Permanent files cannot be destroyed accidentally during normal system operations. They are protected by the system from unauthorized access according to the privacy controls specified when they are created. Permanent files are commonly used since they provide for large mass storage and rapid access to the data.

Described below are some of the more important permanent file functions:

Cycle:

Five permanent files may be catalogued under one permanent file name. Each file is called a cycle and each shares the same user ID and set of passwords. No restriction is imposed on the size or the content of each cycle, since

each is a unique file. The Cycle option is useful in the ATTACH, CATALOG, and PURGE functions.

Subdirectory:

The permanent file directory is divided into subdirectories which are identified numerically. Each permanent file is associated with the number of a subdirectory in which it has been catalogued. The subdirectory reference may be used to expedite a system action on a file. If, however, this option is not specified the whole directory is searched until the desired file is found. The option may be specified while carrying out ATTACH or a PURGE function.

ATTACH:

A permanent file can be accessed directly by means of the ATTACH control card. Depending on the kind of privacy, user's ID and other options, the parameters are specified on this card. In a simpler case, if the file doesn't have any private procedures, the control card would be

```
ATTACH(LOCB,PERMB)
```

where PERMB is the permanent file name and LOCB is the local file name. If no cycle is specified cycle 1 of the file PERMB is attached. If the file is catalogued under a cycle number other than 1, say 2, it is made local by using the following command:

```
ATTACH(LOCB,PERMB,CY=02)
```

PURGING:

Purging causes a permanent file to be removed from its directory and its mass storage space to be released. Anyone who knows the control password and the user ID may purge a file. A file may be purged without such information if during cataloging the file the password and the user's ID are not specified. Only one cycle of a file can be purged at a time, by use of the PURGE function:

```
ATTACH(LOCB,PERMB)
PURGE(LOCB)
```

Cycle 01 of the permanent file PERMB is purged by using the above commands. If, however, some other cycle, e.g. 3 has to be purged, the following command should be used:

```
ATTACH(LOCB,PERMB,CY=03)
PURGE(LOCB)
```

CATALOG:

This function enters a file's location and other pertinent information in the permanent file directory maintained by the system, thereby making the file permanent. RP is the retention period - the number of days the file should be saved. RP=999 indicates that the file has to be saved indefinitely. A local file LOCB can be catalogued under a permanent file name PERMB by the following command:

```
CATALOG(LOCB,PERMB,RP=999)
```

6.8.2 Copying Files and Records

Files or records are written on tape in either external BCD or binary format. On cards, they are punched in either Hollerith coded or binary format. On disks or drums, they are written in binary format.

Copying Files

Coded files can be Hollerith coded on cards or they can be written in external BCD on tape. Regardless of the storage device, they may be copied from one device to another by means of the COPYCF function. The control cards for this procedure are:

```
$SEQUENCE,SCM.
$CHARGE,XXXXXX.
JØBCØPY,CM30000,P6,T50.
REQUEST(TAPE1)
REWIND(TAPE1)
REQUEST(TAPE2)
REWIND(TAPE2)
COPYCF(TAPE1,TAPE2)
(6,7,8,9)
```

Binary files may exist on tapes, cards, disks, or drums. They are copied by using COPYBF function. In the example above, binary files may be copied by using the following control in place of the COPYCF card.

```
COPYBF(TAPE1,TAPE2)
```

Copying Records

Coded records are copied from one device to another in much the same manner. For copying coded records, the request COPYCF is replaced by COPYCR and the request becomes:

```
COPYCR(TAPE1,TAPE2)
```

And, for binary records, the request is:

```
COPYBR(TAPE1,TAPE2)
```

If more than one record has to be copied, this is specified as a third parameter on the above request.

6.8.3 Copying Volumes From Tapes

A volume on a tape consists of one or more logical files. On SCOPE tapes, a logical file can be subdivided into logical records. To copy the entire contents of a reel of tape identified as TAPE 1 onto another reel of tape, called TAPE 2, the following control cards should be used:

```
$SEQUENCE,SCM.
$CHARGE,XXXXXX.
JØBCØPY,CM30000,P6,T50,TP2.
REQUEST,TAPE1,HY          (**YYYY,RING)
REWIND(TAPE1)
REQUEST,TAPE2,HY          (**ZZZZ,RING)
REWIND(TAPE2)
COPY(TAPE1,TAPE2)
(6,7,8,9)
```

Request statements are used to associate input and output

file names, TAPE1 and TAPE2, with specific input/output devices. Such devices must be requested except when input is read from the card reader or disk, or when output is written on the disk or the printer.

6.8.4 Bypassing/Backspacing of Files and Records

Bypassing and backspacing occurs when one or more files or records are extracted from among many.

Suppose there exists a magnetic tape, called TAPEA, containing six files of which files 2,3, and 4 are to be printed. In order to accomplish this, the first file on TAPEA is copied onto a DUMMY file or any other scratch file. The idea is to move TAPEA to the beginning of the second file. The next three files on TAPEA are then copied onto OUTPUT. The control cards would appear as-

```
$SEQUENCE,SCM.
$CHARGE,XXXXXX.
JØBTAPE,CM30000,T50,TP1.
REQUEST,TAPEA,HY.       (**XXXX,RING)
REWIND(TAPEA)
COPYCF(TAPEA,DUMMY)
COPYCF(TAPEA,OUTPUT,3)
(6,7,8,9)
```

The copying of files 2,3, and 4 from TAPEA may also be accomplished by employing the skip-file-request (SKIPF). This request along with the skip-record-request (SKIPR) contains the name of the input file, the number of files or

records to be skipped forward (+n) or backward (-n), the level number in octal, and a code to indicate if the file is written in coded or binary format. In order to skip file 1 and to copy files 2, 3, and 4 from TAPE A to OUTPUT, the following control cards should be used:

```
$SEQUENCE,SCM.
$CHARGE,XXXXXX.
JØBTAPE,CM30000,T50,TP1.
REQUEST,TAPEA,HY.       (**XXXX,RING)
REWIND(TAPEA)
SKIPF(TAPEA,1,17,C)
COPYCF(TAPEA,ØUTPUT,3)
(6,7,8,9)
```

To skip ahead by files within a volume, level number 17 is used; to skip logical records within a file, level number 0 is used.

6.8.5 Library Listing Function

The LIBLIST control card activates the system function that creates a listing of all the binary records in a file. The listing puts all the binary records in sequential order; it also contains an End of File record, the checksum, and the creation date. The following control cards create a library listing of the permanent file PERMB:

```
$SEQUENCE,SCM.
$CHARGE,XXXXXX.
LIBLIST,P10,T10,CM40000,1Ø200.
ATTACH(LØCB,PERMB)
```

CATALOG OF FILE BGO

RECORD NO.	LEVEL NO. OCTAL	LENGTH DECIMAL	LENGTH OCTAL(B)	PACKAGE	CHKSUM	------ CREATION DATE
1	<CARD IMAGE><OVERLAY(OTHR,4,0) >					
2	0	64	100	ALLOTH	5145	UNKNOWN
3	0	1497	2731	PWRPOL	5556	UNKNOWN
4	0	1064	2050	AIRPOL	3514	UNKNOWN
5	0	1326	2456	ACCDNT	1066	UNKNOWN
6	0	211	323	SUM	4007	UNKNOWN
7	0	580	1104	HWYOP	5351	UNKNOWN
8	0	889	1571	TRNOP	11	UNKNOWN
9	0	215	327	SUM3	7301	UNKNOWN
10	0	223	337	SUM2	6067	UNKNOWN
11	0	1705	3251	UMOP	6105	UNKNOWN
12	17	0	0		0	

LEVEL 17GROUP LENGTH IS 7774 17136

***** END FILE *****

FIGURE 6-11 LIBRARY LISTING OF A BINARY FILE

```
LIBLIST(LØCB,OUTPUT)
(7,8,9)
(6,7,8,9)
```

A sample library listing is shown in Figure 6-11.

6.8.6 Creation of Punched Deck File

A punched deck of a complete program is created by use of the full UPDATE procedure and the copy coded file (COPYCF) function. This process first requires the creation of a New Program Library by the specification of option N as the second parameter on the UPDATE control card. Option 8 on the UPDATE card specifies that 80 column card images are to be written on the COMPILE file. The COPYCF function copies the COMPILE file to the card punch Output file. The *SEQUENCE card is also used in this process so that after updating all the decks are resequenced before they are written on the COMPILE file. Because full update option is used all the decks that are updated should be written on the *SEQUENCE card so that the entire punched program is sequenced.

Listed below are the control cards for this process:

```
$SEQUENCE,SCM.
$CHARGE,XXXXXX.
PNCNPHL,P6,T20,CM100000.
ATTACH(OLDPL,PERMS)
ATTACH(LOCB,PERMB)
PURGE(LOCB)
```

```
REQUEST,NEWPL,*PF
UPDATE(F,N,8)
REQUEST,TEMB,*PF.
REQUEST,LGO,*PF.
REWIND(COMPILE)
COPYCF(COMPILE,PUNCH)
REWIND(COMPILE)
RUN(S,,,COMPILE)
REWIND(LGO)
LIBLIST(LGO,OUTPUT)
REWIND(LGO)
CATALOG(LGO,PERMB,RP=999)
CATALOG(NEWPL,PERMB,RP=999)
(7,8,9)
[UPDATE CORRECTION CARDS]
*SEQUENCE DECKNM1,DECKNM2...
(6,7,8,9)
```

6.9 Export/Import UT-200 Terminal Operations

6.9.1 Terminal Description

The Export/Import UT-200 terminal is a remote job entry system that allows users to submit batched jobs (programs and data) to a CDC-6600 computer where the jobs are processed in a multi-programmed environment. All data transmitted back and forth from the remote terminal and the central computer is via telephone lines.

The terminal system is comprised of three primary pieces of equipment that operate in the following manner:

- a card reader that reads each job at a rated speed of 140 cards/minute.
- a printer that receives and prints output from SCOPE at a rate of 180 lines/minutes.
- a combination display screen/keyboard that allows the user to communicate to SCOPE or the central site computer operation. The user types his commands or messages to SCOPE or the operator on the keyboard; in turn, any messages from these sources are displayed to the user on the terminal screen.

6.9.2 Establishing Data Communication with the Central Site

The terminal is connected to the central site (CDC 6600 Data Processing Center) by following these instructions:

a. Remove the handset from its cradle and press the TALK button.

b. Dial the central site telephone number.

c. If the system is up, the central site answers automatically, a steady tone followed by beeps will be heard. Press the data button and hang up.

Soon, the following message will appear on the EXPORT/IMPORT UT-200 terminal screen:

```
TERMINAL IDLE m pp
```

This message indicates that the terminal is on-line to the central site, but not yet active.

m = Multiplexor Number

pp = Port number on the Multiplexor

A steady blinking of the MAN REL light indicates that the terminal is properly connected to the central site.

6.9.3 Activating the Terminal

The UT-200 terminal can be called into operation by depressing CLEAR, and typing IMPORT, CXXX* (where XXX* is the identification code assigned to the terminal) and depressing the SEND key. The above information would be displayed on the screen as:

```
IMPORT, CXXX*Δ
```

Where c = Check digit

XXX* = Terminal identification code

6.9.3 Cont'd

If the word IMPORT is mispelled, the following message appears on the screen:

```
SUBSYSTEM NOT AVAILABLE
```

If there is an error in the terminal identification code, the following message is displayed:

```
INVALID OR MISSING TERMINAL NAME
```

To correct either of the above situations, press the CLEAR key and retype,

```
IMPORT,CXXX*Δ
```

6.9.4 Submission of Jobs

Cards are loaded into the card reader hopper, face down with the 9 edge towards the rear. Cards may be inserted into the hopper during the machine operation, but the card weight would not be removed if less than 100 cards are left. Place the AUTO/MAN switch to AUTO mode by depressing if necessary. Depress the READY/CHECK switch to READY and then depress the ERROR/FEED switch to ready the first card for reading. Then, press MAN/REL key to clear the card reader buffer.

The card-reading operation is initiated by depressing the LOAD button on the Equipment Controller Panel. The LOAD button causes cards to be read and transferred into the card reader's memory

buffer until it is filled (12 card images). After the first buffer is full, the job is transmitted to SCOPE by the following procedure,

a) Depress the CLEAR key

b) Type READ or R on the display's keyboard and press SEND key

Subsequent reading of cards in the hopper is controlled by the central site. After the READ command is sent, the following message may appear on the user's screen.

```
WAITING FOR STORAGE-CR
```

This is a message from the central site indicating that there is momentarily no space in the central site's memory to receive job input. However, the message should disappear quickly and the transmission process continue.

The SCOPE $ control cards (i.e. control cards that guide the SCOPE Operating System) at the beginning of each job are read and checked for validity. If they are not valid, a following message is displayed:

```
JOB CARD ERROR
```

In such case, replace the erroneous cards with the correct ones and resubmit the job.

6.9.5 Ending a READ Operation:

At times, it becomes necessary to end the READ operation before the complete deck is read. This may be required if the user finds an error in his deck after READ operation has been initiated or there is some problem is transmitting the data to the central site. The READ operation is terminated by the END Card Reader command:

```
END, CR △
 (or)

E,CR △
```

To re-initiate reading, reload the cards and use the READ command just as if the job were being initiated for the first time. To ensure that there is no card data left in the reader buffer, the user should press the MAN REL switch to clear the buffer before loading the cards.

6.9.6 Displays

After a job is submitted, the user may want to know its status at the central site. This information can be obtained through the H and B displays.

- H-Display which gives a list of jobs currently in the SCOPE input and output queues, and
- The B-Display indicates what jobs are in execution.

The letters of the appropriate display are typed and SEND key is pressed. If the read or write operations are in progress, the INTR and CLEAR keys are depressed and then the appropriate display message is sent.

The H Display:

To obtain a list of jobs in the input queue, enter the command,

H,I Δ

and jobs in the output queue are displayed by the command,

H,O Δ

Only those jobs submitted by the terminal will appear on the display screen. The jobs are listed in order of priority. For the H,I display, the number of jobs in the input queue are also indicated at the user's terminal.

The B Display:

To display the names of jobs currently in execution at the central site, enter the command

B Δ

The seven control points for jobs are displayed. Each of the control points contains information about the job name, priority, time limit, total time elapsed, relative central memory address,

field length of job, and the most current day file message for the job.

6.9.7 Sending Message To The Central Site

At any time during job processing, messages may be sent to the central site operator. A message must be written in the form:

```
M, mmm...m
```

The message mmm.... may be as large as 28 characters. At the central site, the message is displayed, preceded by the code identifying the terminal which sent the message.

6.9.8 Dropping a Job

If the user wishes to drop a job from execution for some reason, he can do it by using the DROP command. Because only the last two characters are needed to identify a SCOPE job, this command may be abbreviated to:

```
DROP, ii  Δ
 or
D,ii  Δ
```

Where ii are the last two characters of the job sequence number.

6.9.8 Cont'd

When a command (say a DROP command) references a job or file name that cannot be found, the following message is sent back:

```
FILE NOT FOUND
```

This means that the job or file name referenced does not exist, or an incorrect job or file name was entered.

6.9.9 Terminating Printing Process

To completely terminate a job in the process of printing, press the INT and CLEAR keys, and type:

```
END, LP  Δ
 (or)
E, LP   Δ
```

upon receipt of this command, SCOPE skips the printing of the rest of the job and prints the day file.

6.9.10 Rewinding and Reprinting Output Files

The printing of job may be stopped and resumed immediately from the beginning or some other specific point by using the AGAIN command. To start printing from the beginning of a job, type:

```
AGAIN Δ
 (or)
A Δ
```

To start printing again from a point other than the beginning, enter:

```
AGAIN, n Δ
(or)
A, n Δ
```

where n is an octal parameter which tells the system to backspace the print file $10_8 \times n_8$ sectors. If the beginning of the file is reached before $10_8 \times n_8$ sectors, printing is resumed from the beginning.

6.9.11 Suspending Printing in Process

To merely suspend a printing operation currently in progress press the INT and CLEAR keys and enter:

```
SUSPEND Δ
 (or)

S Δ
```

Printing halts immediately, but the printer buffer remains as assigned. To negate the SUSPEND command and continue normal printing operation type:

```
CONTINUE △
 (or)
C △
```

6.9.12 Logout Procedure

If the user has finished submitting his jobs and output has been received, the terminal should be disconnected from the central site. To disconnect from the central site the following command should be sent:

```
LOGOUT △
```

APPENDIX A

PROGRAM LISTING - NEW SYSTEM COST PROGRAM

OVERLAY 0.00
(PRMY)

This overlay contains the following subroutines:

```
                DIMENSION (INPUT,OUTPUT,PUNCH,TAPE5=INPUT,TAPE6=OUTPUT,
                FORTRAN VI PROGRAM DULMODE (INPUT,OUTPUT,PUNCH,TAPE5=INPUT,TAPE6=OUTPUT,
               1TAPE7=PUNCH,TAPE8=TAPE8),TAPE9)
000007          DATA FIL1,FIL2,FIL3,FIL4,LPRMT,LDMCP,LNTWK,LNTWR/
000007          COMMON/RMI/ IPARDT(199)
000007          COMMON/ONM/ MPARDT(1430)
000007          COMMON/SVI/ ISAVDT(100,2)
000007          COMMON/SVR/ MSAVDT(100,2), SMRVDT(320,5)
000007          COMMON/LUPS/ NPI, LPS, NLP, IRC, NLPMI(5), LPSM
000007          COMMON/TITL/ MATTTL(8)
000007          COMMON /OV13/ ACUMR(14,4), DIRTM(, DILNMI, DYTOB
000007          COMMON /OV23/ _OPHRS  , DVTMIP(6,4)  , DTOPMT(4,5), DTOVMT(4,5),
               1 DMPART(5,6,4) , DLVDAT(4,3,4)  ,  DDVMT(150,4,4)  ,
               1 DDFVMT(150,6,4),  DDNVMT(225,6,4)
000007          COMMON /OV33/  AUTONM(5,6,12)  ,  TMCANM(5,6,12) , RNDATA(5,6,12),
               1 DDFPWR(5,6,12) , RPARC(5,6,12),
               1 DNVWM(5,6,12)
000007          COMMON/LST/TULNMI,DILNMI,THLNMI,MALNMI,DYLNMI
000007          COMMON /MM/ MMINUM
000007          MMINDX = 1
000004          NLP = 0
000007          INITIALIZE IPARDT AND MPARDT ARRAYS
000005          DO 1P  K = 1,199
000011       1P IPARDT(K) = MRR
000015          DO 2P  J = 1,1430
000020       2P MPARDT(J) = IXMRR
                INITIALIZE ISAVDT AND MSAVDT ARRAYS
000025          DO 3P  K = 1,100
000030          DO 3P  J = 1,2
000033          ISAVDT(K,J) = 0
000036       3P MSAVDT(K,J) = 0.0
                INITIALIZE SMRVDT ARRAY
000040          DO 4P  K = 1,320
000051          DO 4P  J = 1,5
000054       4P SMRVDT(K,J) = 0.0
                INITIALIZE NLPMI ARRAY
000060          DO 5P  K = 1,5
000071       5P NLPMI(K) = 0
                INITIALIZE LOOP COUNTER
000076          LPS = 1
                PRINT 4001, (TITLE PAGE
                TO BE CHOSEN
000076          CALL MULTIPUT
000077          PRINT 1499
000107    1499 FORMAT(1H1)
000107          PRINT 1415, (MATTTL(L), L = 1,8)
000115    1415 FORMAT(5X,1H ,38A10/)
000115          PRINT 1415, (MATTTL(L), L = 4,8)
000127    1415 FORMAT(1H ,4A10,5X)
000127          IF ((MMMTTTT) .EQ. 1)  GO TO 1490
000131          WRITE(2,1490) MATTTL
000137    1490 FORMAT(8A10)
000137    1490 CONTINUE
```



```
DULMODE

PROGRAM LENGTH (INCLUDING ALL BUFFERS
012447

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
1A      =  000011    4A      =  000070    7A      =  000130    9A      =  000056
5A      =  000071    1400    =  000140    1400    =  000260    1410    =  000262
1415    =  000245    1430    =  000274    1440    =  000272    1450    =  000137
1500    =  000237    1010    =  000151    1005    =  000155    1410    =  000141
1450    =  000149    1000    =  000166

BLOCK NAMES AND LENGTHS
RMI     =  000197    MMM     =  014004    SVI     =  000310    SVR     =  003410
LUPS    =  000024    TITL    =  000108    OV13    =  000127    OV23    =  020431
OV33    =  005500    LST     =  000006    MM      =  100001

VARIABLE ASSIGNMENTS
ASUMR   =  000000C00 AUTONM  =  000000C11 DRMDAT  =  000070C10 DMVMT   =  004730C11
DLVDAT  =  000290C10 DDFPWR  =  000264C11 DDFVMT  =  000311C10 DDNVMT  =  085851C10
DDMPWR  =  002478C11 DDNVMT  =  011011C10 DTOPMT  =  000021C10 DTOVMT  =  000045C10
DYTRIP  =  000001C10 FIL1    =  000321    FIL2    =  000322    FIL3    =  000323
FIL4    =  000324    MPARC   =  004100C11 IPARDT  =  000000C01 ISAVDT  =  000000C03
J       =  000326    JSP     =  000330    K       =  000325    L       =  000327
LPS     =  000001C05 LPSM    =  000023C05 MMINDX  =  000000C13 NLP     =  000002C05
NLPM    =  000000C05 NPI     =  000000C05 MATTTL  =  000000C06 MPARDT  =  000000C02
RSAVDT  =  000000C04 SMRVDT  =  000310C04 TRCKNM  =  000550C11 ANDATA  =  001320C11
RPARC   =  003410C11

START OF CONSTANTS
000242

START OF TEMPORARIES
000314

START OF INTERFACTS
000370

UNUSED COMPILER SPACE
127000
```



```
000137          PRINT 1430
000143    1430 FORMAT(1H ,4HIT IS A REAT ENGINEERING ESTIMATE RUN0.5/)
000149          IF (IPARDT(79) .EQ. 1)  GO TO 1300
000150          CALL PRNTIN
000160    1300 CONTINUE
000160          IF (IPARDT(80) .EQ. 1)  GO TO 1600
000160          GO TO 1610
000151    1400 DO 1605  JRP = 1,75
000155    1605 IPARDT(JRP) = 1
000161    1610 CONTINUE
000161          IF (LPS .EQ. 1)  GO TO 1650
000163          MMINDX = NLP + 1
000165          GO TO 1660
000165    1650 MMINDX = 1
000166    1660 CONTINUE
000166          CALL OVERLAY(FIL2,1,0,0)
000171          CALL OVERLAY(FIL3,2,0,0)
000174          CALL OVERLAY(FIL4,3,0,0)
000177          CALL SUMMARY
000200          CALL SUMMARY
000201          IF (LPS .EQ. LPRM)  GO TO 1500
000203          LPS = LPS + 1
000204          PRINT 1490
000210          PRINT 1415, (MATTTL(L), L = 1,8)
000227          PRINT 1415, (MATTTL(L), L = 4,8)
000234          CALL PARRUN
000235          IF (NPI .NE.999)  GO TO 1300
000237    1500 CONTINUE
000237          CALL EXIT
000240          END
```

```
      SUBROUTINE RDINPUT
      INTEGER DLIFE     , DINT
      INTEGER PARCK3
      REAL  DPKHS , NUMDA
      DIMENSION  DDNLKC(225,4)
      COMMON/RDI/       JSPRNT(80)      , DLIFE(30)       , DINT(30)       ,
     * LECODE           , NYM             , NLM             , NSM             , NYM             ,
     * NIM               , NTM             , NTSM            , NYM             , NOTM            ,
     * NM                , IZ              , JR              , PARCK3(6)       ,
     * NN                , NX              , IPRINT          , IZV             ,
     * (PDLCD(4,2,3), IDACCD(3)            , NMV             , NOTM            ,
     * NVC(9)
      COMMON/RDB/       SCALEF(10)        , DVMCAP(5)       , DVMLDF(5)       ,
     * DLNCAP(5)         , ULANDC(10)      , UTRARC          , USTRUC(4)       ,
     * DRWDTH(10)        , DCNCEN(11)      , DVATCC(5)       , DYARDS(5,13)    ,
     * DNINT(10)         , DVMSUC(4,6)     , DVMUC(4)        , DNUMVM(5)       ,
     * ASSRAT(4)         , TAXRAT(4)       , DNOISE(20,14,2) , RESDEN(4,7)     ,
     * DDNRFL(225,4,4),  DDLKST(225,5)     , BACKDN(225,3,3) , DPKHRF(225,3)   ,
     * DPKIMP(225,3)     ,
     * NPKHS            , NUMDA           , DPKPCT          , DPTRIP(4,3)     ,
     * DPDLYD(2,3)       , DVDLAY(2,3)     , DMVLDF(2,3)     , ZSYSCH(4,5)     ,
     * ZPKPCT            , ZPTRIP(4,4)     , ZDELAY(2,4,2)   , MLNDIR(2,2)     ,
     * ZWMTPC(4)         , DVMTPC(3)       , AUTOCH(11)      , NMCHAR(4)       ,
     * DPCVCH(14,4,2)    , ZPCVCH(4,6)     , HPCVCH(2,6)     , POLRAT(3,3)     ,
     * DMSTAC            , MACRAT(2,5,2)   , ZACRAT(4,4)     , DACRAT(2,3)     ,
     * HMPRAT(2,4)       , HVOPCS(2,6)     , HWTOPC(2,3)     , KOPCU(4,7,4)    ,
     * ANPICU(4,7,4)     ,DDFPFL(150,4,4),DDFLKC(150,3,3)    ,BACKOF(150,3,2),
     * DYIIPC            , UCCOPC          , DHOPC           , Z(4)            ,
     * DVOPUC(6,9,4)     , DTROPC(11,7)    , DGOPC(7)        , FLOCAP(5)       ,
     * DFARE(4,2,2)
      COMMON/SYL/ ISAVDT(100,7)
      COMMON/SVR/ HSAVDT(100,2), SHRYNT(320,5)
      COMMON/LUPS/ NPT, LPS, NLP, IRC, NLPR(15), LPSM
      COMMON/TITLI/ BNTITL(8)
      EQUIVALENCE (LV,NVM) , (LK,NLM)  , (DDNLKC(1),DDLKST(1))
      READ 5, (BNTITL(L), L = 1,8)
    5 FORMAT(8A1B)
      READ 2V, LECODE
   20 FORMAT(I5)
C     READ DUAL MODE LIFE AND INTEREST TABLES
C
      IZM = 0
   1A READ 15, IZ, DLIFE(IZ), DINT(IZ)
      IZM = IZM + 1
      IF (IZ .NE. 30)  GO TO 1A
      IZM = IZM - 1
C     NOTE LAST CARD MUST HAVE A ZERO IN COL 5
      IZ =1
   15 FORMAT(I5,2F10)
C     READ SCALE FACTORS
      READ 30, (SCALEF(JSF),   JSF = 1,10)
   30 FORMAT(10F8.2)
C     READ SUPPRESS PRINT CODES, CODE = 1 = SUPPRESS TABLE
      READ 4V, (JSPRNT(JSP), JSP = 1,80)
```



```
      READ 51V, (DCNCEN(JCD), JCD = 1,7)
   51A FORMAT(4F5.0,F10.2,2F10.0)
C
C     READ WAY COMMAND AND CONTROL INPUTS
      READ 512, (DWAYFC(JWD), JWD = 1,2)
  512 FORMAT(2F10.0)
C
C     READ UM YARD INPUTS AND COUNT NUMBER OF YARDS - NYM
      NYM = 0
   51R READ 51R, NY, (DYARDS(NY,NYD), NYD = 1,7)
      NYM = NYM + 1
      IF (NY .NE. 5)  GO TO 51R
      NYM = NYM - 1
C     NOTE LAST CARD MUST HAVE A ZERO IN COL 5
   51A FORMAT(I5,6F5.1,F10.2,F10.0,F10.2)
C
C     READ UM MAINTENANCE FACILITY INPUTS
      READ 51A, (DMAIN(JMD), JMD = 1,6)
   51A FORMAT(4F5.0,F10.2,F10.0)
C     READ UM VEHICLE UNIT COSTS AND NO. OF VEHICLES
      DO 6A0 NV = 1,NVM
  6A0 READ 602,NVA, (DVMSUC(NV,NVS),NVS=1,6), DVMUC(NV),DNUMVM(NV)
  602 FORMAT(I4,6F10.2,F8.2,F8.0)
C
      ASSRAT(4)=0.0
      TAXRAT(4) =0.0
C     READ ASSESSMENT RATIOS AND TAX RATES BY GEOG. AREA
      READ 1200, (ASSRAT(JGA),JGA =1,3), (TAXRAT(JGA),JGA =1,3)
 1200 FORMAT(3F5.3,3F10.2)
C*********************************************************************
C     READ UM NETWORK INPUT DATA
C*********************************************************************
      READ 9V01, JK, NMS, NR, NMV
 9A01 FORMAT (4I10)
 9A02 FORMAT(5X,F5.0,5X,F5.0,5X,6I1,5X,F5.3)
 9A03 FORMAT(F15.0,5X,F10.1,F10.0,F15.0,5X,F10.1,F10.0)
 9A04 FORMAT(F15.0,5X,F15.0,5X,4F10.0)
 9A05 FORMAT(2(4X,F8.1,4X,F6.1,4X,F6.3))
      JRV=JK+1
      LRV=LK+1
      LVM=V+1
      DO 1A1U II = 1,JRV
      DO 1A1U  LI = 1,LK
      DO 1A1U   NRX = 1,4
 1A1U DDPPFL(IX,LX,NXX) = 0.0
      DO 1A1R   JRX = 1,JK
      DO 1A1D LVX = 1,3
      DO 1A1D NTP = 1,P
 1A1A BACKOF(JRX,LVX,NTP) = 0.0
      DO 1A1A  ITPA = 1,2
      DO 1A1A  LVX = 1,3
 1A1A DMVLOF(ITPA,LVX) = 0.0
      READ 8602,NPKHRS,NUMDA,PARCK3,DPKPCT
      DO 1A03 IX=1,JK
 1A03 READ 8U03,(DDFPFL(IX,LR,3),DDFLKC(IX,LR,2),LR=1,LVMDIN 165
```



```
   6N FORMAT(8D11)
C     READ UM VEHICLE DATA 1 AND COUNT NUMBER OF VEHICLE TYPES NVM
   10 READ 102, NVM, DVMCAP(NV), DVMLDF(NV), ULNCAP(NV)
      NVM = NVM + 1
      IF (NV .NE. 5)  GO TO 10
      NVM = NVM - 1
C     NOTE LAST CARD MUST HAVE A ZERO IN COL 5
  102 FORMAT(I5,F10.0,F10.2,F10.0)
C     READ UM DWMBUIDEWAY DEMAND DATA - SPEED PER LINK AND ADD NO. OF LI
      NLM = 0
  105 READ 1U5, NL,(DDNPFL(NL,NV,3),NV=1,4),(DPKHRF(NL,NV),NV=1,4),
     *(DPKIMP(NL,NV),NV=1,4), DDLKST(NL,1)
      NLM = NLM + 1
      IF(NL .NE. 225)  GO TO 105
      NLM = NLM - 1
C     NOTE LAST CARD MUST HAVE A ZERO IN COL 4
  1U5 FORMAT (I4,4F1N.0,8F4.2,F6.0)
C     READ UM DWMGUIDEWAY BACK-HAUL FACTORS PER LINK AND VEHICLE FLEET
C     FOR DESIGN HOUR, PEAK PERIOD, + OFF- PEAK PERIOD
      DO 17  NL = 1,NLM
      DO 17  NV = 1,NVM
      DO 17  NTP = 1,3
   17 BACKDN(NL,NV,NTP) = 0.0
   1A READ 1V, NL, (IRACKDN(NL,NV,NTP), NTP = 1,3), NV = 1,NVM)
   1V FORMAT(I5,9F6.3)
      IF (NL .NE. 225)  GO TO 1A
C     READ  LAND UNIT COSTS
      READ 1S5, (ULANDC(LC), LC = 1,10)
  1S5 FORMAT(10F8.0)
C     READ ZONES PER NO. OF LANES
      READ 1N5, (DWMNTH(LNS), LNS=1,10)
  1N5 FORMAT(10F8.0)
C     READ UNIT TRACK COST AND UNIT GUIDEWAY STRUCTURES COST + ROW WIDTH
      READ 15N, UTRARC, (USTRUC(K), K = 1,4)
  15N FORMAT(5F10.0)
C     READ UNIT DESIN, DENSITIES PER HES,DENSITY CODES (PERSONS/ACRE)
      DO 115  JHD = 1,6
  115 READ 112U, (HESDEN(JRD,JHD),  JHDC=1,4)
  112U FORMAT(6F10.2)
C     READ NOISE DISTANCE TABLE , DAYINT=11, NIGHT(NT=2)
      DO 150 NT = 1,2
      DO 150  ND = 1,20
      READ 12U, (DNOISE(ND,NDD,NT), NDD = 1,14)
  12U FORMAT(I4,14 2F5.0, 11F4.1)
  150 CONTINUE
      NDM = 20
C*********************************************************************
C
C     READ INPUTS FOR OTHER OM CAPITAL COSTS
C
C*********************************************************************
C
C     READ CONTROL CRITER INPUTS
```



```
      *1
      READ 9U04,(DPTRIP(1,LH),DPTRIP(2,LH),DPDLYD(1,LH),DPDLYD(2,LH),
     *      DVDLAY(1,LB),DVMLDF(1,LH),DMVLDF(2,LH)),LR=1,LV)
 9A04 FORMAT (5I10,F10.0,F10.0,F10.0,F10.0,F10.0,F10.0)
      READ 9U06, JRX,(IRACKOF (JRX,LVX,NTP) ,NTP=1,2),(VXM=1,LV)
 9A06 FORMAT (I5,8F6.3)
      IF (JRX .NE. 15)  GO TO 1020
C*********************************************************************
C     READ  TRANSIT AND HIGHWAY NETWORK DATA
C*********************************************************************
      IAV=NMM+1
      DO 9  IC =1,5
    9 READ 6U0, (ZSYSCH(IX,IC),IX=1,IAV)
 6U0 FORMAT(4F1N.2)
      READ 6U2,ZPKPCT
      DO 11 IA =1,2
   11 READ 6U0, (ZPTRIP(IA,IX),IX=1,IAV)
      DO 12 IT =1,2
   12 READ 6U0, (ZDELAY (IA,IX,IT),IX=1,2),IX =1,IAV)
      READ 93 , (MLNDIR (MMX,ITPA),ITPA=1,2),MMX=1,2)
   93 FORMAT (4F10.3)
      READ 6U4, (ZWMTPC(IX),IX=1,MM)
      READ 9U4, (DWMTPC(IM),IM=1,LV)
   94 FORMAT(4F1N.2)
      READ  1AA (AUTOCH (IK),IK =1,10)
  1AA FORMAT (F10.0,F8.2,4F10.0,2F5.2,2F4.0,F5.0)
      READ 9U, (NMCHAR(IA), IA=1,4)
   9U FORMAT(F10.0,F8.2,2F10.0,F10.0)
C*********************************************************************
C     READ  POWER  INPUT  DATA
C*********************************************************************
      DO  A1 LVX = 1,LV
   A1 READ 82, (HPCVCH(LVX,ICH=1),ICH=1,6)
      DO  A3 LVX = 1,LV
   A3 READ 82, (HPCVCH(LVX,ICH=2),ICH=1,6)
   82 FORMAT(F10.1,2F10.3,2F10.1,F10.0)
      DO  A4 IAV=1,IAV
   A4 READ 92, (HPCVCH(IAVX,ICH),ICH=1,6)
   35 FORMAT (4I3)
   3A READ 3U, (HPCVCH(NMVX,ICH),ICH=1,5)
   3A FORMAT(F10.1,2F10.3,2F10.1)
C*********************************************************************
C     READ  AIR  POLLUTION  INPUT  DATA
C*********************************************************************
      READ 2U0, ((IPOLCD (LVX=IY,1),IY=1,2),LVX=1,LV)
      READ 2U5, (IPOLCD (IXVX,1,2),IXVX=1,IXV)
      READ 2U6, (IPOLCD (NMVX=1,3),NMVX=1,NMV)
  200 FORMAT (4I3)
  2U5 FORMAT (4I3)
  2U6 FORMAT (2I3)
      DO 2U8 IUX=1,3
  2U8 READ 210, (POLWAT (IUX,IY),IY=1,3)
  210 FORMAT (3F10.0)
C*********************************************************************
C     READ  ACCIDENT  INPUT  DATA
C*********************************************************************
```

```
RUN VERSION 2.7 --PSR LEVEL 29A--                    RDINPUT                    06/30/72

001251          READ  400, DMSTAC , (IDACCD (LVX),LVX=1,LV)                     RDIN 221
001264    400 FORMAT (F10.3,I4(5)                                               RDIN 222
0012AA          DO  403 IMX =1,7                                                RDIN 223
001278    403 READ  402 , (IMCRAT (IMX,IA,ITP),IA=1,5),ITP=1,2)                 RDIN 224
00131B    402 FORMAT (10F8.2)                                                   RDIN 225
001318          DO  405 IAVX =1,IAV                                             RDIN 226
001312    405 READ  404 , (ASFRAT (IAVX,IM),IM=1,4)                             RDIN 227
001327    404 FORMAT (2F10.2,2F10.3)                                            RDIN 228
001329          READ 399 , ((UACRAT(IAT,LVX),IAT=1,2),LVX=1,LV)                 RDIN 229
001344    399 FORMAT (8F10.3)                                                   RDIN 230
C*****************************************************************************RDIN 231
C         READ HIGHWAY OPERATING COST  INPUT DATA                               RDIN 232
C*****************************************************************************RDIN 233
001345          READ 800,  ((HOPDAT(MHX,IV),MHX=1,2),IV=1,2)                    RDIN 234
001367    800 FORMAT  (5X,8F10.3)                                               RDIN 235
001367          DO  806  MHX =1,MHV                                             RDIN 236
001389    806 READ 805, (HVOPCS (MHVX,ICT),ICT =1,6)                            RDIN 237
001407    805 FORMAT (6F10.3)                                                   RDIN 238
001407          DO  808  MHX =1,2                                               RDIN 239
001403    808 READ 807, (HWYOPC (MHX,IT),IT =1,3)                               RDIN 240
001417    807 FORMAT (3F10.3)                                                   RDIN 241
C*****************************************************************************RDIN 242
C         READ TRANSIT  OPERATING  COST  INPUT  DATA                            RDIN 243
C*****************************************************************************RDIN 244
C         ZERO  UNIT  INPUT  ARRAYS  BEFORE  READING  INPUT  DATA               RDIN 245
001417          DO 1199  IXXX = 1,4                                             RDIN 246
001421          DO  1199  JXXX = 1,7                                            RDIN 247
001427          DO 1199  KXXX = 1,4                                             RDIN 248
001423          XOPCU  ((IXXX,JXXX,KXXX) = 0.0                                  RDIN 249
001430    1199 XOPICD ((IXXX,JXXX,KXXX) = 0.0                                   RDIN 250
001441          DO 1001  IXVX =1,IXV                                            RDIN 251
001447    1001 READ 1002, (XOPFUI(IXVX,NXX,1), NXX=1,3), ((XOPCU(IXVX,NXX,MX),  RDIN 252
               X  MX=1,4), NXX=4,5)                                             RDIN 253
001473    1002 FORMAT (3F10.0,4F4.0,8X4F4.0)                                    RDIN 254
001477          DO 1006  IXVX =1,IXV                                            RDIN 255
001474          READ 1004 ,((XOPICD (IXVX,1,KX1,KX0),1) ,(XOPICD (IXVX,2,KX),KX =1  RDIN 256
               5,2), (XOPICU (IXVX,JX,1),JX =3,5)                               RDIN 257
001524    1006 READ 1005 , ((XPPICD (IXVX,JX,KX),KX =1,4),JX=6,7)               RDIN 258
001551    1005 FORMAT(4F7.0,4F4.0)                                              RDIN 259
001551    1004 FORMAT  (F4.3,F7.2,F9.0,2F7.3,3F7.0)                             RDIN 260
C*****************************************************************************RDIN 261
C*****************************************************************************RDIN 262
C                                                                               RDIN 263
C         READ  OWN VEHICLE OPERATING COST DATA                                 RDIN 264
001551          DO 810  NV = 1,NVH                                              RDIN 265
001557    810 READ 805,NVA, NVC(NV), (DVOPUC(NV,NVD,1),NVD=1,7), (DVOPUC(NV,NVD,RDIN 266
               2), NVD =1,7)                                                    RDIN 267
001602    805 FORMAT(I2I5,14F5.3)                                               RDIN 268
C                                                                               RDIN 269
C         READ TERMINAL OPER. COST DATA                                         RDIN 270
C         INITIALIZE OTROPC                                                     RDIN 271
001602          DO 845  NDT =1,11                                               RDIN 272
001607          DO 845  NDTV=1,7                                                RDIN 273
001604    845 OTROPC(NDT,NDTD) = 0.0                                            RDIN 274
001614          NDTM = 0                                                        RDIN 275
```

```
RUN VERSION 2.7 --PSR LEVEL 29A--                    RDINPUT                    06/30/72

001614    470 READ 675, NDT,(OTROPC(NDT,NDTD), NDTD =1,4)                       RDIN 276
001633        NDTM = NDTM + 1                                                   RDIN 277
001634        IF (NDT .NE. 11)  GO TO 470                                       RDIN 278
001637        NDTM = NDTM - 1                                                   RDIN 279
C         NDTF LAST CARD MUST HAVE A ZERO IN COL 5                              RDIN 280
001640    470 FORMAT(I3,F5.0,2F15.0,F10.0)                                      RDIN 281
C         READ AND COMPUTE OH GUIDEWAY OPERATING COST                           RDIN 282
001660          READ 700, (UGOPF(JG), JG = 1,3)                                 RDIN 283
001651    700 FORMAT(3F10.0)                                                    RDIN 284
C         READ OPRM COST/YARD ACRES, YEARLY COM.CENTER AND MAIN. FACIL COSTSRDIN 285
001651          READ 715, HYUOPC, OCCOPC, OMOPC                                 RDIN 286
001663    715 FORMAT(F10.2,2F10.0)                                              RDIN 287
C         READ FLEET EQUIVALENT FACTORS AND LOCAL CAPITAL FACTORS               RDIN 288
001669          READ I=01, ((ZI(MV), MV =1,4), (FLOCAP(JLC), JLC =1,5)          RDIN 289
001701    1401 FORMAT(4F5.0,5F4.2)                                              RDIN 290
C                                                                               RDIN 291
C         READ  UNIT FARES                                                      RDIN 292
001701          DO 1400  NV = 1,NVH                                             RDIN 293
001706    1400 READ 15U4, (FARE(NV,1,1), OFARE(NV,2,1), OFARE(NV,1,2),          RDIN 294
               * OFARE(NV,2,2)                                                  RDIN 295
001723    1505 FORMAT(4X3,4F13.3)                                               RDIN 296
001725         RETURN                                                           RDIN 297
001727         END                                                              RDIN 298
```

```
RUN VERSION 2.7 --PSR LEVEL 29A--                    RDINPUT  06/30/72

RDINPUT

SUBPROGRAM LENGTH
002267

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
5      = 001727   1U     = 000823   1S     = 001733   17     = 000162
18     = 001177   19     = 001752   20     = 001731   30     = 001730
36     = 002066   4U     = 001740   60     = 002047   62     = 002062
94     = 002093   95     = 002051   100    = 000066   101    = 000067
102    = 001742   105    = 000111   106    = 001746   130    = 001765
140    = 001747   154    = 001755   15A    = 001761   189    = 002055
200    = 002072   205    = 002074   206    = 002076   210    = 002100
300    = 002112   400    = 002102   402    = 002105   404    = 002107
510    = 001771   512    = 001775   515    = 000355   516    = 001777
51A    = 002003   612    = 002007   645    = 002136   665    = 001604
670    = 001414   674    = 002141   700    = 002145   715    = 002147
800    = 002114   805    = 002117   807    = 002121   1002   = 002173
1004   = 002132   1005   = 002127   1410   = 000503   1014   = 000542
1016   = 000573   1020   = 000667   1120   = 001763   1200   = 002613
1401   = 002152   1505   = 002155   8001   = 002016   8002   = 002020
8003   = 002025   8004   = 002033   8005   = 002037   800A   = 002044

BLOCK NAMES AND LENGTHS
RDI    = 000767   RUN    = 034804   SVI    = 000310   SVH    = 003410
LUPS   = 000020   TITLI  = 000810

VARIABLE ASSIGNMENTS
ASSRAT = 000204C02 AUTOCH = 021415C02 BACKOF = 031470C02 BACKON = 012575C02
DACRAT = 021922C02 DCCOPC = 033275C02 DCHCEN = 000070C02 DFARE  = 033744C02
DGLKST = 010410C02 DGOPC  = 033750C02 DGVDIM = 000050C02 DINT   = 000154C01
DLIFE  = 000120C01 DLWCAP = 000024C02 DMAINT = 000211C02 DMOPC  = 033274C02
DMSTAC = 021955C02 DMVLHF = 021307C02 DNO15E = 000274C02 DNUMVH = 000257C02
DOFLRC = 026742C02 DUFRFL = 022222C02 DONLRC = 010430C02 DONRFL = 001418C02
DPCVCH = 021474C02 DPDLYN = 021273C02 DPKNOF = 020011C02 DPKWHF = 010546C02
DPKPCT = 021284C02 DPTRIP = 021257C02 DTRAPC = 033833C02 DVDLAY = 021301C02
DVHCAP = 000013C02 DVMLHF = 000017C02 DVHSUC = 000223C02 DVMUC  = 000253C02
DVWTPC = 021412C02 DVOPUC = 033303C02 DWAYCC = 000103C02 DYARDS = 000110C02
DYUOPC = 033274C02 FLOCAP = 033757C02 HACWAT = 021556C02 HLNNIR = 021402C02
HNCHAR = 021434C02 HOPDAT = 021630C02 HRCVCH = 021530C02 HVOPCS = 021648C02
HWYOPC = 021464C02 IA     = 002231   IAT    = 002250   IN     = 002247
IC     = 002237   ICH    = 002240   ICT    = 002253   IDACCD = 000271C01
IK     = 002237   IM     = 002230   IMX    = 002245   IPOLCU = 000241C01
ISAVNT = 000000C01 IT     = 002234   ITP    = 002246   ITPX   = 002230
IUX    = 002244   IV     = 002252   IX     = 002223   IXV    = 000248C01
IXVX   = 002241   IXXX   = 002254   IY     = 002243   IZ     = 000225C01
IZM    = 002172   JCH    = 002210   JG     = 002265   JGA    = 002217
JK     = 002234C01 JKX   = 002226   JKY    = 002220   JLC    = 002266
JMD    = 002214   JMD    = 002203   JMDC   = 002204   JSF    = 002173
JSP    = 002174   JSPRNT = 000000C01 JW0    = 002211   JX     = 002261
JXXX   = 002245   K      = 002202   KX     = 002260   KXXX   = 002256
```

```
RUN VERSION 2.7 --PSR LEVEL 29A--                    RDINPUT  06/30/72

L      = 002171   LM     = 002231   LC     = 002280   LECODE = 000214C01
LK     = 002234C01 LKY   = 002221   LKS    = 002201   LV     = 000215C01
LVX    = 002207   LXY    = 002222   LX     = 002224   MM     = 002251
MMX    = 002214   MX     = 002257   MD     = 002280   MDU    = 002207
MDW    = 002234C01 MDZ   = 002263   MDTD   = 002264   MDTM   = 000275C01
MM     = 002234C01 MMX   = 000274C01 MMVX   = 002242   ML     = 002176
MLM    = 002214C01 MLMM  = 000004C05 MPAMBS = 021254C02 MT     = 002205
MTP    = 002177   MUMTR  = 021255C02 MV     = 002175   MVA    = 002215
MVC    = 002234C01 MVH   = 032262   MVW    = 000215C01 MVS    = 002216
MX     = 002234C01 MXX   = 032225   MY     = 002212   MYD    = 002213
MYM    = 002234C01 PHWCH1 = 000227C01 POLMAT = 021544C02 RESDEN = 001354C02
RWTTL  = 000000C04 SSAVIT = 000000C04 SCALEF = 000000C02 SMAVDT = 000310C04
TARMAT = 000234C02 ULAYOF = 000031C02 USTRUC = 000044C02 UTHARC = 000043C02
XACMAT = 021402C02 XDELAY = 021382C02 XMPCU  = 021442C02 XOPICD = 022042C02
XPCVCH = 021540C02 XPRPCT = 021341C02 XPTRIP = 021342C02 XSYSCH = 021315C02
XVHTPC = 021444C02 /      = 033277C02

START OF CONSTANTS
001724

START OF TEMPORARIES
002164

START OF INTERFACTS
002167

UNUSED COMPILER SPACE
12120
```

```
                SUBROUTINE PRNTIN                                       PRIN   2
000002          DIMENSION KA(1)                                         PRIN   3
000002          COMMON/SVI/ ISAVDT(100,2)                               PRIN   4
000002          COMMON/SVR/ RSAVDT(100,2), SMRVDT(320,5)                PRIN   5
000002          COMMON/RD1/ IPARDT(199)                                 PRIN   6
000002          COMMON/RDR/ RPARDT(1a3,8)                               PRIN   7
000002          COMMON/TITL1/ RNTITL(8)                                 PRIN   8
000002          DATA KA/1,2,3,4,5,6,7,8,9,10/                           PRIN   9
000005          PRINT R000                                              PRIN  10
000006   R000 FORMAT(1H1,9 REST ENGINEERING ESTIMATE INPUT DATA9,////)  PRIN  11
C               PRINT INTEGER INPUTS                                    PRIN  12
000004          KH = 1                                                  PRIN  13
000007          KE = 1U                                                 PRIN  14
000010          KC = 0                                                  PRIN  15
000011          PRINT R001, KA                                          PRIN  16
000017   R001 FORMAT(1H ,1WX,10I12,/)                                   PRIN  17
000017          DO R002  KK = 1,19                                      PRIN  18
000021          PRINT R004, KC, (IPARDT(K), K=KH,KE)                    PRIN  19
000036          KC = KC+10                                              PRIN  20
000037          KB = KH + 1U                                            PRIN  21
000041   R002 KE = KE + 1U                                              PRIN  22
000044   R004 FORMAT(1H ,I5,9X,10I12)                                   PRIN  23
C               PRINT REAL INPUTS                                       PRIN  24
000044          KC = 0                                                  PRIN  25
000046          KH = 1                                                  PRIN  26
000046          KE = 1U                                                 PRIN  27
000047          PRINT R005                                              PRIN  28
000052   R005 FORMAT(9X)                                                PRIN  29
000052          PRINT R003, KA                                          PRIN  30
000060          DO R005  KK = 1,1434                                    PRIN  31
000062          PRINT R010,KC, (RPARDT(K), K = KB,KE)                   PRIN  32
000076          KC = KC+10                                              PRIN  33
000100          KB = KB + 1U                                            PRIN  34
000102   R005 KE = KE + 1U                                              PRIN  35
000106   R010 FORMAT(1H , I5,9X,10F12.3)                                PRIN  36
000105          RETURN                                                  PRIN  37
000106          END                                                     PRIN  38
```



```
PRNTIN

SUBPROGRAM LENGTH
000157

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
R000   =  000111    R001   =  000121    R004   =  000124    R010   =  000131
R005   =  000137

BLOCK NAMES AND LENGTHS
SVI    =  000110    SVR    =  003410    RD1    =  000307    RDR    =  034004
TITL1  =  000010

VARIABLE ASSIGNMENTS
IPARDT =  000000C01 ISAVDT =  000000C01 K      =  000156    KA     =  000140
KB     =  000152    KC     =  000154    KE     =  000153    KH     =  000155
RNTITL =  000000C05 RPARDT =  000000C04 RSAVDT =  000000C02 SMRVDT =  000310C02

START OF CONSTANTS
000107

START OF TEMPORARIES
000136

START OF INDIRECTS
000140

UNUSED COMPILED SPACE
130100
```

```
      SUBROUTINE SUMDATA
C
C*********************************************************************
C     THIS SUBROUTINE EQUATES (COMMON) DATA WHICH IS TO BE SAVED FOR
C     SUMMARIES IN SMRYDT ARRAY
C
C*********************************************************************
C
      INTEGER *LIFE    , DINT
      INTEGER  PARCK3
      REAL  NPRMRS , NUMDA
      COMMON/WNT/      JSPRNT(88)    , DLIFE(30)     ,
C     LFCOUF           , NVM         , NLM           , NSM           ,
C     NTM              , NTM         , NTSM          , NVM           ,
C     NMM              , IZ          , JK            , PARCK3(6)     ,
C     NN               , NB          , IPRINT        , IAV           ,
C     IDOLCD(4,2,3),   IDACCD(3)     , NMV           , NUTM          ,
C     NVC(8)
      COMMON/WNW/      SCALEF(10)    , DVMCAP(5)     , DVMLDF(6)     ,
C     DLMCAP(5)        , ULANDF(10)  , UTPARC        , USTRUC(6)     ,
C     DGMUTM(10)       , UCNCFN(11)  , DMAYCC(5)     , UTARDS(5,13)  ,
C     UMATNT(10)       , DVMSUF(4,4) , DYMUC(4)      , DNUMVM(5)     ,
C     ASSWAT(4)        , TARRAT(4)   , UNOISE(20,14,2), RESDEN(4,7)  ,
C     UDUMA(225,4,4),  DGLRST(225,5) , BACRON(225,3,3), DPKMRF(225,3) ,
C     UBKMRF(225,3)    ,
C     NPRMRS           , NUMDA       , DPAPCT        , DPTRIP(4,3)   ,
C     UBMTYH(2,3)      , DVDLAY(2,3) , DMVLDF(2,3)   , ASYSCM(4,5)   ,
C     SUMPCT           , APTRIP(4,4) , ALELAY(2,4,2) , MLMDIR(2,2)   ,
C     AUMTPC(4)        , DVMTPC(3)   , AUTOCH(11)    , MMCHAR(4)     ,
C     DPCVCH(3,4,2)    , APCVCH(4,4) , MMCVCH(2,4)   , POLMAT(3,3)   ,
C     DMSTAC           , MACRAT(2,5,2) , RACRAT(4,4) , UACRAT(2,3)   ,
C     MOPRAT(2,4)      , MUOPCS(2,4) , MAPOPC(2,3)   , ZOPCU(4,7,4)  ,
C     ZOPTCU(4,7,4)    ,DDFPFL(150,4,4),DNFLRC(150,3,3),MACROF(150,3,2),
C     DYMOPC           , DCCOPC      , DMOPC         , Z(4)          ,
C     DYOMU(24,9,4)    , DYMOPC(11,7)  , DBOPC(7)    , FLOCAP(5)     ,
C     UPAMTR(4,7,2)
      COMMON/SYM/ MSAMNT(100,7), SMRYDT(320,5)
      COMMON/LIMS/ APT, LPS, NLP, IRC, NLPMI(5), LPSM
      COMMON /DV12/ ASUMR(14,4), DTATMI, DTLNMI, DTYDA
      COMMON /DV23/ NPRMRS , DVTRIP(4,4)  , UTOPMT(4,5)  ,UTOVMT(4,5),
C     DMMUAT(4,3,4)  , DLVDAT(4,3,4)  ,  DDFVMT(150,4,4)  ,
C     UDFVMT(150,4,4),  DDMVMT(225,4,4)
      COMMON /DV13/  ,UTOMD(5,4,12)  ,  TWCMD(5,4,12)  , XMDATA(5,6,12),
C     DDMPWM(5,4,12) , DDPPWR(5,4,12) , PPWRC(5,6,12) , MPWRC(5,6,12),
C     DMVMT(5,4,12)
      COMMON/ UTTULNMI,DGLNMI,DILNMI,THLNMI,DALNMI,DYLNMI
      COMMON /MM/ MMTEMP
      MM = MMTEMP
      SMRYDT(1,MM) = NLM
      SMRYDT(2,MM) = NSM
      SMRYDT(3,MM) = NJM
      SMRYDT(4,MM) = NTM
      SMRYDT(5,MM) = NTSM
```



```
      SMRYDT(6,MM) = DTATMI
      SMRYDT( 7,MM) = TOLNMI
      SMRYDT( 8,MM) = DNUMVM(1)
      SMRYDT(10,MM) = DVMCAP(1)
      SMRYDT(11,MM) = DVMLDF(1)
      SMRYDT(12,MM) = DLMCAP(1)
      IF (NVM .GT. 1) GO TO 25
      SMRYDT(14,MM) = DNUMVM(2)
      SMRYDT(24,MM)=DYMUC(2)
   25 CONTINUE
      SMRYDT(14,MM) =  DVMCAP(2)
      SMRYDT(15,MM) =  DVMLDF(2)
      SMRYDT(16,MM) =  DLMCAP(2)
      SMRYDT(17,MM) =  NTM
      SMRYDT(18,MM) =  UTMARC
      SMRYDT(19,MM) =  USTRUC(4)
      SMRYDT(23,MM) =  USTRUC(1)
      SMRYDT(21,MM) =  USTRUC(2)
      SMRYDT(22,MM) =  USTRUC(3)
      SMRYDT(23,MM)=DYMUC(1)
      SMRYDT(25,MM)=DILNMI
      SMRYDT(26,MM)=THLNMI
      SMRYDT(27,MM)=TOLNMI
      SMRYDT(28,MM)=DALNMI
      SMRYDT(29,MM)=DYLNMI
      K = IVM + 4
      SMRYDT(31,MM) =  ASUMR(K+5,1)
      SMRYDT(32,MM) =  ASUMR(K+5,3)
      TOVMCC = 0.0
      TOVMAC = 0.0
      PRIVCC=0.0
      DO 110 NW=1,NVM
      IZ = NV + 4
      TOVMAC = TOVMAC + ASUMR(IZ,3)
      TOVMCC = TOVMCC + ASUMR(IZ,1)
      IF (NVC(NV) .LE. 2) GO TO 100
      GO TO 110
  100 CONTINUE
      PRIVCC=PRIVCC+ASUMR(IZ,1)
  110 CONTINUE
      PURCC=ASUMR(K+5,1)-PRIVCC
      SMRYDT(34,MM)=PURCC
      SMRYDT(35,MM)=PRIVCC
      SMRYDT(36,MM) = TOVMAC
      SMRYDT(37,MM) = TOVMCC
      SMRYDT(45,MM) =  ASUMR( 1,3)
      SMRYDT(46,MM) =  ASUMR(K+1,3)
      SMRYDT(47,MM) =  ASUMR(K+2,3)
      SMRYDT(48,MM) =  ASUMR( 2,3)
      SMRYDT(49,MM) =  ASUMR(K+3,3)
      SMRYDT(50,MM) =  ASUMR( 3,3)
      SMRYDT(51,MM) =  ASUMR(K+4,3)
      SMRYDT(52,MM) =  ASUMR( 4,3)
      SMRYDT(53,MM) =  ASUMR( 5,3)
      IF (NVM .EQ. 1)  GO TO 30
```



```
      SMRYDT(54,MM) =  ASUMR( 6,3)
      GO TO 40
   30 SMRYDT(54,MM) = 0.0
   40 CONTINUE
      SMRYDT(55,MM) =  ASUMR( 1,1)
      SMRYDT(56,MM) =  ASUMR(K+1,1)
      SMRYDT(57,MM) =  ASUMR(K+2,1)
      SMRYDT(58,MM) =  ASUMR( 2,1)
      SMRYDT(59,MM) =  ASUMR(K+3,1)
      SMRYDT(70,MM) =  ASUMR( 3,1)
      SMRYDT(71,MM) =  ASUMR(K+4,1)
      SMRYDT(72,MM) =  ASUMR( 4,1)
      SMRYDT(73,MM) =  ASUMR( 5,1)
      IF (NVM .EQ. 1) GO TO 50
      SMRYDT(74,MM) =  ASUMR( 6,1)
      GO TO 60
   50 SMRYDT(74,MM) = 0.0
   60 CONTINUE
      IAUT = IAV +1
      LV = NVM
      LVMO = LV - 1
      SMRYDT ( 81,MM) =  DTOPMT(3,1)
      SMRYDT ( 83,MM) =  DTOVMT(3,1)
      SMRYDT ( 85,MM) =  DTOPMT(4,1)
      SMRYDT ( 87,MM) =  DTOVMT(4,1)
      IF (LV .NE. 2) GO TO 9888
      SMRYDT ( 89,MM) =  DTOPMT(3,2)
      SMRYDT ( 91,MM) =  DTOVMT(3,2)
      SMRYDT ( 93,MM) =  DTOPMT(4,2)
      SMRYDT ( 95,MM) =  DTOVMT(4,2)
      GO TO 9887
 9888 SMRYDT(89,MM) = 0.0
      SMRYDT(91,MM) = 0.0
      SMRYDT(93,MM) = 0.0
      SMRYDT(95,MM) = 0.0
 9887 CONTINUE
      SMRYDT ( 97,MM) =  DPTRIP(3,1)
      SMRYDT ( 98,MM) =  DVTRIP(3,1)
      SMRYDT ( 99,MM) =  DPTRIP(4,1)
      SMRYDT (100,MM) =  DVTRIP(4,1)
      SMRYDT (101,MM) =  DPTRIP(3,2)
      SMRYDT (102,MM) =  DVTRIP(3,2)
      SMRYDT (103,MM) =  DPTRIP(4,2)
      SMRYDT (104,MM) =  DVTRIP(4,2)
      IF (DPTRIP(3,1) .EQ. 0.0) GO TO 9999
      SMRYDT (105,MM) =  SMRYDT(81,MM) / DPTRIP(3,1)
      SMRYDT (106,MM) =  SMRYDT(82,MM) / DPTRIP(3,1)  *60.
      GO TO 9998
 9999 SMRYDT(105,MM) = 0.0
      SMRYDT(106,MM) = 0.0
 9998 IF (DPTRIP(3,2) .EQ.0.0)  GO TO 9997
      SMRYDT (107,MM) =  SMRYDT(89,MM) / DPTRIP(3,2)
      SMRYDT (108,MM) =  SMRYDT(90,MM) / DPTRIP(3,2)  *60.
      GO TO 9996
 9997 SMRYDT(107,MM) = 0.0
```

```
RUN VERSION 2.3 --PSR LEVEL 29A--                                   SUMDATA                    06/30/72

000441          SMRYNT(108,MM) = 0.0                                                           SMDT 166
000444    0006  IXV = IXV + 1                                                                  SMDT 167
000446          SMRYNT(109,MM) =  XNDATA(IXV,3,1)                                              SMDT 168
000479          SMRYNT(110,MM) =  XNDATA(IXV,3,2)                                              SMDT 169
000475          SMRYNT(111,MM) =  XNDATA(IXV,3,3)                                              SMDT 170
000508          SMRYNT(112,MM) =  XNDATA(IXV,3,6)                                              SMDT 171
000503          SMRYNT(113,MM) =  XNDATA(IXV,4,1)                                              SMDT 172
000506          SMRYNT(114,MM) =  XNDATA(IXV,4,2)                                              SMDT 173
000511          SMRYNT(115,MM) =  XNDATA(IXV,4,5)                                              SMDT 174
000514          SMRYNT(116,MM) =  XNDATA(IXV,6,6)                                              SMDT 175
000517          IF (SMRYNT(117,MM) .EQ. 0.0) GO TO 9995                                        SMDT 176
000522          SMRYNT(119,MM) =  XNDATA(IXV,3,1) / SMRYDT(117,MM)                             SMDT 177
000527          SMRYNT(120,MM) =  XNDATA(IXV,3,3) / SMRYDT(117,MM) *60.                        SMDT 178
000534          GO TO 9994                                                                     SMDT 179
000534    9995  SMRYNT(119,MM) = 0.0                                                           SMDT 180
000537          SMRYNT(120,MM) = 0.0                                                           SMDT 181
000542    9994  SMRYDT(124,MM) = AUTONO(3,3,1)                                                 SMDT 182
000546          SMRYNT (124,MM) =  AUTONO(3,3,2)                                               SMDT 183
000551          SMRYNT (127,MM) =  AUTONO(3,3,5)                                               SMDT 184
000554          SMRYNT (128,MM) =  AUTONO(3,3,6)                                               SMDT 185
000557          SMRYNT (129,MM) =  AUTONO(3,4,1)                                               SMDT 186
000562          SMRYNT (130,MM) =  AUTONO(3,4,2)                                               SMDT 187
000565          SMRYNT (131,MM) =  AUTONO(3,4,5)                                               SMDT 188
000570          SMRYNT (132,MM) =  AUTONO(3,4,6)                                               SMDT 189
000573          SMRYNT (133,MM) =  TRCKNO(3,3,2)                                               SMDT 190
000576          SMRYNT (134,MM) =  TRCKNO(3,4,2)                                               SMDT 191
000601          SMRYNT (135,MM) =  AUTONO(3,3,2)                                               SMDT 192
000604          SMRYNT (136,MM) =  TRCKNO(3,3,1)                                               SMDT 193
000607          LV = NVW                                                                       SMDT 194
000610          LVY = LV + 1                                                                   SMDT 195
000612          SMRYNT (138,MM) =  DMVWT (3,3,LVY)                                             SMDT 196
000615          SMRYNT (140,MM) =  AUTONO(3,4,2)                                               SMDT 197
000621          SMRYNT (141,MM) =  TRCKNO(3,4,1)                                               SMDT 198
000624          SMRYNT (143,MM) =  DMVWT (3,4,LVY)                                             SMDT 199
000631          SMRYDT(144,MM) = SMRYDT(84,MM) + SMRYDT(88,MM) + SMRYDT(106,MM) +              SMDT 200
                * SMRYDT(125,MM)
000643          SMRYDT(146,MM) = SMRYDT(83,MM) + SMRYDT(91,MM) + SMRYDT(110,MM) +              SMDT 201
                * SMRYDT(136,MM) + SMRYDT(134,MM)                                              SMDT 202
000651          SMRYDT(151,MM) = SMRYDT(87,MM) + SMRYDT(101,MM) + SMRYDT(117,MM) +             SMDT 203
                * SMRYDT(145,MM)                                                               SMDT 204
000660          SMRYDT(152,MM) = SMRYDT(89,MM) + SMRYDT(93,MM) + SMRYDT(113,MM) +              SMDT 205
                * SMRYDT(129,MM)                                                               SMDT 206
000664          SMRYDT(153,MM) = SMRYDT(97,MM) + SMRYDT(95,MM) + SMRYDT(114,MM) +              SMDT 207
                * SMRYDT(130,MM) + SMRYDT(141,MM)                                              SMDT 208
000674          SMRYDT(154,MM) = SMRYDT(98,MM) + SMRYDT(103,MM) + SMRYDT(118,MM) +             SMDT 209
                * SMRYDT(144,MM)                                                               SMDT 210
000704          IF (SMRYDT(151,MM) .EQ. 0.0) GO TO 9993                                        SMDT 211
000706          SMRYNT (155,MM) = SMRYDT (149,MM) / SMRYDT (151,MM)                            SMDT 212
000711          SMRYNT(156,MM) = (SMRYDT(82,MM) + SMRYDT(90,MM) + SMRYDT(111,MM)               SMDT 213
                * + SMRYDT(127,MM))  / SMRYDT(151,MM) * 60.0                                   SMDT 214
000720          GO TO 9992                                                                     SMDT 215
000721    9993  SMRYNT(155,MM) = 0.0                                                           SMDT 216
000724          SMRYNT(156,MM) = 0.0                                                           SMDT 217
000727    9992  SMRYNT(157,MM) = 0.0                                                           SMDT 218
000732          SMRYNT (158,MM) =  0.0                                                         SMDT 219
                                                                                               SMDT 220
```

```
RUN VERSION 2.3 --PSR LEVEL 29A--                                   SUMDATA                    06/30/72

000735          SMRYNT (159,MM) =  0.0                                                         SMDT 221
000737          SMRYNT (160,MM) =  0.0                                                         SMDT 222
000742          RETURN                                                                         SMDT 223
000742          END                                                                            SMDT 224
```

```
RUN VERSION 2.3 --PSR LEVEL 29A--                                   SUMDATA  06/30/72

SUMDATA

SUBPROGRAM LENGTH
001164

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
25    -  000040    J4    -  000251    4A    -  000254    5A    -  000317
6A    -  000392    IJ6   -  000104    110   -  000167    9B47  -  00037A
9B95  -  000534    9996  -  000464    9997  -  000721    9994  -  000542
9999  -  000434                                                  -  000444

BLOCK NAMES AND LENGTHS
RD1   -  000307    MIN   -  034884    SVR   -  003410    LUPS  -  000824
DV17  -  000127    DV23  -  028631    DV33  -  005508    LNT   -  000006
MM    -  000001

VARIABLE ASSIGNMENTS
ASSMT -  000244C07 ASJMP -  000000C09 AUTOCH -  021415C02 AUTONO -  000000C07
BACKOF -  011475C02 BACKON -  012575C02 DACHAT -  021022C02 DALNMI -  000004C10
DRMHAT -  000071C04 DLNCEN -  000076C02 DFARF -  033764C02 DGLKST -  010430C02
DSLNMI -  000001C10 DWUPC -  033756C02 DGNDTH -  000050C02 DHVWT -  004739C07
DLVDAT -  000201C06 DMAINT -  000100C01 DLIFE -  001200C01 DLNCAP -  000024C02
DMUWVH -  000297C07 DUFLRC -  026762C02 DOFPFL -  022222C02 DOFPWR -  002644C07
DOFVHT -  000411C04 DUFVHT -  050851C06 DONPFL -  001418C02 DONPWR -  002970C07
DONVHT -  011411C04 DPCVCH -  021434C07 DPDLYN -  021273C02 DPRDRF -  029811C02
DPRHRF -  016544C02 DPTRIP -  021257C07 DTOPHT -  000021C06 DTOVHT -  000045C06
DTRDPF -  013613C02 DTUTMI -  000126C05 DVDLAY -  021381C02 DVMCAP -  000012C02
DVLMLF -  000017C07 DVMSIC -  000223C02 DVMUC -  000253C02 DVMTPC -  021412C02
DVOPUC -  033343C07 DVTRIP -  000001C06 DWATCC -  000103C02 DYARDS -  000110C02
DYLNMI -  000005C10 FLOCAP -  033757C02 HACRAT -  021556C02 HLMDTR -  021482C02
HMCHAR -  021440C02 HOPDAT -  021630C02 HPCVCH -  021536C02 HPHRC -  024160C07
HVOPCS -  021654C02 HWTOPC -  021054C02 IDACCO -  000271C01 IPOLCD -  000241C01
IXV    -  000240C01 IXVT   -  001641    JSPRNT -  000000C01 JZ    -  001097
K      -  001042    LV     -  001042    LVY    -  001043    MM     -  001001
MMINDX -  000008C11 NIM    -  000220C01 NLM    -  000214C01 NLPM   -  000004C04
NPRHRS -  021294C02 NSM    -  000217C01 NTM    -  000221C01 NTSM   -  000222C01
NTW    -  000293C01 NVW    -  000276C01 NVM    -  000215C01 NVC    -  000276C01
PANCRS -  000227C01 POLRAT -  021544C02 PRIVCC -  001985
PUMCC  -  001142    RESDEN -  001354C02 RSAVDT -  000000C03 SCALEF -  000000C02
SMRYDT -  000319C03 TAXRAT -  000270C02 TOLNMI -  000000C10 TOVHAC -  001054
TRCKNO -  001093    TWCKMD -  000550C07 TRLNMI -  000003C10 ULANDC -  000031C02
USTRUC -  000044C09 UTRANC -  000043C02 XACRAT -  021462C02 XDELAY -  021382C02
XNDATA -  001320C07 XUPCU  -  021442C02 XDPTCD -  022042C02 XPCVCH -  021308C02
XPTRIP -  021342C02 XPWRC  -  023410C07 XSYSCH -  021319C02 XVHTPC -  021406C02
Z      -  033277C02

START OF CONSTANTS
000744

START OF TEMPORARIES
```

```
RUN VERSION 2.3 --PSR LEVEL 29A--                                                          06/30/72

                SUBROUTINE SUMMARY                                                         SMRY  2
C                                                                                          SMRY  3
C*****************************************************************************************SMRY  4
C                                                                                          SMRY  5
C     THIS SUBROUTINE PRINTS THE SUMMARY OF MAJOR VARIABLES AT THE END                     SMRY  6
C     OF EACH LOOP-- PARAMETER SUMMARY AFTER VARIATION OF EACH PARAMETER                   SMRY  7
C     --AND PUNCHES SUMMARY CARDS AFTER N POINT VARIATION OF EACH VALUE                    SMRY  8
C     BEING VARIED                                                                         SMRY  9
C                                                                                          SMRY 10
C*****************************************************************************************SMRY 11
C                                                                                          SMRY 12
C          NLP  = COUNTER OF TIMES A VALUE IS VARIED  1 TO 5                               SMRY 13
C          NLPM = MAXIMUM TIMES A VALUE IS VARIED                                          SMRY 14
C          LPS  = COUNTER OF LOOPS THRU MODEL                                              SMRY 15
C          LPSM = TOTAL LOOPS THRU MODEL INCLUDING BEST ESTIMATE                           SMRY 16
C                                                                                          SMRY 17
C*****************************************************************************************SMRY 18
000007          COMMON/XYT/ ST1(80),ST2(80),ST3(80),ST4(80),ST5(80),ST6(80),               SMRY 19
              * ST7(80),ST8(80)                                                            SMRY 20
000007          COMMON/SVR/ MSAVDT(100,2), SMRYDT(320,5)                                   SMRY 21
000007          COMMON/LUPS/ NPT, LPS, NLP, IBC, NLPM(15), LPSM                            SMRY 22
000007          COMMON/PL/ KKM                                                             SMRY 23
000007          COMMON/RD1/ IPARDT(99)                                                     SMRY 24
000007          DATA ST1/10HNO. OF LIN,10HKS      ,10HNO. OF SFG,10HMENTS                  , SMRY 25
              *       10HNO. OF INT,10HERCHANGES ,10HNO. OF TFR,10HMINALS                  , SMRY 26
              *       10HNO. OF TER,10HM. SEGM.  ,10HTOTAL ROUT,10HE MILES                 , SMRY 27
              *       10HTOTAL LANE,10H MILES    ,10HTOTAL REQU,10HIRED ACRES              , SMRY 28
              *       10HNO. OF VEH,10HS.-TYPE1  ,10HVEH. CAPAC,10HITY TYPE1               , SMRY 29
              *       10HVEH. LO-FA,10HCT. TYPE 1,10HVEH. LN-CA,10HP. TYPE1                , SMRY 30
              *       10HNO. OF VEH,10HS.-TYPE2  ,10HVEH. CAPAC,10HITY-TYPE2               , SMRY 31
              *       10HVEH. LO-FA,10HCT TYPE2  ,10HVEH. LN-CA,10HP. TYPE2                , SMRY 32
              *       10HNO. OF YAR,10HDS        ,10HTHACK COST,10H  S/LN-MI               , SMRY 33
              *       10H1H CONS. U,10HC  S/LN-MI,10H2H-B CONS. ,10HUC  S/LN-MI            , SMRY 34
              *       10H4H-B CONS. ,10HUC S/LN-MI,10HEL. CONS. ,10HUC S/LN-MI             , SMRY 35
              *       10HVEH. COST ,10HTYPE1 S/VH,10HVEH. COST ,10HTYPE2 S/VH              , SMRY 36
              *       10HGUIDEWAY L,10HMANE MILES ,10HINTERCHANG,10HE LN MILES             , SMRY 37
              *       10HTERMINAL L,10HANE MILES ,10HWACCEL/DECE,10HL LN MILES             , SMRY 38
              *       10HYARDS LANE,10H MILES    ,10H    ,10H                              , SMRY 39
              *       10H     ,10H    ,10H    ,10H                                         , SMRY 40
              *       10H     ,10H    ,10H    ,10H                                         , SMRY 41
              *       10H     ,10H    ,10H    ,10H                                         , SMRY 42
              *       10H     ,10H    ,10H    ,10H                                         , SMRY 43
              *       10H     ,10H    ,10H    ,10H                                         , SMRY 44
000007          DATA ST2/10HTOTAL DM C,10HMAP. COST S,10HTOTAL DM A,10HCC      S/YR,       / SMRY 45
              *       10HTOTAL PUB,10H ACC  S/YR,10HTOTAL PRIV,10H ACC  S/YR,              , SMRY 46
              *       10HGUIDEWAY A,10HCC    S/YR,10HINTERCHANG,10HES ACC  S/YR,          , SMRY 47
              *       10HTERMINAL A,10HCC    S/YR,10HA-D LANES ,10HACC     S/YR,          , SMRY 48
              *       10HCONTROL CE,10HANTER ACC S/YR,10HYARDS      A,10HCC     S/YR,      , SMRY 49
              *       10HMAINT FACI,10HL ACC S/YR,10HWAY C-C  A,10HCC      S/YR,          , SMRY 50
              *       10HVEH. TYPE1,10H ACC  S/YR,10HVEH. TYPE2,10H ACC  S/YR,            , SMRY 51
              *       10HVEH. TYPE3,10H ACC  S/YR,10HTOTAL VEH,,10H ACC  S/YR,            , SMRY 52
              *       10H     ,10H    ,10H    ,10H                                         , SMRY 53
              *       10H     ,10H    ,10H    ,10H                                         , SMRY 54
              *       10H     ,10H    ,10H    ,10H                                         , SMRY 55
```

```
RUN VERSION 2.3 --PSR LEVEL 29A--                                   SUMMARY                    06/30/72

              *       10H     ,10H    ,10H    ,10H                                         , SMRY 56
              *       10HGUIDEWAY    ,10H   CAP.  S,10HINTERCHANG,10HES CAP.  S,           , SMRY 57
              *       10HTERMINALS ,10H    CAP.  S,10HA-D LANES ,10H   CAP.  S,            , SMRY 58
              *       10HCONTROL CE,10HANTER CAP.S,10HYARDS      ,10H     CAP.S,           , SMRY 59
              *       10HMAI-T FACI,10HL     CAP.S,10HWAY C-C    ,10H     CAP.S,           , SMRY 60
              *       10HVEH. TYPE1,10H      CAP.S,10HVEH. TYPE2,10H     CAP.S,           , SMRY 61
              *       10HVEH. TYPE3,10H      CAP.S,10HTOTAL VEH,,10H     CAP.S,           , SMRY 62
              *       10HTOTAL LAND,10H      CAP.S,10HSTRUCTURES,10H     CAP.S,           , SMRY 63
              *       10HTHACK     ,10H      CAP.S,10H           ,10H                     , SMRY 64
000007          DATA ST3/10HDW=V1  DAI,10HLY   PHT ,10HDW=V1  DAI,10HLY   A=PHT ,          / SMRY 65
              *       10HDW=V1  ANNU,10HAL   PHT ,10HDW=V1  DAI,10HLY   A=VHT ,            , SMRY 66
              *       10HDW=V1  ANNU,10HAL   VHT ,10HDW=V1  ANNU,10HAL   A=PHT ,           , SMRY 67
              *       10HDW=V2  DAI,10HLY    PHT ,10HDW=V2  DAI,10HLY   A=VHT ,            , SMRY 68
              *       10HDW=V2  ANNU,10HAL   PHT ,10HDW=V2  ANNU,10HAL   A=VHT ,           , SMRY 69
              *       10HDW=V2  DAI,10HLY    VHT ,10HDW=V2  DAI,10HLY   A=PHT ,            , SMRY 70
              *       10HDW=V2  ANNU,10HAL   PHT ,10HDW=V2  ANNU,10HAL   A=PHT ,           , SMRY 71
              *       10HDW=V2  ANNU,10HAL   VHT ,10HDW=V2  ANNU,10HAL   A=VHT ,           , SMRY 72
              *       10HDV1 DAILY ,10H PER TRIP ,10HDV1 DAILY ,10H VEH TRIP ,             , SMRY 73
              *       10HDV1 ANNUAL,10H PER TRIP ,10HDV1 ANNUAL,10H VEH TRIP ,             , SMRY 74
              *       10HDV2 DAILY ,10H PER TRIP ,10HDV2 DAILY ,10H VEH TRIP ,             , SMRY 75
              *       10HDV2 ANNUAL,10H PER TRIP ,10HDV2 ANNUAL,10H VEH TRIP ,             , SMRY 76
              *       10HDV1 AVE TR,10HIP LENGTH ,10HDV1 TRIP T,10HIME (MIN) ,             , SMRY 77
              *       10HDV2 AVE TR,10HIP LENGTH ,10HDV2 TRIP T,10HIME (MIN) ,             , SMRY 78
              *       10HDAILY TRAN,10HSIT   PHT ,10HDAILY TRAN,10HSIT   VHT ,            , SMRY 79
              *       10HDAILY TRAN,10HSIT A-PHT ,10HDAILY TRAN,10HSIT A-VHT ,            , SMRY 80
              *       10HANNUAL TRA,10HNSIT  PHT ,10HANNUAL TRA,10HNSIT  VHT ,            , SMRY 81
              *       10HANNUAL TRA,10HNSIT APHT ,10HANNUAL TRA,10HNSIT AVHT ,            , SMRY 82
              *       10HDAILY ASIT,10H PER TRIP ,10HANN. ASIT,10H PER TRIP ,             , SMRY 83
              *       10HASIT TRIP ,10HLENGTH    ,10HASIT TRIP ,10HTIME-MIN. /            , SMRY 84
000007          DATA ST4/10HAVE. FWY S,10HPEED PEAK ,10HAVE. FWY S,10HPEED OPK  ,          , SMRY 85
              *       10HAVE. SA  S,10HPEED PEAK ,10HAVE. SA  S,10HPEED OPK  ,             , SMRY 86
              *       10HAUTO DAILY,10H      PHT ,10HAUTO DAILY,10H      VHT ,            , SMRY 87
              *       10HAUTO DAILY,10H    A-PHT ,10HAUTO DAILY,10H    A-VHT ,            , SMRY 88
              *       10HAUTO ANNUA,10HL     PHT ,10HAUTO ANNUA,10HL     VHT ,            , SMRY 89
              *       10HAUTO ANNUA,10HL   A-PHT ,10HAUTO ANNUA,10HL   A-VHT ,            , SMRY 90
              *       10HDAILY  TRU,10HNCK   VHT ,10HANNUAL TRU,10HNCK    VHT ,           , SMRY 91
              *       10HHWY DAILY ,10H AUTO  VHT ,10HHWY DAILY ,10H TRUCK VHT ,          , SMRY 92
              *       10HHWY DAILY ,10H ASIT  VHT ,10HHWY DAILY ,10H DM   VHT ,           , SMRY 93
              *       10HTOTAL HWY ,10HDAILY  VHT ,10HHWY ANNUAL,10H AUTO  VHT ,          , SMRY 94
              *       10HHWY ANNUAL,10H TRUCK VHT ,10HHWY ANNUAL,10H ASIT  VHT ,          , SMRY 95
              *       10HHWY ANNUAL,10H DM   VHT ,10HTOTAL HWY ,10HANNUAL VHT ,           , SMRY 96
              *       10HAUTO DAILY,10H  PER TRIP ,10HAUTO ANNUA,10HL PER TRIP ,          , SMRY 97
              *       10HAVE AUTO T,10HRIP LENGTH ,10HAUTO TRIP ,10HTIME -MIN.,           , SMRY 98
              *       10HTOT. SYS. ,10HDAILY  PHT ,10HTOT. SYS. ,10HDAILY  VHT ,          , SMRY 99
              *       10HTOT. SYS. ,10HDAILY  PT.,10HTOT. SYS. ,10HANNUAL PHT ,           , SMRY 100
              *       10HTOT. SYS. ,10HANNUAL VHT,10HTOT. SYS. ,10HANNUAL PT.,            , SMRY 101
              *       10HSYSTEM  TR,10HIP LENGTH ,10HSYSTEM TRI,10HP TIME-MIN.            , SMRY 102
              *       10H     ,10H    ,10H    ,10H                                         , SMRY 103
              *       10H     ,10H    ,10H    ,10H                                         , SMRY 104
000007          DATA ST5/10HAUTO OPER.,10HCOST  S/YR,10HTRUCK OPER,10H.COST S/YR,          / SMRY 105
              *       10HTOT. VEH. ,10HOPER. COST,10HFWY WAY OP,10HER.  COST ,            , SMRY 106
              *       10HSA WAY OPE,10HR.  COST  ,10HTOT. WAY O,10HPER.  COST ,            , SMRY 107
              *       10HTOT. HWY O,10HPER. COST ,10HPCT. VEH. ,10HOF TOTAL  ,            , SMRY 108
              *       10HPCT. WAY O,10HF     TOTAL ,10H      ,10H                         , SMRY 109
              *       10H     ,10H    ,10H    ,10H                                         , SMRY 110
```

```
RUN VERSION 2.3 --PSR LEVEL 208--                          SUMMARY                    06/30/72

                 X          10H          ,10H                ,10HTS-1 WAY  ,10HOPER. COST, SMRY 111
                 X          1OHTS-1 VEH. ,10HOPER. COST,10HTS-1 MANPO,10HWER          , SMRY 112
                 X          1OHTS-1 STORA,10HGE          ,10HTS-1 MAINT,10HENANCE     , SMRY 113
                 X          1OHTS-1 TERMI,10HNALS        ,10HTS-1 PARKI,10HNG         , SMRY 114
                 X          1OHTOTAL TS-1,10HOPER. COST,10HTS-2 WAY  ,10HOPER. COST, SMRY 115
                 X          1OHTS-2 VEH. ,10HOPER. COST,10HTS-2 MANPO,10HWER          , SMRY 116
                 X          1OHTS-2 STORA,10HGE          ,10HTS-2 MAINT,10HENANCE     , SMRY 117
                 X          1OHTS-2 TERM,10HINALS        ,10HTS-2 PARK,10HING         , SMRY 118
                 X          1OHTOTAL TS-2,10HOPER. COST,10HTS-3 WAY  ,10HOPER. COST, SMRY 119
                 X          1OHTS-3 VEH. ,10HOPER. COST,10HTS-3 MANPO,10HWER          , SMRY 120
                 X          1OHTS-3 STOR,10HAGE          ,10HTS-3 MAINT,10HENANCE     , SMRY 121
                 X          1OHTS-3 TERM,10HINALS        ,10HTS-3 PARK,10HING         , SMRY 122
                 X          1OHTOTAL TS-3,10HOPER. COST,10HTOTAL TRAN,10HSIT OPER, , SMRY 123
                 X          1OHPCT. TS-1 ,10HOF TOTAL    ,10HPCT. TS-2 ,10HOF TOTAL  , SMRY 124
                 X          1OHPCT. TS-3 ,10HOF TOTAL    ,10H                        / SMRY 125
000087       DATA ST6/10HANNUAL DM ,10HOPER. COST,10HANN PUB-DM,10HOPER. COST, SMRY 126
                 X          1OHANN PRI-DM,10HOPER. COST,10HANN DM-NON,10HVEH OPER.$, SMRY 127
                 X          1OHANN PRI-DM,10HVEH OPER.$,10HANN DMV1  ,10HVEH OPER.$, SMRY 128
                 X          1OHANN DMV2  ,10HVEH OPER.$,10HANN DMV3  ,10HVEH OPER.$, SMRY 129
                 X          1OHTOTAL DM $,10HEH. OPER.$,10HANN DM TP,10HH. OPER. $, SMRY 130
                 X          1OHANN DM GDH,10HY OPER.  $,10HANN DM YAR,10HD  OPER. $, SMRY 131
                 X          1OHANN DM C-C,10HOPER. COST,10HANN DM MAI,10HNT OPER. $, SMRY 132
                 X          1OH          ,10H                ,10H                ,10H          , SMRY 133
                 X          1OH          ,10H                ,10H                ,10H          , SMRY 134
                 X          1OH          ,10H                ,10H                ,10H          , SMRY 135
                 X          1OH          ,10H                ,10H                ,10H          , SMRY 136
                 X          1OH          ,10H                ,10H                ,10H          , SMRY 137
                 X          1OH          ,10H                ,10HANN. EXPEN,10HD.  AT 0.0, SMRY 138
                 X          1OHANN. EXPEN,10HD.  AT 0.1,10HANN. EXPEN,10HD.  AT 0.3, SMRY 139
                 X          1OHANN. EXPEN,10HD.  AT 0.6,10HANN. EXPEN,10HD.  AT 1.0, SMRY 140
                 X          1OHPROFIT FLA,10HT-FARE 0.0,10HPROFIT FLA,10HT-FARE 0.1, SMRY 141
                 X          1OHPROFIT FLA,10HT-FARE 0.3,10HPROFIT FLA,10HT-FARE 0.6, SMRY 142
                 X          1OHPROFIT FLA,10HT-FARE 1.1,10HPROFIT FLA,10HT-RATE 0.0, SMRY 143
                 X          1OHPROFIT FLA,10HT-RATE 0.1,10HPROFIT FLA,10HT-RATE 0.3, SMRY 144
                 X          1OHPROFIT FLA,10HT-RATE 0.6,10HPROFIT FLA,10HT-RATE 1.0/ SMRY 145
000087       DATA ST7/10HDAILY XSIT,10HPOWER KWHR,10HDAILY XSIT,10HPOWER  LBS, SMRY 146
                 X          1OHDAILY XSIT,10HPOWER  BTU,10HDAILY MHY ,10HPOWER  LBS, SMRY 147
                 X          1OHDAILY MHY ,10HPOWER  BTU,10HDAILY DM  ,10HPOWER KWHR, SMRY 148
                 X          1OHDAILY DM  ,10HPOWER  LBS,10HDAILY DM  ,10HPOWER  BTU, SMRY 149
                 X          1OHDAILY DMV1,10HLBS. POL. ,10HDAILY DMV2,10HLBS. POL. , SMRY 150
                 X          1OHDAILY TS-1,10HLBS. POL. ,10HDAILY TS-2,10HLBS. POL. , SMRY 151
                 X          1OHDAILY TS-3,10HLBS. POL. ,10HDAILY AUTO,10HR LBS. POL. , SMRY 152
                 X          1OHDAILY TRUC,10HK LBS. POL.,10HANNUAL DMV,10H1 LBS. POL. , SMRY 153
                 X          1OHANNUAL DMV,10H2 LBS. POL.,10HANNUAL TS-,10H1 LBS. POL. , SMRY 154
                 X          1OHANNUAL TS-,10H2 LBS. POL.,10HANNUAL TS-,10H3 LBS. POL. , SMRY 155
                 X          1OHANNUAL AUT,10HO LBS. POL.,10HANNUAL TRU,10HCK LBS POL. , SMRY 156
                 X          1OHANNUAL AUT,10HO ACCIDENT,10HANNUAL AUT,10HO FATAL.  , SMRY 157
                 X          1OHANNUAL AUT,10HO INJURIES,10HANNUAL RSI,10HNT ACCIDENT, SMRY 158
                 X          1OHANNUAL RSI,10HNT FATAL.  ,10HANNUAL OVI,10H ACC. -OFF, SMRY 159
                 X          1OHANNUAL OVI,10H FATAL-OFF,10HANNUAL OVI,10H ACC. -ON , SMRY 160
                 X          1OHANNUAL OVI,10H FATAL.-ON,10HTOTAL OVI ,10HACCIDENTS , SMRY 161
                 X          1OHTOTAL OVI ,10HFATALITIES,10HANNUAL OVZ,10H ACC. -OFF, SMRY 162
                 X          1OHANNUAL OVZ,10H FATAL-OFF,10HANNUAL OVZ,10H ACC. -ON , SMRY 163
                 X          1OHANNUAL OVZ,10H FATAL.-ON,10HTOTAL OVZ ,10HACCIDENTS , SMRY 164
                 X          1OHTOTAL OVZ ,10HFATALITIES,10HANNU. DM ST,10HATION ACC./ SMRY 165
```

```
RUN VERSION 2.3 --PSR LEVEL 208--                          SUMMARY                    06/30/72

000002       DATA ST8/10HTOTAL TARA,10HBLE  ACHES,10HAVG. COST-,10HTAX. LAND , SMRY 166
                 X          1OHMKT. VALUE,10H-TAX. LAND,10HTAX  LOSS ,10H          , SMRY 167
                 X          1OH          ,10H          ,10H          ,10H          , SMRY 168
                 X          1OH          ,10H          ,10H          ,10H          , SMRY 169
                 X          1OH          ,10H          ,10H          ,10H          , SMRY 170
                 X          1OHANT. DISPL,10HACEM  AREA,10HNO. WHITE ,10HDWELLINGS , SMRY 171
                 X          1OHNO. BLACK ,10HDWELLINGS ,10HNO. OTHER ,10HDWELLINGS , SMRY 172
                 X          1OHNO. LOW-IN,10HC.DWELLING,10HNO. MED-IN,10HC.DWELLING, SMRY 173
                 X          1OHNO. HI-INC,10H. DWELLING,10HTOTAL NO. ,10HDWELLINGS , SMRY 174
                 X          1OH          ,10H          ,10HNOISE IMPA,10HCT AREA-DY, SMRY 175
                 X          1OHNO. WHITE ,10HDWELLINGS ,10HNO. BLACK ,10HDWELLINGS , SMRY 176
                 X          1OHNO. OTHER ,10HDWELLINGS ,10HNO. LOW-IN,10HC.DWELLING, SMRY 177
                 X          1OHNO. MED-IN,10HC.DWELLING,10HNO. HI-INC,10H.DWELLINGS, SMRY 178
                 X          1OHTOTAL NO. ,10HDWELLINGS ,10H          ,10H          , SMRY 179
                 X          1OH          ,10H          ,10H          ,10H          , SMRY 180
                 X          1OHNOISE IMPA,10HCT AREA-NT,10HNO. WHITE ,10HDWELLINGS , SMRY 181
                 X          1OHNO. BLACK ,10HDWELLINGS ,10HNO. OTHER ,10HDWELLINGS , SMRY 182
                 X          1OHNO. LOW-IN,10HC.DWELLING,10HNO. MED-IN,10HC.DWELLING, SMRY 183
                 X          1OHNO. HI-INC,10H. DWELLING,10HTOTAL NO. ,10HDWELLINGS , SMRY 184
                 X          1OH          ,10H          ,10HDEMAND SCA,10HLE FACTORS/ SMRY 185
                                                                                     SMRY 186
000087         IF(LPS .GT. 1)  GO TO 615                                            SMRY 187
000088         PRINT 610                                                            SMRY 188
000011     A10 FORMAT(1H1,3X,*  THIS IS THE SUMMARY OF THE WEST ENGINEERING ESTIMSMRY 189
                 ATE RUN*,///)                                                       SMRY 190
000011         WP = 1                                                               SMRY 191
000012         KL = 1                                                               SMRY 192
           C   READ NU. OF PARAMETERS BEING VARIED IN THE RUN AND NO. OF TIMES      SMRY 193
           C   EACH OF THEM IS VARIED. AND TOTAL NO. OF LOOPS THRU MODEL            SMRY 194
000017         READ 62, LPS*, (NC, P*(K),K=1,10C)                                   SMRY 195
000032     62  FORMAT(16I5)                                                         SMRY 196
           C   INITIALIZE NLP                                                       SMRY 197
000032         NLP = 0                                                              SMRY 198
000033         GO TO 622                                                            SMRY 199
000034     A19 PRINT 620                                                            SMRY 200
000040     A20 FORMAT(1H1,3X,*  THIS IS THE SUMMARY OF A PARAMETER VARIATION LOOPSMRY 201
                 * ,//)                                                              SMRY 202
000040         WF = NLP + 1                                                         SMRY 203
000042         NL = NLP + 1                                                         SMRY 204
000043     622 CONTINUE                                                             SMRY 205
000043         PRINT 625                                                            SMRY 206
000047     A25 FORMAT(1H ,* CAPITAL COSTS SECTION*,//)                              SMRY 207
000047         DO 630  K = 1,40                                                     SMRY 208
000051         KL = K +2                                                            SMRY 209
000057         KF = KL - 1                                                          SMRY 210
000054     A30 PRINT 635, K, (ST1(J),JWKF,KL), (SHRYDT(K,M),MWKF,ML)                SMRY 211
000104     A35 FORMAT(1H,I5,5X, 2A10,5F20.3)                                        SMRY 212
000104         PRINT 640                                                            SMRY 213
000110     A40 FORMAT(1H1,3/)                                                       SMRY 214
000110         PRINT 645                                                            SMRY 215
000114         DO 650  K = 41,80                                                    SMRY 216
000114         KL = (K-40) * 2                                                      SMRY 217
000120         KF = KL - 1                                                          SMRY 218
000127     A65 PRINT 635, K, (ST2(J),JWKF,KL), (SHRYDT(K,M),MWKF,ML)                SMRY 219
000157         PRINT 610                                                            SMRY 220
000158     610 FORMAT(1H1,3/,* DUAL MODE, HIGHWAY, AND TRANSIT NETWORK DATA *,//)SMRY 221
```

```
000150          DO 815  K = 81,120                                  SMRY 221
000160          KL =(K-80) * 2                                      SMRY 222
000162          KF =KL - 1                                          SMRY 223
000164     815 PRINT 835, K, (ST3(J),J=KF,KL), (SMRYDT(K,M),M=MF,ML) SMRY 224
000214          PRINT 810                                           SMRY 225
000220          DO 825 K = 121,160                                  SMRY 226
000222          KL =(K-120) *2                                      SMRY 227
000224          KF =KL-1                                            SMRY 228
000226     825 PRINT 835, K, (ST4(J),J=KF,KL), (SMRYDT(K,M),M=MF,ML) SMRY 229
000258          PRINT 640                                           SMRY 230
000262     640 FORMAT(1H1,3/,* OPERATING COSTS SECTION*,//)         SMRY 231
000262          M = MLP + 1                                         SMRY 232
000264          DO 845  K = 161,200                                 SMRY 233
000266          KL = (K-160) * 2                                    SMRY 234
000270          KF = KL - 1                                         SMRY 235
000272     845 PRINT 835, K, (ST5(J),J=KF,KL), (SMRYDT(K,M),M=MF,ML) SMRY 236
000322          PRINT 640                                           SMRY 237
000324          DO 835 K = 201,240                                  SMRY 238
000330          KL = (K - 200) * 2                                  SMRY 239
000332          KF = KL - 1                                         SMRY 240
000334     835 PRINT 835, K, (ST6(J),J=KF,KL), (SMRYDT(K,M),M=MF,ML) SMRY 241
000364          PRINT 650                                           SMRY 242
000370     850 FORMAT(1H1,3/,* INDIRECT BENEFITS AND COSTS*,//)     SMRY 243
000370          DO 855 K = 241,280                                  SMRY 244
000372          KL =(K - 240) * 2                                   SMRY 245
000374          KF = KL - 1                                         SMRY 246
000376     855 PRINT 835, K, (ST7(J),J=KF,KL), (SMRYDT(K,M),M=MF,ML) SMRY 247
000426          PRINT 650                                           SMRY 248
000432          DO 865  K = 281,320                                 SMRY 249
000434          KL = (K-280) * 2                                    SMRY 250
000436          KF = KL - 1                                         SMRY 251
000440     865 PRINT 835, K, (ST8(J),J=KF,KL), (SMRYDT(K,M),M=MF,ML) SMRY 252
000470          IF (MLPP .EQ. 7)  GO TO 725                         SMRY 253
000472          IF (LPS .EQ. 1)  GO TO 720                          SMRY 254
000474          IF(MLP .EQ. MLPM(IBC))  GO TO 670                   SMRY 255
000477          IF (LPS .EQ. LPSM)  GO TO 730                       SMRY 256
000500          GO TO 805                                           SMRY 257
000501     870 CONTINUE                                             SMRY 258
000502          MF = 1                                              SMRY 259
000504          ML = MLPM(IBC) + 1                                  SMRY 260
000506          MLP = 0                                             SMRY 261
000508          MLPP = 7                                            SMRY 262
C               PUNCH SUMMARY DATA CARDS                            SMRY 263
000507          IF (IPARDT(77) .EQ. 1)  GO TO 873                   SMRY 264
000510          WRITE(7,871) IBC                                    SMRY 265
000514     871 FORMAT(*VARIATION SET NUMBER*,I3)                    SMRY 266
000516          DO 940  K = 1,320                                   SMRY 267
000524          IF (SMRYDT(K,1) .GT. 1000.0)  GO TO 915             SMRY 268
000527          WRITE(7,910)  K, (SMRYDT(K,M),M=1,ML)               SMRY 269
000542     910 FORMAT(I5,5F15.3)                                    SMRY 270
000546          GO TO 940                                           SMRY 271
000548     915 WRITE(7,920)  K, (SMRYDT(K,M),M=1,ML)               SMRY 272
000552     920 FORMAT (I5,5F15.0)                                   SMRY 273
000552     940 CONTINUE                                             SMRY 274
000554     873 CONTINUE                                             SMRY 275
```



```
SUMMARY

SUBPROGRAM LENGTH
000774

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
02    = 000446   200   = 000732   810   = 000834   815   = 000834
820   = 000890   822   = 000843   825   = 000860   826   = 000871
871   = 000717   873   = 000564   875   = 000741   878   = 000574
879   = 000804   895   = 000885   710   = 000815   720   = 000807
915   = 000543   920   = 000727   940   = 000562

BLOCK NAMES AND LENGTHS
ZYT   = 001200   SVH   = 003410   LUPS  = 000824   PL    = 000001

VARIABLE ASSIGNMENTS
IBC   = 0000003C03 IPARDT = 000000C05 J     = 000770   K     = 000765
KF    = 000747   KKM   = 000000C04 KL    = 000766   L     = 000773
LPS   = 000001C03 LPSM  = 000023C03 M     = 000771   MF    = 000763
ML    = 000744   MLP   = 000002C03 MLPM  = 000000C03 MLPP  = 000772
SMRVDT = 000000CA2 SMRYDT = 000310C02 ST3   = 000000C01 ST2   = 000120C01
ST7   = 000740C01 ST4   = 000380C01 ST5   = 000580C01 ST6   = 000020C01
                  ST8   = 001680C01

START OF CONSTANTS
000632

START OF TEMPORARIES
000754

START OF INDIRECTS
000762

UNUSED COMPILER SPACE
128480
```



```
000564          KKM = ML                                            SMRY 276
000566          IF (IPARDT(78) .EQ. 1)  GO TO 678                   SMRY 277
000570          PRINT 200                                           SMRY 278
000573     200 FORMAT(1H1,3/,*  THE FOLLOWING GRAPHS HAVE BEEN PLOTTED*,//) SMRY 279
000577          CALL PHPLOT                                         SMRY 280
000574     678 CONTINUE                                             SMRY 281
000574          PRINT 875, MLPM(IBC)                                SMRY 282
000602     875 FORMAT(1H1,3/,*  THIS IS THE SUMMARY OF A PARAMETRIC RUN WITH A VA SMRY 283
                VARIABLE ASSUMING*,I3,* VALUES*,//)                 SMRY 284
000602          IBC = IBC + 1                                       SMRY 285
000604     879 CONTINUE                                             SMRY 286
000604          GO TO 822                                           SMRY 287
000606     895 CONTINUE                                             SMRY 288
000606          MLP = MLP + 1                                       SMRY 289
000607          GO TO 710                                           SMRY 290
000607     720 CONTINUE                                             SMRY 291
000607          MLP = 1                                             SMRY 292
000610          IBC = 1                                             SMRY 293
000611          GO TO 730                                           SMRY 294
000612     725 CONTINUE                                             SMRY 295
C               INITIALIZE SUMMARY DATA ARRAY FOR NEXT PARAMETER VARIATION SMRY 296
000612          DO 710  K = 1,320                                   SMRY 297
000614          DO 710  L = 2,5                                     SMRY 298
000616     710 SMRYDT(K,L) = 0.00                                   SMRY 299
000624          MLPP = 0                                            SMRY 300
000626          MLP = 1                                             SMRY 301
000627     730 CONTINUE                                             SMRY 302
000627          RETURN                                              SMRY 303
000630          END                                                 SMRY 304
```

```
                SUBROUTINE PRPLOT
000002          DIMENSION INUF(1024), XARRAY(7), YARRAY(7), XLABLE(2), YLABLE(2)
000002          DIMENSION ATITLE(4)
000002          DIMENSION STITLE(10,4)
000002          COMMON/PL/ XXM
000002          COMMON/SVR/ HSAVDT(100,2), SMAYDT(320,4)
000002          COMMON/STT/ STT(64)
          C     INITIALIZE X AND Y ARRAYS
000002          DO 20 J = 1,7
000004          XARRAY(J) = 0.0
000005       20 YARRAY(J) = 0.0
          C     READ IN ALTERNATE GRAPH TITLES
000010          DO 105 NTI=1,10
000011      105 READ 110,(ATITLE(NTI,NTIP),NTIP=1,4)
000024      110 FORMAT(4A10)
000025          CALL SETPLTS (4)
000027          CALL PLOT(4.0,0.,-3)
          C     READ CURVES THAT ARE TO BE PLOTTED
          C     INDEXX = INDEX NO. OF X VARIABLE
          C     INDEXY = INDEX NO. OF Y VARIABLE
          C     NPTS = NO. OF POINTS IN EACH CURVE
          C     NTIT = TITLE DESIRED FOR THIS GRAPH
000032       50 READ 100, INDEXX, INDEXY, NPTS, NTIT
000044      100 FORMAT(4I5)
000044          IF (INDEXX .EQ. 999)  GO TO 500
000046          DO 105 JVALUE = 1,NPTS
000050          XARRAY(JVALUE)=SMAYDT(INDEXX,JVALUE)
000054      105 YARRAY(JVALUE)=SMAYDT(INDEXY,JVALUE)
          C     THIS SECTION PUTS XARRAY AND YARRAY VALUES IN INCREASING SEQUENTIAL
000063          NE=NPTS+1
000064          NL=NPTS-1
000066          I=1
000067          NP=
000070        4 CONTINUE
000070          DO 10 KP=1,NPTS
000072          IF (XARRAY(I).LT.XARRAY(KP)) GO TO 10
000074          XARRAY(NE)=XARRAY(I)
000077          XARRAY(I)=XARRAY(KP)
000101          XARRAY(KP)=XARRAY(NE)
000103          YARRAY(NE)=YARRAY(I)
000105          YARRAY(I)=YARRAY(KP)
000107          YARRAY(KP)=YARRAY(NE)
000110       10 CONTINUE
000113          IF (I.EQ.NL) GO TO 15
000114          I=I+1
000116          NP=NP+1
000117          GO TO 5
000120       15 CONTINUE
          C     COMPUTE LOCATION OF X AND Y AXIS TITLES
000120          INDEXX = (INDEXX * 2) - 2
000123          INDEXY = (INDEXY * 2) - 2
000124          DO 130  J = 1,2
000126          JX = INDEXX + J
000130          XLABLE(J) = STT(JX)
```



```
PRPLOT

SUBPROGRAM LENGTH
002459

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
5      - 000070   10     - 000110   15     - 000120   50     - 000032
100    - 000247   120    - 000140   210    - 000241   220    - 000275
230    - 000302   310    - 000241   500    - 000233

BLOCK NAMES AND LENGTHS
PL     - 000001   SVM    - 003410   XYT    - 001600

VARIABLE ASSIGNMENTS
ATITLE - 002397   STITLE - 002353   I      - 002441   INUF   - 000331
INDEXX - 002432   INDEXY - 002433   INDXSX - 002444   INDXSY - 002445
ISYMM  - 002451   J      - 002427   JT     - 002456   JVALUE - 002436
JX     - 002444   JY     - 002447   K      - 002452   KP     - 002463
N      - 002442   NE     - 002437   NL     - 002440   NPTS   - 002434
NTI    - 002430   NTIP   - 002431   NTIT   - 002435   RSAVDT - 000000C02
SMAYDT - 000110C42 STT    - 000000C03 XARRAY - 002331   XLABLE - 002347
YARRAY - 002340   YLABLE - 002351

START OF CONSTANTS
000237

START OF TEMPORARIES
000305

START OF INPUT/OUTPUT
000313

UNUSED COMPUTER SPACE
127300
```



```
000132          JY = INDXSY + J
000133      130 YLABLE(J) = STT(JY)
          C     CALL PLOT STATEMENTS HERE
          C     COMPUTE TITLE OF THIS GRAPH
000137          DO 120 JT = 1,4
000140      120 ATITLE(JT)=ATITLE(NTIT,JT)
000147          ISYMM = 0
          C     PLOT FIRST CURVE ON A GRAPH
000150          CALL XYPLOT (XARRAY(1),YARRAY(1),NPTS,1,1, 0, 0, 5.5,0.5,
         *  XLABLE, 20, YLABLE, 20, ATITLE, 40)
          C     PRINT THE X-COORDINATE TITLE; THE Y-COORDINATE TITLE, THE X ARRAY,
          C     AND THE Y ARRAY OF THE GRAPHS TO BE PLOTTED
000164          PRINT 210
000172      210 FORMAT(1H ,3/,1H ,0 X-COORDINATE TITLE          Y-COORDINATE TITLE 0
         *  X ,AA,  0X-COORDINATES0,12X,0Y-COORDINATES0,/)
000177          PRINT 220, XLABLE(1), XLABLE(2), YLABLE(1), YLABLE(2), XARRAY(1),
         *  Y YARRAY(1)
000212      220 FORMAT(1H ,2A10,6X,2A10,F19.3,F25.3)
000212          DO 225  K = 2,NPTS
000214      225 PRINT 230, XARRAY(K), YARRAY(K)
000224      230 FORMAT(1H ,40X, 2F25.3)
          C     RE-INITIALIZE X AND Y ARRAYS
000224          DO 250  K = 1,7
000227          XARRAY(K) = 0.0
000230      250 YARRAY(K) = 0.0
000232          GO TO 50
000234      500 CONTINUE
000234          CALL ENDPLTS
000236          RETURN
000236          END
```

```
RUN VERSION 2.1 --PSH LEVEL 20A--                                                    06/30/72

                SUBROUTINE PARRUN
                COMMON/SVI/ ISAVDT(100,2)
                COMMON/SVR/ RSAVDT(100,2), SNRYDT(320,5)
                COMMON/RDI/ IPARDT(100)
                COMMON/RDR/ RPARDT(103,8)
                COMMON/LUPS/ NPI, LPS, NLPH, IOC, NLPH(15), LPSH
          C
          C*****************************************************************
          C
          C     THIS SECTION CHECKS FOR PARAMETRIC RUNS AND READS THE DATA IF IT
          C     IS A PARAMETRIC RUN
          C
          C     PERMITS VARYING UP TO 15 VARIABLES AT ONE TIME
          C
          C*****************************************************************
          C
          C
          C
          C     REPLACE SAVED DATA INTO PARDAT ARRAY FOR NEXT PARAMETRIC RUN AND
          C             NI = INDEX LOCATION OF VARIABLE IN IPARDT ARRAY (INTEGER)
          C             NR = INDEX LOCATION OF VARIABLE IN RPARDT ARRAY (REAL)
000007          DO 75  J = 1,100
000004          NI = ISAVDT(J,1)
000006          IF (NI .EQ. 0)  GO TO 65
000007          IPARDT(NI) = ISAVDT(J,2)
000010    65 CONTINUE
000011          NR = RSAVDT(J,1)
000013          IF (NR .EQ. 0)  GO TO 75
000014          RPARDT(NR) = RSAVDT(J,2)
000014    75 CONTINUE
          C
          C     INITIALIZE SAVDAT FOR THE NEXT RUN
000017          DO 80  J = 1,100
000021          DO 80  K = 1,2
000022          ISAVDT(J,K) = 0
000025    80 RSAVDT(J,K) = 0.0
          C
          C     READ PARAMETRIC DATA FOR NEXT LOOP THRU THE MODEL AND SAVE THE
          C     ORIGINAL GOOD DATA
          C             NPI = NO. OF INTEGER VALUES TO BE VARIED THIS LOOP
          C             NPR = NO. OF REAL VALUES TO BE VARIED THIS LOOP
          C             999 = NO MORE RUNS
          C     READ NUMBER OF INTEGER VALUES TO BE VARIED
000034          READ 84, NPI
000042    84 FORMAT(I5)
000043          IF (NPI .EQ. 0)  GO TO 90
000049          IF (NPI.EQ. 999)  GO TO 95
          C     PRINT HEADING FOR PARAMETRIC RUN
000058          PRINT 9000, NPI
000057   9000 FORMAT(1H -1W/,95X,*THIS IS A PARAMETRIC RUN WITH THE FOLLOWING*,
            * I4,* INTEGER INPUT VALUES BEING VARIED*3/)
000057          DO 90  K = 1,NPI
000064          READ 86, NI
000061    86 FORMAT(I10)
```

```
RUN VERSION 2.1 --PSH LEVEL 20A--                                    PARRUN      06/30/72

PARRUN

SUBPROGRAM LENGTH
000PA1

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
85    =  000010    75    =  000015    84    =  000166    86    =  000205
88    =  000207    90    =  000102    95    =  000161    150   =  000216
152   =  000220    155   =  000222    160   =  000236    170   =  000240
175   =  000242    190   =  000156    9000  =  000171    9010  =  000211

BLOCK NAMES AND LENGTHS
SVI   =  000310    SVR   =  003410    RDI   =  000307    RDR   =  034004

VARIABLE ASSIGNMENTS
IPARDT =  0000000C03 ISAVDT =  000000C01 J     =  000254    K     =  000257
NI     =  000255     NLPH   =  0000004C05 NPI   =  0000000C05 NPR   =  000260
NR     =  000256     RPARDT =  000000C04 RSAVDT =  000080C02 SNRYDT =  000310C02

START OF CONSTANTS
000164

START OF TEMPORARIES
000247

START OF INDIRECTS
000251

UNUSED COMPILER SPACE
12750A
```

```
RUN VERSION 2.1 --PSH LEVEL 20A--                                    PARRUN      06/30/72

000PA1          ISAVDT(K,1) = NI
000PA7          ISAVDT(K,2) = IPARDT(NI)
000PA8          READ 88, IPARDT(NI)
000PB2    88 FORMAT(I20)
000PB7          PRINT 9010, NI , IPARDT(NI)
000PC2   9010 FORMAT(5X,*IPARDT(*, I5, *) =*, I20,/)
000PC2    90 CONTINUE
          C     READ NUMBER OF REAL VALUES TO BE VARIED
000PC6          READ 150, NPR
000PC7   150 FORMAT(I5)
000PC7          IF (NPR .EQ. 0)  GO TO 190
          C     PRINT HEADING FOR PARAMETRIC RUN
000PC3          PRINT 152
000PD7   152 FORMAT(5/)
000PD7          PRINT 155, NPR
000PD6   155 FORMAT(1H -10/,25X,*THIS IS A PARAMETRIC RUN WITH THE FOLLOWING*,
            * I4,* REAL INPUT VALUES BEING VARIED*,3/)
000PD6          DO 190  K = 1,NPR
000PE7          READ 160, NR
000PE4   160 FORMAT(I10)
000PE4          RSAVDT(K,1) = NR
000PE7          RSAVDT(K,2) = RPARDT(NR)
000PE8          READ 170,RPARDT(NR)
000PE8   170 FORMAT(F20.4)
000PE8          PRINT 175, NR, RPARDT(NR)
000PF4   175 FORMAT(5X,*RPARDT(*, I4, *) =*,F20.3,/)
000PF4   190 CONTINUE
000PF1    95 CONTINUE
000PF1          RETURN
000PF2          END
```

OVERLAY 1.00
(DMCP)

This overlay contains the following subroutines:

```
RUN VERSION 2.7 --PSW LEVEL 28A--                                                    06/30/72

      OVERLAY(DMCP,1,0)
      PROGRAM DMCAPC                                                                 DMSCAPT2
C                                                                                    DMSCAPT3
C                                                                                    DMSCAPT4
C     DUAL MODE CAPITAL COST PROGRAM                                                 DMSCAPT5
C                                                                                    DMSCAPT6
C                                                                                    DMSCAPT7
C        MAJOR COMPUTATION SECTIONS                                                  DMSCAPT8
C              1.  GUIDEWAY POWER                                                    DMSCAPT9
C              2.  INTERCHANGES                                                      DMSCAPT10
C              3.  TERMINAL POWER                                                    DMSCAPT11
C              4.  TERMINAL ACCEL/DECEL LANES                                        DMSCAPT12
C              5.  CONTROL CENTER                                                    DMSCAPT13
C              6.  MAINTENANCE FACILITY                                              DMSCAPT14
C              7.  YARDS                                                             DMSCAPT15
C              8.  WAY COMMAND AND CONTROL                                           DMSCAPT16
C              9.  VEHICLES                                                          DMSCAPT17
C             10.  TAX LOSSES                                                        DMSCAPT18
C             11.  GRAND TOTALS                                                      DMSCAPT19
C                                                                                    DMSCAPT20
C                                                                                    DMSCAPT21
C                                                                                    DMSCAPT22
C                                                                                    DMSCAPT23
C                                                                                    DMSCAPT24
000001      INTEGER ULIFE    , DINT                                                  DMSCAPT25
000001      INTEGER PANCK3                                                           DMSCAPT26
000001      REAL  NPKWPS , NUMDA                                                     DMSCAPT27
000001      COMMON/VAR1/        JSPANT(8A)     , ULIFE(30)     , DINT(30)      ,     DMSCAPT28
     *        LFCUIF          , NWM          , NLM           , NSM           ,     DMSCAPT29
     *        NTM             , NTM          , NTSM          , NTM           ,     DMSCAPT30
     *        NUM             , IZ           , JK            , PARCK3(6)     ,     DMSCAPT31
     *        NN              , NB           , IPHINT        , IBV           ,     DMSCAPT32
     *        TMULC(14,2,3)   , IDACCD(3)    , MMV           , NDTM          ,     DMSCAPT33
     *        NVC(9)                                                              ,DMSCAPT34
000001      COMMON/WHW/         SCALEF(3A)     , DVMCAP(5)     , DVMLUF(5)     ,     DMSCAPT35
     * ULNPAR(5)       , ULANDC(10)     , UTRAKC        , USTRUC(4)     ,DMSCAPT36
     * UPWNTM(14)      , UCMCFN(11)     , UWAYCC(5)     , UYARDS(5,13)  ,DMSCAPT37
     * UMAINT(10)      , DVMSUC(4,6)    , DVMUC(4)      , DMUMWM(5)     ,DMSCAPT38
     * ASSUAT(6)       , TAXRAT(4)      , DNOISE(20,14,2), RESDEN(4,7)  ,DMSCAPT39
     * UPMPFL(225,4,4) , DGLAST(225,5)  , BACKON(225,3,3), DPKMRF(225,3) ,DMSCAPT40
     * UPKUMP(225,3)   ,                                                   ,DMSCAPT41
     * NPKWPS          , NUMDA          , DPKPCT        , DPTRIP(4,3)   ,DMSCAPT42
     * NPALTU(2,3)     , DVDLAY(2,3)    , DMLUF(2,3)    , ASYSCM(4,5)   ,DMSCAPT43
     * APKPCT          , APTRIP(4,4)    , AUELAY(2,4,2) , MLNDIR(2,2)   ,DMSCAPT44
     * AVMTPC(4)       , DVMTPC(3)      , AUTOCM(3)     , MMCHAR(4)     ,DMSCAPT45
     * UPPVCC(13,4,2)  , APCVCM(4,6)    , MPCVCM(2,6)   , POLRAT(3,3)   ,DMSCAPT46
     * UMSTAC          , MACRAT(2,5,2)  , XACMAT(4,4)   , DACRAT(2,3)   ,DMSCAPT47
     * AMPLAT(2,4)     , MVOPCS(2,6)    , MWYOPC(2,3)   , XOPCU(4,7,4)  ,DMSCAPT48
     * XOPIC(14,7,4)   ,DOFPFL(150,4,4),DOFLKC(150,3,3) ,RACROF(150,3,2),DMSCAPT49
     * DMUMMC          , DCCOPC         , OMOPC         , Z(4)          ,DMSCAPT50
     * DVMPUC(4,9,4)   , DTROMC(11,7)   , NGOPC(7)      , FLOCAP(5)     ,DMSCAPT51
     * UFAIL(4,2,2)                                                        ,DMSCAPT52
000001      COMMON/VIC1/  GROUP2(32,5,3)), GROUP3(32,5,1))                           DMSCAPT53
000001      COMMON/ACT/ ANNCS(4,10), ANNCST(10,4)                                    DMSCAPT54
                                                                                     DMSCAPT55
```

```
RUN VERSION 2.7 --PSW LEVEL 28A--                                                    06/30/72

000003      COMMON/SG/ DSTAT(400,22)                                                 DMSCAPT56
000003      COMMON /DV13/ ASUMR(14,4), DTRTMI, DTLNMI, DTYDA                         DMSCAPT57
000003      COMMON/SVA/ WSAVDT(180,2), SMRVDT(320,5)                                 DMSCAPT58
000003      COMMON/LNT/TOLNMI,DBLNMI,DILNMI,TRLNMI,DALNMI,DYLNMI                     DMSCAPT59
000003      COMMON/MM/MMINMI                                                         DMSCAPT60
000003      CALL GUIDWAY                                                             DMSCAPT61
000004      CALL INTRCHG                                                             DMSCAPT62
000005      CALL TERMNAL                                                             DMSCAPT63
000006      CALL ADLANES                                                             DMSCAPT64
000007      CALL OTHRCAP                                                             DMSCAPT65
000010      RETURN                                                                   DMSCAPT66
000012      END                                                                      DMSCAPT67
```

```
RUN VERSION 2.7 --PSW LEVEL 28A--                                                    06/30/72

DMCAPC

PROGRAM LENGTH INCLUDING I/O BUFFERS
000029

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

BLOCK NAMES AND LENGTHS
BH1   =  000147   BUW   =  034004   DC1   =  000780   DC7   =  000230
SG    =  021140   DV13  =  000127   SVA   =  003410   LNT   =  000006
MM    =  000001

VARIABLE ASSIGNMENTS
ANNCS  =  000008C04 ANNCST =  000050C04 ASSUAT =  000264C02 ASUMR  =  000080C04
AUTOCM =  021419C02 BACROF =  031470C02 BACMAW =  012575C02 DACRAT =  021622C02
DCMCFN =  000076C02 DFAUF  =  033764C02 DGLAST =  010430C02 DGOPC  =  033750C02
DGWTM  =  000056C02 DINT   =  000156C01 DLIFE  =  000120C01 DLNCAP =  000024C02
DMAINT =  000211C02 DMVLUF =  021327C02 DNOISE =  000274C02 DNUMWM =  000257C02
DOFLKC =  021742C02 DOFPFL =  022222C02 DONPFL =  001410C02 DPCVCM =  021434C02
DPOLYM =  021277C02 DVMDPF =  020011C02 DPKMRF =  010546C02 DPTRIP =  021257C02
DSTAT  =  000008C04 DTRDPC =  033633C02 DVDLAY =  021301C02 DVMCAP =  000012C02
DVMLDF =  000017C02 DVMSUC =  000223C02 DVMUC  =  000253C02 DVMTPC =  021412C02
DVUPUC =  033303C02 DWAYCC =  000103C02 DYARDS =  000110C02 FLOCAP =  032757C02
GROUP2 =  000008C03 GROUP3 =  003340C03 HACRAT =  021556C02 MLNDIR =  021402C02
MMCHAR =  021470C02 MUPDAT =  021630C02 MPCVCM =  021530C02 MVOPCS =  021640C02
MWYOPC =  021454C02 IUACCM =  000271C01 IPOLCD =  000241C01 JSPANT =  000080C01
NPKWPS =  021754C02 NUMDA  =  021255C02 NVC    =  000276C01 PARCK3 =  000227C01
POLRAT =  021544C02 RESDEN =  001354C02 RSAVDT =  000800C07 SCALEF =  000080C02
SMRVDT =  000310C07 TAXRAT =  000276C02 ULANDC =  000031C02 USTRUC =  000044C02
APCVCM =  021542C02 AUELAY =  021362C02 XOPCU  =  021662C02 XOPICD =  022642C02
APTRIP =  021362C02 ASYSCM =  021315C02 AVMTPC =  021440C02
Z      =  033277C02

START OF CONSTANTS
000014

START OF TEMPORARIES
000021

START OF INDIRECTS
000023

UNUSED COMPILER SPACE
127400
```

```
                SUBROUTINE GUIDWAY                                      GDWY  2
C                                                                       GDWY  3
C*********************************************************************GDWY  4
C                                                                       GDWY  5
C       MAIN OR GUIDEWAY PROPER SECTION                                 GDWY  6
C                                                                       GDWY  7
C*********************************************************************GDWY  8
C                                                                       GDWY  9
000002          INTEGER OLIFE     , DINT                                GDWY 10
000002          INTEGER  PACK3                                          GDWY 11
000002          REAL  NPKHRS , MIMDA                                    GDWY 12
000002          COMMON/SG/ DGSGST(400,27)                               GDWY 13
000002          DIMENSION DVMLOD(3), DONFLO(225,3,6), MCHTMI(5,6,12)    GDWY 14
000002          DIMENSION TUNIT(5),TCOST(5)                             GDWY 15
000002          DIMENSION AVB(5,6,12),AVB2(5,6,12),PRCT(32,5,11)        GDWY 16
000002          DIMENSION DSTAT(400,22)                                 GDWY 17
000002          COMMON/H9/ UNDIP0(5,6,12)                               GDWY 18
000002          COMMON/HD1/       JSPRNT(80)     , OLIFE(30)     , DINT(30)     ,  GDWY 19
     *    LECONE          , NVM           , NLM           , NSM           , NYM           ,  GDWY 20
     *    NJM             , NTM           , NTSM          , NYM           ,  GDWY 21
     *    NDM             , IZ            , JK            , PARCK3(6)     ,  GDWY 22
     *    MM              , NS            , IPRINT        , ISV           ,  GDWY 23
     *    IPOLCD(4,2,3)   , IDACCD(3)     , NMV           , NDTM          ,  GDWY 24
     *    NVC(9)                                                          GDWY 25
000007          COMMON/HDH/       SCALEF(10)     , DVMCAP(5)     , DVMLOF(5)     ,  GDWY 26
     * DLNCAP(5)       , DLANDC(10)    , UTRARC        , USTRUC(4)     ,  GDWY 27
     * DRWNTM(10)      , DCNCPM(11)    , DVATCC(5)     , DYARDS(5,13)  ,  GDWY 28
     * DMAINT(10)      , DVMSUC(4,6)   , DVMUC(4)      , DMUNVM(5)     ,  GDWY 29
     * AKSPAT(4)       , TARRAT(4)     , DMDISE(20,14,2), RESDEN(4,7)  ,  GDWY 30
     * DONPFL(225,4,4), DGLKST(225,5)  , BACKON(225,3,3), DPKHRF(225,3) ,  GDWY 31
     * DPKONF(225,3)   ,                                                 GDWY 32
     * NPKHRS          , NUMDA         , DPKPCT        , DPTRIP(4,3)   ,  GDWY 33
     * DPOLTO(2,3)     , DVDLAY(2,3)   , DMVLDF(2,3)   , XSYSCH(4,5)   ,  GDWY 34
     * XPKPC           , XPTRIP(4,4)   , XDELAY(2,4,2) , MLNDIM(2,2)   ,  GDWY 35
     * AVMTPC(4)       , DVMTPC(3)     , AUTOCH(11)    , MNCHAR(4)     ,  GDWY 36
     * DPCVCM(3,4,2)   , XPCVPM(4,6)   , MPCVCH(2,6)   , POLRAT(3,3)   ,  GDWY 37
     * DWSTAC          , MACRAT(2,5,2) , XACRAT(4,4)   , DACRAT(2,3)   ,  GDWY 38
     * MOPDAT(2,4)     , MVOPCS(2,6)   , MVYOPC(2,3)   , XOPCU(4,7,4)  ,  GDWY 39
     * XNPICU(4,7,4)   ,DOFPFL(150,4,4),DOFLRC(150,3,3) ,MACKOF(150,3,2),  GDWY 40
     * DVHNPC          , DCCOPC        , DHOPC         , Z(4)          ,  GDWY 41
     * DVOPUC(6,9,4)   , DTROPC(11,7)  , DGOPC(7)      , FLOCAP(5)     ,  GDWY 42
     * DFARE(4,2,2)                                                      GDWY 43
000007          COMMON/DC1/  GROUP2(32,4,11), GROUP3(32,5,11)           GDWY 44
000007          COMMON/DCS/TITLE1(16), TITLE2(16), ANCM(3), UNTNAM(5,9),  GDWY 45
     *TNAME(5,12)                                                       GDWY 46
000007          COMMON/UC7/ ANNCST(4,10), ANNCST1(4,14)                 GDWY 47
000007          COMMON /OV13/ AK,MR(14,6), DTATMI, DTLNMI, DTYDA        GDWY 48
000007          COMMON/LNT/TOLNMI,DELNMI,DILNMI,TRLNMI,DALNMI,DTLNMI    GDWY 49
000007          EQUIVALENCE (DSTAT(1), DGSGST(1))                       GDWY 50
C               INITIALIZE IZ FOR ANNUAL COST ROUTINES                  GDWY 51
000007          IZ = 1                                                  GDWY 52
C               INITIALIZE GROUP 2 AND 3                                GDWY 53
000003          DO 700 K=1,5                                            GDWY 54
000008          DO 700 L=1,11                                           GDWY 55
```



```
000229          IF(JSPRNT( 2) .EQ. 1) GO TO 62                          GDWY 111
000227          DO 125  NV = 1,NVM                                      GDWY 112
000231          WRITE 12A , NV                                          GDWY 113
000234          WRITE 124                                               GDWY 114
000242   124 FORMAT(1H ,0 LINK*,10X,*DAILY  PEAK  PEAK*,2(8X,*PEAK HOUR*,8X,  GDWY 115
     *0PEAK VIR,*,8X,*DPP, DIR,*)/1H ,*  NO.    PERSON FLOW  HOUR  DIR,*0GDWY 116
     *,3(*    PERSON FLOW*,3(8X,*VEH. FLOW*)/1H ,10X,*(PERS/DAY) FACTR  GDWY 117
     *,FACT*,3(6X,*(PERS,/HR,)*),3(5X,*(VEH./HR,)*)//)                  GDWY 118
                                                                        GDWY 119
000247          DO 125  NL = 1,NLM                                      GDWY 120
000244   125 WRITE 127 ,NL, DONPFL(NL,NV,3), DPKHRF(NL,NV), DPKDRF(NL,NV),  GDWY 121
     *(DONFLO(NL,NV,IDI), IDI = 1,6)                                    GDWY 122
000303   12A FORMAT(1H ,5HNDM ON GUIDEWAY DESIGN HOUR LINK FLOWS FOR VEHICLE TYGDWY 123
     *PE, I3///)                                                        GDWY 124
000303   127 FORMAT(1H ,15, F15.0,2F4.2, 6F15.0)                        GDWY 125
000303    62 CONTINUE                                                   GDWY 126
C                                                                       GDWY 127
C       COMPUTE NUMBER OF LANES REQUIRED PER LINK                       GDWY 128
C                                                                       GDWY 129
000303          DO 150  NL = 1,NLM                                      GDWY 130
000304          ITOVFL = 0                                              GDWY 131
000304          ITPFLO = 0                                              GDWY 132
000307          DO  120  NV = 1,NVM                                     GDWY 133
000310          ITPFLO = ITPFLO + DONPFL(NL,NV,3)                       GDWY 134
000314   120 ITOVFL = ITOVFL + DONFLO(NL,NV,6)                          GDWY 135
000325          IF (ITPFLO .EQ. 0) GO TO 148                            GDWY 136
C       COMPUTE NO. OF LANES IN PEAK DIRECTION                          GDWY 137
C       NO. OF LANES = (I/C)*D2/C2) ROUNDED UP                          GDWY 138
000324          PKPNLN = 0.                                             GDWY 139
000327          DO 130  NV = 1,NVM                                      GDWY 140
000331   130 PKPNLN = PKPNLN + DONFLO(NL,NV,5) / DLNCAP(NV)             GDWY 141
000342          LNPKPR = PKPNLN + 1.0                                   GDWY 142
000344          DGLKST(NL,3) = LNPKPR                                   GDWY 143
C       COMPUTE NO. OF LANES IN THE OPPOSITE DIRECTION                  GDWY 144
000347          IF (ITOVFL .EQ. 0) GO TO 145                            GDWY 145
000354          OPNPLN = 0.                                             GDWY 146
000351          DO 1.0  NVM 1,NVM                                       GDWY 147
000352   140 OPNPLN = OPNPLN + DONFLO(NL,NV,6) / DLNCAP(NV)             GDWY 148
000363          LNOPPR = OPNPLN + 1.0                                   GDWY 149
000366          GO TO 142                                               GDWY 150
000366   145 LNOPPR = 0                                                 GDWY 151
000367   142 CONTINUE                                                   GDWY 152
000367          DGLKST(NL,4) = LNOPDR                                   GDWY 153
000377          DGLKST(NL,5) = LNPKDR + LNOPDR                          GDWY 154
C       FOR SYSTEMS WITHOUT EMERGENCY LANES DONT CHECK CONDITION        GDWY 155
000374          IF (LECODE .EQ. 0)  GO TO 150                           GDWY 156
C       COMPUTE TOTAL NO. OF LANES REQUIRED INCLUDING EMERGENCY LANES   GDWY 157
C               MINIMUM LANES FOR ONE-WAY LINKS = 2                     GDWY 158
C               MINIMUM LANES FOR TWO-WAY LINKS = 3                     GDWY 159
000374          LANEST = LNPKDR + LNOPDR                                GDWY 160
000374          IF (ITOVFL .EQ. 0 .AND. LANEST .EQ. 1) GO TO 145        GDWY 161
000405          IF (ITOVFL .GT. 0 .AND. LANEST .LE. 2) DGLKST(NL,5) = 3.0  GDWY 162
000414          GO TO 150                                               GDWY 163
C       IF THERE IS NO FLOW THEN THERE ARE NO LANES                     GDWY 164
000417   148 DGLKST(NL,3) = 0.0                                         GDWY 165
000421          DGLKST(NL,4) = 0.0
```



```
                                                                        GDWY  56
000008          DO 700 M=1,32                                           GDWY  57
000007          DWNHP2(M,K,L)=0.0                                       GDWY  58
000014   700 DWNHP3(M,K,L)=0.0                                          GDWY  59
000024          REWIND 11                                               GDWY  60
C                                                                       GDWY  61
C       READ AND COMPUTE GUIDEWAY SEGMENT STATISTICS                    GDWY  62
C                                                                       GDWY  63
C       READ SEGMENT INPUT DATA FOR EACH SEGM NS, TOT. SEGM. = NSM      GDWY  64
000027          NSM = 0                                                 GDWY  65
000030   151 READ(11,152)NS,(DGSGST(NS,NSD),NSD=12,13),(DGSGST(NS,NSD),NSD=1,3)GDWY  66
     *,(DGSGST(NS,NSD),NSD=14,15),(DGSGST(NS,NSD),NSD=18,20)            GDWY  67
000074          NSM = NSM + 1                                           GDWY  68
000074          IF (NS .NE. 400)  GO TO 151                             GDWY  69
000100          NSM = NSM - 1                                           GDWY  70
000102          NOTE LAST CARD MUST HAVE A ZERO IN COL 5                GDWY  71
000103   152 FORMAT(I5,4F5.0,F10.2,F10.2,2F5.0)                         GDWY  72
C       COMPUTE DESIGN HOUR VEHICLE LOADINGS                            GDWY  73
000103          DO 110  NV = 1,NVM                                      GDWY  75
C       PRINT VEHICLE DATA 1                                            GDWY  74
000104   110 DVMLOD(NV) = DVMCAP(NV) * DVMLOF(NV)                       GDWY  76
000114          IF(JSPRNT( 1) .EQ. 1) GO TO 61                          GDWY  77
000112          WRITE 10A                                               GDWY  78
000114   10A FORMAT(1H1,14HVEHICLE DATA 1*///)                          GDWY  79
000114          WRITE 107                                               GDWY  80
000127   107 FORMAT(1H ,45X  OH   VEHICLE   VEHICLE     LANE   VEHICLE/1H ,  GDWY  81
     *45X VEH.  CAPACITY     LOAD  CAPACITY   LOADING/1H ,              GDWY  82
     *45XTNUIE  PERS/VEH     FACTOR    VEH/HR  PERS/VEH,//)              GDWY  83
000127          DO 103  NV = 1,NVM                                      GDWY  84
000124   103 WRITE 104, NV, DVMCAP(NV), DVMLOF(NV), DLNCAP(NV) ,DVMLOD(NV)  GDWY  85
000144   104 FORMAT (1H , I5,F10.0,F10.2,F10.0,F10.2)                   GDWY  86
000144    61 CONTINUE                                                   GDWY  87
000144          DO 112 NL = 1,NLM                                       GDWY  88
000144          DGLKST(NL,1) = DGLKST(NL,1) * SCALEF(2)                 GDWY  89
000148          DO 112 NV = 1,NVM                                       GDWY  90
000152          NIX = NV + 4                                            GDWY  91
000154   112 DONPFL(NL,NV,3) = DONPFL(NL,NV,3) * SCALEF(1) * SCALEF(NIX)GDWY  92
C       COMPUTE DESIGN HOUR PERSON AND VEHICLE FLOWS PER LINK AND VEH TYPEGDWY  93
000164          DO 120  NL = 1,NLM                                      GDWY  95
C       COMPUTE BIDIRECTIONAL DESIGN HOUR PERSON FLOWS                  GDWY  94
000167          DO 120  NV = 1,NVM                                      GDWY  96
000170          DONFLO(NL,NV,1) = DONPFL(NL,NV,3) * DPKHRF(NL,NV)       GDWY  97
000174          DONFLO(NL,NV,2) = DONFLO(NL,NV,1) * DPKDRF(NL,NV)       GDWY  98
C       COMPUTE PEAK DIRECTION DESIGN HOUR PERSON FLOWS                 GDWY  99
C       COMPUTE NON-PEAK DIRECTION DESIGN HOUR PERSON FLOWS             GDWY 100
000202          DONFLO(NL,NV,3) = DONFLO(NL,NV,1) - DONFLO(NL,NV,2)     GDWY 101
C       COMPUTE PEAK DIRECTION DESIGN HOUR VEHICLE FLOWS                GDWY 102
000205          DONFLO(NL,NV,5) = DONFLO(NL,NV,2) / DVMLOD(NV)          GDWY 103
C       COMPUTE NON-PEAK DIRECTION DESIGN HOUR VEHICLE FLOWS (LOADED)   GDWY 104
000210          DPFLLO          = DONFLO(NL,NV,3) / DVMLOD(NV)          GDWY 105
C       COMPUTE LOADED + EMPTY VEHICLE FLOWS IN NON-PEAK DIRECTION      GDWY 106
000212          DONFLO(NL,NV,6) = DPFLLO + BACKON(NL,NV,1) * DONFLO(NL,NV,5)GDWY 107
C       COMPUTE BI-DIRECTIONAL DESIGN HOUR VEHICLE FLOWS--INCLUDE EMPTIES GDWY 108
000214          DONFLO(NL,NV,6) = DONFLO(NL,NV,5) + DONFLO(NL,NV,6)     GDWY 109
000221   12A CONTINUE                                                   GDWY 110
C       PRINT DESIGN HOUR PERSON AND VEHICLE FLOWS PER LINK AND VEH TYPE
```

```
000429          DGLKST(NL,5) = 0.0                                      GDWY 166
000429          GO TO 150                                               GDWY 167
000429    149 DGLKST(NL,5) = 2.0                                        GDWY 168
000429    150 CONTINUE                                                  GDWY 169
000430          IF(JSPRNT( 3) .EQ. 1) GO TO 63                          GDWY 170
000432          PRINT 153                                               GDWY 171
000434    153 FORMAT(1H1,6X,9HANDW.11X, 8HUNIT*, / 1H ,8X,8HCOST*, 11X, 8HLAND*,  GDWY 172
            * /,1H , 6X, 8HCODE*, 11X, 8HCOST*,//)                       GDWY 173
000435          DO 155 LC = 1,10                                        GDWY 174
000437    155 PRINT 156, LC, ULANDC(LC)                                 GDWY 175
000459    156 FORMAT(1H , I10, F15.0)                                   GDWY 176
000459          PRINT 157, UTRARC                                       GDWY 177
00045A    157 FORMAT(7X, 1H , 8HTRACK COST = $*, F 10.0,3X, 8HPER LANE MILE*,$/)  GDWY 178
000454          PRINT 163                                               GDWY 179
000462    163 FORMAT(1H ,6X, 8HCONS*, 5X, 8HSTRUCTURAL*, /1H ,6X,8HTYPE*, 11X,  GDWY 180
            * 8HCOST*, / 1H , 4X, 8HCODE*, 8X, 8H($/LN/MI)*,//)           GDWY 181
000462          DO 163  K = 1,4                                         GDWY 182
000466    163 PRINT 165, K, USTRUC(K)                                   GDWY 183
00047E    165 FORMAT(1H , I10, F15.0)                                   GDWY 184
00047E          PRINT 169                                               GDWY 185
000501    169 FORMAT(7X, 1H ,4X, 8HNO. OF*, 7X, 8HGUIDEWAY*,/ 1H , 5X, 8HLANES*,  GDWY 186
            * 6X, 8HINTH(FT)*, //)                                        GDWY 187
000501          DO 148  LN5 = 1,16                                      GDWY 188
000503    148 PRINT 149, LN5, DGWDTH(LN5)                               GDWY 189
000514    149 FORMAT(1H , I10, F15.0)                                   GDWY 190
000514    43  CONTINUE                                                  GDWY 191
            C     COMPUTE SEGMENT STATISTICS                            GDWY 192
000514          DO 140  NS = 1,NSM                                      GDWY 193
            C     COMPUTE NO. OF LANES THIS SEGM.                       GDWY 194
00051A          NL = DGSGST(NS,12)                                      GDWY 195
00052A          DGSGST(NS,16) = DGLKST(NL,5)                            GDWY 196
            C     COMPUTE LANE MILES THIS SEGM.                         GDWY 197
000522          DGSGST(NS,5) = DGSGST(NS,16) * DGSGST(NS,15)            GDWY 198
            C     COMPUTE ROW WIDTH OF THIS SEGM. (FEET)                GDWY 199
00052A          LN5 = DGSGST(NS,16)                                     GDWY 200
000527          IF (LN5.EQ. 0) GO TO 159                                GDWY 201
00053A          DGSGST(NS,17) = DGWDTH(LN5)                             GDWY 202
000437          GO TO 41                                                GDWY 203
000532    159 DGSGST(NS,17) = 0.0                                       GDWY 204
000534    41  CONTINUE                                                  GDWY 205
            C     COMPUTE ROW AREA OF THIS SEGM. (ACRES)                GDWY 206
000534          DGSGST(NS,4) = DGSGST(NS,15) * DGSGST(NS,17) * (5280./43560.)  GDWY 207
            C     COMPUTE GUIDEWAY STRUCTURAL COST                      GDWY 208
000541          K = DGSGST(NS,2)                                        GDWY 209
000543          DGSGST(NS,6) = DSGST(NS,5) * USTRUC(K)                  GDWY 210
            C     COMPUTE LAND ACQUISITION COST                         GDWY 211
00054A          LC = DGSGST(NS,1)                                       GDWY 212
00055A          DGSGST(NS,7) = DGSGST(NS,4) * ULANDC(LC)                GDWY 213
            C     CHECK IF LAND IS TAXABLE                              GDWY 214
000552          IF (LC.GT.5) GO TO 162                                  GDWY 215
            C     ENTER ZEROS FOR TAXABLE ACRES AND MARKET VALUE        GDWY 216
00055A          DGSGST(NS,10) = 0.0                                     GDWY 217
000557          DGSGST(NS,11) = 0.0                                     GDWY 218
000560          GO TO 164                                               GDWY 219
000561    162 DGSGST(NS,10) = DGSGST(NS,4)                              GDWY 220
```



```
000563          DGSGST(NS,11) = DGSGST(NS,7)                            GDWY 221
000565    164 CONTINUE                                                  GDWY 222
            C     COMPUTE TRACK COST                                    GDWY 223
000565          DGSGST(NS,8) = DGSGST(NS,5) * UTRARC                    GDWY 224
            C     COMPUTE TOTAL COST OF GUIDEWAY PROPER                 GDWY 225
000570          DGSGST(NS,9) = DGSGST(NS,6) + DGSGST(NS,7) + DGSGST(NS,8)  GDWY 226
00057A    140 CONTINUE                                                  GDWY 227
            C COMPUTE ALL GUIDEWAY PROPER TOTALS                        GDWY 228
00057A          NST=NSM+1                                               GDWY 229
00060A          DO 17 NSO=4,11                                          GDWY 230
000A01    17  DGSGST(NST,NSO)=0.0                                       GDWY 231
00061A          DO 18 NS=1,NSM                                          GDWY 232
000611          DO 18 NSO=4,11                                          GDWY 233
000612    18  DGSGST(NST,NSO)=DGSGST(NST,NSO)+DGSGST(NS,NSO)            GDWY 234
000627          DGLN=DGSGST(NST,5)                                      GDWY 235
            C     PRINT GUIDEWAY SEGMENT STATISTICS                     GDWY 236
000631          IF(JSPRNT( 6) .EQ. 1) GO TO 66                          GDWY 237
000633          PRINT 166                                               GDWY 238
00063A    166 FORMAT(1H1,30HGW GUIDEWAY SEGMENT STATISTICS,///)         GDWY 239
00063A          PRINT 167                                               GDWY 240
000647    167 FORMAT(1H ,4HSEGM. LINK SEGM GEOG CONS LAND LAND  SEGM.  NO.  ROW  GDWY 241
            * LANE  TRACK      STRUC.      LAND      TRACK      TOTAL    TAX.  GDWY 242
            * MARKET*/1H ,  4HINDEX  NO.  NO. AREA TYPE  USE COST LENGTH   OF W  GDWY 243
            6NTH   AREA LENGTH*,4(7X,4HCOST*),*   AREA       VALUE*/ 1H ,15X,  GDWY 244
            *4HCODE*,*   MILES*,* LANE FEET  ACRES LN,MI,*,4(8X,*($)*),*  ACRGDWY 245
            *ES       ($)*,//)                                            GDWY 246
000647          DO 168  NS = 1,NSM                                      GDWY 247
000644    168 PRINT 178, NS,(DGSGST(NS,NSO), NSO=12,13), (DGSGST(NS,NSO), NSO=1,GDWY 248
            *11),(DGSGST(NS,NSO), NSO=14,17), (DGSGST(NS,NSO), NSO=4,11)  GDWY 249
000714    178 FORMAT(1H,I5,6F5.0,F7.2,2F5.0,2F7.2,4F11.0,F7.2,F11.0)    GDWY 250
000714          PRINT 175,(DGSGST(NST,NSO),NSO=4,11)                    GDWY 251
000727    175 FORMAT(//,* TOTALS*,45X,2F7.2,4F11.0,F7.2,F11.0)          GDWY 252
000727    66  CONTINUE                                                  GDWY 253
            C                                                           GDWY 254
            C     COMPUTE ROUTE MILES PER LINK AND TOTAL ROUTE MILES    GDWY 255
000727          DO 176  NL= 1,NLM                                       GDWY 256
000731    176 DGLKST(NL,2) = 0.0                                        GDWY 257
000735          DTRTMI = 0.0                                            GDWY 258
000736          DO 178  NS= 1,NSM                                       GDWY 259
000737          NL = DGSGST(NS,12)                                      GDWY 260
000741          DGLKST(NL,2) = DGLKST(NL,2) + DGSGST(NS,15)             GDWY 261
000744    178 DTRTMI = DTRTMI + DGSGWT(NS,15)                           GDWY 262
            C                                                           GDWY 263
            C     PRINT GUIDEWAY LINK STATISTICS                        GDWY 264
000750          IF(JSPRNT( 5) .EQ. 1) GO TO 65                          GDWY 265
000752          PRINT 172                                               GDWY 266
000755    172 FORMAT(1H1,27HGW GUIDEWAY LINK STATISTICS,///)            GDWY 267
000755          PRINT 171                                               GDWY 268
000761    171 FORMAT(1H ,* LINK       LINK      ROUTE  NO.LANES  NO.LANES      TOTAGDWY 269
            *L* / 1H ,*  NO.     SPEED    LENGTH      PEAK  OPPOSITE     NO. OF*GDWY 270
            */ 1H ,10X,*(MPH)    (MILES) DIRECTION DIRECTION      LANES*,//)  GDWY 271
000761          DO 173  NL = 1,NLM                                      GDWY 272
000763    173 PRINT 174, NL,(DGLKST(NL,NLD), NLD=1,5)                   GDWY 273
001003    174 FORMAT(1H ,I5,2F10.2,3F10.0)                              GDWY 274
001003          PRINT 180, DTRTMI                                       GDWY 275
```



```
001010    180 FORMAT(//,1H ,18HTOTAL ROUTE MILES =,F6.2)                GDWY 276
001010    65  CONTINUE                                                  GDWY 277
            C     INITIALIZE GROUP9 ARRAY                               GDWY 278
001010          DO 182  M = 1,4                                         GDWY 279
001012          DO 182  K = 1,9                                         GDWY 280
001019          DO 182  L = 1,11                                        GDWY 281
001014          RCRTMI(M,K,L) = 0.0                                     GDWY 282
001021    182 GRNUP9(M,K,L) = 0.0                                       GDWY 283
            C                                                           GDWY 284
            C     COMPUTE ROUTE MILE GROUPING TABLE                     GDWY 285
001032          DO 140  NS = 1,NSM                                      GDWY 286
001039          M = DGSGST(NS,1)                                        GDWY 287
001036          K = DGSGST(NS,2)                                        GDWY 288
001037          L = DGSGST(NS,3)                                        GDWY 289
001041    184 GRNUP9(M,K,L) = GROUP9(M,K,L) +    DGSGST(NS,15)          GDWY 290
            C     SUM GROUP TABLE                                       GDWY 291
001051          CALL SUM(3,4,10,GROUP9)                                 GDWY 292
            C     COMPUTE PERCENTS OF SYSTEM TOTALS                     GDWY 293
001054          CALL PRCNT(3,4,10,GROUP9,PCRTM)                         GDWY 294
            C                                                           GDWY 295
            C     PRINT ROUTE MILE GROUPING TABLE                       GDWY 296
001064          IF(JSPRNT( 6) .EQ. 1) GO TO 66                          GDWY 297
001062          PRINT 190                                               GDWY 298
001064    190 FORMAT(1H1,19X,27HSYSTEM ROUTE MILE SUMMARIES,///)        GDWY 299
001064          PRINT 192                                               GDWY 300
001077    192 FORMAT(1H ,10X,*GEOG      CONS      LAND     ROUTE   PERCENT* /1H  GDWY 301
            *, 14X,*AREA      TYPE      USE    LENGTH    OF SYS* / 1H , 18X, GDWY 302
            * 7(4X,*CODE*),*    (MILES)      TOTAL*,//)                   GDWY 303
001077          ICNT = 0                                                GDWY 304
001079          DO 194  M = 1,4                                         GDWY 305
001074          DO 194  K = 1,9                                         GDWY 306
001074          DO 194  L = 1,11                                        GDWY 307
001077          PRINT 193, M,K,L, GROUP9(M,K,L), PCRTM(M,K,L)           GDWY 308
001124          ICNT = ICNT +1                                          GDWY 309
001124          IF (ICNT.EQ.44) GO TO 194                               GDWY 310
001127          GO TO 149                                               GDWY 311
001127    194 PRINT 197                                                 GDWY 312
001137          ICNT = 0                                                GDWY 313
001134    194 CONTINUE                                                  GDWY 314
001142    197 FORMAT(1H1,*/)                                            GDWY 315
001142    193 FORMAT(1H ,10X,9I10, F10.2,F10.0)                         GDWY 316
001142    44  CONTINUE                                                  GDWY 317
001142          CALL SUMT(NSM ,NSTAT,GROUP2,PRCT,AVG),AVG2)             GDWY 318
001144    919 FORMAT( 1X,4A1A,3I7)                                      GDWY 319
001144          TUNITS(1)=GROUP2(4,5,11)                                GDWY 320
001150          TUNITS(2)=GROUP2(8,5,11)                                GDWY 321
001151          TUNITS(3)=GROUP2(8,5,11)                                GDWY 322
001152          TUNITS(4)=1000000000000.                                GDWY 323
001154          TCOST(1)=GROUP2(4,5,11)                                 GDWY 324
001155          TCOST(2)=GROUP2(20,5,11)                                GDWY 325
001157          TCOST(3)=GROUP2(12,5,11)                                GDWY 326
001140          TCOST(4)=GROUP2(24,5,11)                                GDWY 327
001162          PRINT 918                                               GDWY 328
001164    918 FORMAT(1H1)                                               GDWY 329
001164          PRINT 919, TITLF1(1),TITLE1(2), ANCH                    GDWY 330
```

```
RUN VERSION 2.3 --PSH LEVEL 20A--                                  GUIDWAY          06/30/72

001177          CALL ANNUALU(1,I2,3,TUNITS,TCOST,DINT,DLIFE,UNTNAM(1,1),          GDWY 331
                XTNAME(1,1),ANNCST)                                                GDWY 332
001214          CALL DISPLAC                                                       GDWY 333
001215          CALL NOISE                                                         GDWY 334
001216          RETURN                                                             GDWY 335
001217          END                                                                GDWY 336
```

```
RUN VERSION 2.3 --PSH LEVEL 20A--                                  GUIDWAY          06/30/72

TITLE7 = 000070C0A TNAME  = 000111C0A TUNITS = 012353    ULAWDC = 000031C04
UNTNAM = 000074C0A USTRUC = 000044C04 UTRANC = 000043C04 AACRAT = 021602C04
XDELAY = 021342C04 XDPCU  = 021602C04 XDPICD = 022042C04 XPCVCH = 021500C04
XPTRIP = 021342C04 XSYSCH = 021315C04 XVHTPC = 021400C04 Z      = 033277C04

START OF CONSTANTS
001221

START OF TEMPORARIES
001624

START OF INTERFCTS
001637

UNUSED COMPILER SPACE
122300
```

```
RUN VERSION 2.3 --PSH LEVEL 20A--                                  GUIDWAY   06/30/72

GUIDWAY

SUBPROGRAM LENGTH
017777

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
17    = 000001   1A    = 000612   61    = 000144   62    = 000303
63    = 000516   66    = 000727   65    = 001010   96    = 001142
91    = 000934   104   = 001257   107   = 001234   108   = 001230
110   = 000104   124   = 001264   126   = 001314   127   = 001327
130   = 000331   140   = 000352   141   = 000366   142   = 000367
145   = 000423   148   = 000417   150   = 000425   151   = 000030
152   = 001273   153   = 001337   156   = 001352   157   = 001355
159   = 000537   161   = 001364   162   = 000561   164   = 000565
165   = 001400   166   = 001422   167   = 001430   169   = 001403
170   = 001474   171   = 001510   172   = 001510   174   = 001542
175   = 001582   176   = 000731   180   = 001546   189   = 001415
190   = 001554   192   = 001584   196   = 001618   197   = 001600
198   = 001127   198   = 001136   910   = 001620   919   = 001614

BLOCK NAMES AND LENGTHS
SA    = 001140   GV    = 000550   RDI   = 000387   RDW   = 034004
DC1   = 000700   DC5   = 000285   DC7   = 000230   OV13  = 000127
LNT   = 000004

VARIABLE ASSIGNMENTS
ANCH  = 000012C06 ANNCS  = 000008C07 ANNCST = 000058C07 ASSRAT = 000264C04
ASUMR = 000000C10 AUTOCH = 021415C04 AVD1   = 012345    AVD2   = 013135
BACKOF = 031476C04 BACKON = 012575C04 DACRAT = 021622C04 DCHCEN = 000070C04
DFARF = 033746C04 DULKST = 010430C04 DBLNMI = 000001C11 DGOPC  = 033750C04
DGSOST = 000000C01 DGWJTM = 000050C04 DINT   = 000156C03 DLIFE  = 000120C03
DLNCAR = 000024C04 DMAINT = 000211C04 DMVLDF = 021307C04 DNOISE = 000274C04
DNUMVH = 000057C04 DOFLKC = 026762C04 DOFPFL = 022222C04 DOWFLD = 001001
DOWPFL = 001410C04 DPCVCH = 021434C04 DPDLYO = 021273C04 DPKORF = 020011C04
DPRHRF = 010546C04 DPTRIP = 021257C04 DSTAT  = 000000C01 DTRPC  = 033633C04
DTRTMI = 000124C10 DVULAY = 021301C04 DVHCAP = 000012C04 DVHLDF = 000017C04
DVHLDP = 001456    DVHSUC = 000223C04 DVHUC  = 000253C04 DVHTPC = 021612C04
DVOPUF = 033343C04 DWAYCC = 000103C04 DYARDS = 000110C04 FLDCAP = 033757C04
GROUP2 = 000000C05 GROUP3 = 003340C05 GROUP4 = 000000C02 HACRAT = 021556C04
HLNMIR = 021402C04 HNCHAR = 021430C04 HOPUAT = 021030C04 HPCVCH = 021530C04
HVOPCS = 021644C04 HWTOPC = 021454C04 ICNT   = 017272    IDACCO = 000271C03
INX   = 017246    IPDLCD = 000241C03 ITOVFL = 017257    ITPFLD = 017260
I7    = 000275C03 JSMNT  = 000000C03 K      = 017245    L      = 017246
LANEST = 017245   LC     = 017266    LECODE = 000214C03 LHOPDR = 017264
LHPKDR = 017242   LHS    = 017267    M      = 017247    ML     = 017253
MLD   = 017271    MLM    = 000216C03 NPKHRS = 021254C04 MS     = 017256
MSD   = 017251    MSM    = 000217C03 NST    = 017270    NTX    = 017254
NUMDA = 021255C04 NV     = 017252    NVC    = 000276C03 NVM    = 000215C03
OPURLN = 017243   OPFLLD = 017255    PARCK3 = 000227C03 PCRTMI = 011603
PKORLN = 017241   PULRAT = 021544C04 PRCT   = 013765    RESDEN = 001354C04
SCALEF = 000000C04 TARRAT = 000278C04 TCOST  = 012360    TITLE1 = 000000C04
```

```
RUN VERSION 2.3 --PSK LEVEL 298--                                   06/30/72

      SUBROUTINE INTRCHG                                                  RDS   2
C                                                                         RDS   3
C*****************************************************************      RDS   4
C                                                                         RDS   5
C     BEGIN DW INTERCHANGE SECTION                                        RDS   6
C                                                                         RDS   7
C*****************************************************************      RDS   8
      INTEGER DLIFE   , DINT                                              RDS   9
      INTEGER  PANCK3                                                     RDS  10
      REAL  DPKHRS , NUMDA                                                RDS  11
      DIMENSION DSTAT(400,22)                                             RDS  12
      DIMENSION AVG1(5,8,12),AVG2(5,8,12),PMCT(32,5,1)                    RDS  13
      DIMENSION TUNITS(5),TCOST(5)                                        RDS  14
      COMMON/HIT/                JSPRNT(80)   , DLIFE(30)    , DINT(30)  ,RDS  15
     *  IFCODE          , NVH          , NLM          , NSM          ,   RDS  16
     *  NIM             , NTM          , NTSM         , NTM          ,   RDS  17
     *  NMP             , IZ           , JK           , PARCK3(6)    ,   RDS  18
     *  NH              , NS           , IPRINT       , IAV          ,   RDS  19
     *  IPULCD(4,2,3)   , IDACCD(3)    , NHV          , NOTH         ,   RDS  20
     *  NVC(9)                                                           RDS  21
      COMMON/HDR/                SCALEF(10)   , DVMCAP(5)    , DVMLDF(5) ,RDS  22
     * ULNCAM(5)        , ULANCC(10)   , UTRANC       , USTRUC(4)    ,RDS  23
     * DONPT=(14)       , DCNCFN(11)   , DWATCC(5)    , DYARDS(5,13) ,RDS  24
     * UMAINT(10)       , DVMSUC(4,6)  , DVMUCI(4)    , DNUMVM(5)    ,RDS  25
     * ARCRAT(4)        , TARRAT(4)    , DNOISE(20,14,2), RESDEN(4,7) ,RDS  26
     * DNNPFL(225,4,4)  , DOLRDT(225,4) , HACKON(225,3,3), DPKHRF(225,3) ,RDS  27
     * DPKHNP(225,3)    ,                                                RDS  28
     * NPRHRS           , NUMDA        , DPKPCT       , DPTRIP(4,3)  ,RDS  29
     * DPRNTO(2,3)      , DVOLAT(2,3)  , DMVLDF(2,3)  , ASYSCH(4,5)  ,RDS  30
     * APKPCF           , XPTRIP(4,4)  , AQELAT(2,4,2) , WLNOTR(2,2)  ,RDS  31
     * AWTPC(4)         , DWMTPC(3)    , AWTOCH(11)   , MNCHAR(6)    ,RDS  32
     * HPCVCH(14,6,2)   , HPCVCH(4,6)  , HPCVCH(2,6)  , POLRAT(3,3)  ,RDS  33
     * DWSTAC           , HACRAT(2,5,2) , AACRAT(4,4)  , DACRAT(2,3)  ,RDS  34
     * HOPRAT(2,4)      , MVOPCS(2,6)  , MVYOPC(2,3)  , XOPCU(4,7,4)  ,RDS  35
     * APRICU(4,7,4)    ,DOPPFL(150,4,4),DOFLKC(150,3,3) ,BACKOF(150,3,2),RDS  36
     * DMMNVC           , DCCOPC       , DWOPC        , Z(6)         ,RDS  37
     * DVOPUC(4,9,4)    , DTROPC(11,7) , DUOPC(7)     , FLOCAP(5)    ,RDS  38
     * DFAWL(4,2,2)                                                     RDS  39
      COMMON/HLV/  GROUP2(32,5,11), GROUP3(32,5,11)                       RDS  40
      COMMON/HCS/TITLF1(16), TITLE2(10), ANCH(2), UNTNAM(5,9),            RDS  41
     *TNAME(5,12)                                                         RDS  42
      COMMON/VC7/ ANNCS(4,10), ANNCST(8,14)                               RDS  43
      COMMON/SG/ DINTST(400,2P)                                           RDS  44
      COMMON/LNT/DLNM(,DBLNM(,DILNM(,14LNM(,DALNM(,DYLNM(                 RDS  45
      EQUIVALENCE (DSTAT(1), DINTST(1))                                   RDS  46
C     INPUT DUAL-MODE INTERCHANGE DATA AND COUNT NUMBER OF INTERCHANGES   RDS  47
C     (NIM)                                                               RDS  48
      NIM=0                                                               RDS  49
  201 READ(11,202)NI,DINTST(NI,15),(DINTST(NI,MID),MID=1,3),DINTST(NI,16,RDS  50
     *3), (DINTST(NI,M),MID=4,6)                                          RDS  51
      NIM=NIM+1                                                           RDS  52
      IF (NI .NE. 400)  GO TO 201                                         RDS  53
C     INTERCHANGES MUST BE NUMBERED SEQUENTIALLY BEGINNING AT 1 WITH NO   RDS  54
```

```
RUN VERSION 2.3 --PSK LEVEL 298--                                   06/30/72

                                                                          RDS 111
                TYMHICT.    P.O.D.      TRACK        TOTAL   TAXABLE    MARKET  PCT  RDS 112
                PT  WCE=/ IM . O  NO. DRAW AREA TYPE  USE COST      AREA     WAMPSRDS 113
                ro. 3I=X,*CDST*1, 8X,*CDST*, 8X,*AREA          VALUE  ROW TRCK STR.*,/  RDS 114
                 2IM . 8X,*TYPE*, 6I* CONF*1, *    (ACHES)  (LN,MI,I*, 3I7X,*(S)*),  RDS 115
                 E 9X, *(S)*, *    (ACRES)        (S)*, 3I* COST*),//)              RDS 116
0001A2          DO 266 NI=1,NIM                                           RDS 117
0001A4          WRITE 266,NI,(DINTST(NI,15),(DINTST(NI,MID),MID=1,3),     RDS 118
                (DINTST(NI ,M) ,(MID=NTST(NI,M),MID=4,6)                  RDS 119
0001F6    266 CONTINUE                                                    RDS 120
0001FA    266 FORMAT(1M ,I5,4F5.0,2F10.2,3F10.0,F12.0, F10.2,F10.0,3F5.0) RDS 121
0001FC        WRITE 266, (DINTST(NIT,MID),MID=1,6)                        RDS 122
0001J4    268 FORMAT(//,1M ,*MTOTALS,7X,2F10.2,3F10.0,F12.0, F10.2,F10.0,3F5.0)RDS 123
0001J4     67 CONTINUE                                                    RDS 124
0001J4        TCOST(1)=DINTST(NIT,7)                                      RDS 125
0001J6        TCOST(2)=DINTST(NIT,8)                                      RDS 126
0001J7        TCOST(3)=DINTST(NIT,6)                                      RDS 127
0001J8        TCOST(4)=DINTST(NIT,9)                                      RDS 128
0001J9        WRITE *(9)                                                  RDS 129
0001J9    91A FORMAT(1(-1,5/)                                             RDS 130
0001J9        WRITE *(9), TITLF2(1),TITLE2(2), ANCH                       RDS 131
0001J9    919 FORMAT(1X,4A10,*(/))                                        RDS 132
0001J9        CALL ANNIAL2(1,1,2,3,TCOST,DINT,DLIFE,TNAME(1,2),ANNCS)     RDS 133
0001J9        CALL SUMT(NIM ,DSTAT,GROUP2,PMCT,AVG1,AVG2)                 RDS 134
0001J9        IF(JSPRNT(1) .EQ. 1) GO TO 7A                               RDS 135
0001J9        CALL PWRTAR(GROUP2,PMCT,AVG1,AVG2)                          RDS 136
0001J9     7A CONTINUE                                                    RDS 137
C                                                                         RDS 138
C       SAVE GROUP2 IN GROUP3                                             RDS 139
C                                                                         RDS 140
0001J9        DO 749 M=1,5                                                RDS 141
0001J9        DO 749 NI=1,11                                              RDS 142
0001J9        DO 749 MI=1,32                                              RDS 143
0001J9    749 GROUP3(MI,M,NI)=GROUP2(MI,M,NI)                             RDS 144
0001J9        RETURN                                                      RDS 145
0001J9        END
```

```
RUN VERSION 2.3 --PSK LEVEL 298--                                   06/30/72

C     BAR, HOWEVER THE INPUT CARDS NEED NOT BE IN ORDER. LAST CARD MUST   RDS  56
C     HAVE A 9 IN COLUMN 5.                                               RDS  57
0000A1        NIM=NIM+1                                                   RDS  58
0000A2    202 FORMAT(I5,5F5.0,2F10.2,F10.0)                               RDS  59
C     COMPUTE DUAL MODE INTERCHANGE STATISTICS . AREAS, RAMPMILES,        RDS  60
C     AND STRUCTURAL COSTS ARE THOSE IN ADDITION TO BASE COSTS OF MAVING  RDS  61
C     MAIN LINES INTERSECTING AT GRADE (I.E. BASE DATA ARE INCLUDED IN    RDS  62
C     QUITEWAY PROPER CALCULATIONS)                                       RDS  63
0000A2        DO 210 NI=1,NIM                                             RDS  64
C     COMPUTE ADDITIONAL ROW COST                                         RDS  65
0000A3        LCM=DINTST(NI,1A)                                           RDS  66
0000A5        DINTST(NI,7)=DINTST(NI,4)*DLANCC(LC)                        RDS  67
C     COMPUTE ADDITIONAL COST OF TRACK FOR RAMPS                          RDS  68
0000A5        DINTST(NI,8)=DINTST(NI,5)*UTRANC                            RDS  69
C     COMPUTE TOTAL ADDITIONAL INTERCHANGE COST ABOVE BASE COST           RDS  70
0000A5        DINTST(NI,9)=DINTST(NI,6)+DINTST(NI,7)+DINTST(NI,8)         RDS  71
C     COMPUTE PERCENT ROW COST IS OF TOTAL INTERCHANGE COST               RDS  72
0000A5        DINTST(NI,12)=(DINTST(NI,7)/DINTST(NI,9))*100.              RDS  73
C     COMPUTE PERCENT TRACK COST IS OF TOTAL INTERCHANGE COST             RDS  74
0000A6        DINTST(NI,13)=(DINTST(NI,8)/DINTST(NI,9))*100.              RDS  75
C     COMPUTE PERCENT STRUCTURAL COST IS OF TOTAL INTERCHANGE COST.       RDS  76
0000A6        DINTST(NI,14)=(DINTST(NI,6)/DINTST(NI,9))*100.              RDS  77
C     DETERMINE IF LAND ACQUIRED WAS TAXABLE (LC=6 THRU 10 IS TAXABLE     RDS  78
C     LAND) AND DETERMINE AREA AND MARKET VALUE OF THISLAND.              RDS  79
0000A7        IF(LC.GT.5)GO TO 204                                        RDS  80
0000A9        DINTST(NI,10)=0.0                                           RDS  81
0000AA        DINTST(NI,11)=0.0                                           RDS  82
0000AB        GO TO 210                                                   RDS  83
0000AC    204 DINTST(NI,10)=DINTST(NI,4)                                  RDS  84
0000B0        DINTST(NI,11)=DINTST(NI,7)                                  RDS  85
0000B2    210 CONTINUE                                                    RDS  86
C     COMPUTE ALL INTERCHANGE TOTALS                                      RDS  87
0000B5        NITM=NIM+1                                                  RDS  88
0000B7        DO 220 MID=4,11                                             RDS  89
0000B8    220 DINTST(NITM,MID)=0.0                                        RDS  90
0000B7        DO 230 NI=1,NIM                                             RDS  91
0000B8        DO 230 MID=4,11                                             RDS  92
0000B9        DINTST(NITM,MID)=DINTST(NITM,MID)+DINTST(NI,MID)            RDS  93
0000BA    230 CONTINUE                                                    RDS  94
C     COMPUTE PERCENT ALL INTER CHANGES ROW COST IS OF ALL INTERCHANGES   RDS  95
C     COST                                                                RDS  96
0000BB        DINTST(NITM,12)=(DINTST(NITM,7)/DINTST(NITM,9))*100.        RDS  97
C     COMPUTE PERCENT ALL INTERCHANGES TRACK COST IS OF ALL INTERCHANGES  RDS  98
C     COST                                                                RDS  99
0000C1        DINTST(NITM,13) = (DINTST(NITM,8)/DINTST(NITM,9)) * 100.    RDS 100
C     COMPUTE PERCENT ALL INTERCHANGES COST (STRUCTURAL) IS OF ALL        RDS 101
C     INTERCHANGES COST                                                   RDS 102
0000C4        DINTST(NITM,14)=(DINTST(NITM,6)/DINTST(NITM,9))*100.        RDS 103
0000C7        DLNM=DINTST(NITM,5)                                         RDS 104
C     PRINT INTERCHANGE STASTICS                                          RDS 105
0000C9        IF(JSPRNT( 7) .EQ. 1) GO TO 67                              RDS 106
0000C9        WRITE 266                                                   RDS 107
0000CA    266 FORMAT(1H1,35X   DUAL MODE INTERCHANGE STATISTICS,///)      RDS 108
0000CA        WRITE 262                                                   RDS 109
0000CD    262 FORMAT(1H ,*  INT  INT AREA CONS LAND LAND      ADDN.    LENGTH  RDS 110
```

```
RUN VERSION 2.3 --PSK LEVEL 298--                                   06/30/72

INTRCHG

SUBPROGRAM LENGTH
005409

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
67    =  000213    7A    =  000301    266   =  000002    201   =  000003
262   =  000331    294   =  000076    210   =  000102    220   =  000110
240   =  000337    242   =  000345    246   =  000414    24A   =  000423
749   =  000305    91A   =  000433    919   =  000436

BLOCK NAMES AND LENGTHS
RN1   =  000347    RIM   =  034804    DC1   =  000708    DC5   =  000205
DC7   =  000270    SG    =  021140    LNT   =  000006

VARIABLE ASSIGNMENTS
ANCH  =  000072C04 ANNCS  =  000000C05 ANNCST =  000050C05 ASSRAT =  000264C02
AUTOCH =  021415C02 AVG1  =  000501    AVG2  =  001251    BACKOF =  031470C02
BACKON =  012575C02 UACRAT =  021828C02 DCNCFN =  000070C02 DFARE  =  033764C02
DGLKST =  010439C02 DGOPC  =  033750C02 DRWOTH =  000050C02 DILNM1 =  000002C07
DINT   =  000154C01 DINTST =  000000C04 DLIFE  =  000120C01 DLNCAP =  000024C02
DMAINT =  000211C02 DMVLDF =  013B7C02 DNOISE =  000274C02 DNUMVM =  000257C02
DOFLKC =  028742C02 DOPPFL =  022222C02 DONPFL =  001410C02 DPCVCH =  021434C02
DPKLYD =  021273C02 DPKDRF =  020011C02 DPKHRF =  010540C02 DPTRIP =  021257C02
DSTAT  =  000000C04 DTROPC =  033633C02 DVOLAT =  021301C02 DVMCAP =  000012C02
DVMLDF =  000017C02 DVMSUC =  000223C02 DVMUCI =  000253C02 DWMTPC =  021412C02
DVOPUC =  033303C02 DWATCC =  000103C02 DYARDS =  000110C02 FLOCAP =  033757C02
GROUP2 =  000000C03 GROUP3 =  003340C03 HACRAT =  021550C02 HLNDTR =  021402C02
HNCHAR =  021470C02 HOPDAT =  021630C02 HPCVCH =  021530C02 HVOPCS =  021440C02
HWYOPC =  021644C02 IUACCD =  000271C01 IPOLCD =  000241C01 IZ     =  000225C01
JSPRNT =  000000C01 K      =  005377    L      =  005400    LC     =  005375
M      =  005401    MI     =  005373    MID    =  005376    MIM    =  000228C01
MIT    =  005376    NPRHRS =  021254C02 NUMDA  =  021255C02 NVC    =  000276C01
PARCK3 =  000297C01 POLRAT =  021544C02 PRCT   =  002821    RESDEN =  001354C02
SCALEF =  000000C02 TARRAT =  000270C02 TCOST  =  003366    TITLE1 =  000000C04
TITLE2 =  000000C04 TNAME  =  000111C04 TUNITS =  005361    ULANDC =  000031C02
UNTNAM =  000076C04 USTRUC =  000044C02 UTRANC =  000043C02 XACRAT =  021402C02
XDELAY =  021342C02 XOPCU  =  021062C02 XOPICD =  022242C02 XPCVCH =  021540C02
XPTRIP =  021342C02 XSYSCH =  021315C02 XWMTPC =  021408C02 Z      =  033277C02

START OF CONSTANTS
000327

START OF TEMPORARIES
000443

START OF INDIRECTS
000454

UNUSED COMPILER SPACE
120000
```

```
RUN VERSION 2.3 --PSR LEVEL 20A--                                          06/30/72

      $SUBROUTINE TERMINAL                                                  TERM   2
C                                                                           TERM   3
C*****************************************************************************TERM   4
C                                                                           TERM   5
C     HEADING ON TERMINAL PROPER SECTION                                    TERM   6
C                                                                           TERM   7
C*****************************************************************************TERM   8
C                                                                           TERM   9
      INTEGER  DLIFE     , DINT                                             TERM  10
      INTEGER  PANCK3                                                       TERM  11
      REAL  .PCKR5 , NUMDA                                                  TERM  12
      DIMENSION AVB(5,6,12),AVA2(5,6,12),PRCT(32,5,11)                      TERM  13
      DIMENSION TUNIT(5),TCUST(4)                                           TERM  14
      DIMENSION DSTAT(400,22)                                               TERM  15
      COMMON/MOD1/      JSPRNT(80)     , DLIFE(30)     , DINT(30)     ,     TERM  16
     * IFCODE          , NVR           , NLM           , NSM          ,     TERM  17
     * NIM             , NTM           , NTSM          , NTM          ,     TERM  18
     * NMM             , IZ            , JK            , PARCK3(6)    ,     TERM  19
     * NN              , NX            , IPRINT        , IXV          ,     TERM  20
     * IPULC(14,2,3)   , DACCD(3)      , NHV           , NOTM         ,     TERM  21
     * NVC(4)                                                               TERM  22
      COMMON/NOR/       SCALEF(10)     , DVMCAP(5)     , DVMLOF(5)    ,TERM  23
     * DLNCAP(5)        , ULANDC(10)    , UTRANC        , USTRUC(4)    ,TERM  24
     * DANTM(16)        , DCNCEV(11)    , DWAYCC(5)     , DYARDS(9,13) ,TERM  25
     * DMAIN(11)        , DVMSUC(14,6)  , DVMUC(4)      , DMUNVM(5)    ,TERM  26
     * ASSCAT(4)        , TARRAT(4)     , DWDISE(20,14,2), RESDEN(4,7) ,TERM  27
     * UPMPFL(225,4,4), DOLKDT(225,5), HACKON(225,3,3), DPKMRF(225,3) ,TERM  28
     * UPKINP(225,3)   ,                                                    TERM  29
     * NPRMNS          , NUMDA         , DPKPCT        , DPTRIP(4,3)  ,TERM  30
     * UPRLTU(12,3)    , DVDLAY(2,3)   , DWYLOF(2,3)   , ESYSCH(4,5)  ,TERM  31
     * APRPCT          , APTRIP(4,4)   , XDELAY(2,4,2) , MLNDIR(2,2)  ,TERM  32
     * AVMTPC(4)       , DVMTPC(3)     , AUTOCH(11)    , HMCHAR(4)    ,TERM  33
     * UPPVLM(13,6,2)  , APCVCM(4,6)   , MPCVCM(2,6)   , POLRAT(3,3)  ,TERM  34
     * UMSTAC          , MACRAT(2,5,2) , AACRAT(4,4)   , DACRAT(2,3)  ,TERM  35
     * MAPIAT(2,4)     , MVOPCS(2,6)   , MMVOPC(2,3)   , KOPCU(4,7,4) ,TERM  36
     * ANPICU(4,7,4)   ,DOPPFL(150,4,4),DOFLKC(150,3,3) ,HACKOF(150,3,2),TERM  37
     * DVMMPC          , DCCOPC        , DMOPC         , Z(6)         ,TERM  38
     * DVPMUC(4,9,4)   , DTROMC(11,7)  , DGOPC(7)      , FLOCAP(5)    ,TERM  39
     * UFAM(4,2,2)                                                          TERM  40
      COMMON/ICI/ GROUP2(32,5,11), GROUP3(32,5,11)                          TERM  41
      COMMON/DCS/TITLE1(16), TITLE2(10), ANCM(2), UNTNAM(5,9),              TERM  42
     *TNAME(5,12)                                                           TERM  43
      COMMON/DCT/ ANNCS(4,18), ANNCST(8,14)                                 TERM  44
      COMMON/SGF UTERST(400,22)                                             TERM  45
      COMMON/LNT/TULNM1,DOLNM1,DTLNM1,THLNM1,DALNM1,DYLNM1                  TERM  46
      EQUIVALENCE (DSTAT(1), DTERST(1))                                     TERM  47
C     INITIALIZE GROUP 2                                                    TERM  48
      DO 740 NI=1,5                                                         TERM  49
      DO 740 LC=1,11                                                        TERM  50
      DO 740 NI=1,42                                                        TERM  51
  740 GROUP2(NI,LC,NI)=0.0                                                  TERM  52
C     READ IN TERMINAL PROPER DATA AND COUNT NUMBER OF TERMINALS            TERM  53
      NTM = 0                                                               TERM  54
  741 READ(11,301)NT,DTERST(NT,12), (DTERST(NT+NTD),NTD=1,3), (DTERST(NTTERM  55
```

```
RUN VERSION 2.3 --PSR LEVEL 20A--                                          06/30/72

     *,NTD),NTD=13,20)                                                      TERM  56
      NTM = NTM + 1                                                         TERM  57
      IF (NT .NE. 400)  GO TO 741                                           TERM  58
      NTM = NTM + 1                                                         TERM  59
C     NOTE LAST CARD MUST HAVE A ZERO IN COL 5                              TERM  60
  302 FORMAT(I5,5F5.0,F8.0,3F6.2,3F4.0)                                     TERM  61
C     APPLY SCALE FACTORS                                                   TERM  62
      DO 745  NT = 1,NTM                                                    TERM  63
      DTERST(NT,14) = DTERST(NT,14) * SCALEF(10) * SCALEF(9)                TERM  64
      DTERST(NT,15) = DTERST(NT,15) * SCALEF(10) * SCALEF(9)                TERM  65
      DTERST(NT,18) = DTERST(NT,18) * SCALEF(10) * SCALEF(9)                TERM  66
      DTERST(NT,20) = DTERST(NT,20) * SCALEF(10) * SCALEF(9)                TERM  67
  745 CONTINUE                                                              TERM  68
C                                                                           TERM  69
C     COMPUTE ON TERMINAL PROPER STATISTICS                                 TERM  70
C                                                                           TERM  71
      DO 750  NT = 1,NTM                                                    TERM  72
C     COMPUTE TERMINAL PROPER AREAS AND TRACK MILES FROM EQUATIONS BASEDTERM  73
C     ON TERMINAL DESIGN TYPE (JTD)                                         TERM  74
      JTD = DTERST(NT,12)                                                   TERM  75
      GO TO (310,312,314,316,318,320,322,324,326,328,329), JTD               TERM  76
C     DESIGN TYPE 1                                                         TERM  77
  310 DTERST(NT,21) = (1.1 * (68750. + (20.8*DTERST(NT,14)))) / 43560.      TERM  78
      DTERST(NT,22) = (2395. + (2.2R * DTERST(NT,14))) / 5280.              TERM  79
      GO TO 330                                                             TERM  80
C     DESIGN TYPE 2                                                         TERM  81
  312 DTERST(NT,21) = (0.505 *(68750.+ (20.8*DTERST(NT,14)))) / 43560.      TERM  82
      DTERST(NT,22) = (2395. + (2.2R * DTERST(NT,14))) / 5280.              TERM  83
      GO TO 330                                                             TERM  84
C     DESIGN TYPE 3                                                         TERM  85
  314 DTERST(NT,21) = (1.1 * (22500. + (20.8*DTERST(NT,14)))) / 43560.      TERM  86
      DTERST(NT,22) = (1530. + (2.2R * DTERST(NT,14))) / 5280.              TERM  87
      GO TO 330                                                             TERM  88
C     DESIGN TYPE 4                                                         TERM  89
  316 DTERST(NT,21) = (0.505 *(22500. + (20.8*DTERST(NT,14)))) / 43560.     TERM  90
      DTERST(NT,22) = (1530. + (2.2R * DTERST(NT,14))) / 5280.              TERM  91
      GO TO 330                                                             TERM  92
C     DESIGN TYPE 5                                                         TERM  93
  318 DTERST(NT,21) = (1.1 * (68625. + (24.3*DTERST(NT,14)))) / 43560.      TERM  94
      DTERST(NT,22) = (1945. + (0.6 * DTERST(NT,14))) / 5280.               TERM  95
      GO TO 330                                                             TERM  96
C     DESIGN TYPE 6                                                         TERM  97
  320 DTERST(NT,21) = (0.505 *(68625. +(24.3*DTERST(NT,14)))) / 43560.      TERM  98
      DTERST(NT,22) = (1945. + (0.6 * DTERST(NT,14))) / 5280.               TERM  99
      GO TO 330                                                             TERM 100
C     DESIGN TYPE 7                                                         TERM 101
  322 DTERST(NT,21) = 41250./43560.                                         TERM 102
      DTERST(NT,22) = 765. / 5280.                                          TERM 103
      GO TO 330                                                             TERM 104
C     DESIGN TYPE 8                                                         TERM 105
  324 DTERST(NT,21) = 21000. / 43560.                                       TERM 106
      DTERST(NT,22) = 765. / 5280.                                          TERM 107
      GO TO 330                                                             TERM 108
C     DESIGN TYPE 9                                                         TERM 109
  326 DTERST(NT,21) = (1.1 * (22500. + (20.8*DTERST(NT,14)))) / 43560.      TERM 110
```

```
RUN VERSION 2.3 --PSR LEVEL 20A--                                          06/30/72

      DTERST(NT,22) = (1230. + (1.73 * DTERST(NT,14))) / 5280.              TERM 111
      GO TO 330                                                             TERM 112
C     DESIGN TYPE 10                                                        TERM 113
  328 DTERST(NT,21) = (0.505 * (22500. + (20.8*DTERST(NT,14)))) / 43560.    TERM 114
      DTERST(NT,22) = (1230. + (2.2R * DTERST(NT,14))) / 5280.              TERM 115
      GO TO 330                                                             TERM 116
C     DESIGN TYPE 11 = DUMMY FOR BALANCE                                    TERM 117
  329 DTERST(NT,21) = 0.0                                                   TERM 118
      DTERST(NT,22) = 0.0                                                   TERM 119
  330 CONTINUE                                                              TERM 120
C     COMPUTE TOTAL TERMINAL AREA (ACRES)                                   TERM 121
      DTERST(NT,4) = DTERST(NT,15) + DTERST(NT,18) + DTERST(NT,21)          TERM 122
C     COMPUTE TOTAL TERMINAL TRACK LENGTH (LANE MILES)                      TERM 123
      DTERST(NT,5) = DTERST(NT,17) + DTERST(NT,22)                          TERM 124
C     COMPUTE TOTAL STRUCTURAL COST                                         TERM 125
      DTERST(NT,6) = DTERST(NT,19) + DTERST(NT,20)                          TERM 126
C     COMPUTE TOTAL TERMINAL LAND ACQUISITION COST                          TERM 127
      LC = DTERST(NT,13)                                                    TERM 128
      DTERST(NT,7) = DTERST(NT,4) * ULANDC(LC)                              TERM 129
C     COMPUTE TOTAL TRACK COST                                              TERM 130
      DTERST(NT,8) = DTERST(NT,5) * UTRANC                                  TERM 131
C     COMPUTE TOTAL TERMINAL COST                                           TERM 132
      DTERST(NT,9) = DTERST(NT,6) + DTERST(NT,7) + DTERST(NT,8)             TERM 133
C     CHECK IF LANU IS TARABLE  LC=6 TO 10 IS TARABLE                       TERM 134
      IF (LC.GT.5) GO TO 340                                                TERM 135
C     IF NOT TARABLE ENTER ZEROS FOR ACHES AND MARKET VALUE                 TERM 136
      DTERST(NT,10) = 0.0                                                   TERM 137
      DTERST(NT,11) = 5.0                                                   TERM 138
      GO TO 350                                                             TERM 139
  340 DTERST(NT,10) = DTERST(NT,4)                                          TERM 140
      DTERST(NT,11) = DTERST(NT,7)                                          TERM 141
  350 CONTINUE                                                              TERM 142
C     COMPUTE SYSTEM TOTALS FOR EACH TERMINAL STATISTIC                     TERM 143
      NTT = NTM + 1                                                         TERM 144
C     INITIALIZE TOTALS                                                     TERM 145
      DO 355  NTD = 1,22                                                    TERM 146
      DTERST(NTT,NTD) = 0.0                                                 TERM 147
C     COMPUTE TOTALS                                                        TERM 148
      DO 360  NTD = 4,22                                                    TERM 149
      DO 360  NT = 1,NTM                                                    TERM 150
  360 DTERST(NTT,NTD) = DTERST(NTT,NTD) + DTERST(NT,NTD)                    TERM 151
      THLNM1=DTERST(NTT,5)                                                  TERM 152
C                                                                           TERM 153
C     PRINT ON TERMINAL INPUT DATA WITH TOTALS                              TERM 154
      IF(JSPRNT( 8) .EQ. 1) GO TO 48                                        TERM 155
      PRINT 362                                                             TERM 156
  362 FORMAT(1H1,34HON TERMINAL INPUT DATA WITH TOTALS,///)                 TERM 157
      PRINT 363                                                             TERM 158
  363 FORMAT(1H ,* TERM TERM BEDS CONS LAND LAND    DEMAND   PARKING  STERM 159
     *TORAGE   STORAGE        PARKING        STORAGE        TERMINAL*,/  TERM 160
     *1H , * NO. TYPE AREA TYPE  USE COST    IVEN,/      AREA      AREATERM 161
     *      TRACK*,  3(8X,*CONS,COST*), / 1H , 18X, 6(* CODE*), *     MOUTERM 162
     *R)   (ACRES)   (ACRES)   (LN.MI.)*, 3(12X,*($)*), //)                TERM 163
      DO 364  NT = 1,NTM                                                    TERM 164
  364 PRINT 366, NT, DTERST(NT,12), (DTERST(NT+NTD),NTD=1,3), (DTERST(NTTERM 165
```

```
          X,NTD), NTD = 13,20)                                          TERM 166
000429    76A FORMAT(1H ,15,5F5.0,F10.0,3F10.2,3F15.0)                  TERM 167
00042A        WRITE 3A8, (DTERST(NTT,NTD), NTD = 1A,20)                 TERM 168
000441    78A FORMAT(//,1H ,6HTOTALS,24X,F10.0,3F10.2,3F15.0)           TERM 169
000441     6A CONTINUE                                                  TERM 170
        C                                                               TERM 171
        C     PRINT OUT TERMINAL STATISTICS                             TERM 172
000441        IF(JSPRNT( 4) .EQ. 1) GO TO 69                            TERM 173
000447        WRITE 378                                                 TERM 174
000447    370 FORMAT (1H1,27HOUT TERMINAL UNITS AND COSTS,///)          TERM 175
000447        PRINT 371                                                 TERM 176
000453    371 FORMAT(1H ,* TERM PARK, STOR, TERM, TOTAL STOR, TERM, TOTAL   PARKTERM 177
              X ING   STORAGE  TERMINAL     TOTAL      LAND   TRACK    TOTAL  TARTERM 178
              X,    MARKET*, / 1H ,*  NO.*,  4(* AREA*),  3(* TRACK*)  ,   TERM 179
              X  4(* CONS,CUST*),  *      COST    COST TERM,COST  AREA    VALUE*TERM 180
              X / 1H , 5X, 4(* ACRES*), 3(* LN,MI*), 5(7A,*($)*), *    ($)  TERM 181
              X ($) ACRES          ($)*, //)                              TERM 182
000453        DO 372  NT = 1,NTM                                        TERM 183
000455    372 WRITE 374, NT, (DTERST(NT,NTD),NTD=15,16), DTERST(NT,21), DTERST(NTERM 184
              XT,4), DTERST(NT,17), DTERST(NT,22), DTERST(NT,5),(DTERST(NT,NTD), TERM 185
              X NTD =18,20), (DTERST(NT,NTD), NTD =8,11)                TERM 186
000527    374 FORMAT(1H ,15,7F6.2,5F10.0,F8.0,F8.2,F10.0)               TERM 187
000527        WRITE 376, (DTERST(NTT,NTD), NTD=15,16), DTERST(NTT,21), DTERST(NTTERM 188
              XT,4), DTERST(NTT,17), DTERST(NTT,22), DTERST(NTT,5), (DTERST(NTT,NTERM 189
              XTD), NTD = 18,20), (DTERST(NTT,NTD), NTD = 8,11)         TERM 190
000574    376 FORMAT(//,1H ,5HTOTAL,7F4.2,5F10.0,F8.0,F10.0,F8.2,F10.0) TERM 191
000574     69 CONTINUE                                                  TERM 192
000574        TCOST(1)=DTERST(NTT,7)                                    TERM 193
000576        TCOST(2)=DTERST(NTT,8)                                    TERM 194
000600        TCOST(3)=DTERST(NTT,6)                                    TERM 195
000601        TCOST(4)=DTERST(NTT,9)                                    TERM 196
000603        WRITE 910                                                 TERM 197
000604    910 FORMAT(1H1,//////)                                        TERM 198
000604        WRITE 919, TITLE2(3),TITLE2(4), ANCH                      TERM 199
000629    919 FORMAT(1X,4A10,3I/))                                      TERM 200
000629        CALL ANNUAL(12,TZ,3,TCOST,DINT,DLIFE,TNAME(1,3),ANNCS)    TERM 201
000631        CALL SORT(NTM ,DSTAT,GROUP2,PRCT,AVG1,AVG2)               TERM 202
000639        RETURN                                                    TERM 203
00063A        END                                                       TERM 204
```



```
001122

UNUSED COMPILER SPACE
124500
```



```
TERMINAL

SUBPROGRAM LENGTH
000659

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
8A    =  000441    6Y    =  000574    3B0   =  000021    3B1   =  000022
3B2   =  000442    31H   =  000117    312   =  000132    314   =  000145
316   =  000146    31K   =  000173    32B   =  000206    322   =  000221
324   =  000224    32A   =  000233    32B   =  000246    329   =  000261
33B   =  000264    36U   =  000317    350   =  000323    355   =  000331
360   =  000342    36P   =  000724    363   =  000732    366   =  000773
36A   =  001000    37U   =  001006    371   =  001014    37A   =  001061
376   =  001067    75U   =  000800    910   =  001076    919   =  001102

BLOCK NAMES AND LENGTHS
RHI   =  000787    HUH   =  034884    DC1   =  000780    DC5   =  000205
DC7   =  000230    5U    =  021140    LNT   =  000006

VARIABLE ASSIGNMENTS
ANCH  =  000092C04 ANNCS  =  000000C05 ANNCST =  000050C05 ASSGAT =  000264C02
AUTNCH =  021415C02 AVG1   =  001151    AVG2   =  001721    BACKOF =  031478C02
BACKON =  012575C02 DACRAT =  021622C02 DCNCEN =  000070C02 DFARE  =  033764C02
DGLRST =  010639C02 DGNPC  =  033750C02 DGWNTH =  000050C02 DINT   =  000158C01
DLIFE  =  000190C01 DLNCAP =  000024C02 DMAINT =  000211C02 DMVLOF =  891387C02
DNUISF =  000274C02 DNUMVH =  000257C02 DOFLRC =  027462C02 DOFPFL =  022222C02
DOMPFI =  001410C02 DPCVCH =  021434C02 DPOLYD =  021273C02 DPRGRF =  028011C02
DPKHRF =  010546C02 DPTHIP =  021257C02 DSTAT  =  000000C04 DTERST =  000000C04
DTROPF =  013433C02 DVDLAY =  021301C02 DVHCAP =  000012C02 DVMLOF =  000017C02
DVMSUC =  000225C02 DVMUC  =  000253C02 DVMTPC =  021412C02 DVOPUC =  033383C02
DWAYCF =  000193C02 DYAHDS =  000118C02 FLOCAP =  033757C02 GROUP2 =  000000C03
GROUP3 =  003140C03 HACRAT =  021558C02 HLNDIR =  021482C02 HNCHAR =  021438C02
HOPIAT =  021478C02 HPCVCH =  021538C02 HVOPCS =  021449C02 HWTOPC =  021654C02
IMACCN =  000271C01 IMOLCN =  000241C01 IZ     =  000225C01 JSPRNT =  000000C01
JTH    =  000059    K      =  000043    L      =  000044    LC     =  000051
M      =  000045    NMKHUS =  021254C02 NT     =  000046    NTD    =  000047
NTM    =  000231C01 NTT    =  000052    NUMDA  =  021255C02 NVC    =  000276C01
PARCK3 =  000237C01 PHLWAT =  021544C02 PRCT   =  002471    RESDEN =  001354C02
SCALFF =  000090C02 TAXWAT =  000270C02 TCOST  =  000036    TITLE1 =  000000C04
TITLE2 =  000090C04 TNAME  =  000111C04 TRLNMI =  000003C07 TUNITS =  000031
ULANDC =  000031C02 UNTHAW =  000034C04 USTRUC =  000044C02 UTHARC =  000043C02
XACRAT =  021442C02 XDLAY  =  021382C02 XDPCU  =  021662C02 XDPICD =  022042C02
XPCVCH =  021560C02 XPTHIP =  021342C02 XSYSCH =  021315C02 XVMTPC =  021408C02
Z      =  013277C02

START OF CONSTANTS
000640

START OF TEMPORARIES
001100

START OF INPUT/OUTPUT
```

```
      SUBROUTINE ADLANFS                                                ADLN   2
C                                                                       ADLN   3
C*********************************************************************ADLN   4
C                                                                       ADLN   5
C     BEGIN OW TERMINAL ACCEL / DECEL SECTION                           ADLN   6
C                                                                       ADLN   7
C*********************************************************************ADLN   8
C                                                                       ADLN   9
      INTEGER NLIFE    , DINT                                          ADLN  10
      INTEGER PANCK3                                                   ADLN  11
      REAL  NMKMHS , NUMDA                                             ADLN  12
      DIMENSION DTADST(225,8)                                          ADLN  13
      DIMENSION AVG1(5,6,12),AVG2(5,6,12),PRCT(32,5,11)                ADLN  14
      DIMENSION DSTAT(400,22)                                          ADLN  15
      DIMENSION TUNITS(5),TCOST(5)                                     ADLN  16
      COMMON/WOI/        JSPRNT(80)     , DLIFE(30)     , DINT(30)     ,ADLN  17
     X   LFCUOF          , NVM           , NLM           , NSM          ,ADLN  18
     X   NTM             , NTSM          , NTM           , NTM          ,ADLN  19
     X   NDM             , IZ            , JR            , PARCK3(16)   ,ADLN  20
     X   NM              , NS            , IPHINT        , IRV          ,ADLN  21
     X   IPULCD(4,2,3)   , IDACCD(3)     , NHV           , NDTM         ,ADLN  22
     X   NVC(9)                                                         ,ADLN  23
      COMMON/WOR/        SCALEF(10)      , DVMCAP(5)     , DVMLDF(5)    ,ADLN  24
     X ULNCAP(5)         , ULANDC(10)    , UTRAKC        , USTRUC(4)    ,ADLN  25
     X DRWNTM(16)        , DCNCEN(11)    , OWATCC(5)     , DYAROS(5,13) ,ADLN  26
     X DMAINT(10)        , DVMSUC(4,6)   , DVMUC(6)      , DNUMVM(5)    ,ADLN  27
     X ASSMAT(6)         , TAXRAT(6)     , DNOISE(20,10,2), RESDEN(4,7) ,ADLN  28
     X URNDFL(22,4,4),   DGLRST(225,5)   , RACRDN(22,3,3), DPRMRF(225,3),ADLN  29
     X URRDNP(225,3)     ,                                              ,ADLN  30
     X URKMRS            , NUMDA         , DPKPCT        , DPTRIP(4,3)  ,ADLN  31
     X (PDLTU(2,3)       , DVDLAY(2,3)   , DMVLDF(2,3)   , XSYSCH(4,5)  ,ADLN  32
     X APKPLT            , XPTRIP(4,4)   , ADELAY(2,4,2) , MLNOIR(2,2)  ,ADLN  33
     X AVMTPC(4)         , DVMTPC(3)     , AUTOCH(11)    , MNCHAR(4)    ,ADLN  34
     X (PRCVCH(3,4,2)    , RPCVCH(4,4)   , MPCVCH(2,4)   , POLRAT(3,3)  ,ADLN  35
     X UMSTAC            , MACRAT(2,5,2) , AACRAT(4,4)   , DACRAT(2,3)  ,ADLN  36
     X MRMNAT(2,4)       , MVDPCS(2,6)   , MHYOPC(2,3)   , XOPCU(4,7,4) ,ADLN  37
     X RPRICU(4,7,4)     ,DOPPFL(150,4,4),DOFLKC(150,3,3) ,MACKDF(150,3,2),ADLN  38
     X UVRRPC            , DCCOPC        , DMOPC         , Z(4)         ,ADLN  39
     X DVRRUC(6,9,4)     , DTROPC(11,7)  , DGOPC(7)      , FLOCAP(5)    ,ADLN  40
     X UFARE(4,2,2)                                                     ,ADLN  41
      COMMON/WCI/  GROUP2(32,5,11), GROUP3(32,5,11)                    ADLN  42
      COMMON/DCS/TITLE1(16), TITLE2(10), ANCH(2), UNTNAM(5,9),         ADLN  43
     XTNAME(5,12)                                                      ADLN  44
      COMMON/DCT/ ANNCS(4,10), ANNCST(8,14)                            ADLN  45
      COMMON/SG/ DTSGST(400,22)                                        ADLN  46
      COMMON/LNT/TOLNMI,DOLNMI,DILNMI,TMLNMI,DALNMI,DYLNMI             ADLN  47
      EQUIVALENCE (DSTAT(1),DTSBST(1))                                 ADLN  48
C     READ TERMINAL SEGMENT DATA AND COUNT NUMBER OF SEGMENTS           ADLN  49
  408 NTSM = 0                                                         ADLN  50
  401 READ(11,402)NTS,(DTSGST(NTS,NTSD), NTSD=12,13), (DTSGST(NTS,NTSD),ADLN  51
     XNTSD=1,3), DTSGST(NTS,14), (DTSGST(NTS,NTSD), NTSD=4,5)          ADLN  52
      NTSM = NTSM + 1                                                  ADLN  53
      IF (NTS .NE. 400) GO TO 401                                      ADLN  54
      NTSM = NTSM - 1                                                  ADLN  55
```



```
      DO 444  NTD = 1,6                                                ADLN 111
      DO 444  NT = 1,NTM                                               ADLN 112
444   DTADST(NTT,NTD) = DTADST(NTT,NTD) + DTADST(NT,NTD)               ADLN 113
      DALNMI=DUTADST(NTT,2)                                            ADLN 114
C                                                                      ADLN 115
C     PRINT OW TERMINAL ACCEL / DECEL LANE UNITS AND COSTS             ADLN 116
      IF(JSPRNT(1) .EQ. 1) GO TO 71                                    ADLN 117
      PRINT 447                                                        ADLN 118
447   FORMAT(1H1,30HTERMINAL ACCEL DECEL LANE STATISTICS,///)          ADLN 119
      PRINT 443                                                        ADLN 120
443   FORMAT(1H ,9 TERM        LAND       TRACK=, 9X,9STRUC,9X,11X,9LANDP,10XADLN 121
     X,9TRACK=,1X,9TOTAL,9X,9TAX,9X,9X,9MARKET=,/ 1H , 9   NO.      AREAADLN 122
     X     LENGTH=, 4(11X,9CDST),9      AREA=,10X,9VALUE=,/ 1H , 5X,9  ADLN 123
     X(ACRES)  (LN,MI,)=, 4(12X,9($)=),9    (ACRES)=,12X,9($)=, //)   ADLN 124
      DO 444  NT = 1,NTM                                               ADLN 125
444   PRINT 445, NT, (DTADST(NT,NTD),NTD = 1,8)                        ADLN 126
445   FORMAT(1H ,15,2X,8F15.0,2X,F10.2,F15.0)                          ADLN 127
      PRINT 446,  (DTADST(NTT,NTD), NTD = 1,8)                         ADLN 128
446   FORMAT(//,1H ,9TOTALS,F9.2,F10.2,4F15.0,F10.2,F15.0)             ADLN 129
   71 CONTINUE                                                         ADLN 130
      TUNITS(1)=DTADST(NTT,1)                                          ADLN 131
      TUNITS(2)=DTADST(NTT,2)                                          ADLN 132
      TUNITS(3)=DTADST(NTT,2)                                          ADLN 133
      TUNITS(4) = 1000000000000.                                       ADLN 134
      TCOST(1)=DTADST(NTT,4)                                           ADLN 135
      TCOST(2)=DTADST(NTT,5)                                           ADLN 136
      TCOST(3)=DTADST(NTT,3)                                           ADLN 137
      TCOST(4)=DTADST(NTT,6)                                           ADLN 138
      PRINT 410                                                        ADLN 139
410   FORMAT(1H1,5/)                                                   ADLN 140
      PRINT 9[X, TITLF1(3),TITLE1(4), ANCH                             ADLN 141
919   FORMAT(1X,A8,0,3/)1)                                             ADLN 142
      CALL ANNUALO(2,IZ,3,TUNITS,TCOST,DINT,ULIFE,UNTNAM(1),41,        ADLN 143
     X     TNAME(1,4),ANNCST )                                         ADLN 144
      CALL SUNTINTSM,DSTAT,GRNHP2,PRCT,AVG1,AVG2                       ADLN 145
      IF(JSPRNT(1) .EQ. 1) GO TO 79                                    ADLN 146
      CALL PUNTARIGRNHP2,PRCT,AVG1,AVG2)                               ADLN 147
   79 CONTINUE                                                         ADLN 148
C                                                                      ADLN 149
C     UPDATE NWDWP3                                                    ADLN 150
C                                                                      ADLN 151
      DO 511 K=1,5                                                     ADLN 152
      DO 511 L=1,11                                                    ADLN 153
      DO 511 M=1,32                                                    ADLN 154
511   GRNHP3(M,K,L)=GRNHP3(M,K,L)+GRNHP2(M,K,L)                        ADLN 155
      RETURN                                                           ADLN 156
      END                                                              ADLN 157
```



```
C     NOTE LAST CARD MUST HAVE A ZERO IN COL 5                          ADLN  56
  409 FORMAT(1H,4F5.0,F10.2,F10.2)                                     ADLN  57
C                                                                      ADLN  58
C     COMPUTE TERMINAL SEGMENT STATISTICS                              ADLN  59
      DO 410  NTS = 1,NTSM                                             ADLN  60
C     COMPUTE ACCEL / DECEL STRUCTURAL COST THIS SEGM                  ADLN  61
      K = DTSGST(NTS,2)                                                ADLN  62
      DTSGST(NTS,6) = DTSGST(NTS,5) * USTRUC(K)                        ADLN  63
C     COMPUTE LAND ACQUISITION COST                                    ADLN  64
      LC = DTSGST(NTS,4)                                               ADLN  65
      DTSGST(NTS,7) = DTSGST(NTS,6) * ULANDC(LC)                       ADLN  66
C     COMPUTE TRACK COST                                               ADLN  67
      DTSGST(NTS,8) = DTSGST(NTS,5) * UTRAKC                           ADLN  68
C     COMPUTE TOTAL COST THIS SEGM                                     ADLN  69
      DTSGST(NTS,9) = DTSGST(NTS,6) + DTSGST(NTS,7) + DTSGST(NTS,8)    ADLN  70
C     CHECK IF LAND IS TAXABLE  ( LC=6 TO 18  IS TAXABLE )             ADLN  71
      IF (LC.GT.5) GO TO 405                                           ADLN  72
C     IF NOT TAXABLE ENTER ZEROS FOR ACRES AND MARKET VALUE            ADLN  73
      DTSGST(NTS,10) = 0.0                                             ADLN  74
      DTSGST(NTS,11) = 0.0                                             ADLN  75
      GO TO 410                                                        ADLN  76
  405 DTSGST(NTS,10) = DTSGST(NTS,4)                                   ADLN  77
      DTSGST(NTS,11) = DTSGST(NTS,7)                                   ADLN  78
  410 CONTINUE                                                         ADLN  79
C                                                                      ADLN  80
C     PRINT TERMINAL SEGMENT STATISTICS                                ADLN  81
      IF(JSPRNT(10) .EQ. 1) GO TO 70                                   ADLN  82
      PRINT 414                                                        ADLN  83
414   FORMAT(1H1,27HTERMINAL SEGMENT STATISTICS,///)                   ADLN  84
      PRINT 416                                                        ADLN  85
416   FORMAT(1H ,9 SEGM TERM   A/D GEOG CONS LAND LAND     LAND   TRACK ADLN  86
     X   STRUC,        LAND        TRACK        TOTAL    TAX,     MARKETADLN  87
     X / 1H , 9INDEX  NO. SEGM AREA TYPE  USE COST     AREA  LENGTH9,  ADLN  88
     X 4(8X,9COST9), 9     AREA        VALUE9, / 1H ,  10X, 9  NO.9,   ADLN  89
     X 4(9 CODE9), 9   ACRES  LN,MI,9, 4(8X,9($)=),9    ACRES9,9X,9($)9ADLN  90
     X //)                                                             ADLN  91
      DO 420  NTS = 1,NTSM                                             ADLN  92
420   PRINT 419, NTS,(DTSGST(NTS,NTSD), NTSD=12,13), (DTSGST(NTS,NTSD),ADLN  93
     XNTSD=1,3), DTSGST(NTS,14), (DTSGST(NTS,NTSD), NTSD=4,11)         ADLN  94
419   FORMAT(1H ,I5,8F5.0,2F8.2,4F12.0,F8.2,F12.0)                     ADLN  95
   70 CONTINUE                                                         ADLN  96
C                                                                      ADLN  97
C     ADD SEGMENTS PER TERMINAL                                        ADLN  98
C     INITIALIZE DTADST                                                ADLN  99
      NTT = NTM + 1                                                    ADLN 100
      DO 425  NT = 1,NTT                                               ADLN 101
      DO 425  NTD = 1,8                                                ADLN 102
425   DTADST(NT,NTD) = 0.0                                             ADLN 103
      DO 430  NTS = 1,NTSM                                             ADLN 104
      NT = DTSGST(NTS,12)                                              ADLN 105
      DO 430  NTSD = 4,11                                              ADLN 106
      NTD = NTSD - 3                                                   ADLN 107
430   DTADST(NT,NTD) = DTADST(NT,NTD) + DTSGST(NTS,NTSD)               ADLN 108
C                                                                      ADLN 109
C     COMPUTE ALL TERMINAL TOTALS                                      ADLN 110
```



```
ADLANFS

SUBPROGRAM LENGTH
011131

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
78    -  000143   71    -  000311   79    -  000371   400   -  000002
401   -  000003   402   -  000422   405   -  000077   410   -  000103
414   -  000477   416   -  000435   419   -  000477   425   -  000170
440   -  000275   442   -  000504   443   -  000513   444   -  000540
446   -  000593   511   -  000375   919   -  000562   919   -  000565

BLOCK NAMES AND LENGTHS
WDI   -  000307   WWH   -  034004   DCI   -  000700   DCS   -  000205
DCT   -  000730   SG    -  021140   LNT   -  000006

VARIABLE ASSIGNMENTS
ANCH  -  000032C04 ANNCS  -  000000C05 ANNCST -  000050C05 ASSMAT -  000264C02
AUTOCH -  021415C02 AVG1  -  004226   AVG2  -  004776   BACKDF -  011470C02
BACKOM -  012575C02 DACRAT -  021022C02 DALNMI -  000004C07 DCNCEN -  000070C02
DFARE -  013744C02 DGLRST -  010430C02 DGOPC  -  033735C02 DRWNTM -  000050C02
DINT  -  000156C01 DLIFE  -  000120C01 DLNCAP -  000024C02 DMAINT -  000211C02
DMVLDF -  021367C02 DNOISE -  000274C02 DNUMVM -  000257C02 DOFLKC -  026762C02
DOPPFL -  022222C02 DRWNFL -  001410C02 DPCVCH -  021434C02 DPDLYD -  021273C02
DPKOMF -  020011C02 DPKMRF -  016548C02 DPTRIP -  021257C02 DSTAT  -  000000C04
DTAMST -  000616   DTROPC -  033603C02 DTSGST -  000000C04 DVDLAY -  021301C02
DVMCAP -  000012C02 DVMLDF -  000017C02 DVMSUC -  000223C02 DVMUC  -  000253C02
DVMTPC -  021412C02 DVRRUC -  033303C02 DWATCC -  000103C02 DYAROS -  000110C02
FLOCAP -  033747C02 GROUP2 -  000000C03 GROUP3 -  003348C03 MACRAT -  021550C02
MLNOIR -  021482C02 MNCHAR -  021430C02 MRMNAT -  021038C02 MPCVCH -  021538C02
MVDPCS -  021448C02 MHYOPC -  021454C02 IDACCD -  000271C01 IPDLCD -  000241C01
IZ    -  002295C01 JSPRNT -  000000C01 K     -  011122   L     -  011127
LC    -  011123   M     -  011130   NPKMRS -  021254C02 NT    -  011125
NTD   -  011124   NTM   -  000221C01 NTS   -  011120   NTSD  -  011121
NTSM  -  000232C01 NTT   -  011124   NUMDA  -  021255C02 NVC   -  000276C01
PARCK3 -  000297C01 PULRAT -  021544C02 PRCT  -  005546   RESDEN -  001354C02
SCALFF -  000000C02 TAXRAT -  000278C02 TCOST -  011113   TITLE1 -  000000C04
TITLF2 -  000095C04 TNAME  -  000111C04 TUNITS -  011106   ULANDC -  000031C02
UNTNAM -  000044C04 USTRUC -  000844C02 UTRAKC -  000043C02 XACRAT -  021602C02
ADELAY -  021342C02 XOPCU  -  021662C02 XOPICD -  022042C02 XPCVCH -  021508C02
XPTRIP -  021342C02 XSYSCH -  021315C02 XVMTPC -  021408C02 Z     -  033277C02

START OF CONSTANTS
000420

START OF TEMPORARIES
000571

START OF INDIRECTS
000574

UNUSED COMPILER SPACE
```

```
                SUBROUTINE OTHRCAP                                      OTCP   2
C                                                                       OTCP   3
C*********************************************************************OTCP   4
C                                                                       OTCP   5
C       COMPUTE OTHER DM CAPITAL COSTS                                  OTCP   6
C                                                                       OTCP   7
C*********************************************************************OTCP   8
C                                                                       OTCP   9
C       COMPUTE DM CONTROL CENTER STATISTICS                            OTCP  10
C                                                                       OTCP  11
C*********************************************************************OTCP  12
C                                                                       OTCP  13
000002          INTEGER DLIFE   , DINT                                  OTCP  14
000002          INTEGER  PARCK3                                         OTCP  15
000002          REAL NPRKWS , NUMDA                                     OTCP  16
000002          DIMENSION DVMCST(5)                                     OTCP  17
000002          DIMENSION TUNITS(5),TCOST(5)                            OTCP  18
000002          COMMON/UNI/          JSPRNT(8A)   , DLIFE(30)   , DINT(30)   ,OTCP  19
     X    LECODE          , NYM           , NLM           , NSM           ,OTCP  20
     X    NIM             , NYM           , NTSM          , NTM           ,OTCP  21
     X    NOM             , IZ            , JK            , PARCK3(6)     ,OTCP  22
     X    NW              , NS            , IPRINT        , IXV           ,OTCP  23
     X    IPULC(4,2,3), IDACC(3)    , NMV           , NDYM          ,OTCP  24
     X    NVC(9)                                                        OTCP  25
000003          COMMON/NOW/          SCALEF(10)   , DVMCAP(5)     , DVMLDF(5)     ,OTCP  26
     X DLNCAP(5)       , ULANDC(10)    , UTRARC        , USTRUC(4)     ,OTCP  27
     X DRNTT(16)       , DCNCEN(11)    , DWATCC(5)     , DYARDS(5,13)  ,OTCP  28
     X DMAINT(18)      , DVMSUC(4,6)   , DVMUC(4)      , DMUNWW(5)     ,OTCP  29
     X ASSWAT(4)       , TARRAT(4)     , DNOISE(28,14,2), RESDEN(4,7)  ,OTCP  30
     X DONPFL(225,4,4), DGLKRT(225,5), BACKON(225,3,3), DPRMRF(225,3) ,OTCP  31
     X UPRMRF(225,3)   ,                                                OTCP  32
     X NPKWS           , NUMDA         , DPKPCT        , DPTRIP(4,3)   ,OTCP  33
     X DPDLY(12,3)     , DVDLAY(2,3)   , DMVLDF(2,3)   , XSYSCM(4,5)   ,OTCP  34
     X XPRPCT          , RPTRIP(4,4)   , ADELAY(2,4,2) , HLNDIR(2,2)   ,OTCP  35
     X XVMTPC(4)       , DVMTPC(3)     , AUTOCM(11)    , MNCHAR(4)     ,OTCP  36
     X DPCVCM(3,6,2)   , XPCVCM(4,6)   , MPCVCM(2,6)   , PDLRAT(3,3)   ,OTCP  37
     X DMSTAC          , MACRAT(2,5,2) , XACRAT(4,4)   , DACRAT(2,3)   ,OTCP  38
     X MNPDAT(2,4)     , MVDPCS(7,6)   , MNYDPC(2,3)   , XDPCU(4,7,4)  ,OTCP  39
     X APPICU(4,7,4)   ,DOFPFL(19,4,4),DOFLRC(150,3,3) ,BACKOF(150,3,2),OTCP  40
     X DYMPC           , DCCOPC        , DMOPC         , Z(4)          ,OTCP  41
     X DVDPUC(6,9,4)   , DYRDPC(11,7)  , DODPC(7)      , FLOCAP(5)     ,OTCP  42
     X UPANE(4,2,2)                                                     OTCP  43
000002          COMMON/DC1/  GROUP2(32,5,11), GROUP3(32,5,11)           OTCP  44
000002          COMMON/DCS/TITLE1(16), TITLE2(10), ANCM(2), UNTNAM(5,9),OTCP  45
     XTNAME(5,12)                                                       OTCP  46
000002          COMMON/UCT/ ANNCST(4,10), ANWCST(8,14)                  OTCP  47
000002          COMMON /GV13/ ASUMR(14,6), DTRTMI, DTLNMI, DTYDA        OTCP  48
000002          COMMON/LNT/TOLNMI,DBLNMI,DTLNMI,TRLNMI,DALNMI,DYLNMI    OTCP  49
000007          LC = DCNCEN(4)                                          OTCP  50
000014          DCNCEN(6) = DCNCEN(5) * ULANDC(LC)                      OTCP  51
C       COMPUTE TOTAL DM CONTROL CENTER COST                            OTCP  52
000027          DCNCEN(9) = DCNCEN(6) + DCNCEN(7) + DCNCEN(8)           OTCP  53
C       COMPUTE TAXABLE AREA AND MARKET VALUE                           OTCP  54
C       CHECK IF LAND IS TAXABLE  (LC=6 TO 10 IS TAXABLE)               OTCP  55
```



```
000032          IF (LC.GT.5) GO TO 520                                  OTCP  56
000034          DCNCEN(10) = 0.0                                        OTCP  57
000036          DCNCEN(11) = 0.0                                        OTCP  58
000040          GO TO 522                                               OTCP  59
000047    520 DCNCEN(10) = DCNCEN(5)                                    OTCP  60
000051          DCNCEN(11) = DCNCEN(8)                                  OTCP  61
000052    522 CONTINUE                                                  OTCP  62
C                                                                       OTCP  63
C       PRINT DM CONTROL CENTER STATISTICS                              OTCP  64
000052          IF(JSPRNT(12) .EQ. 1) GO TO 72                          OTCP  65
000054          PRINT 524                                               OTCP  66
000060    524 FORMAT(1H1,///,20H DM CONTROL CENTER STATISTICS,///)      OTCP  67
000070          PRINT 526                                               OTCP  68
000076    526 FORMAT(1H ,9 GEOG CONS LAND LAND*,     8X,*LAND*,11X, *LAND*,8X,  OTCP  69
     X*STRUC.*, 7X,*COMPUTER*, 10X, *TOTAL*, 8X,*TAX.*, 8X, *MARKET*,/  OTCP  70
     X1H , * AREA TYPE  USE COST      AREA*, 2(11X,*COST*), *       FACIOTCP  71
     X*LITY*, 11X, *COST*, 8X, *AREA*, 10X, *VALUE*, / 1H ,  6I* CODE*), OTCP  72
     X *     (ACRES)*, 4(12X,*($)*), *    (ACRES)*, 12X, *(S)*, //)    OTCP  73
000034          PRINT 528, (DCNCEN(JCD), JCD =1,5), DCNCEN(8), DCNCEN(6),DCNCEN(7)OTCP  74
     X,(DCNCEN(JCD), JCD = 9,11)                                        OTCP  75
000062    528 FORMAT(1H ,*F5.0,F10.2,*F15.0,F10.2,F15.0)                OTCP  76
000067     72 CONTINUE                                                  OTCP  77
000067          TCOST(1)=DCNCEN(8)                                      OTCP  78
000064          TCOST(2)=DCNCEN(7)                                      OTCP  79
000065          TCOST(3)=DCNCEN(6)                                      OTCP  80
000067          TCOST(4)=DCNCEN(9)                                      OTCP  81
000070          PRINT 900                                               OTCP  82
000074    900 FORMAT(20(/))                                             OTCP  83
000074          PRINT 919, TITLE2(5),TITLE2(6), ANCM                    OTCP  84
000104          CALL ANNUAL2(3,IZ,3,TCOST,DINT,DLIFE,TNAME(1,5),ANNCS)  OTCP  85
C                                                                       OTCP  86
C*********************************************************************OTCP  87
C                                                                       OTCP  88
C       COMPUTE DM MAINTENANCE FACILITY STATISTICS                      OTCP  89
C                                                                       OTCP  90
C*********************************************************************OTCP  91
C                                                                       OTCP  92
C       COMPUTE LAND ACQUISITION COST                                   OTCP  93
000117          LC = DMAINT(4)                                          OTCP  94
000121          DMAINT(7) = DMAINT(5) * ULANDC(LC)                      OTCP  95
C       COMPUTE TOTAL MAINT FACIL COST                                  OTCP  96
000124          DMAINT(8) = DMAINT(6) + DMAINT(7)                       OTCP  97
C       COMPUTE TAXABLE ACRES AND MARKET VALUE                          OTCP  98
000126          IF (LC.GT.5) GO TO 540                                  OTCP  99
000131          DMAINT(9)  = 0.0                                        OTCP 100
000131          DMAINT(10) = 0.0                                        OTCP 101
000132          GO TO 542                                               OTCP 102
000133    540 DMAINT(9)  = DMAINT(5)                                    OTCP 103
000135          DMAINT(10) = DMAINT(7)                                  OTCP 104
000136    542 CONTINUE                                                  OTCP 105
C       PRINT DM MAINTENANCE FACIL STAT                                 OTCP 106
000136          IF(JSPRNT(13) .EQ. 1) GO TO 73                          OTCP 107
000140          PRINT 544                                               OTCP 108
000144    544 FORMAT(1H1,///,* DM MAINTENANCE FACILITY STATISTICS*,///) OTCP 109
000144          PRINT 545                                               OTCP 110
```



```
000150    545 FORMAT(1H ,* GEOG CONS LAND LAND      LAND      LAND    STRUC.   OTCP 111
     X TOTAL      TAX.    MARKET* / 1H , * AREA TYPE  USE COST      AREAOTCP 112
     X      COST  * EQUIP.      COST      AREA      VALUE* / 1H ,          OTCP 113
     X *(* CODE*), *    (ACRES)*, 7X, *($)*, *  COST ($)*, 7X, *($)   (ACOTCP 114
     X*ES)*, 7X, *($)*, //)                                             OTCP 115
000190          PRINT 546, (DMAINT(JMD), JMD=1,5), DMAINT(7),(DMAINT(6),(DMAINT(JMD)OTCP 116
     X), JMD =6,10)                                                     OTCP 117
000174    546 FORMAT(1H ,*F5.0,F10.2,3F10.0,F10.2,F18.0)                OTCP 118
000174     73 CONTINUE                                                  OTCP 119
000174          TCOST(1)=DMAINT(7)                                      OTCP 120
000174          TCOST(2)=DMAINT(6)                                      OTCP 121
000177          TCOST(3)=DMAINT(8)                                      OTCP 122
000201          PRINT 900                                               OTCP 123
000204          PRINT 919, TITLE2(7),TITLE2(8), ANCM                    OTCP 124
000214          CALL ANNUAL2(4,IZ,2,TCOST,DINT,DLIFE,TNAME(1,6),ANNCS)  OTCP 125
C                                                                       OTCP 126
C*********************************************************************OTCP 127
C                                                                       OTCP 128
C       COMPUTE DM YARDS STATISTICS                                     OTCP 129
C                                                                       OTCP 130
C*********************************************************************OTCP 131
C                                                                       OTCP 132
000227          DO 555  NY = 1,NYM                                      OTCP 133
C       COMPUTE LAND ACQUISITION COST THIS YARD                         OTCP 134
C       COMPUTE TRACK COST                                              OTCP 135
C       COMPUTE STRUCTURAL COST                                         OTCP 136
000231          LC = DYARDS(NY,4)                                       OTCP 137
000231          DYARDS(NY,8) = DYARDS(NY,5) * ULANDC(LC)                OTCP 138
000234          DYARDS(NY,9) = DYARDS(NY,7) * UTRARC                    OTCP 139
000240          DYARDS(NY,10)= DYARDS(NY,5) * DYARDS(NY,6)              OTCP 140
C       COMPUTE TOTAL YARD COST                                         OTCP 141
000242          DYARDS(NY,11) = DYARDS(NY,8) + DYARDS(NY,9) + DYARDS(NY,10)  .  OTCP 142
C       COMPUTE TAXABLE AREA AND MARKET VALUE                           OTCP 143
000246          IF (LC.GT.5)  GO TO 554                                 OTCP 144
000251          DYARDS(NY,12) = 0.0                                     OTCP 145
000257          DYARDS(NY,13) = 0.0                                     OTCP 146
000261          GO TO 556                                               OTCP 147
000264    554 DYARDS(NY,12) = DYARDS(NY,5)                              OTCP 148
000266          DYARDS(NY,13) = DYARDS(NY,8)                            OTCP 149
000270    555 CONTINUE                                                  OTCP 150
C       INITIALIZE  YARD TOTALS                                         OTCP 151
000263          NYT= NYM +1                                             OTCP 152
000265          DO 560  NYD = 5,13                                      OTCP 153
000266    560 DYARDS(NYT,NYD) = 0.0                                     OTCP 154
C       COMPUTE TOTALS FOR ALL YARDS                                    OTCP 155
000273          DO 565  NYD = 5,13                                      OTCP 156
000276          DO 565  NY = 1,NYM                                      OTCP 157
000274    565 DYARDS(NYT,NYD) = DYARDS(NYT,NYD) + DYARDS(NY,NYD)        OTCP 158
000310          DYLNMI=DYARDS(NYT,7)                                    OTCP 159
000312          DTYDA= DYARDS (NYT, 5)                                  OTCP 160
C                                                                       OTCP 161
C       PRINT DM YARD STATISTICS                                        OTCP 162
000314          IF(JSPRNT(14) .EQ. 1) GO TO 74                          OTCP 163
000314          PRINT 568                                               OTCP 164
000321    568 FORMAT(1H1,///,* DM YARD STATISTICS*,///)                 OTCP 165
```

```
000321          PRINT 569                                                OTCP 166
000325    569 FORMAT(1H ,* YARD GEOG CONS LAND LAND*, 2(6X,*LAND*), 2(5X,*TRACK*OTCP 167
              1;  2(6X,*STRUC,*), 5X, *TOTAL*, 6X, *TAX,*, 6X, *MARKET*, / 1H , OTCP 168
              2*  NO. AREA TYPE  USE COST      AREA      COST    LENGTH      COSTOTCP 169
              3 UNIT COST      COST      COST          AREA      VALUE* / 1H , 5X,  OTCP 170
              4 4(* CUBE*), *    (ACRES)        ($)  (LN.MI.)        ($)  ($/ACRE)* OTCP 171
              5 2(7X,*($)*), *    (ACRES)*, 7X, *($)*, //)                      OTCP 172
000325          DO 570  NY = 1,NYM                                       OTCP 173
000327    570 PRINT 572, NY, (DYARDS(NY,NYD), NYD=1,5), DYARUS(NY,8), DYARDS(NY,OTCP 174
              *71, DYARDS(NY,9), DYARUS(NY,6), (DYARUS(NY,NYD), NYD = 10,13)   OTCP 175
000336    572 FORMAT(1H ,I5,4F5.0,F10.2,F10.0,F10.2,4F10.0,F10.2,F10.0)      OTCP 176
000364          PRINT 574, DYARDS(NYT,5), DYARDS(NYT,8), DYARDS(NYT,7), DYARDS(NYTOTCP 177
              *,9), DYARDS(NYT,6), (DYARDS(NYT,NYD), NYD = 10, 13)             OTCP 178
000411    574 FORMAT(//,1H ,6HTOTALS,19X,F10.2,F10.0,F10.2,4F10.0,F10.2,F10.0) OTCP 179
000411     74 CONTINUE                                                     OTCP 180
000411        TCOST(1)=DYARDS(NYT,8)                                       OTCP 181
000413        TCOST(2)=DYARDS(NYT,10)                                      OTCP 182
000414        TCOST(3)=DYARDS(NYT,9)                                       OTCP 183
000416        TCOST(4)=DYARDS(NYT,11)                                      OTCP 184
000420        TUNITS(1)=DYARDS(NYT,5)                                      OTCP 185
000421        TUNITS(2)=DYARDS(NYT,5)                                      OTCP 186
000423        TUNITS(3)=DYARDS(NYT,7)                                      OTCP 187
000424        TUNITS(4) = 1000000000000.                                   OTCP 188
000426        PRINT 900                                                    OTCP 189
000431        PRINT 919, TITLE1(5),TITLE1(6), ANCH                         OTCP 190
000447        CALL ANNUALD(3,IZ,3,TUNITS,TCOST,DINT,DLIFE,UNTNAM(1,7),     OTCP 191
             X  TNAME(1,7),ANNCST )                                        OTCP 192
000460        NTLNM=GROUP3(8,5,1)                                          OTCP 193
C             ADD MISCELLANEOUS UNITS AND COSTS TO GROUP3 ARRAY FOR SYSTEM OTCP 194
C             TOTALS NY GEOG. AREA, CONSTRUCTION TYPE AND LAND USE CATAGORIES OTCP 195
000462        N = NCNCEN(1)                                                OTCP 196
000464        K = NCNCEN(2)                                                OTCP 197
000466        L = NCNCEN(3)                                                OTCP 198
000470        GROUP3(N,K,L) = GROUP3(N,K,L) + DCNCEN(5)                    OTCP 199
000498        GROUP3(N+8,K,L) = GROUP3(N+8,K,L) +DCNCEN(6)                 OTCP 200
000502        GROUP3(N+12,K,L) = GROUP3(N+12,K,L) + DCNCEN(8)              OTCP 201
000507        GROUP3(N+20,K,L) = GROUP3(N+20,K,L) + DCNCEN( 9)             OTCP 202
000513        GROUP3(N+24,K,L) = GROUP3(N+24,K,L) + DCNCEN(10)             OTCP 203
000528        GROUP3(N+28,K,L) = GROUP3(N+28,K,L) + DCNCEN(11)             OTCP 204
C                                                                          OTCP 205
000526        N = DMAINT(1)                                                OTCP 206
000528        K = DMAINT(2)                                                OTCP 207
000530        L = DMAINT(3)                                                OTCP 208
000532        GROUP3(N,K,L   ) = GROUP3(N,K,L   ) + DMAINT( 5)             OTCP 209
000540        GROUP3(N+8,K,L ) = GROUP3(N+8,K,L ) + DMAINT( 6)             OTCP 210
000544        GROUP3(N+12,K,L) = GROUP3(N+12,K,L) + DMAINT(7)              OTCP 211
000551        GROUP3(N+20,K,L) = GROUP3(N+20,K,L) + DMAINT( 8)             OTCP 212
000555        GROUP3(N+24,K,L) = GROUP3(N+24,K,L) + DMAINT( 9)             OTCP 213
000562        GROUP3(N+28,K,L) = GROUP3(N+28,K,L) + DMAINT(10)             OTCP 214
000566        DO 580  NY = 1,NYM                                           OTCP 215
000567        N = DYARDS(NY,1)                                             OTCP 216
000571        K = DYARDS(NY,2)                                             OTCP 217
000573        L = DYARDS(NY,3)                                             OTCP 218
000575        GROUP3(N,K,L   ) = GROUP3(N,K,L   ) + DYARDS(NY,5 )          OTCP 219
000601        GROUP3(N+4,K,L ) = GROUP3(N+4,K,L ) + DYARDS(NY,7 )          OTCP 220
```



```
              GROUP3(N+8,K,L ) = GROUP3(N+8,K,L ) + DYARDS(NY,10)          OTCP 221
              GROUP3(N+12,K,L) = GROUP3(N+12,K,L) + DYARDS(NY,8 )          OTCP 222
              GROUP3(N+16,K,L) = GROUP3(N+16,K,L) + DYARDS(NY,9 )          OTCP 223
              GROUP3(N+20,K,L) = GROUP3(N+20,K,L) + DYARDS(NY,11)          OTCP 224
              GROUP3(N+24,K,L) = GROUP3(N+24,K,L) + DYARDS(NY,12)          OTCP 225
    680       GROUP3(N+28,K,L) = GROUP3(N+28,K,L) + DYARDS(NY,13)          OTCP 226
              TOLNMI=GLNMI+DLNMI+TMLNMI+DALNMI+UTLNMI                      OTCP 227
C                                                                          OTCP 228
C*************************************************************************OTCP 229
C                                                                          OTCP 230
C     COMPUTE ON WAY COMMAND AND CONTROL STATISTICS                        OTCP 231
C                                                                          OTCP 232
C*************************************************************************OTCP 233
C                                                                          OTCP 234
C     COMPUTE TOTAL EQUIPMENT COST                                         OTCP 235
000678        OWAYCC(3) = OWAYCC(1) + OTOTWI                               OTCP 236
C     COMPUTE  TOTAL SYSTEM LANE MILES OF TRACK                            OTCP 237
C     COMPUTE TOTAL DETECTOR COST                                          OTCP 238
000672        OWAYCC(4) = OWAYCC(2) + TOLNMI                               OTCP 239
C     COMPUTE TOTAL WAY CC COST                                            OTCP 240
000674        OWAYCC(5) = OWAYCC(3) + OWAYCC(4)                            OTCP 241
C                                                                          OTCP 242
C     PRINT ON WAY COMMAND AND CONTROL STATISTICS                          OTCP 243
000678        IF(JSPHNT(15) .EQ. 1) GO TO 75                               OTCP 244
000680        PRINT 530                                                    OTCP 245
000703    530 FORMAT(1H1,///,* ON WAY COMMAND AND CONTROL STATISTICS*,///)  OTCP 246
000705        PRINT 531                                                    OTCP 247
000707    531 FORMAT(1H , *      ROUTE C+C EQUIP TOT EQUIP       LANE  DETECTOR  TOTCP 248
             XNT.OPT.      TOTAL* / 1H ,                 21(5X,*MILES*,6X,*COST*,6X, OTCP 249
             X*COST*), *  C+C COST*, / 1H , 12X, *S/RT.MI.*, 7X, *($)*, 12X,  OTCP 250
             X *$/LN.MI.*,          21(7X,*($)*), //)                         OTCP 251
000707        PRINT 532, (TATMI, OWAYCC(I),I=1,5), TOLNMI, OWAYCC(2),      OTCP 252
             XOWAYCC(4), OWAYCC(5)                                         OTCP 253
000731    532 FORMAT(1H ,F10.1,12F10.0,F10.2,3F10.0)                       OTCP 254
000731     75 CONTINUE                                                     OTCP 255
000731        TUNITS(1)=OTOTWI                                             OTCP 256
000733        TUNITS(2)=TOLNMI                                             OTCP 257
000734        TUNITS(3) = 1000000000000.                                   OTCP 258
000736        PRINT 900                                                    OTCP 259
000741        PRINT 919, TITLE1(7),TITLE1(8), ANCH                         OTCP 260
000753        PRINT 1234, IZ,NYM,DINT(IZ),DLIFE(IZ)                        OTCP 261
000767   1234 FORMAT(5I10)                                                 OTCP 262
000767        CALL ANNUALD(4,IZ,2,TUNITS,OWAYCC(3),DINT,DLIFE,UNTNAM(1,8),OTCP 263
             X  TNAME(1,8),ANNCST)                                         OTCP 264
001004        PRINT 1234, IZ,NYM,DINT(IZ),DLIFE(IZ)                        OTCP 265
C             COMPUTE PERCENTS EACH CATAGORY OF GRAND TOTALS               OTCP 266
001020        CALL USORT (GROUP3)                                          OTCP 267
C                                                                          OTCP 268
C*************************************************************************OTCP 269
C                                                                          OTCP 270
C     BEGIN ON VEHICLE SECTION                                             OTCP 271
C                                                                          OTCP 272
C*************************************************************************OTCP 273
C                                                                          OTCP 274
C     COMPUTE ON VEHICLE CAPITAL COSTS                                     OTCP 275
```



```
C                                                                          OTCP 276
C     CHECK IF TOTAL ON VEHICLE SUBSYSTEM UNIT COSTS HERE INPUT            OTCP 277
C APPLY SCALE FACTORS                                                      OTCP 278
001022        DO 595 NV=1,NVM                                              OTCP 279
001024        NTRANV=0                                                     OTCP 280
001029    595 NMUMVM(NV)=DMUMVM(NV)*SCALEF(1)*SCALEF(NTA)                  OTCP 281
001033        NVT=NVM+1                                                    OTCP 282
001036        DO 612 NV=1,NVM                                              OTCP 283
001038        IF (DVMUC(NV).NE.0.0) GO TO 612                              OTCP 284
001037        DVMUC(NV) = 0.0                                              OTCP 285
001040        DO 610 NV5 = 1,4                                             OTCP 286
001042        DVMUC(NV) = DVMUC(NV) + DUMSUC(NV,NV5)                       OTCP 287
001047    610 CONTINUE                                                     OTCP 288
001051    612 CONTINUE                                                     OTCP 289
C             COMPUTE COST OF EACH VEHICLE FLEET                           OTCP 290
001054        DO 615  NV = 1,NVM                                           OTCP 291
001055    615 DVMCST(NV) = DVMUC(NV) * DMUMVM(NV)                          OTCP 292
001061        DVMCST(NVT)=0.0                                              OTCP 293
001063        DMUMVM(NVT)=0.0                                              OTCP 294
001064        DO 620 NVP=1,NVM                                             OTCP 295
001068        DVMCST(NVT)=DVMCST(NVT) + DVMCST(NVP)                        OTCP 296
001070    620 DMUMVM(NVT) = DMUMVM(NVT) +DMUMVM (NVP)                      OTCP 297
001074        DO 617 NV = 1,NVM                                            OTCP 298
001076        PRINT 619                                                    OTCP 299
001101    619 FORMAT(1H1)                                                  OTCP 300
001101        PRINT 900                                                    OTCP 301
001104        NV2 = NV + 4                                                 OTCP 302
001107        NP = 2*NV - 1                                                OTCP 303
001111        PRINT 919, TITLE1(NP+8), TITLE1(NP+9), ANCH                  OTCP 304
001122    619 FORMAT(1X,4A10,5(/))                                         OTCP 305
001127        NXX = NV + 8                                                 OTCP 306
001124        TUNITS(1) = DMUMVM(NV)                                       OTCP 307
001126        TUNITS(2) = DMUMVM(NV)                                       OTCP 308
001127        TCOST(1) = DVMCST(NV)                                        OTCP 309
001131        TCOST(2) = DVMCST(NV)                                        OTCP 310
001132        PRINT 1234, IZ,NVM,DINT(IZ),DLIFE(IZ)                        OTCP 311
001146        CALL ANNUALD(NV2,IZ,1,TUNITS,TCOST,DINT,DLIFE,               OTCP 312
             X UNTNAM(1,9), TNAME(1,NXX), ANNCST)                          OTCP 313
001165        PRINT 1234, IZ,NVM,DINT(IZ),DLIFE(IZ)                        OTCP 314
001201    617 CONTINUE                                                     OTCP 315
C                                                                          OTCP 316
C*************************************************************************OTCP 317
C                                                                          OTCP 318
C     COMPUTE TAX LOSSES                                                   OTCP 319
C                                                                          OTCP 320
C*************************************************************************OTCP 321
C                                                                          OTCP 322
001204        CALL TAXLOSS (GROUP3 )                                       OTCP 323
C                                                                          OTCP 324
C*************************************************************************OTCP 325
C                                                                          OTCP 326
C     COMPUTE GRAND TOTALS                                                 OTCP 327
C                                                                          OTCP 328
C*************************************************************************OTCP 329
C                                                                          OTCP 330
```

```
RUN VERSION 2.7 --PSR LEVEL 20A--                    OTHRCAP          06/30/72

001204          CALL ANSUMI(NVM+6, ANACST, 4, ANNCS, ASUMR)          OTCP 331
001213          OTCAPC =ASUMR(1)+3)                                  OTCP 332
001215          RETURN                                               OTCP 333
001216          END                                                  OTCP 334
```

```
RUN VERSION 2.7 --PSR LEVEL 20A--                    OTHRCAP          06/30/72

START OF TEMPORARIES
001527

START OF LABELS
001524

UNUSED COMPILER SPACE
122700
```

```
RUN VERSION 2.7 --PSR LEVEL 20A--                    OTHRCAP          06/30/72

OTHRCAP

SUBPROGRAM LENGTH
001012

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
72    = 000042    73    = 000174    74    = 000411    75    = 000731
520   = 000017    522   = 000022    524   = 001221    526   = 001227
528   = 001276    530   = 001446    531   = 001455    532   = 001505
540   = 000133    542   = 000136    544   = 001301    545   = 001310
546   = 001344    550   = 000254    555   = 000260    560   = 000260
565   = 000274    568   = 001354    569   = 001361    572   = 001427
576   = 001436    612   = 001851    615   = 001055    619   = 001514
900   = 001274    919   = 001516    1234  = 001512

BLOCK NAMES AND LENGTHS
RD1   = 000307    HUW   = 034004    DC1   = 000700    DC5   = 000205
DC7   = 000730    UW13  = 000127    LNT   = 000000

VARIABLE ASSIGNMENTS
ANCH   = 000032C04 ANNCS  = 000000C05 ANNCST = 000050C05 ASSRAT = 000264C02
ASUMR  = 000000C04 AUTOCH = 021415C02 BACKNF = 031470C02 BACKON = 012575C02
DACRAT = 021422C02 DALNMI = 000004C07 DCNCFN = 000070C02 DFABE  = 033746C02
DDLKST = 010470C02 DGLNMI = 000001C07 DGOPC  = 033750C02 DGWDTH = 000050C02
DILNMI = 000032C07 DINT   = 000150C01 DLIFF  = 000120C01 DLNCAP = 000024C02
DMAINT = 000211C02 DMVLDF = 021307C02 DNOISE = 000274C02 DNUMVM = 000257C02
DOFLMC = 028742C02 DOFRFL = 022222C02 DONPFL = 001410C02 DPCVCM = 021434C02
DPOLYD = 021273C02 DVRDBF = 020011C02 DPRMBF = 016546C02 DPTOIP = 021757C02
DTCAPC = 001411    DTLNMI = 000125C06 DTHOPC = 033633C02 DTRTMI = 000174C06
DTYDA  = 000176C00 DVDLAY = 021301C02 DVMCAP = 000012C02 DVMCST = 001551
DVMLDF = 000017C02 DVMSUC = 000223C02 DVMUC  = 000253C02 DVMTPT = 021412C02
DVOPUC = 033035C02 DWATCC = 000103C02 DVARDS = 000110C02 DYLNMI = 000085C07
FLOCAP = 033757C02 GHOUPJ = 000000C03 GROUPJ = 033360C03 HACRAT = 021556C02
HLNDIM = 021402C02 HMCHAD = 021430C02 HOPDAT = 021030C02 HPCVCH = 021530C02
HVOPCS = 021440C02 HWTOPC = 021054C02 IDACCD = 000271C01 IPOLCD = 000741C01
IZ     = 000235C01 JCD    = 001571    JMD    = 001572    JSPANT = 000000C01
K      = 001577    L      = 001000    LC     = 001570    M      = 001576
NP     = 001607    NPRNWS = 021254C02 NTX    = 001002    NUMDA  = 021255C02
NV     = 001601    NVC    = 000276C01 NVM    = 000215C01 NVP    = 001605
NVS    = 001604    NVT    = 001603    NVZ    = 001006    NXX    = 001610
NY     = 001573    NYD    = 001575    NYM    = 000223C01 NYT    = 001574
PARCKS = 000237C01 POLRAT = 021544C02 RESOFN = 001354C02 SCALEF = 000000C02
TAXRAT = 000270C02 TCHST  = 001563    TITLE1 = 000000C04 TITLE2 = 000020C04
TNAME  = 000111C04 TULNMI = 000000C07 TRLNMI = 000003C07 TUNITS = 001556
ULANDC = 000031C02 UNTNAM = 000034C04 USTRUC = 000044C02 UTRAKP = 000043C02
XACRAT = 021402C02 XULLAY = 021362C02 XNPCU  = 021642C02 XOPTCD = 022642C02
XPCVCM = 021500C02 XVTAIP = 021342C02 XSYSCM = 021315C02 XVMTPC = 021408C02
Z      = 033277C02

START OF CONSTANTS
001217
```

```
RUN VERSION 2.1 --PSR LEVEL 29A--                                    06/30/72

            FORTRAN VI SUBROUTINE OSORT (GROUP3)                      OMOSORT2
000006      DIMENSION GROUP3(32,5,11), AVRG1(5,6,12),AVRG2(5,6,12)    OMOSORT3
000007      DIMENSION UNT5(5,6,12), GRPCN3(32,5,11), PCT5(5,6,12)     OMOSORT4
000008      DIMENSION UNIT1(5,6,12), UNIT4(5,6,12), UNIT7(5,6,12),UNIT8(5,6,
           *          12)                                             OMOSORT6
000003      COMMON /AN/ UNIT1, UNIT4                                  OMOSORT7
000009      COMMON/HOI/ IPARDT(199)                                   OMOSORT8
000003      COMMON/SVR/HSAVDT(100,2), SHRYDT(320,5)                   OMOSORT9
000009      COMMON/BM/MNINDX                                          OMOSORT10
000004      EQUIVALENCE (UNIT1(1),UNIT7(1),UNT5(1)),(UNIT4(1),UNIT8(1),PCT5(1
           * ) )                                                      OMOSORT12
000003      MM=MMINDX                                                 OMOSORT13
C                                                                     OMOSORT14
000009      DO 9A6 IS=1,8                                             OMOSORT15
000010      JNK=4*(IS-1)                                              OMOSORT16
C                                                                     OMOSORT17
000017      DO 9A2 IA=1,4                                             OMOSORT18
00001A      DO 9A2 IR=1,5                                             OMOSORT19
000021      DO 9A2 IC=1,11                                            OMOSORT20
000024      JNM=JNK+IA                                                OMOSORT21
000026  9A2 UNT5(IA,IR,IC)=GROUP3(JNM,IR,IC)                         OMOSORT22
000045      CALL SUM (3,4,10,UNT5)                                    OMOSORT23
000058      CALL PMCNT (3,4,10,UNT5,PCT5 )                            OMOSORT24
000054      DO 9A5 JK=1,4                                             OMOSORT25
000061      DO 9A4 JL=1,5                                             OMOSORT26
000064      DO 9A4 JM=1,11                                            OMOSORT27
000067      JM5=JNK+JK                                                OMOSORT28
000071      AVRG2(JM5,JL,JM)=UNT5 (JK,JL,JM )                         OMOSORT29
000101  9A4 AVRCN3(JM5,JL,JM) = PCT5(JK,JL,JM)                       OMOSORT30
000120  9A5 CONTINUE                                                  OMOSORT31
000123      DO 13U JK=1,4                                             OMOSORT32
00012A      DO 13U JL=1,5                                             OMOSORT33
000131      DO 13U JM=1,11                                            OMOSORT34
000134      UNIT1(JK,JL,JM)=GROUP3(JK,JL,JM)                          OMOSORT35
00014A  13A UNIT4(JK,JL,JM)=GROUP3(JK+12,JL,JM)                       OMOSORT36
000163      CALL AVRG(4,5,11,UNIT4,UNIT1,AVRG1)                       OMOSORT37
000167      DO 1A0 JK=1,4                                             OMOSORT38
000174      DO 1A0 JL=1,5                                             OMOSORT39
000177      DO 1A0 JM=1,11                                            OMOSORT40
000202      UNIT7(JK,JL,JM)=GROUP3(JK+24,JL,JM)                       OMOSORT41
000217  1A0 UNIT8(JK,JL,JM)=GROUP3(JK+28,JL,JM)                       OMOSORT42
000231      CALL AVRG(4,5,11,UNIT8,UNIT7,AVRG2)                       OMOSORT43
000239      SHRYDT (201,MM) = GROUP3(28,5,11)                         OMOSORT44
000242      SHRYDT (202,MM) = AVRG1(4,5,11)                           OMOSORT45
000245      SHRYDT (203,MM) = GROUP3 (32,5,11)                        OMOSORT46
000256      SHRYDT(8,MM)=GROUP3(4,5,11)                               OMOSORT47
000253      SHRYDT(77,MM)=GROUP3(16,5,11)                             OMOSORT48
000256      SHRYDT(78,MM)=GROUP3(12,5,11)                             OMOSORT49
000261      SHRYDT(79,MM)=GROUP3(20,5,11)                             OMOSORT50
000264      IF(IPARDT(16) .EQ. 1) GO TO 200                           OMOSORT51
000266      CALL PMNTAM (GROUP3,GRPCN3,AVRG1,AVRG2 )                  OMOSORT52
000270  200 CONTINUE                                                  OMOSORT53
000270      RETURN                                                    OMOSORT54
000271      END                                                       OMOSORT55
```

```
RUN VERSION 2.1 --PSR LEVEL 29A--                    OSORT     06/30/72

OSORT

SUBROUTINE LENGTH
000219

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
13A   =  000144   16A   =  000212   200   =  000270   902   =  000024
9A4   =  000120   9A4   =  000101

BLOCK NAMES AND LENGTHS
AN    =  001320   HOI   =  000387   SVR   =  003410   MM    =  000001

VARIABLE ASSIGNMENTS
AVRG1 =  000320   AVRG2 =  001070   GRPCN3 =  001040   IA    =  005203
IR    =  005204   IC    =  005205   IPARDT =  000000C02 IS    =  005201
JK    =  005207   JL    =  005210   JM    =  005211   JNK   =  005202
JNM   =  005206   JM5   =  005212   MM    =  005200   MMINDX =  000000C04
PCT5  =  000550C01 HSAVDT =  000080C03 SHRYDT =  000310C03 UNIT1 =  000000C01
UNIT4 =  000550C01 UNIT7 =  000080C01 UNIT8 =  000550C01 UNT5  =  000000C01

START OF CONSTANTS
000279

START OF TEMPORARIES
000277

START OF INTRFCTS
000367

UNUSED COMPILER SPACE
127280
```

```
RUN VERSION 2.3 --PSW LEVEL 298--                                                    06/30/72

              FORTRAN VI SUBROUTINE TXRLOSS (GROUP3 )                                  DMTXL052
000003        INTEGER DLIFE, DINT                                                      DMTXL053
000003        INTEGER PARCKS                                                           DMTXL054
000003        REAL NPKHRS , NUMDA                                                      DMTXL055
000003        DIMENSION GROUP3(32,5,11)                                                DMTXL056
000003        DIMENSION TXLOSS(5,5,12)                                                 DMTXL057
000003        COMMON/RDI/      JSPRNT(80)     , DLIFE(30)      , DINT(30)      ,       DMTXL058
              *   LECODE       , NVM          , NLM            , NSM           ,       DMTXL059
              *   NIM          , NTM          , NTSM           , NTM           ,       DMTXL010
              *   NM           , NZ           , IPRINT         , IRV           ,       DMTXL011
              *   IPOLCD(4,2,3), IDACCD(3)    , MHV            , NDTM          ,       DMTXL012
              *   NVC(8)                                                               DMTXL013
000003        COMMON/VAR/      SCALEF(10)     , DVMCAP(5)      , DVMLDF(5)     ,       DMTXL014
              * DLNCAP(5)     , ULANDC(10)     , UTRANC         , USTRUC(4)     ,       DMTXL015
              * DWNTMI(6)     , DCMCEN(11)     , DWATCC(5)      , DYARDS(5,13)  ,       DMTXL016
              * DMAINT(10)    , DVMSUC(4,6)    , DVMUC(4)       , DNUMVM(5)     ,       DMTXL017
              * ASSRAT(6)     , TAXRAT(4)      , DNOISE(20,14,2), RESDEN(4,7)   ,       DMTXL018
              * DDMPFL(225,4,6), DBLRST(225,5) , BACKON(225,3,3), DPKHRF(225,3) ,       DMTXL019
              * DPKHRS(225,3)  ,                                                       DMTXL020
              * NPKHRS         , NUMDA          , DPRPCT         , DPTRIP(4,3)   ,       DMTXL021
              * DPDLTU(2,3)    , DMVLDF(2,3)    , DMVLDF(2,3)    , XSYSCM(4,5)   ,       DMTXL022
              * XPRPCT         , XPTRIP(4,4)    , XDELAY(2,4,2)  , HLNDIR(2,2)   ,       DMTXL023
              * EVMTPC(4)      , DVMTPC(3)      , AUTOCM(11)     , MMCHAR(4)     ,       DMTXL024
              * DPCVCM(3,6,2)  , XPCVCM(4,6)    , HPCVCM(2,6)    , POLRAT(3,3)   ,       DMTXL025
              * DMSTAC         , MACRAT(2,5,2)  , MACRAT(4,4)    , DACRAT(2,3)   ,       DMTXL026
              * MDPRAT(2,4)    , MVOPCS(2,6)    , MHTOPC(2,3)    , XOPCU(4,7,4)  ,       DMTXL027
              * XMPICD(4,7,4)  ,DDPPFL(150,4,4),DDPLRC(150,3,3),MACKOF(150,3,2),       DMTXL028
              * DVIMPC         , DCCOPC         , DMOPC          , Z(4)          ,       DMTXL029
              * DVOPUC(6,9,4)  , DTROPC(11,7)   , DOOPC(7)       , FLOCAP(5)     ,       DMTXL030
              * DFARE(4,2,2)                                                            DMTXL031
000003        COMMON/SVR/ HSAVDT(100,2), SHRYDT(320,5)                                 DMTXL032
000009        COMMON/MM/MMINDX                                                         DMTXL033
000009        MMMMINDX                                                                 DMTXL034
              C                                                                        DMTXL035
              C       MAIN TAX LOSS COMPUTATIONS                                       DMTXL036
              C                                                                        DMTXL037
              C       COMPUTE TAX LOSS                                                 DMTXL038
000009                DO 1220  N = 20,31                                               DMTXL039
000010                M = N - 28                                                       DMTXL040
000012                DO 1220  K = 1,4                                                 DMTXL041
000015                DO 1220  L = 1,10                                                DMTXL042
000020        1220 TXLOSS(M,K,L) = ((GROUP3(N,K,L)*ASSRAT(M)) / 1000.) * TAXRAT(M)     DMTXL043
000047                CALL SUM (5,4,10,TXLOSS)                                         DMTXL044
              C                                                                        DMTXL045
000052                SHRYDT (284,MM) = TXLOSS (4,5,11)                                DMTXL046
000054                IF(KPRNT(17) .EQ. 1)  GO TO 1249                                 DMTXL047
              C       PRINT TAX LOSS TABLE                                             DMTXL048
000061                PRINT 1240                                                       DMTXL049
000064        1240 FORMAT(///1H1,3HTAX LOSS FROM DUAL MODE CONSTRUCTION,////)          DMTXL050
000064                PRINT 1245                                                       DMTXL051
000070        1245 FORMAT(1H ,15H REOG CONS LAND,9X,6HMARKET,6X,6HASSESS,7X,3HTAX,12X  DMTXL052
                  *,3HTAX/1H ,15H AREA CODE   USE,10X,5HVALUE,5X,5HRATIO,6X,6HRATE,11X DMTXL053
                  *,4HLOSS//)                                                          DMTXL054
                                                                                       DMTXL055
```

```
RUN VERSION 2.3 --PSW LEVEL 298--                                                    TXRLOSS   06/30/72

000070                ICNT = 0                                                         DMTXL056
000071                DO 1248  M =  1,4                                                DMTXL057
000074                N = M + 28                                                       DMTXL058
000100                DO 1248  K = 1,4                                                 DMTXL059
000103                DO 1248  L = 1,11                                                DMTXL060
000104                PRINT 1250, M,K,L,GROUP3(N,K,L),ASSRAT(M),TAXRAT(M),TXLOSS(M,K,L) DMTXL061
000147                ICNT = ICNT + 1                                                  DMTXL062
000149                IF (ICNT.EQ.44) GO TO 1270                                       DMTXL063
000147                GO TO 1248                                                       DMTXL064
000150        1270 PRINT 1240                                                          DMTXL065
000154                PRINT 1245                                                       DMTXL066
000160                ICNT = 0                                                         DMTXL067
000161        1248 CONTINUE                                                            DMTXL068
000171        1250 CONTINUE                                                            DMTXL069
000171        1280 FORMAT(1H1,6/)                                                      DMTXL070
000171        1250 FORMAT(1H ,3I5,F15.0,F10.3,F10.2,F15.0)                             DMTXL071
000171                RETURN                                                           DMTXL072
000172                END                                                              DMTXL073
```

```
RUN VERSION 2.3 --PSW LEVEL 298--                                                    TXRLOSS   06/30/72

TXRLOSS

SUBPROGRAM LENGTH
001024

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
1220  =  000070    1245  =  000210    1248  =  000161
1249  =  000171    1250  =  000231    1270  =  000150    1280  =  000227

BLOCK NAMES AND LENGTHS
RDI   =  000307    RVR   =  034004    SVR   =  003410    MM    =  000001

VARIABLE ASSIGNMENTS
ASSRAT = 000244C02 AUTOCM = 021415C02 BACKOF = 031478C02 BACKON = 012575C02
DACRAT = 021422C02 DCMCEN = 000070C02 DFARE  = 033746C02 DDLRST = 010430C02
DOOPC  = 033746C02 DVMDTM = 000850C02 DINT   = 000156C01 DLIFE  = 000120C01
DLNCAP = 000074C02 DMAINT = 000211C02 DMVLDF = 021307C02 DNOISE = 000274C02
DNUMVM = 000257C02 DVFLRC = 026762C02 DOPPFL = 022222C02 DOMPFL = 001410C02
DPCVCM = 021474C02 DMHLYD = 021273C02 DPKDRF = 020011C02 DPKHRF = 016546C02
DPTRIP = 021297C02 DTROPC = 033633C02 DVDLAY = 021301C02 DVMCAP = 000012C02
DVMLDF = 000017C02 DVMSUC = 000223C02 DVMUC  = 000253C02 DVMTPC = 021412C02
DVOPUC = 033303C02 DWATCC = 000103C02 DYARDS = 000110C02 FLOCAP = 033757C02
HACRAT = 021596C02 HLNDIR = 021402C02 HMCHAR = 021430C02 HOPRAT = 021630C02
HPCVCM = 021430C02 HVOPCS = 021404C02 HVTOPC = 021654C02 ICNT   = 001024
IDACCD = 000271C01 IPOLCD = 000241C01 JSPRNT = 000080C01 K      = 001022
L      = 000103    M      = 001021    MM     = 001017    MMINDX  = 000000C06
N      = 000104    NPKHRS = 021254C02 NUMDA  = 021255C02 NVC    = 000270C01
PARCKS = 000277C01 PULRAT = 021544C02 RESDEN = 001354C02 HSAVDT  = 000000C03
SCALEF = 000000C02 SHRYDT = 000310C03 TAXRAT = 000270C02 TXLOSS  = 000000
ULANDC = 000017C02 USTRUC = 000440C03 XACRAT = 000270C02 XDELAY  = 021362C02
XOPCU  = 021442C02 XOPICD = 022042C02 XPCVCM = 021540C02 XPTRIP  = 021342C02
ASYSCM = 021314C02 AVMTPC = 021400C02 Z      = 033277C02

START OF CONSTANTS
000173

START OF TEMPORARIES
000239

START OF INDIRECTS
000247

UNUSED COMPILER SPACE
126780
```

```
RUN VERSION 2.9 --PSW LEVEL 298--                                      06/30/72

            SUBROUTINE SUM(I,J,M,UNIT)                                 DMSUM  2
      C
      C     SUBROUTINE TO EVALUATE THE SUMMATIONS.                     DMSUM  3
      C                                                                DMSUM  4
      C                                                                DMSUM  5
      C                                                                DMSUM  6
000007      DIMENSION UNIT(9,6,12)                                     DMSUM  7
      C                                                                DMSUM  8
      C     SUMMATION OF ROWS.                                         DMSUM  9
000007      DO 45 KM=1,M                                               DMSUM 10
000010      DO 45 KI=1,I                                               DMSUM 11
000011      UNIT(KI,J+1,KM)=0.0                                        DMSUM 12
00001A      DO 45 KJ=1,J                                               DMSUM 13
000017   45 UNIT(KI,J+1,KM)=UNIT(KI,J+1,KM)+UNIT(KI,KJ,KM)             DMSUM 14
      C                                                                DMSUM 15
      C     SUMMATION OF COLUMNS.                                      DMSUM 16
000037      DO 46 KM=1,M                                               DMSUM 17
00004A      DO 46 KJ=1,J                                               DMSUM 18
000041      UNIT(I+1,KJ,KM)=0.0                                        DMSUM 19
00004A      DO 46 KI=1,I                                               DMSUM 20
000047   46  UNIT(I+1,KJ,KM)=UNIT(I+1,KJ,KM)+UNIT(KI,KJ,KM)           DMSUM 21
      C                                                                DMSUM 22
      C     SUMMATION TOWARDS REAR.                                    DMSUM 23
      C                                                                DMSUM 24
000067      DO 47 KJ=1,J                                               DMSUM 25
00007A      DO 47 KI=1,I                                               DMSUM 26
000071      UNIT(KI,KJ,M+1)=0.0                                        DMSUM 27
000074      DO 47 KM=1,M                                               DMSUM 28
000079   47 UNIT(KI,KJ,M+1)=UNIT(KI,KJ,M+1)+UNIT(KI,KJ,KM)            DMSUM 29
      C                                                                DMSUM 30
      C     SUMMATION OF ROWS AND COLUMNS                              DMSUM 31
      C                                                                DMSUM 32
000117      DO 51 KM=1,M                                               DMSUM 33
00012A      UNIT(I+1,J+1,KM)=0.0                                       DMSUM 34
000124      DO 51 KI=1,I                                               DMSUM 35
00012A   51 UNIT(I+1,J+1,KM)=UNIT(I+1,J+1,KM)+UNIT(KI,J+1,KM)         DMSUM 36
      C                                                                DMSUM 37
      C     SUMMATION OF ROWS AND REAR ELEMENTS.                       DMSUM 38
      C                                                                DMSUM 39
000144      DO 54  KI=1,I                                              DMSUM 40
00014A      UNIT(KI,J+1,M+1)=0.0                                       DMSUM 41
000151      DO 54  KM=1,M                                              DMSUM 42
000199   54 UNIT(KI,J+1,M+1)=UNIT(KI,J+1,M+1)+UNIT(KI,J+1,KM)         DMSUM 43
      C                                                                DMSUM 44
      C     SUMMATION OF COLUMNS AND THE REAR ELEMENTS.                DMSUM 45
      C                                                                DMSUM 46
000171      DO 56  KJ=1,J                                              DMSUM 47
000172      UNIT(I+1,KJ,M+1)=0.0                                       DMSUM 48
00017A      DO 56  KM=1,M                                              DMSUM 49
000200   56 UNIT(I+1,KJ,M+1)=UNIT(I+1,KJ,M+1)+UNIT(I+1,KJ,KM)         DMSUM 50
      C                                                                DMSUM 51
      C     GRAND TOTAL                                                DMSUM 52
      C                                                                DMSUM 53
000214      UNIT(I+1,J+1,M+1)=0.0                                      DMSUM 54
000227      DO 57 KM=1,M                                               DMSUM 55
```

```
RUN VERSION 2.9 --PSW LEVEL 298--                  SUM        06/30/72

000223   57 UNIT(I+1,J+1,M+1)=UNIT(I+1,J+1,M+1)+UNIT(I+1,J+1,KM)       DMSUM 56
      C                                                                DMSUM 57
000239      RETURN                                                     DMSUM 58
000236      END                                                        DMSUM 59
```

```
RUN VERSION 2.9 --PSW LEVEL 298--                  SUM        06/30/72

SUM

SUBPROGRAM LENGTH
000293

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
45     =  000017    46     =  000047    47     =  000077    51     =  000124
54     =  000199    56     =  000200    57     =  000293

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS
KI     =  000291    KJ     =  000292    KM     =  000258

START OF CONSTANTS
000240

START OF TEMPORARIES
000241

START OF INDIRECTS
000248

UNUSED COMPILER SPACE
127500
```

```
                SUBROUTINE DISPLAC
C
C*********************************************************************
C
C       DISPLACEMENT SECTION
C
C*********************************************************************
C
000007          INTEGER NLIFE, NINT
000007          INTEGER PANCK
000007          REAL  NPKHRS , MINDA
000007          COMMON/G9/ GROUP9(5,6,12)
000002          COMMON/SO/ DUSGST(400,22)
000007          COMMON/HDI/        JSPANT(80)     , DLIFE(30)     , DINT(30)      ,
                X   LECODE         , NYM          , NLM           , NSM           ,
                X   NTM            , NTSM         , NTM           , NTM           ,
                X   NDM            , IZ           , JK            , PARCK3(6)     ,
                X   NW             , NX           , IPRINT        , IRV           ,
                X   IPOLCD(4,2,1), IDACCD(3)      , NHV           , NOTM          ,
                X   NVC(9)
000002          COMMON/HDR/        SCALEF(10)     , DVMCAP(5)     , DVMLDF(5)     ,
                X DLNCAP(5)        , VLANDC(10)   , UTRANC        , USTRUC(4)     ,
                X DANTNH(16)       , DCNCEN(11)   , DWATCC(5)     , DYARDS(5,13)  ,
                X DWAINT(10)       , DVMBUC(4,6)  , DVMUC(4)      , DNUMVH(5)     ,
                X ASSHAT(4)        , TARRAT(4)    , DNOISE(20,14,2), RESDEN(4,7)  ,
                X DPMDFL(225,4,4), DBLKRT(225,5)  , BACKDN(225,3,3), DPKHRF(225,3),
                X DPKHRF(225,3)    ,
                X NPKHRS           , NUMDA        , DPRPCT        , DPTRIP(4,3)   ,
                X DPDLYU(2,3)      , DVDLAY(2,3)  , DMVLUF(2,3)   , RSYSCH(4,5)   ,
                X XPRPC(           , BPYRIP(4,4)  , XDELAY(2,4,2) , MLNDIR(2,2)   ,
                X XVHTPC(4)        , DVHTPC(3)    , AUTOCH(11)    , NMCHAR(4)     ,
                X DPCVCH(3,4,2)    , APCVCH(4,4)  , NPCVCH(2,4)   , POLRA(3,3)    ,
                X DHSTAC           , HACRAT(2,5,2), XACRAT(2,8)   , DACRAT(2,3)   ,
                X HOPRAT(2,4)      , HVDPCS(2,4)  , NHVDPC(2,3)   , RDPCDU(4,7,4) ,
                X XDPICU(4,7,4)    ,DDOFPL(150,4,4),DDOFLKC(150,3,3) ,BACKOF(150,3,2),
                X DYUDPC           , DCCDPC       , DMDPC         , Z(4)          ,
                X DVDMUC(4,9,4)    , DTRDWC(11,7) , DBOPC(7)      , FLOCAP(5)     ,
                X DPARE(4,2,2)
000007          COMMON/SVA/ HSAVDT(100,2), SWAYDT(320,5)
000007          COMMON/MW/WINDX
000007          MMWWINDX
C
C       COMPUTE DISPLACEMENT AREA PER SEGMENT
000004          DO 1000 NS = 1,NSM
C       IF CONSTRUCTION IS TUNNEL, THEN THERE IS NO DISPLACEMENT
C       IF LAND USE CODE IS 9 OR 10 (OPEN SPACE), THEN NO DISPLACEMENTS
000005          IF (DGSGST(NS,2) .EQ. 4.0)  GO TO 1100
000007          IF (DGSGST(NS,3) .EQ. 9.0 .OR. DGSGST(NS,3) .EQ. 10.0)  GO TO 1100
C       IF THERE IS NO EXISTING SYSTEM ROW, THEN DISPLACEMENT AREA EQUALS
C       THE  ROW AREA OF THE SEGMENT
000017          IF (DGSGST(NS,18) .EQ. 0.0)  GO TO 1095
C       IF THE EXISTING SYSTEM ROW IS GREATER THAN THE NEW SYSTEM ROW,
C       THEN THERE IS NO NEW DISPLACEMENT
00002D          IF (DGSGST(NS,18) .GT. DGSGST(NS,17)) GO TO 1100
```



```
C       IF THE EXISTING SYSTEM ROW IS LESS THAN THE NEW SYSTE ROW,
C       THEN THE DISPLACEMENT IS THE DIFFERENCE OF THE TWO AREAS. WE WIDTH
000024          IF (DGSGST(NS,18) .LT. DGSGST(NS,17)) W = DGSGST(NS,17) - DGSGST
                X (NS,18)
C       DISPLACEMENT AREA EQUALS W * SEGMENT LENGTH
00003D          DGSGST(NS,21) = W * DGSGST(NS,15) = (5280./43560.)
000036          GO TO 1000
000035    1095  DGSGST(NS,21) = DGSGST(NS,4)
000037          GO TO 1000
000040    1100  DGSGST(NS,21) = 0.0
000042    1000  CONTINUE
C       INITIALIZE GROUP9
000044          DO 1105  K=1,5
000046          DO 1105  L=1,5
000047          DO 1105  M=1,5
000050    1105  GROUP9(K,L,M) = 0.0
C       GROUP DISPLACEMENT AREAS BY GEOG AREA, CONS, TYPE, * RESID DENSITY
000063          DO 1110  NS = 1,NSM
000064          K = DGSGST(NS,1)
000066          L = DGSGST(NS,2)
000070          M = DGSGST(NS,20)
000072          IF(M.EQ.0)  GO TO 1110
000073          GROUP9(K,L,M) = GROUP9(K,L,M) + DGSGST(NS,21)
000101    1110  CONTINUE
C       COMPUTE TOTAL RESID. DENSITY PER CODE
000104          DO 1125  JRD = 1,4
000108    1125  RESDEN(JRD,7) = RESDEN(JRD,1) + RESDEN(JRD,2) + RESDEN(JRD,3)
C       PRINT RESIDENTIAL DENSITIES (PERSONS/ACRE)
00011K          IF(JSPRNT(2)) .EQ. 1) GO TO 01
000117          PRINT 1150
000129    1150  FORMAT(1H1,U/,* RESIDENTIAL DENSITIES --(PERSONS PER ACRE)*,///)
000129          PRINT 1152
00012A    1152  FORMAT(1H , *  RES.DEN.*, 18X, *ETHNIC BREAKDOWN*, 20X, *INCOME BRD
                X*ARDOWN*, 17X, *TOTAL*, /1H , 6X, *CODE*, 7X, *WHITE*, 7X,*BLACK*,
                X 7X, *OTHER*, 9X, *LOW*, 9X, *MED*, 8X, *HIGH*, 5X, *DENSITY*,/)
00012A          DO 1155  JRD = 1,4
00013D    1154  FORMAT(1H , I10, 7F12.2,/)
00013D    1155  PRINT 1154, JRD, (RESDEN(JRD,JRDC), JRDC= 1,7)
00014A    R1    CONTINUE
C       PRINT MEANINGS
00014A          IF(JSPRNT(22) .EQ. 1) GO TO R2
00015A          PRINT 1174
000154    1174  FORMAT(1H1,///,*  PERSONS DISPLACED BY UN GUIDEWAY PROPER*,///)
000154          PRINT 1177
00016A    1177  FORMAT(1H ,4X,*ETHNIC BREAKDOWN*, 29X, *INCOME BREAKDOWN*, 19X, DISPL100
                X *TOTAL* /1H , * GEOG CONS RES.*, 4X, *DISPLAC.*, 5X, *WHITE  PCT*DISPL101
                X *, 4X, *BLACK  PCT*, 5X, *OTHER  PCT*, 7X,*LOW  PCT*, 7X,*MED  PCTDISPL102
                X *, 4X, *HIGH  PCT  DISPLAC.*,/1H , * AREA TYPE DENS* , 8X,*AREA*,DISPL103
                X 6( *    (NO. OF   OF*), *    (NO. OF*,/1H,  3(* CODE*), 5X,*(ACRES)DISPL104
00016A    R2    CONTINUE                                                        DISPL105
C       COMPUTE NUMBER OF PEOPLE DISPLACED PER CATAGORY AND PERCENT OF          DISPL106
C       TOTAL AND PRINT PER GROUP CATAGORY                                      DISPL107
00016A          SUM1 = 0.0                                                      DISPL108
00016A          SUM2 = 0.0                                                      DISPL109
                                                                                DISPL110
```

```
000162          SUM3 = 0.0                                          DISPL111
000163          SUM4 = 0.0                                          DISPL112
000164          SUM5 = 0.0                                          DISPL113
000165          SUM6 = 0.0                                          DISPL114
000166          SUM7 = 0.0                                          DISPL115
000167          SUM8 = 0.0                                          DISPL116
000170          DO 1145  K = 1,3                                    DISPL117
000171          DO 1145  L = 1,4                                    DISPL118
000172          DO 1145  M = 1,4                                    DISPL119
000173          IF (GROUP9(K,L,M) .EQ. 0.0)  GO TO 18               DISPL120
000177          A = GROUP9(K,L,M) * RESDEN(M,1)                     DISPL121
000204          B = GROUP9(K,L,M) * RESDEN(M,2)                     DISPL122
000210          C = GROUP9(K,L,M) * RESDEN(M,3)                     DISPL123
000214          D = GROUP9(K,L,M) * RESDEN(M,4)                     DISPL124
000220          E = GROUP9(K,L,M) * RESDEN(M,5)                     DISPL125
000224          F = GROUP9(K,L,M) * RESDEN(M,6)                     DISPL126
000230          G = GROUP9(K,L,M) * RESDEN(M,7)                     DISPL127
000234          SUM1 = SUM1 + GROUP9(K,L,M)                         DISPL128
000240          SUM2 = SUM2 + A                                     DISPL129
000247          SUM3 = SUM3 + B                                     DISPL130
000244          SUM4 = SUM4 + C                                     DISPL131
000246          SUM5 = SUM5 + D                                     DISPL132
000250          SUM6 = SUM6 + E                                     DISPL133
000252          SUM7 = SUM7 + F                                     DISPL134
000254          SUM8 = SUM8 + G                                     DISPL135
000256          AP=  A / G  * 100.                                  DISPL136
000260          BP=  B / G  * 100.                                  DISPL137
000262          CP=  C / G  * 100.                                  DISPL138
000264          DP=  D / G  * 100.                                  DISPL139
000266          EP=  E / G  * 100.                                  DISPL140
000270          FP=  F / G  * 100.                                  DISPL141
000272          GO TO 15                                            DISPL142
000273       18 A  = 0.0                                            DISPL143
000274          AP = 0.0                                            DISPL144
000276          B  = 0.0                                            DISPL145
000276          BP = 0.0                                            DISPL146
000277          C  = 0.0                                            DISPL147
000300          CP = 0.0                                            DISPL148
000301          D  = 0.0                                            DISPL149
000302          DP = 0.0                                            DISPL150
000303          E  = 0.0                                            DISPL151
000304          EP = 0.0                                            DISPL152
000305          F  = 0.0                                            DISPL153
000306          FP = 0.0                                            DISPL154
000307          G  = 0.0                                            DISPL155
000310       15 CONTINUE                                            DISPL156
000310          IF(JSPRNT(22) .EQ. 1)  GO TO 1145                   DISPL157
000312          PRINT 1135, K,L,M, GROUP9(K,L,M), A,AP,B,BP,C,CP,D,DP,E,EP,F,FP,G DISPL158
000364    1135  FORMAT(1H ,3I5, F12.2, 8(F10.0,F5.0), F10.0)        DISPL159
000364    1145  CONTINUE                                            DISPL160
000372          PC2 = SUM2 / SUM8 * 100.                            DISPL161
000374          PC3 = SUM3 / SUM8 * 100.                            DISPL162
000377          PC4 = SUM4 / SUM8 * 100.                            DISPL163
000401          PC5 = SUM5 / SUM8 * 100.                            DISPL164
000403          PC6 = SUM6 / SUM8 * 100.                            DISPL165
```



```
000405          PC7 = SUM7 / SUM8 * 100.                            DISPL166
000407          WRITE (291,MM) *  SUM1                              DISPL167
000413          WRITE (292,MM) *  SUM2                              DISPL168
000416          WRITE (293,MM) *  SUM3                              DISPL169
000421          WRITE (294,MM) *  SUM4                              DISPL170
000424          WRITE (295,MM) *  SUM5                              DISPL171
000427          WRITE (296,MM) *  SUM6                              DISPL172
000432          WRITE (297,MM) *  SUM7                              DISPL173
000434          WRITE (298,MM) *  SUM8                              DISPL174
000437          IF (JSPRNT(22) .EQ. 1)  GO TO 2A2                   DISPL175
000441          PRINT 1137, SUM1, SUM2, PC2, SUM3, PC3, SUM4, PC4, SUM5, PC5,  DISPL176
                1 SUM6, PC6, SUM7, PC7, SUM8                        DISPL177
000500    1137  FORMAT(/,1H ,*TOTALS*,9X,F12.2, 6(F10.0,F5.0) F10.0)  DISPL178
000500    2A2   CONTINUE                                            DISPL179
000500          RETURN                                              DISPL180
000501          END                                                 DISPL181
```



```
DISPLAC

SUBPROGRAM LENGTH
000734

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
10    - 000273   15    - 000318   81    - 000146   82    - 000160
282   - 000548   1000  - 000042   1095  - 000035   1100  - 000040
1105  - 000050   1110  - 000101   1125  - 000185   1126  - 000553
1127  - 000542   1135  - 000631   1137  - 000638   1145  - 000364
1150  - 000512   1152  - 000522   1154  - 000547

BLOCK NAMES AND LENGTHS
GO    - 000590   SW    - 021140   WDI   - 000307   RDA   - 034004
SVR   - 003410   MM    - 000001

VARIABLE ASSIGNMENTS
A     - 000712   AP    - 000721   ASSRAT - 000264C04 AUTOCM - 021415C04
B     - 000713   BACKOF - 316730C04 BACKON - 012575C04 BP    - 000722
C     - 000714   CP    - 000723   D     - 000715   DACRAT - 021622C04
DCNCEW - 000730C04 DFAWE - 033764C04 DGLKST - 010430C04 DGOPC - 033750C04
DGSGST - 000000C02 DGWNTH - 000050C04 DINT  - 000150C03 DLIFE - 000120C03
DLNCAP - 000074C04 DMAINT - 002110C04 DMVLDF - 021307C04 DNOISE - 000274C04
DNUMVH - 000297C04 DOFLRC - 026702C04 DOFPFL - 022222C04 DOMPFL - 001410C04
DP    - 000724   DPCVCH - 021434C04 DPDLYO - 021273C04 DPKORF - 020011C04
DPKWRF - 010546C04 DPTRIP - 021257C04 DTROPC - 033633C04 DVOLAY - 021301C04
DVHCAP - 000012C04 DVHLDF - 000017C04 DVHSUC - 000223C04 DVHUC - 000253C04
DVHTPC - 021412C04 DVOPUC - 033303C04 DWAYCC - 000103C04 DYAMOS - 000110C04
E     - 000716   EP    - 000725   F     - 000717   FLOCAP - 033307C04
FP    - 000726   G     - 000720   GROUP9 - 000000C01 HACRAT - 021556C04
HLNDIP - 021402C04 HMCHAR - 021430C04 HMPDAT - 021030C04 HPCVCH - 021530C04
HVOPCS - 021044C04 HWYPPC - 021454C04 IDACCD - 000271C03 IPOLCD - 000241C03
JRD   - 000700   JRDC    - 000701   JSPRNT - 000000C03 K     - 000675
L     - 000676   M       - 000677   MM    - 000672   MMINDX - 000000C06
MPKWRS - 021254C04 NS    - 000673   NSW   - 000217C03 NUMDA - 021255C04
NYC   - 000276C03 PAHCK3 - 000227C03 PC2   - 000727   PC3   - 000730
PC4   - 000731   PC5     - 000732   PC6   - 000733   PC7   - 000734
POLRAT - 021544C04 RESDEN - 001354C04 RSAVDT - 000000C05 SCALEF - 000000C04
SWRYDT - 000310C05 SUM1  - 000702   SUM2  - 000703   SUM3  - 000704
SUM4  - 000705   SUM5    - 000706   SUM6  - 000707   SUM7  - 000710
SUM8  - 000711   TARRAT  - 000276C04 ULANDC - 000031C04 USTRUC - 000044C04
W     - 000674   AACRAT  - 021402C04 XDELAY - 021362C04 XOPCU - 021662C04
XOPICD - 022042C04 XPCVCH - 021500C04 XPTRIP - 021342C04 XSYSCH - 021315C04
XWNTPC - 021406C04 Z     - 033277C04

START OF CONSTANTS
000803

START OF TEMPORARIES
000644

START OF INDIRECTS
```

```
                 SUBROUTINE PRNTAR (GROUP,PRCT,AVB2,AVB1)
000007           DIMENSION GROUP(32,5,11),PRCT(36,5,11),AVB1(5,6,12),AVB2(5,6,12)
000007           COMMON/SVB/ RSAVDT(100,2), SMRYDT(320,5)
000007           COMMON/MM/MMINDX
000007           MM=MMINDX
                 PRINT LAND SUMMARIES
000010           PRINT 770
000014       770 FORMAT(1H1,40HLAND ACQUISITION AND MARKET VALUE SUMMARIES BY GEOG.
                 1AREA- CONSTRUCTION TYPE- AND LAND USE ////)
000014           PRINT 800
000020       800 FORMAT(1H ,9 GEOG CONS LAND9,(4X,9TAXABLE LAND DATA9,27X,9TOTAL LAD
                 1ND ACQUISITION DATA9,/1H , 9 AREA TYPE  USE9,   8X,9AREA  PRCNT AVD
                 2EN,COST   MARKET VALUE  PRCNT9 +8X,9AREA  PRCNT AVER.COST     R,0.W
                 3, COST  PRCNT9,                                  / 1H , 31X CODE9),
000020           1 21X   (ACRES)  TOTAL  (9/ACRE)                  (B)      TOTAL9),///)
000021           ICNT=0
000027           DO 740 N=1,4
000030           WRN
000034           DO 740 K=1,5
000037           DO 740 L=1,11
                 PRINT 792,9,K,L,GROUP (M=24,K,L),PRCT(M=24,K,L),AVB1(N,K,L),
                 1AVB2(N,K,L),PRCT(M=28,K,L),GROUP (M,K,L),PRCT(M,K,L),
                 2GROUP (M=12,K,L),PRCT(M=12,K,L)
000164           ICNT=ICNT+1
00016A           IF(ICNT.EQ.44)GO TO 776
000172           GO TO 780
000172       776 PRINT 792
00017A           PRINT 800
000202           ICNT=0
000203       780 CONTINUE
000214       772 FORMAT(1H ,31X,2(F10.2,F7.0,F10.0,F15.0,F7.0))
000214       792 FORMAT(1H1,W2)
                 PRINT DATA SUMMARIES BY GEOGRAPHICAL AREA ,CONSTRUCTION TYPE
                 AND LAND USE
000214           PRINT 780
000217       780 FORMAT(1H1,31HDATA SUMMARIES BY GEOG. AREA- CONSTRUCTION TYPE- AND
                 1LAND USE ////)
000217           PRINT 810
000221       810 FORMAT(1H ,9 GEOG CONS LAND      LANE  PRCNT9, 10X,9TRACK  PRCNT9,
                 2 8X,9STRUCT.  PRCNT9, 8X,9R.O.W.  PRCNT9, 10X,9TOTAL  PRCNT9,/1H ,
                 3 9 AREA TYPE  USE      LENGTH      DE9, 4(11X,9COST,5X,9OF9),/ 1H ,
000223           ICNT = 0
000224           DO 780  N=1,4
000232           WRN
000233           DO 780 K=1,5
000237           DO 780 L=1,11
000242           PRINT 792,9,K,L,GROUP (M=4,K,L),PRCT(M=4,K,L),GROUP (M=16,K,L),
                 2PRCT(M=16,K,L),GROUP (M=8,K,L),PRCT(M=8,K,L),GROUP (M=12,K,L),
                 3PRCT(M=12,K,L),GROUP (M=20,K,L),PRCT(M=20,K,L)
000364           ICNT = ICNT + 1
00036A           IF (ICNT.EQ.44) GO TO 784
000372           GO TO 780
000372       784 PRINT 792
```



```
PRNTAR

SUBPROGRAM LENGTH
000370

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
780   =  000594    792   =  000550    784   =  000372    768   =  000403
770   =  000417    772   =  000474    776   =  000172    780   =  000203
792   =  000592    800   =  000433    810   =  000515

BLOCK NAMES AND LENGTHS
SVB   =  003410    MM    =  000001

VARIABLE ASSIGNMENTS
ICNT  =  000543    K     =  000566    L     =  000567    M     =  000565
MM    =  000542    MMINDX =  00080C02 N     =  000564    RSAVDT =  000080C8
SMRYDT =  00031CA1

START OF CONSTANTS
000419

START OF TEMPORARIES
000554

START OF INTRFCTS
000542

UNUSED COMPILER SPACE
12468A
```



```
000376           PRINT 810
000402           ICNT=0
000403       784 CONTINUE
000414       792 FORMAT(1H ,31X,F10.2,F7.0,4(F15.0,F7.0))
000414           RETURN
000414           END
```

```
                SUBROUTINE NOISE                                        NOIS   2
C                                                                       NOIS   3
C*********************************************************************NOIS   4
C                                                                       NOIS   5
C     DM GUIDEWAY PHOPER NOISE SECTION                                  NOIS   6
C                                                                       NOIS   7
C*********************************************************************NOIS   8
C                                                                       NOIS   9
000007          INTEGER DLIFE, DINT                                     NOIS  10
000007          INTEGER PARCK3                                          NOIS  11
000007          REAL  NPKHRS , NUMDA                                    NOIS  12
000007          COMMON/CGV/ GWDUP9(5,6,12)                              NOIS  13
000007          COMMON/SGV/ DGSGST(400,22)                              NOIS  14
000007          COMMON/NHI/          JSPRNT(20)     , DLIFE(30)     , DINT(30)      ,NOIS  15
                *    LECDUE          , NYM          , NLM           , NSM           ,NOIS  16
                *    NIM             , NTM          , NTSM          , NTM           ,NOIS  17
                *    NUM             , IZ           , JK            , PARCK3(6)     ,NOIS  18
                *    NW              , NZ           , IPHINT        , IRV           ,NOIS  19
                *    IPOLCH(4,2,4), IDACCD(3)       , NHV           , NDTM          ,NOIS  20
                *    NVC(9)                                                         ,NOIS  21
000007          COMMON/RUR/          SCALEF(10)     , DVMCAP(5)     , DVMLOF(5)     ,NOIS  22
                * DLNCAP(5)          , ULANDC(10)   , UTRAKC        , USTRUC(4)     ,NOIS  23
                * DOWNTH(16)         , DCNCEN(11)   , DWAYCC(5)     , DVARGS(5,13)  ,NOIS  24
                * DWAINT(16)         , DVHSUC(4,6)  , DVHUC(4)      , DNUMVH(5)     ,NOIS  25
                * ASSWAT(4)          , TAXWAT(4)    , DNOISE(20,14,2), RESDEN(4,7)  ,NOIS  26
                * DONPFL(225,4,4), DGLKST(225,5) , BACKON(225,3), DPKWRF(225,3) ,NOIS  27
                * DPKPRF(225,3)                                                  ,NOIS  28
                * NPKHRS             , NUMDA        , DPRPCT        , DPTRIP(4,3)   ,NOIS  29
                * DPOLYU(2,3)        , DVDLAY(2,3)  , DMYLOF(2,3)   , XSYSCH(4,5)   ,NOIS  30
                * APKPCT             , APTRIP(4,4)  , AUELAY(2,4,2) , MLNOIR(2,2)   ,NOIS  31
                * AVHTPC(4)          , DVHTPC(3)    , AUTOCH(11)    , MNCHAR(4)     ,NOIS  32
                * DPCYCH(3,4,2)      , APCYCH(4,6)  , HPCYCH(2,6)   , POLRAT(3,3)   ,NOIS  33
                * DWSTAC             , WACRAT(2,5,2) , XACRAT(4,4)  , UACRAT(2,3)   ,NOIS  34
                * HNPDAT(2,4)        , HNYDPC5(2,6) , HNYDPC(2,3)   , XOPCU(4,7,6)  ,NOIS  35
                * AOPTCU(4,7,4)      ,DOFPFL(150,4,4),DOFLRC(150,3,3) ,RACKOF(150,3,2),NOIS  36
                * DYUOPC             , DCCOPC       , DNOPC         , Z(4)          ,NOIS  37
                * DVPPUC(6,9,4)      , DTNOPC(11,7) , DGOPC(7)      , FLOCAP(5)     ,NOIS  38
                * DFARE(6,2,2)                                                    ,NOIS  39
000002          COMMON/SVR/ MSAWDT(100,2), SWRYDT(320,5)                NOIS  40
000002          COMMON/RR/WRIND                                         NOIS  41
000002          WRWRN(NOIS                                              NOIS  42
000004          DO 2000  NT = 1,2                                       NOIS  43
C               COMPUTE NOISE AREA PER SEGMENT                          NOIS  44
000009          DO 500  NS = 1,NSM                                      NOIS  45
C               INITIALIZE NOISE AREA                                   NOIS  46
C               IF CONSTRUCTION TYPE IS TUNNEL, THEN THERE IS NO NOISE  NOIS  48
000009          DGSGST(NS,22) = 0.0                                     NOIS  47
000007          IF (DGSGST(NS,2) .EQ. 4.0) GO TO490                     NOIS  49
C               IF CONSTRUCTION IS BELOW GRADE THAN THERE IS NO NOISE   NOIS  50
000011          IF (DGSGST(NS,2) .EQ. 7.0) GO TO 490                    NOIS  51
C               IF LAND USE CODE IS 9 OR 10 (OPEN SPACE), THEN THERE IS NO NOISE NOIS  52
000013          IF (DGSGST(NS,3) .EQ. 9.0 .OR. DGSGST(NS,3) .EQ. 10.0)  GO TO 490 NOIS  53
C               COMPUTE LARGEST WIDTH - W1 (EITHER DM WIDTH OR EXISTING SYS WIDTH)NOIS  54
000023          IF(DGSGST(NS,18) .GT. DGSGST(NS,17)) GO TO 100          NOIS  55
```



```
000224          PRINT 270                                               NOIS 111
000231     270 FORMAT(1H ,*  CONS   NO. OPER.*, 4X, *STATUS OF EXISTING ROW*/ NOIS 112
                * 1H , *  TYPE    OF SPEED*, 5X, *ABAN-*, 6X, *WITH AT-GRADE HIGHWAYNOIS 113
                *Y HAVING*, 13X, *WITH ELEVATED HIGHWAY HAVING*, 7X, *COMMUTER*,   NOIS 114
                * 5X, *RAPID*/1H , *  CODE LANES (MPH)*, 5X, *DONED*,2X,*   2 LANES NOIS 115
                * 4 LANES    4 LANES    8 LANES*), 6X, *RAIL*, 3X, *TRANSIT*,//)  NOIS 116
000231          DO 280  ND = 1,NDM                                      NOIS 117
000233          PRINT 300, (DNOISE(ND,NNO,NT), NOD = 1,14)              NOIS 118
000250     300 FORMAT(1H , 3F6.0, 11F10.1,/)                            NOIS 119
000250     280 CONTINUE                                                 NOIS 120
000253     A3 CONTINUE                                                  NOIS 121
C               INITIALIZE GWDUP9                                       NOIS 122
000253          DO 1105  K=1,6                                          NOIS 123
000254          DO 1105  L=1,5                                          NOIS 124
000256          DO 1105  M=1,5                                          NOIS 125
000257     1105 GROUP9(K,L,M) = 0.0                                     NOIS 126
C               GROUP    NOISE    AREAS BY GEOG AREA, CONS, TYPE, + RESID DENSITYNOIS 127
000272          DO 1110  NS = 1,NSM                                     NOIS 128
000273          K = DGSGST(NS,1)                                        NOIS 129
000275          L = DGSGST(NS,2)                                        NOIS 130
000277          M = DGSGST(NS,20)                                       NOIS 131
000301          IF(M.EQ.0) GO TO 1110                                   NOIS 132
000302          GROUP9(K,L,M) = GROUP9(K,L,M) + DGSGST(NS,22)           NOIS 133
000310     1110 CONTINUE                                                NOIS 134
C               PRINT HEADINGS                                          NOIS 135
000313          IF(JSPRNT(26) .EQ. 1) GO TO 82                          NOIS 136
000315          IF (NT .EQ.  1)  GO TO 1120                             NOIS 137
000317          PRINT 1126                                              NOIS 138
000327     1126 FORMAT(1H1,///,*  PERSONS AFFECTED BY NOISE FROM DM GUIDEWAY PROPERNOIS 139
                * (DAY)*,////)                                          NOIS 140
000322          GO TO 1130                                              NOIS 141
000323     1120 PRINT 1125                                              NOIS 142
000327     1125 FORMAT(1H1,///,*  PERSONS AFFECTED BY NOISE FROM DM GUIDEWAY PROPERNOIS 143
                * (NIGHT)*,////)                                        NOIS 144
000327     1130 CONTINUE                                                NOIS 145
000327          PRINT 1127                                              NOIS 146
000333     1127 FORMAT(1H ,*2A, *ETHNIC BREAKDOWN*, 29X, *INCOME BREAKDOWN*, 10X, NOIS 147
                * *TOTAL* /1H , * GEOG CONS RES.*, 4X, *   NOISE*, 5X, *WHITE  PCT*NOIS 148
                * , 5X, *BLACK  PCT*, 5X, *OTHER  PCT*, 7X,*LOW  PCT*, 7X,*MED  PCTNOIS 149
                * *, 5X, *HIGH  PCT    NO. OF*,/1H , * AREA TYPE DENS* , 4X,*AREA*,*NOIS 150
                * &I *   (NO. OF   OF*), *    PERSONS*/1H1  3(* CODE*), 5X,*(ACRES)NOIS 151
                *0 , A10  PERSONS)  TOT*), *  IMPACTED*,///)             NOIS 152
000333     82 CONTINUE                                                  NOIS 153
C               COMPUTE NUMBER OF PEOPLE IMPACTED  PER CATAGORY AND PERCENT OF NOIS 154
C               TOTAL AND PRINT PER GROUP CATAGORY                      NOIS 155
000333          SUM1 = 0.0                                              NOIS 156
000334          SUM2 = 0.0                                              NOIS 157
000335          SUM3 = 0.0                                              NOIS 158
000336          SUM4 = 0.0                                              NOIS 159
000337          SUM5 = 0.0                                              NOIS 160
000340          SUM6 = 0.0                                              NOIS 161
000341          SUM7 = 0.0                                              NOIS 162
000342          SUMA = 0.0                                              NOIS 163
000343          DO 1145  K = 1,6                                        NOIS 164
000344          DO 1145  L = 1,6                                        NOIS 165
```



```
000027          W1 = DGSGST(NS,17)                                      NOIS  56
000038          GO TO 110                                               NOIS  57
000039     100 W1 = DGSGST(NS,18)                                       NOIS  58
000039     110 CONTINUE                                                 NOIS  59
C               SEARCH INPUT NOISE DISTANCE TABLE TO FIND APPROPRIATE NOISE NOIS  60
C               DISTANCE IN FEET FROM CENTERLINE (DISTNS)               NOIS  61
C                    I = II = CONS TYPE                                 NOIS  62
C                    J = JJ = NO. OF LANES                              NOIS  63
C                    K = KK = SPEED                                     NOIS  64
000039          I = DGSGST(NS, 2)                                       NOIS  65
000039          J = DGSGST(NS,1A)                                       NOIS  66
000037          NL= DGSGST(NS,12)                                       NOIS  67
000041          K = DGLKST(NL,1)                                        NOIS  68
000043          DO 130  ND = 1,NDM                                      NOIS  69
000046          II = DNOISE(ND,I,NT)                                    NOIS  70
000051          JJ = DNOISE(ND,J,NT)                                    NOIS  71
000059          KK = DNOISE(ND,K,NT)                                    NOIS  72
000060          IF(I.EQ.II.AND.J.EQ.JJ.AND.K.EQ.KK)  GO TO 120          NOIS  73
000073          GO TO 130                                               NOIS  74
C               COMPUTE TOTAL NOISE WIDTH                               NOIS  75
000073     120 WDN = DGSGST(NS,19)  + 3.0                               NOIS  76
000077          IF (DNOISE(ND,NNO,NT) .LT. W1)  GO TO 490               NOIS  77
000100          W2 = 2 * DNOISE(ND,NDO, NT) - W1                        NOIS  78
000114     130 CONTINUE                                                 NOIS  79
C               COMPUTE NOISE AREA                                      NOIS  80
000117          DGSGST(NS,22) = W2 * DGSGST(NS,15) * (5280./43560.)     NOIS  81
000123          GO TO 500                                               NOIS  82
000124     490 DGSGST(NS,22) = 0.0                                      NOIS  83
000124     500 CONTINUE                                                 NOIS  84
C               PRINT DM GUIDEWAY SEGMENT DISPLACEMENT + NOISE STATISTICS NOIS  85
000131          IF(JSPRNT(20) .EQ. 1) GO TO 80                          NOIS  86
000133          PRINT 1160                                              NOIS  87
000136     1160 FORMAT(1H1,// *  DM GUIDEWAY SEGMENT DISPLACEMENT AND NOISE STATISNOIS  88
                *TICS*,///)                                             NOIS  89
000136          PRINT 1162                                              NOIS  90
000142     1162 FORMAT(1H ,*   SEGM.    GEOG    CONS    LAND    RES.   STATUS NOIS  91
                *M ANW  EXISTING  DISPLACEMENT  NOISE IMPACT* / 1H , 5X, *NO.  NOIS  92
                *AREA    TYPE    USE    DEN.  EXIST.    WIDTH  SYS WDTH*,      NOIS  93
                * 2(1X,*ABAR*) / 1H ,      8X, A(4A,*CODE*), *  SYSTEM*,      NOIS  94
                * 2(4X,*(FEET)*), 2I8X,*(ACRES)*) *///                       NOIS  95
000142          DO 1160  NS = 1,NSM                                     NOIS  96
000144     1164 PRINT 1165, NS, (DGSGST(NS,N5D), N5D = 1,3), DGSGST(NS,20), NOIS  97
                * DGSGST(NS,19), (DGSGST(NS,N5D), N5D = 17,18),          NOIS  98
                *(DGSGST(NS,N5D), N5D = 21,22)                          NOIS  99
000210     1165 FORMAT(1H , I8, 5F8.0, 2F10.2, 2F15.2)                  NOIS 100
000210     80 CONTINUE                                                  NOIS 101
C               PRINT NOISE DISTANCE TABLE (DAY + NIGHT)                NOIS 102
000210          IF(JSPRNT(23) .EQ. 1) GO TO 83                          NOIS 103
000212          IF (NT .EQ. 2)  GO TO 250                               NOIS 104
000214          PRINT 260                                               NOIS 105
000220     260 FORMAT(1H1,///,* NOISE DISTANCE INPUT TABLE (DAY)*,////) NOIS 106
000220          GO TO 280                                               NOIS 107
000221     250 PRINT 255                                                NOIS 108
000224     255 FORMAT(5/,1H ,*NOISE DISTANCE INPUT TABLE (NIGHT)*,////) NOIS 109
000224     280 CONTINUE                                                 NOIS 110
```



```
000348          DO 1145  M = 1,6                                        NOIS 166
000348          IF (GROUP9(K,L,M) .EQ. 0.0)  GO TO 18                   NOIS 167
000350          A = GROUP9(K,L,M) * RESDEN(M,1)                         NOIS 168
000357          B = GROUP9(K,L,M) * RESDEN(M,2)                         NOIS 169
000365          C = GROUP9(K,L,M) * RESDEN(M,3)                         NOIS 170
000367          D = GROUP9(K,L,M) * RESDEN(M,4)                         NOIS 171
000371          E = GROUP9(K,L,M) * RESDEN(M,5)                         NOIS 172
000377          F = GROUP9(K,L,M) * RESDEN(M,6)                         NOIS 173
000403          G = GROUP9(K,L,M) * RESDEN(M,7)                         NOIS 174
000407          SUM1 = SUM1 + GROUP9(K,L,M)                             NOIS 175
000413          SUM2 = SUM2 + A                                         NOIS 176
000415          SUM3 = SUM3 + B                                         NOIS 177
000417          SUM4 = SUM4 + C                                         NOIS 178
000421          SUM5 = SUM5 + D                                         NOIS 179
000423          SUM6 = SUM6 + E                                         NOIS 180
000426          SUM7 = SUM7 + F                                         NOIS 181
000427          SUMA = SUMA + G                                         NOIS 182
000431          AP= A / G  * 100.                                       NOIS 183
000433          BP= B / G  * 100.                                       NOIS 184
000435          CP= C / G  * 100.                                       NOIS 185
000437          DP= D / G  * 100.                                       NOIS 186
000441          EP= E / G  * 100.                                       NOIS 187
000443          FP= F / G  * 100.                                       NOIS 188
000445          GO TO 19                                                NOIS 189
000446     18 A  = 0.0                                                  NOIS 190
000447          AP = 0.0                                                NOIS 191
000450          B  = 0.0                                                NOIS 192
000451          BP = 0.0                                                NOIS 193
000452          C  = 0.0                                                NOIS 194
000453          CP = 0.0                                                NOIS 195
000454          D  = 0.0                                                NOIS 196
000455          DP = 0.0                                                NOIS 197
000456          E  = 0.0                                                NOIS 198
000457          EP = 0.0                                                NOIS 199
000460          F  = 0.0                                                NOIS 200
000461          FP = 0.0                                                NOIS 201
000462          G  = 0.0                                                NOIS 202
000463     19 CONTINUE                                                  NOIS 203
000463          IF(JSPRNT(24) .EQ. 1)  GO TO 1145                       NOIS 204
000465          PRINT 1135, K,L,M, GROUP9(K,L,M), A,AP,B,BP,C,CP,D,DP,E,EP,F,FP,G NOIS 205
000537     1135 FORMAT(1H ,3I5, F12.2, 6(F10.0,F5.0), F10.0)            NOIS 206
000537     1145 CONTINUE                                                NOIS 207
000544          IF (SUMA .EQ. 0.0)  GO TO 1136                          NOIS 208
000546          PC2 = SUM2 / SUMA * 100.                                NOIS 209
000551          PC3 = SUM3 / SUMA * 100.                                NOIS 210
000553          PC4 = SUM4 / SUMA * 100.                                NOIS 211
000555          PC5 = SUM5 / SUMA * 100.                                NOIS 212
000557          PC6 = SUM6 / SUMA * 100.                                NOIS 213
000561          PC7 = SUM7 / SUMA * 100.                                NOIS 214
000563          GO TO 1138                                              NOIS 215
000564     1136 PC2 = 0.0                                               NOIS 216
000565          PC3 = 0.0                                               NOIS 217
000566          PC4 = 0.0                                               NOIS 218
000567          PC5 = 0.0                                               NOIS 219
000570          PC6 = 0.0                                               NOIS 220
```

```
RUN VERSION 2.1 --PSW LEVEL 29A--                                   NOISE                    06/30/72

000571          PC7 = 0.0                                                                    NOIS 221
000572     113A CONTINUE                                                                     NOIS 222
000572          IF (NT .EQ. 2) GO TO 999                                                     NOIS 223
000574          SMRYNT (300,MM) = SUM1                                                       NOIS 224
000600          SMRYNT (301,MM) = SUM2                                                       NOIS 225
000603          SMRYNT (302,MM) = SUM3                                                       NOIS 226
000606          SMRYNT (303,MM) = SUM4                                                       NOIS 227
000611          SMRYNT (304,MM) = SUM5                                                       NOIS 228
000614          SMRYNT (305,MM) = SUM6                                                       NOIS 229
000617          SMRYNT (306,MM) = SUM7                                                       NOIS 230
000621          SMRYNT (307,MM) = SUM8                                                       NOIS 231
000624          GO TO 999                                                                    NOIS 232
000625      999 CONTINUE                                                                     NOIS 233
000625          SMRYNT (311,MM) = SUM1                                                       NOIS 234
000631          SMRYNT (312,MM) = SUM2                                                       NOIS 235
000634          SMRYNT (313,MM) = SUM3                                                       NOIS 236
000637          SMRYNT (314,MM) = SUM4                                                       NOIS 237
000642          SMRYNT (315,MM) = SUM5                                                       NOIS 238
000645          SMRYNT (316,MM) = SUM6                                                       NOIS 239
000650          SMRYNT (317,MM) = SUM7                                                       NOIS 240
000653          SMRYNT (318,MM) = SUM8                                                       NOIS 241
000656      99A CONTINUE                                                                     NOIS 242
000656          IF(JSPRNT(24) .EQ. 1)  GO TO 282                                             NOIS 243
000660          PRINT 1137, SUM1, SUM2, PC2, SUM3, PC3, SUM4, PC4, SUM5, PC5,                NOIS 244
               X SUM6, PC6, SUM7, PC7, SUM8                                                  NOIS 245
000720     1137 FORMAT(/,1H ,*TOTALS*,9X,F12.2, 6(F10.0,F5.0) F10.0)                         NOIS 246
000720      282 CONTINUE                                                                     NOIS 247
000720     2000 CONTINUE                                                                     NOIS 248
000722          RETURN                                                                       NOIS 249
000723          END                                                                          NOIS 250
```

```
RUN VERSION 2.1 --PSW LEVEL 29A--                                   NOISE                    06/30/72

000725

START OF TEMPORARIES
001202

START OF INDIRECTS
001207

UNUSED COMPILER SPACE
124000
```

```
RUN VERSION 2.1 --PSW LEVEL 29A--                                   NOISE                    06/30/72


NOISE

SUBPROGRAM LENGTH
001303

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
10      =  000446    15      =  000463    A0      =  000210    B2      =  000333
B3      =  000293    100     =  000830    110     =  000032    120     =  000073
130     =  000114    240     =  001013    250     =  000221    255     =  001021
260     =  000275    270     =  001038    282     =  000728    300     =  001072
490     =  000174    500     =  000126    900     =  000056    999     =  000625
1105    =  000257    1110    =  000310    1120    =  000323    1125    =  001107
1126    =  001076    1127    =  001128    1130    =  000327    1135    =  001167
1136    =  000546    1137    =  001174    1138    =  000572    1145    =  000537
1160    =  000736    1162    =  000747    1165    =  001006    2000    =  000728

BLOCK NAMES AND LENGTHS
GO      =  000550    SG      =  021140    ADI     =  000307    ADR     =  034004
SVR     =  003410    MM      =  000001

VARIABLE ASSIGNMENTS
A       =  001240    AM      =  001267    ASSRAT  =  000264C04 AUTOCH  =  071415C04
B       =  001241    BACKOF  =  031470C04 BACKON  =  012575C04 BP      =  001270
C       =  001242    CP      =  001271    D       =  001263    DACMAT  =  071622C04
DCMCEN  =  000070C04 DFARE   =  033764C04 DBLKST  =  010430C04 DBOPC   =  033750C04
DGSOST  =  000000C02 DGWDTH  =  000050C04 DINT    =  000150C03 DLIFE   =  000120C03
DLHCAP  =  000024C04 DMAINT  =  000211C04 DMVLOF  =  021307C04 DMOISE  =  000274C04
DNUMVH  =  000257C04 DOFLRC  =  026762C04 DOFPFL  =  022222C04 DONPFL  =  001410C04
DP      =  001272    DPCVCH  =  021434C04 DPDLYD  =  021273C04 DPKDRF  =  020011C04
DPKWRF  =  016544C04 DPTRIP  =  021257C04 DTRDPC  =  033033C04 DVDLAY  =  021301C04
DVHCAP  =  000012C04 DVMLOF  =  000017C04 DVMSUC  =  000223C04 DVMUC   =  000253C04
DVMTPC  =  071412C04 DVNPUC  =  033003C04 DWATEC  =  000103C04 DYARDS  =  000110C04
E       =  001244    EP      =  001273    F       =  001265    FLOCAP  =  033757C04
FP      =  001274    G       =  001260    GABUPO  =  000000C01 HACRAT  =  021554C04
HLWDIP  =  021402C04 HMCMAN  =  021438C04 HOPUAT  =  021030C04 HPCVCM  =  021530C04
HVOPES  =  021640C04 HWYOPC  =  021654C04 I       =  001233    IDACCD  =  000271C03
II      =  001240    IPOLCD  =  000241C03 J       =  001234    JJ      =  001241
JSPRNT  =  000000C03 K       =  001236    KK      =  001242    L       =  001246
M       =  001247    MM      =  001227    MMINDX  =  000008C06 ND      =  001237
NDD     =  001243    NUM     =  000224C03 NL      =  001235    NPKHRS  =  021254C04
NS      =  001231    NSD     =  001245    NSW     =  000217C03 NT      =  001230
NUMDA   =  021255C04 NVC     =  000270C03 PARCK3 =  000227C03 PC2     =  001275
PC3     =  001276    PC4     =  001277    PC5     =  001300    PC6     =  001301
PC7     =  001302    PULWAT  =  021544C04 RESDEN  =  001354C04 RSAVDT  =  000000C05
SCALEF  =  000000C04 SMRYNT  =  000318C05 SUM1    =  001250    SUM2    =  001251
SUM3    =  001252    SUM4    =  001253    SUM5    =  001254    SUM6    =  001255
SUM7    =  001256    SUM8    =  001257    TAXRAT  =  000270C04 ULANDC  =  000031C04
USTRUC  =  000044C04 W1      =  001232    W2      =  001244    XACRAT  =  021602C04
XDELAY  =  021342C04 XUPCU   =  021602C04 XOPICD  =  022042C04 XPCVCM  =  021500C04
XPTRIP  =  021342C04 XSYSCH  =  021315C04 XVHTPC  =  021408C04 Z       =  033277C04

START OF CONSTANTS
```

```
                SUBROUTINE SORT (NO,STAT,GROUP2,GRPCNT,AVRG1,AVRG2)
      C
000011          DIMENSION GROUP1(32,5,11),GROUP2(32,5,11)
000011          DIMENSION UNIT4(5,6,12),AVRG1(5,6,12),AVRG2(5,6,12),PCT4(5,6,12),
               *          UNIT4(5,6,12),UNIT5(5,6,12),UNIT6(5,6,12)
000011          DIMENSION GRPCNT(32,5,11)
000011          DIMENSION  STAT(480,22)
000011          COMMON /ANY UNIT1, UNIT4
000011          COMMON/NOI/PARDT(190)
000011          EQUIVALENCE (UNIT1(1),UNIT7(1),UNT4(1)), (UNIT6(1),UNIT8(1),
               *     PCT4(1))
      C
000011          DO 7A IW=1,32
000014          DO 7A JW=1,5
000017          DO 7A MW=1,11
000022          GRPCNT(IW,JW,MW)=0.0
000027       7A GROUP1(IW,JW,MW)=0.0
000043          DO 110 INW4=1,
00004A          JNW4=(INW-3)-4
      C
000051          DO 4A I=1,4
00005A          DO 4A IW=1,5
00006A          DO 4A IC=1,11
000063       4A UNT4(IW,IW,IC)=0.0
00010A          DO 5A IW=1,NO
000183          IK=STAT(IW,1)
00018A          IL=STAT(IW,2)
000110          IM=STAT(IW,3)
000113       5A UNT4(IK,IL,IM)=UNT4(IK,IL,IM)+STAT(IW,1RD)
000127          CALL SUM (3,4,10,UNT4)
00013F          CALL PHCNT (3,4,10,UNT4,PCT4)
00014A          DO 8A JK=1,4
00014R          DO 8A JL=1,5
000159          DO 8A JM=1,11
000153          JNW=JNW+JK
000159          GROUP2(JNW,JL,JM) = GROUP2(JNW,JL,JM) + UNT4 (JK,JL,JM )
00016A          GROUP1(JNW,JL,JM)=GROUP1(JNW,JL,JM)+UNT4(JK,JL,JM)
00017A       8A GRPCNT(JNW,JL,JM)=GRPCNT(JNW,JL,JM)+PCT4(JK,JL,JM)
000214      110 CONTINUE
000217          DO 12A JK=1,4
000227          DO 12A JL=1,5
000229          DO 12A JM=1,11
00023A          UNIT1(JK,JL,JM)=GROUP1(JK,JL,JM)
000240      12A UNIT4(JK,JL,JM)=GROUP1(JK+12,JL,JM)
000254          CALL AVRG    (4,5,11,UNIT4,UNIT1,AVRG1)
000262          DO 15A JK=1,4
000271          DO 15A JL=1,5
00027A          DO 15A JM=1,11
000277          UNIT7(JK,JL,JM)=GROUP1(JK+24,JL,JM)
000387      15A UNIT8(JK,JL,JM)=GROUP1(JK+28,JL,JM)
00032A          CALL  AVRG   (4,5,11,UNIT8,UNIT7,AVRG2)
00033A          DO  13A  JK = 1,4
000337          DO  13A  JL= 1,
000347          DO  13A  JM = 1,11
```



```
000349          UNIT1(JK,JL,JM) = GROUP2(JK,JL,JM)
000355      13A UNIT4(JK,JL,JM) = GROUP2(JK+12,JL,JM)
000374          CALL AVRG (4,5,11,UNIT4,UNIT1,AVRG1)
000400          DO 16A  JK = 1,4
000407          DO 16A  JL = 1,5
000412          DO 16A  JM = 1,11
000414          UNIT7(JK,JL,JM) = GROUP2(JK+24,JL,JM)
000428      16A UNIT8(JK,JL,JM) = GROUP2(JK+28,JL,JM)
000444          CALL AVRG(4,5,11,UNIT8,UNIT7,AVRG2)
000447          IF (IPARDT(26).EQ.1) GO TO 200
000455          CALL PRNTARIGH(IP1,GRPCNT,AVRG1,AVRG2)
000457      200 CONTINUE
000457          RETURN
000460          END
```



```
SORT

SUBPROGRAM LENGTH
000459

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
40    =  000047    50    =  000113    70    =  000027    90    =  000175
110   =  000214    120   =  000240    130   =  000355    150   =  000387
180   =  000429    200   =  000457

BLOCK NAMES AND LENGTHS
AN    =  001320    NOI   =  000307

VARIABLE ASSIGNMENTS
GROUP1 =  000474   IA    =  004043   IB    =  004044   IC    =  004045
IK    =  004046   IL    =  004047   IM    =  004050   IPARDT =  000000C02
IR    =  004074   IWD   =  004041   JK    =  004051   JL    =  004052
JM    =  004099   JNW   =  004042   JNW   =  004054   JR    =  004037
NO    =  004040   PCT4  =  000550C01 UNIT1 =  000880C01 UNIT4 =  000550C01
UNIT7 =  000880C01 UNIT8 =  000550C01 UNT4  =  000880C01

START OF CONSTANTS
000461

START OF TEMPORARIES
00046A

START OF INDIRECTS
000472

UNUSED COMPILER SPACE
126480
```

```
RUN VERSION 2.1 --PSW LEVEL 298--                                        06/30/72

              FORTRAN VI SUBROUTINE AVRG (I,J,M,ITEMS,UNITS,AVERG)        DMAVRG 2
        C                                                                 DMAVRG 3
000011        REAL ITEMS                                                  DMAVRG 4
000011        DIMENSION ITEMS(5,6,12), UNITS(5,6,12), AVERG(5,6,12)       DMAVRG 5
        C                                                                 DMAVRG 6
000011        DO 70 KI=1,I                                                DMAVRG 7
000014        DO 70 KJ=1,J                                                DMAVRG 8
000017        DO 70 KM=1,M                                                DMAVRG 9
000022        IF (UNITS(KI,KJ,KM))40,40,50                                DMAVRG10
000027     50 AVERG(KI,KJ,KM)=ITEMS(KI,KJ,KM)/UNITS(KI,KJ,KM)            DMAVRG11
000042        GO TO 75                                                    DMAVRG12
000043     40 AVERG(KI,KJ,KM)=0.0                                         DMAVRG13
000051     75 CONTINUE                                                    DMAVRG14
000051     70 CONTINUE                                                    DMAVRG15
00006A        RETURN                                                      DMAVRG16
00006A        END                                                         DMAVRG17
```

```
RUN VERSION 2.1 --PSW LEVEL 298--                    AVRG          06/30/72

AVRG

SUBPROGRAM LENGTH
000079

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
40     = 000047    50     = 000027    70     = 000051    75     = 000051

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS
KI     = 000070    KJ     = 000071    KM     = 000072

START OF CONSTANTS
000061

START OF TEMPORARIES
000061

START OF INDIRECTS
000069

UNUSED COMPILER SPACE
13020A
```

```
                BLOCK DATA                                                      DMBLKU72
000007          DIMENSION H1(14),H2(14),H3(14),H4(14)                           DMBLKU73
000007          COMMON/ICS/TITLE1(10), TITLE2(10), ANCM(2), UNTNAM(5,8),        DMBLKU74
                XTNAME(5,12)                                                    DMBLKU75
000007          COMMON/EF/TYPNAM(11,6),ANMEAD(14,4),UNTNM(10),AC(10),ACI(10),   DMBLKU76
                X          ANZMED(14,4)                                         DMBLKU77
000007          EQUIVALENCE(M1(1),ANMEAD(1,1)),(M2(1),ANMEAD(1,2))              DMBLKU78
000007          EQUIVALENCE(M3(1),ANMEAD(1,3)),(M4(1),ANMEAD(1,4))              DMBLKU79
000007          DATA ANCM /10HANNUAL COS,10HT TABLE    /                        DMBLKU80
000007          DATA TITLE1,TITLE2 /10H BUIDEWAY ,10HPROPER     ,               DMBLKU81
                X                  10H ACCEL/DEC,10HEL LANES  ,                 DMBLKU82
                X                  10H YARDS    ,10H          ,                 DMBLKU83
                X                  10H WAY COMM.,10HAND CONT. ,                 DMBLKU84
                X                  10H DM VEHICL,10HE TYPE1   ,                 DMBLKU85
                X                  10H DM VEHICL,10HE TYPE2   ,                 DMBLKU86
                X                  10H DM VEHICL,10HE TYPE3   ,                 DMBLKU87
                X                  10H DM VEHICL,10HE TYPE4   ,                 DMBLKU88
                X                  10H INTERCHAN,10HGES       ,                 DMBLKU89
                X                  10H TERMINALS,10H          ,                 DMBLKU90
                X                  10H CONTROL C,10HENTER     ,                 DMBLKU91
                X                  10H MAINTENAN,10HCE FACIL. ,                 DMBLKU92
                X                  10H TOTALS   ,10H          /                 DMBLKU93
000007          DATA M1/10HTYPE    ,10H UNIT NAME,10H    TOTAL ,10H          T, DMBLKU94
                X        10HOTAL   LIF,10HE  INTER- ,10H    ANNUAL,10H  ANNUAL C, DMBLKU95
                X        10HUST TOTAL ,10H    ANNUAL,10H    ANNUAL,10H     PCT. , DMBLKU96
                X        10H,PCT.OF PC,10HT. OF     /                          DMBLKU97
000007          DATA M2/10H        ,10H          ,10H    UNITS ,10H          C, DMBLKU98
                X        10HOST    YEA,10HRS  EST   ,10H    COST  ,10H  WITH INT, DMBLKU99
                X        10H.   COST/U,10HNIT COST/U,10HNIT COST W,10HITH  OF  , DMBLKU100
                X        10H TOTAL  TO,10HT.AN.     /                           DMBLKU101
000007          DATA M3/10H        ,10H          ,10H                ,10H      , DMBLKU102
                X        10H           ,10H          ,10H                ,10H  , DMBLKU103
                X        10H           ,10H          ,10H    INTERE,10HST  TOTAL , DMBLKU104
                X        10H ANNUAL CO,10HST WT     /                           DMBLKU105
000007          DATA M4/10H        ,10H          ,10H                ,10H      , DMBLKU106
                X        10H           ,10H          ,10H                ,10H  , DMBLKU107
                X        10H           ,10H          ,10H    PER UN,10HIT  COST  , DMBLKU108
                X        10H COST   -T,10HAN INT    /                           DMBLKU109
000007          DATA ANZMED/                                                    DMBLKU110
                X  10HTYPE         ,10H      TOTA,10HL      LIF,10HE  INTER- ,   DMBLKU111
                X  10H   ANNUAL ,10H  ANNUAL C,10HOST  PCT. ,10H PCT.OF PC,     DMBLKU112
                X  10HT. OF     ,10H          ,10H          ,10H          ,     DMBLKU113
                X  10H          ,10H          ,             ,             ,     DMBLKU114
                X  10H          ,10H      COST,10H      YEA,10HRS  EST  ,      DMBLKU115
                X  10H   COST   ,10H   WITH IN,10HT.   OF  ,10H TOTAL  TO,     DMBLKU116
                X  10HT.AN.     ,10H          ,10H          ,10H          ,     DMBLKU117
                X  10H          ,10H          ,             ,             ,     DMBLKU118
                X  10H          ,10H          ,10H          ,10H          ,     DMBLKU119
                X  10H          ,10H          ,10H      TOTAL,10H ANNUAL CO,    DMBLKU120
                X  10HST WT     ,10H          ,10H          ,10H          ,     DMBLKU121
                X  10H          ,10H          ,             ,             ,     DMBLKU122
                X  10H          ,10H          ,10H          ,10H          ,     DMBLKU123
                X  10H          ,10H          ,10H     COST ,10H COST   -T,    DMBLKU124
                X  10HM INT     ,10H          ,10H          ,10H          ,     DMBLKU125
```



```
                X   10H         ,10H          /                                DMBLKU56
000007          DATA TNAME /10H LAND      ,10H TRACK    ,10H STRUCT.  ,10H TOTALS  DMBLKU57
                X .        10H           ,10H LAND     ,10H TRACK .  ,10H STRUCT. DMBLKU58
                X .        10H TOTALS    ,10H          ,10H LAND     ,10H TRACK   DMBLKU59
                X .        10H STRUCT.   ,10H TOTALS   ,10H          ,10H LAND    DMBLKU60
                X .        10H TRACK     ,10H STRUCT.  ,10H TOTALS   ,10H         DMBLKU61
                X .        10H LAND      ,10H COMPUTERS,10H STRUCT.  ,10H TOTALS  DMBLKU62
                X .        10H           ,10H LAND     ,10HSTR.,-EQUIP,10H TOTALS DMBLKU63
                X .        10H           ,10H          ,10H LAND     ,10H STRUCT. DMBLKU64
                X .        10H TRACK     ,10H TOTALS   ,10H          ,10HCC-EQUIP DMBLKU65
                X .        10H DETECT.   ,10H TOTALS   ,10H          ,10H         DMBLKU66
                X . 10H VEH. 1   ,10H TOTALS   ,10H          ,10H          ,      DMBLKU67
                X   10H          ,10H VEH. 1   ,10H TOTALS   ,10H          ,      DMBLKU68
                X   10H          ,10H          ,10H VEH. 3   ,10H TOTALS   ,      DMBLKU69
                X   10H          ,10H          ,10H          ,10H VEH. 4   ,      DMBLKU70
                X   10H TOTALS   ,10H          ,10H          ,10H          /      DMBLKU71
000007          DATA UNTNAM/10H ACRES     ,10H LN-MILES ,10H LN-MILES ,10H ------ DMBLKU72
                X .        10H           ,10H  ACRES   ,10H LN-MILES ,10H LN-MILESDMBLKU73
                X .        10H ------    ,10H          ,10H ACRES    ,10H LN-MILESDMBLKU74
                X .        10H LN-MILES  ,10H ------   ,10H          ,10H ACRES   DMBLKU75
                X .        10H LN-MILES  ,10H LN-MILES ,10H ------   ,10H         DMBLKU76
                X .        10H           ,10H          ,10H          ,10H         DMBLKU77
                X .        10H           ,10H          ,10H          ,10H         DMBLKU78
                X .        10H           ,10H          ,             ,            DMBLKU79
                X                         10H ACRES     ,10H ACRES    ,10H LN-MILESDMBLKU80
                X .        10H ------    ,10H          ,10H RT.MILES ,10H LN-MILESDMBLKU81
                X .        10H ------    ,10H          ,10H          ,10H VEH.    DMBLKU82
                X .        10H VEH.      ,10H VEH.     ,10H VEH.     ,10H VEH.    DMBLKU83
                X /                                                             DMBLKU84
000007          END                                                             DMBLKU85
```



```
BLKDATA

SUBPROGRAM LENGTH
000007

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

BLOCK NAMES AND LENGTHS
DCS    - 000285    EF     - 000320

VARIABLE ASSIGNMENTS
AC     - 000244C02 ACI    - 000210C02 ANCM   - 000032C01 ANMEAD - 000102C02
ANZMED - 000338C02 M1     - 000102C02 M2     - 000128C02 M3    - 000130C02
M4     - 000154C02 TITLE1 - 000008C01 TITLE2 - 000020C01 TNAME  - 000011C01
TYPNAM - 000008C02 UNTNAM - 000034C01 UNTNM  - 0001T2C02

START OF CONSTANTS
000004

START OF TEMPORARIES
000005

START OF INDIRECTS
000007

UNUSED COMPILER SPACE
12600A
```

```
                FORTRAN IV SUBROUTINE ANSUMD (M,ANNCST,N, ANNCS, A)
000010          DIMENSION ANNCST(8,14), ANNCS(4,10), A(14,6)
C
000010          COMMON/UCS/TITLE1(16), TITLE2(10), ANCH(2), UNTNAM(5,9),
               1TNAME(5,12)
C
000010          DO 4A IM=1,M
000013          A(IM,1)=ANNCST(IM,6)
000016          DO 5A J=2,9
000021       5A A(IM,J)=ANNCST(IM,J+5)
000034       4A CONTINUE
000037          MN=M+N
000040          MM=M+1
000042          DO 7A IN=1,N
000045          IM=IN+M
000047          A(IM,1)=ANNCS(IN,2)
000052          DO 8A J=2,3
000055       8A A(IM,J)=ANNCS(IN,J+3)
000071       7A CONTINUE
000074          MN=M+N
000076          DO 9A J=1,3
000101          A(MN+1,J)=0.0
000104          DO 10A I=1,MN
000107      10A A(MN+1,J)=A(MN+1,J)+A(I,J)
000122       9A CONTINUE
000125          DO 12A J=1,3
000130          DO 125 I=1,MN
000133     125 A(I,J+3)=A(I,J)/A(MN+1,J)*100.00
000154     12A A (MN+1,J+3) = 100.00
000163          PRINT 130
000164     130 FORMAT(*1*,* *,*OM CAPITAL COST ANNUAL SUMMARY TABLE*///)
000186     132 FORMAT(12X,7A10,*(F12.0,2X)/)
000188          PRINT 133
000172          MR=M+N+1
000177          DO 4A IS=1,M
000202          IP=2*IS-1
000204       4B PRINT 132, TITLE1(IP),TITLE1(IP+1),(A(IS,JK),JK=1,6)
000243          DO 47 IS=MM,MR
000246          IK=IS-MM+1
000251          IP=2*IK-1
000252       47 PRINT 132, TITLE2(IP),TITLE2(IP+1),(A(IS,JK),JK=1,6)
000311     133 FORMAT (21/)+23X,*TOTAL COST  ANNUAL COST     ANN.COST+INT.*,
               1     *        PCT.OF TOT.  PCT. OF ANN.  PCT.OF ANN.+INT.*/)
000311          RETURN
000311          END
```



```
ANSUMD

SUBPROGRAM LENGTH
000369

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
46     =  000204    47     =  000252    5A     =  000021    8A     =  000034
7A     =  000071    8B     =  000055    9A     =  000122    10A    =  000107
12A    =  000144    125    =  000133    130    =  000315    132    =  000324

BLOCK NAMES AND LENGTHS
DCS    =  000249

VARIABLE ASSIGNMENTS
ANCH   =  000032C01 I      =  000357    IK     =  000364    IM     =  000352
IN     =  000356    IP     =  000362    IS     =  000361    J      =  000353
JK     =  000363    MM     =  000355    MN     =  000354    MR     =  000360
TITLE1 =  000000C01 TITLE2 =  000020C01 TNAME  =  000111C01 UNTNAM =  000034C01

START OF CONSTANTS
000312

START OF TEMPORARIES
000344

START OF INDIRECTS
000347

UNUSED COMPILER SPACE
12710A
```

```
RUN VERSION 2.1 --PSH LEVEL 298--                                        06/30/72

      SUBROUTINE ANNUAL2(IS,IZ,NN,TCOST,INTR,LIF,ATYPE,ANNCST)          DMANN2 2
C                                                                       DMANN2 3
C     INPUT ARGUMENTS                                                   DMANN2 4
C                                                                       DMANN2 5
C       INTEGER  IS, INDEX FOR STORING RESULTS IN ANNCST                DMANN2 6
C                IZ, INDEX FOR ACCESSING INT, LIF                       DMANN2 7
C                NN, NO. OF TABLE ENTRYS IN TUNITS,TCOST,ATYPE          DMANN2 8
C                                                                       DMANN2 9
C       INTEGER ARRAYS  INT, INTEREST TABLE                             DMANN210
C                       LIF, LIFE TABLE                                 DMANN211
C                                                                       DMANN212
C       REAL ARRAY   TCOST, TOTAL COST FOR UNITS IN TUNITS              DMANN213
C                                                                       DMANN214
C       LITERAL ARRAYS  ATYPE, TYPE NAME (10 CHARACTERS)                DMANN215
C                                                                       DMANN216
C     COMMON                                                            DMANN217
C                                                                       DMANN218
C       LITERAL ARRAYS  ANMEAD, TABLE COLUMN HEADINGS, ALPHA STRINGS    DMANN219
C                                                                       DMANN220
      DIMENSION INTR(10),LIF(10),TCOST(5),ANNCST(4,10),ATYPE(5)         DMANN221
      COMMON /EF/ TYPNAM(10), ANMEAD(14,4), UNTNM(10),AC(10),ACI(10),   DMANN222
     1  AN2HLD(14,4)                                                    DMANN223
      PRINT 101                                                         DMANN224
  101 FORMAT(1H , 85H-----------------------------------------------------DMANN225
     1-----------------------------------------------------------------)DMANN226
      PRINT 102,AN2HED                                                  DMANN227
  102 FORMAT(1H ,13A10,A5)                                              DMANN228
      PRINT 103                                                         DMANN229
      ANNCST(IS,1)=ATYPE(NN+1)                                          DMANN230
      ANNCST(IS,2)=TCOST(NN+1)                                          DMANN231
      ANNCST(IS,3)=0.0                                                  DMANN232
      ANNCST(IS,4)=0.0                                                  DMANN233
      ANNCST(IS,5)=0.0                                                  DMANN234
      ANNCST(IS,6)=0.0                                                  DMANN235
      ANNCST(IS, 7)=100.0                                               DMANN236
      ANNCST(IS, 8)=100.0                                               DMANN237
      ANNCST(IS, 9)=100.0                                               DMANN238
      DO 1 NA=1,NN                                                      DMANN239
      NZ=NN+1-I                                                         DMANN240
      XINT=INTR(NZ)/100.0                                               DMANN241
      CRF=XINT*(1.0+XINT)**LIF(NZ)/((1.0+XINT)**LIF(NZ)-1.0)            DMANN242
      AC(NA)=TCOST(NZ)/LIF(NZ)                                          DMANN243
      ACI(NA)=TCOST(NZ)*CRF                                             DMANN244
      ANNCST(IS,5)=ANNCST(IS,5)+AC(NA)                                  DMANN245
      ANNCST(IS,6)=ANNCST(IS,6)+ACI(NA)                                 DMANN246
    1 CONTINUE                                                          DMANN247
      DO 3 NA=1,NN                                                      DMANN248
      PCT=TCOST(NA)/TCOST(NN+1)*100.0                                   DMANN249
      PTAC=AC(NA)/ANNCST(IS,5)*100.0                                    DMANN250
      PTACI=ACI(NA)/ANNCST(IS,6)*100.0                                  DMANN251
      NZ=IZ+NN-1                                                        DMANN252
      XINT=INTR(NZ)                                                     DMANN253
    3 PRINT 103,ATYPE(NA),            TCOST(NA),LIF(NZ),XINT,           DMANN254
     1        AC(NA),ACI(NA),                PCT,PTAC,PTACI             DMANN255
```

```
RUN VERSION 2.1 --PSH LEVEL 298--                          ANNUAL2      06/30/72

  103 FORMAT(1H0,A10,                  1X,F14.0,1X,I6,1X,F6.2,1X,F11.0,1X,DMANN256
     1      F11.0,                          1X,F6.0,1X,F6.0,1X,F6.0)    DMANN257
      PRINT 101                                                         DMANN258
      PRINT 104,ATYPE(NN+1),            TCOST(NN+1),                    DMANN259
     1        (ANNCST(IS,IX),IX=5,9)                                    DMANN260
  104 FORMAT(1H0,A10,                  1X,F14.0,15X,F11.0,1X,F11.0,      DMANN261
     1                          1X,F6.0,1X,F6.0,14X,F6.0)               DMANN262
      PRINT 101                                                         DMANN263
      IZ=IZ+NN                                                          DMANN264
      RETURN                                                            DMANN265
      END                                                               DMANN266
```

```
RUN VERSION 2.1 --PSH LEVEL 298--                                        ANNUAL2

ANNUAL2

SUBPROGRAM LENGTH
000404

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
101   -  000334    102   -  000312    103   -  000320    104   -  000335

BLOCK NAMES AND LENGTHS
EF    -  000320

VARIABLE ASSIGNMENTS
AC    -  00021AC01 ACI   -  00021BC01 ANMEAD -  000102C01 ANNCST -  000001
AN2HED -  000210C01 ATYPE -  000000    CRF   -  000400    IS    -  000405
NN    -  000375    NZ    -  000376    PCT   -  000401    PTAC  -  000402
PTACI -  000403    XINT  -  000404    TYPNAM -  000008C01 UNTNM -  000172C01
XINT  -  000377

START OF CONSTANTS
000278

START OF TEMPORARIES
000351

START OF INDIRECTS
000367

UNUSED COMPILER SPACE
127208
```

```
RUN VERSION 2.3 --PSW LEVEL 298--                                    08/30/72

                FORTRAN VI SUBROUTINE PRCNT(I,J,M,UNITS,PERCNT)          DMPRCNT2
000010          DIMENSION UNITS(5,6,12),PERCNT(5,6,12)                   DMPRCNT3
000015          IIX=I+1                                                  DMPRCNT4
000012          JJX=J+1                                                  DMPRCNT5
000013          MMX=M+1                                                  DMPRCNT6
000014          DO 1 IA=1,IIX                                            DMPRCNT7
000017          DO 1 JA=1,JJA                                            DMPRCNT8
000022          DO 1 MA=1,MMA                                            DMPRCNT9
000024          IF (UNITS(IIA,JJX,MMX )) 2,2,5                           DMPRCNT10
000033        5 PERCNT(IX,JA,MX)=UNITS(IX,JA,MX)/UNITS(IIA,JJA,MMX)*100.0  DMPRCNT11
000053          GO TO 1                                                  DMPRCNT12
000053        2 PERCNT (IX,JA,MX ) = 0.0                                 DMPRCNT13
000061        1 CONTINUE                                                 DMPRCNT14
000072          IF (PERCNT(1)[X,JJX,MMX)) 6,6,4                          DMPRCNT15
000100        4 PERCNT(IIA,JJA,MMX)=100.0                                DMPRCNT16
000106        6 CONTINUE                                                 DMPRCNT17
000106          RETURN                                                   DMPRCNT18
000107          END                                                      DMPRCNT19
```

```
RUN VERSION 2.3 --PSW LEVEL 298--                PRCNT                 08/30/72

PRCNT

SUBPROGRAM LENGTH
000124

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
1       =  000041    2       =  000053    4       =  000100    5       =  000033
6       =  000104

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS
IIX     =  000120    IA      =  000123    JJX     =  000121    JX      =  000124
MMX     =  000122    MA      =  000125

START OF CONSTANTS
000110

START OF TEMPORARIES
000111

START OF INTERCTS
000114

UNUSED COMPILER SPACE
130100
```

```
                SUBROUTINE ANNUALD(IS,IZ,NN,TUNITS,TCOST,INTR,LIF,AUNT,ATYPE,ANNCS
               XT)
C
C               INPUT ARGUMENTS
C
C                 INTEGER  IS, INDEX FOR STORING RESULTS IN ANNCST
C                          IZ, INDEX FOR ACCESSING INT, LIF
C                          NN, NO. OF TABLE ENTRYS IN TUNITS,TCOST,ATYPE
C
C                 INTEGER ARRAYS  INT, INTEREST TABLE
C                                 LIF, LIFE TABLE
C
C                 REAL ARRAYS  TUNITS, NO. OF UNITS TABLE
C                              TCOST, TOTAL COST FOR UNITS IN TUNITS
C
C                 LITERAL  AUNT, UNIT NAME (10 CHARACTERS)
C
C               OUTPUT ARGUMENTS
C
C                 REAL ARRAYS  ANNCST, OUTPUT AREA FOR ENTIRE TABLE
C                                (15 TO (5*NN)   ) TO 14)
C
C               COMMON
C
C                 LITERAL ARRAYS  ANMEAN, TABLE COLUMN HEADINGS, ALPHA STRINGS
C
C
                DIMENSION INTR(10),LIF(30),TUNITS(5),TCOST(5),ANNCST(8,14),AUNT(5)
000015          X        ,ATYPE(5)
000015          COMMON /EF/ TYPNAM(1,6), ANMEAD(14,4), UNTNM(10),AC(10),ACI(10)
                X ,ANPMED(14,4)
000015          PRINT 101
000026     1n1  FORMAT(1H ,135H-----------------------------------------------------------
                X-----------------------------------------------------------
                X-----------------)
000028          PRINT 102,ANMEAN
00002A     1n2  FORMAT(1H ,13A10,A5)
00002A          PRINT 101
000032          ANNCST(IS,1)=ATYPE(NN+1)
000041          ANNCST(IS,2)=AUNT(NN+1)
000044          ANNCST(IS,3)=TUNITS(NN+1)
00004A          ANNCST(IS,4)=TCOST(NN+1)
000050          ANNCST(IS,5)=0.0
000057          ANNCST(IS,6)=0.0
000053          ANNCST(IS,7)=0.0
000064          ANNCST(IS,8)=0.0
00005A          ANNCST(IS,12)=100.0
000057          ANNCST(IS,13)=100.0
00006A          ANNCST(IS,14)=100.0
000067          DO 1 NX=1,NN
000067          NZ=NX+IZ-1
00006E          RINT=INTR(NZ)/100.0
000071          CRF=(RINT*(1.0+RINT)**LIF(NZ))/((1.0+RINT)**LIF(NZ)-1.0)
000103          AC(NX)=TCOST(NX)/LIF(NZ)
000112          ACI(NX)=TCOST(NX)*CRF
```



```
ANNUALD

SUBPROGRAM LENGTH
000514

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
101    =  000747    1u2    =  00041A    1n3    =  00041A    1n4    =  000433

BLOCK NAMES AND LENGTHS
EF     =  000370

VARIABLE ASSIGNMENTS
AC     =  00020A(C) ACI    =  00021C(C) ACU    =  000504    ACUI   =  000505
ANMEAN =  00012C(C) ANNCST =  000003    ANPMED =  00023C(C) ATYPE  =  000002
AUNT   =  000001    CRF    =  000502    CU     =  000503    IS     =  000513
KUNIT  =  000511    LIF    =  000000    NX     =  000477    NZ     =  000500
PCT    =  000504    PTAC   =  000507    PTACI  =  000510    RINT   =  000512
TYPNAM =  00008C(C) UNTNM  =  00017C(C) AINT   =  000501

START OF CONSTANTS
000304

START OF TEMPORARIES
000447

START OF INDIRECTS
000460

UNUSED COMPILER SPACE
126720
```



```
000115          ANNCST(IS,7)=ANNCST(IS,7)+AC(NX)
00012F          ANNCST(IS,8)=ANNCST(IS,8)+ACI(NX)
00123  1        CONTINUE
00124           DO 2 NX=1,NN
00124           CU=TCOST(NX)/TUNITS(NX)
000131          ACU=AC(NX)/TUNITS(NX)
000134          ACUI=ACI(NX)/TUNITS(NX)
00013A          PCT=CU/TCOST(NX)*100.0
000147          PTAC=AC(NX)/ANNCST(IS,7)*100.0
00014A          PTACI=ACI(NX)/ANNCST(IS,8)*100.0
000152          KUNIT=TUNITS(NX)
000155          NZ=IZ+NX-1
000157          RINT=INTR(NZ)
000162        2 PRINT 103,ATYPE(NX),AUNT(NX),KUNIT,TCOST(NX),LIF(NZ),RINT,
               X        AC(NX),ACI(NX),CU,ACU,ACUI,PCT,PTAC,PTACI
000267    1n3 FORMAT(1H0,A10,1X,A10,1X,I8,1X,F14.0,1X,I6,1X,F8.2,1X,F11.0,1X,
               X     F11.0,1X,F9.0,1X,F9.0,1X,F8.0,1X,F8.0,1X,F8.0)
000267          ANNCST(IS,9)=ANNCST(IS,4)/ANNCST(IS,3)
00026A          ANNCST(IS,10)=ANNCST(IS,7)/ANNCST(IS,3)
000271          ANNCST(IS,11)=ANNCST(IS,8)/ANNCST(IS,3)
000275          KUNIT=TUNITS(NN+1)
000277          PRINT 101
000303          PRINT 104,ATYPE(NN+1),AUNT(NN+1),KUNIT,TCOST(NN+1),
               X        (ANNCST(IS,I),I=7,14)
000352    1n4 FORMAT(1H0,A10,1X,A10,1X,I8,1X,F14.0,15X,F11.0,1X,F11.0,1X,F9.0,
               X        1X,F9.0,1X,F9.0,1X,F8.0,1X,F8.0,1X,F8.0)
000352          PRINT 101
00035A          IZ=IZ+NN
000363          RETURN
000363          END
```

OVERLAY 2.00
(NTWK)

This overlay contains the following subroutine:

```
                OVERLAY(NTWK,2,6)                                                DNET   2
                FORTRAN VI PROGRAM NTWORK                                        DNET   3
000003          INTEGER PARCK3                                                   DNET   4
000003          REAL NPKHRS,NUMDA,NOPHRS                                         DNET   5
000003          DIMENSION  HITP(4) , PKPCT(3)                                    DNET   6
000003          DIMENSION  DOFPC1(4,4,4) , DOFVFL(150,4,4) , DONPC1(4,4,4)  ,    DNET   7
               X    DONPC2(4,4,4) , DONVFL(225,4,4) , DPDLYI(2,4)    ,           DNET   8
               X  DVHOCC(2,4)   , TOPDLY(4,4,2)   , TOVDLY(4,4,2)   ,           DNET   9
               X    DADPHT(4,4)   ,  DADVHT(4,4)   ,  DAVPTL(3,4)   ,           DNET  10
               X    DAVPTT(3,4)   , DAVVTL(3,4)   ,  DAVVTT(3,4)   ,            DNET  11
               X  DOFPMT(150,4,4) , DONPHT(225,4,4) , DONPMT(225,4,4),          DNET  12
               X    DTOPHT(4,5)   ,DOFPMT(150,4,4) , DTOVHT(4,5)   ,            DNET  13
               X  DONVHT(225,4,4), DONDAT(4,4,4), DONLKC(225,5)                 DNET  14
000003          COMMON/RDI/        JSPRNT(80)    , DLIFE(30)     , DINT(30)     ,  DNET  15
               X   LFCODE        , NVM           , NLM           , NSM          ,  DNET  16
               X   NIM           , NTM           , NTSM          , NYM          ,  DNET  17
               X   NDM           , IZ            , JK            , PARCK3(6)    ,  DNET  18
               X   NH            , NX            , IPRINT        , IXV          ,  DNET  19
               X   IPOLCD(4,2,3), IDACCD(3)     , NHV           , NDTM         ,  DNET  20
               X   NVC(9)                                                        DNET  21
000003          COMMON/RDR/        SCALEF(10)    , DVHCAP(5)       , DVHLDF(5)      ,DNET  22
               X DLNCAP(5)       , ULANDC(10)    , UTRAKC          , USTRUC(4)      ,DNET  23
               X DGWDTH(16)      , DCNCFN(11)    , DWAYCC(5)       , DYARDS(5,13)   ,DNET  24
               X DMAINT(10)      , DVHSUC(4,6)   , DVHUC(4)        , DNUMVH(5)      ,DNET  25
               X ASSRAT(4)       , TAXRAT(4)     , DNOISE(20,14,2), RESDEN(4,7)     ,DNET  26
               X DONPFL(225,4,4), DGLKST(225,5) , BACKON(225,3,3), DPKHRF(225,3)  ,DNET  27
               X DPKDRF(225,3)   ,                                                DNET  28
               X NPKHRS          , NUMDA         , DPKPCT          , DPTRIP(4,3)    ,DNET  29
               X DPDLYO(2,3)     , DVDLAY(2,3)   , DMVLDF(2,3)     , XSYSCH(4,5)    ,DNET  30
               X XPKPCT          , XPTRIP(4,4)   , XDELAY(2,4,2)   , HLNDIR(2,2)    ,DNET  31
               X XVMTPC(4)       , DVMTPC(3)     , AUTOCH(11)      , HNCHAR(4)      ,DNET  32
               X DPCVCH(3,6,2)   , XPCVCH(4,6)   , HPCVCH(2,6)     , POLRAT(3,3)    ,DNET  33
               X DMSTAC          , HACRAT(2,5,2) , XACRAT(4,4)     , DACRAT(2,3)    ,DNET  34
               X HOPDAT(2,4)     , HVOPCS(2,6)   , HWYOPC(2,3)     , XOPCU(4,7,4)   ,DNET  35
               X XOPICU(4,7,4)   ,DOFPFL(150,4,4),DOFLKC(150,3,3) ,BACKOF(150,3,2),DNET  36
               X DYUOPC          , DCCOPC        , DMOPC           , Z(4)           ,DNET  37
               X DVOPUC(6,9,4)   , DTROPC(11,7)  , DGOPC(7)        , FLOCAP(5)      ,DNET  38
               X DFARE(4,2,2)                                                    DNET  39
000003          COMMON /OV23/  NOPHRS  , DVTRIP(4,4)  , DTOPMT(4,5)  ,DTOVMT(4,5),DNET  40
               X  DRHDAT(6,3,4) , DLVDAT(6,3,4)  ,  DOFVHT(150,4,4)  ,           DNET  41
               X DOFVMT(150,4,4),  DONVMT(225,4,4)                               DNET  42
000003          COMMON/SVR/ HSAVDT(100,2), SMRYDT(320,5)                         DNET  43
000003          COMMON /MM/ MMINDX                                               DNET  44
000003          EQUIVALENCE  (DONLKC(1),DGLKST(1))                               DNET  45
000003          EQUIVALENCE (DONPMT(1),DOFPMT(1)),                               DNET  46
               X  (DONPHT(1),DOFPHT(1))     ,                                    DNET  47
               X   (DONVFL(1),DOFVFL(1))      ,                                  DNET  48
               X    (DONVHT(1),DOFVHT(1))                                       DNET  49
000003          DATA  HITP /10H  PEAK     ,10H  OFF PEAK,10H  DAILY    ,10H  ANNUAL DNET  50
               X /                                                               DNET  51
000003          MM = MMINDX                                                      DNET  52
000005          LV = NVM                                                         DNET  53
000006          LK = NLM                                                         DNET  54
          C                                                                      DNET  55
```

```
          C       ZERO  OUT  COMPUTATION  ARRAYS                                DNET  56
          C                                                                    DNET  57
000010          DO 1  LKX = 1,225                                              DNET  58
000013          DO  1  LVX = 1,4                                               DNET  59
000016          DO  1 ITPX = 1,4                                               DNET  60
000021          DONPHT(LKX,LVX,ITPX)  = 0.0                                    DNET  61
000026          DONPMT(LKX,LVX,ITPX)  = 0.0                                    DNET  62
000033          DONVFL(LKX,LVX,ITPX)  = 0.0                                    DNET  63
000040          DONVHT(LKX,LVX,ITPX)  = 0.0                                    DNET  64
000045        1 DONVMT(LKX,LVX,ITPX)  = 0.0                                    DNET  65
000061          DO 5  ITPX = 1,4                                               DNET  66
000064          DO 5  LVX = 1,4                                                DNET  67
000067          DADPHT(ITPX,LVX)=0.0                                           DNET  68
000072          DADVHT(ITPX,LVX)=0.0                                           DNET  69
000074        5 DVTRIP(ITPX,LVX) = 0.0                                         DNET  70
000103          DO 3 ITPX = 1,2                                                DNET  71
000106          DO 3 LVX = 1,4                                                 DNET  72
000111        3 DVHOCC(ITPX,LVX) = 0.0                                         DNET  73
          C                                                                    DNET  74
          C                                                                    DNET  75
000122          DO 10  ITPX = 1,2                                              DNET  76
000125          DO 10 LVX=1,LV                                                 DNET  77
000130       10 DVHOCC(ITPX,LVX) = DVHCAP(LVX) * DMVLDF(ITPX,LVX)              DNET  78
000145          NOPHRS=(24.-NPKHRS)                                            DNET  79
          C     APPLY SCALE FACTORS TO DM PERSON TRIPS                         DNET  80
000147          DO 35 LVX = 1,LV                                               DNET  81
000152          DO 35 ITPX = 1,2                                               DNET  82
000155          NTX = LVX + 4                                                  DNET  83
000157       35 DPTRIP(ITPX,LVX) = DPTRIP(ITPX,LVX) * SCALEF(1) * SCALEF(NTX)  DNET  84
          C     CALCULATE DM PEAK  PERCENTAGE  FACTORS PER DM VEHICLE TYPE     DNET  85
000172          DO  15  LVX= 1,LV                                              DNET  86
000175       15 PKPCT(LVX) =  DPTRIP(1,LVX) / (DPTRIP(1,LVX) + DPTRIP(2,LVX))  DNET  87
          C     CALCULATE  ON-GDWY LINK LOADINGS PER PEAK AND OFF PEAK         DNET  88
000211          DO  20 LKX = 1,LK                                              DNET  89
000214          DO  20 LVX = 1,LV                                              DNET  90
000217          DONPFL (LKX,LVX,1) = DONPFL(LKX,LVX,3) *  PKPCT(LVX)           DNET  91
000225       20 DONPFL(LKX,LVX,2) =  DONPFL(LKX,LVX,3) * (1.0 - PKPCT(LVX))    DNET  92
          C     APPLY SCALE FACTORS TO OFF-GDWY LINK LOADINGS AND CALCULATE    DNET  93
          C      LOADINGS PER PEAK AND OFF-PEAK                                DNET  94
000243          DO  30 JKX = 1,JK                                              DNET  95
000246          DO 30 LVX = 1,LV                                               DNET  96
000251          NTX = LVX + 4                                                  DNET  97
000253          DOFPFL(JKX,LVX,3) = DOFPFL(JKX,LVX,3) * SCALEF(1) * SCALEF(NTX) DNET  98
000260          DOFPFL (JKX,LVX,1) = DOFPFL(JKX,LVX,3) * PKPCT(LVX)            DNET  99
000266       30  DOFPFL(JKX,LVX,2) =DOFPFL(JKX,LVX,3) * (1.0 - PKPCT(LVX))    DNET 100
          C     CALCULATE  DM ON-GDWY PMT-PHT PER LINK                         DNET 101
000304          DO 65  ITPX =1,2                                               DNET 102
000307          DO 65  LVX = 1,LV                                              DNET 103
000312          DO 65  LKX = 1,LK                                              DNET 104
000315          DONPMT(LKX,LVX,ITPX) = DONPFL(LKX, LVX,ITPX) * DONLKC (LKX, 2) DNET 105
000326       65 DONPHT(LKX,LVX,ITPX) = DONPMT(LKX,LVX,ITPX) / DONLKC (LKX,1)   DNET 106
          C     SUM ARRAYS FOR DAILY AND COMPUTE ANNUAL                        DNET 107
000350          DO 70 LVX = 1,LV                                               DNET 108
000353          DO 70 LKX = 1,LK                                               DNET 109
000356          DONPFL(LKX,LVX,3) = DONPFL(LKX,LVX,1)+ DONPFL(LKX,LVX,2)       DNET 110
```

```
000364          DONPMT(LKX,LVX,3) = DONPMT(LKX,LVX,1)+ DONPMT(LKX,LVX,2)          DNET 111
000370          DONPHT(LKX,LVX,3) = DONPHT(LKX,LVX,1)+ DONPHT(LKX,LVX,2)          DNET 112
000374          DONPFL(LKX,LVX,4) = DONPFL(LKX,LVX,3) * NUMDA                     DNET 113
000377          DONPMT(LKX,LVX,4) = DONPMT(LKX,LVX,3) * NUMDA                     DNET 114
000402       70 DONPHT(LKX,LVX,4) = DONPHT(LKX,LVX,3) * NUMDA                     DNET 115
                                                                                  DNET 116
000416          LVY = LV + 1                                                      DNET 117
000420          DO 71 LKX = 1,LK                                                  DNET 118
000423          DO 71 ITPX = 1,4                                                  DNET 119
000426       71 DONPFL(LKX,LVY,ITPX) = 0.0                                        DNET 120
          C     SUM ARRAYS FOR ALL LK  AND FOR ALL LV                             DNET 121
000442          LVY = LV +1                                                       DNET 122
000444          LKY = LK+1                                                        DNET 123
000445          DO 75 ITPX =1,4                                                   DNET 124
000450          DO 75 LKX  =1,LK                                                  DNET 125
000453          DO 75 LVX  =1,LV                                                  DNET 126
000456          DONPFL(LKX,LVY,ITPX) =DONPFL(LKX,LVY,ITPX)+ DONPFL(LKX,LVX,ITPX)  DNET 127
000470          DONPMT(LKX,LVY,ITPX) =DONPMT(LKX,LVY,ITPX)+ DONPMT(LKX,LVX,ITPX)  DNET 128
000477       75 DONPHT(LKX,LVY,ITPX) =DONPHT(LKX,LVY,ITPX)+ DONPHT(LKX,LVX,ITPX)  DNET 129
000521          DO 77 ITPX= 1,4                                                   DNET 130
000524          DO 77 LVX = 1,LVY                                                 DNET 131
000527          DO 77 LKX = 1,LK                                                  DNET 132
000532          DONPMT(LKY,LVX,ITPX)= DONPMT(LKY,LVX,ITPX)+ DONPMT(LKX,LVX,ITPX)  DNET 133
000544       77   DONPHT(LKY,LVX,ITPX)= DONPHT(LKY,LVX,ITPX)+ DONPHT(LKX,LVX,ITPX)DNET 134
          C                                                                       DNET 135
          C     ZERO OUT  DBHDAT AND  DLVDAT  ARRAYS  ************************** DNET 136
000566          DO 40  ND = 1,6                                                   DNET 137
000571          DO 40 ITPX = 1,4                                                  DNET 138
000574          DO 40  LVX = 1,3                                                  DNET 139
000577          DBHDAT(ND,LVX,ITPX) = 0.0                                         DNET 140
000604       40 DLVDAT(ND,LVX,ITPX) = 0.0                                         DNET 141
          C     CALCULATE  NUMBER OF LOADED VEHICLES  PER LINK  ****************  DNET 142
000620          DO 45 ITPX =1,2                                                   DNET 143
000623          DO 45 LKX  =1,LK                                                  DNET 144
000626          DO 45 LVX  =1,LV                                                  DNET 145
000631       45 DONVFL (LKX,LVX,ITPX) = DONPFL(LKX,LVX,ITPX) / DVHOCC(ITPX,LVX)   DNET 146
          C     CALCULATE        BACKHAUL VMT - VHT FOR ALL ON-GDY LINKS          DNET 147
000656          DO 50 ITPX = 1,2                                                  DNET 148
000661          NTP = ITPX + 1                                                    DNET 149
000663          DO 50 LVX = 1,LV                                                  DNET 150
000666          DO 50 LKX = 1,LK                                                  DNET 151
000671          DBHDAT(1,LVX,ITPX) = DBHDAT(1,LVX,ITPX)+ (DONVFL(LKX,LVX,ITPX) *  DNET 152
               X BACKON(LKX,LVX,NTP) * DONLKC(LKX,2))                             DNET 153
000710          DBHDAT(2,LVX,ITPX) = DBHDAT(2,LVX,ITPX) + ((DONVFL(LKX,LVX,ITPX)* DNET 154
               X BACKON(LKX,LVX,NTP) * DONLKC(LKX,2)) / DONLKC(LKX,1))            DNET 155
000726       50 CONTINUE                                                          DNET 156
          C     CALCULATE LOADED VEH. VMT-VHT FOR ALL ON-GDWY LINKS               DNET 157
000736          DO 55  ITPX =1,2                                                  DNET 158
000741          DO 55  LVX = 1,LV                                                 DNET 159
000744          DO 55  LKX = 1,LK                                                 DNET 160
000747          DLVDAT (1,LVX,ITPX) = DLVDAT(1,LVX,ITPX)+ (DONVFL(LKX,LVX,ITPX) * DNET 161
               X     DONLKC(LKX,2))                                              DNET 162
000763          DLVDAT (2,LVX,ITPX) = DLVDAT(2,LVX,ITPX) + ((DONVFL(LKX,LVX,ITPX)*DNET 163
               X     DONLKC(LKX,2)) / DONLKC (LKX,1))                            DNET 164
000776       55 CONTINUE                                                          DNET 165
          C     CALCULATE  LOADED + BACKHAUL VEHS.-VMT-VHT PER ON-GDWY LINK       DNET 166
```

```
001006          DO 60 ITPX = 1,2                                          DNET 166
001011          NTP = ITPX + 1                                            DNET 167
001013          DO 60 LVX = 1,LV                                          DNET 168
001016          DO 60 LKX  = 1,LK                                         DNET 169
001021          DONVFL(LKX,LVX,ITPX)= DONVFL(LKX,LVX,ITPX)+(DONVFL(LKX,LVX,ITPX)* DNET 170
               X BACKON(LKX,LVX,NTP))                                     DNET 171
001033          DONVMT (LKX,LVX,ITPX) =  DONVFL (LKX,LVX,ITPX)* DONLKC(LKX,2)  DNET 172
001043       60 DONVHT(LKX,LVX,ITPX) = DONVMT(LKX,LVX,ITPX) / DONLKC(LKX,1)  DNET 173
          C                                                               DNET 174
          C     SUM VEH.-VMT-VHT ARRAYS -ONGDWY FOR DAY AND COMPUTE ANNUAL  DNET 175
          C                                                               DNET 176
001065          DO  80 LVX = 1,LV                                         DNET 177
001070          DO  80 LKX = 1,LK                                         DNET 178
001073          DONVFL(LKX,LVX,3) = DONVFL(LKX,LVX,1) + DONVFL(LKX,LVX,2) DNET 179
001101          DONVMT(LKX,LVX,3) = DONVMT(LKX,LVX,1) + DONVMT(LKX,LVX,2) DNET 180
001106          DONVHT(LKX,LVX,3) = DONVHT(LKX,LVX,1) + DONVHT(LKX,LVX,2) DNET 181
001111          DONVFL(LKX,LVX,4) = DONVFL(LKX,LVX,3) *NUMDA              DNET 182
001114          DONVMT(LKX,LVX,4) = DONVMT(LKX,LVX,3) * NUMDA             DNET 183
001120       80 DONVHT(LKX,LVX,4) = DONVHT(LKX,LVX,3) * NUMDA             DNET 184
          C     SUM ARRAYS FOR ALL LK AND ALL LV                          DNET 185
001134          DO  85  ITPX=1,4                                          DNET 186
001137          DO  85  LKX =1,LK                                         DNET 187
001142          DO  85  LVX =1,LV                                         DNET 188
001145          DONVFL(LKX,LVY,ITPX) = DONVFL(LKX,LVY,ITPX) + DONVFL(LKX,LVX,ITPX)DNET 189
001157          DONVMT(LKX,LVY,ITPX) = DONVMT(LKX,LVY,ITPX)+ DONVMT(LKX,LVX,ITPX) DNET 190
001167       85 DONVHT(LKX,LVY,ITPX)= DONVHT(LKX,LVY,ITPX)+ DONVHT(LKX,LVX,ITPX)  DNET 191
001211          DO 90 ITPX= 1,4                                           DNET 192
001214          DO 90 LVX = 1,LVY                                         DNET 193
001217          DO 90 LKX = 1,LK                                          DNET 194
001222          DONVMT(LKY,LVX,ITPX)=DONVMT(LKY,LVX,ITPX)+ DONVMT(LKX,LVX,ITPX)  DNET 195
001234       90 DONVHT(LKY,LVX,ITPX)=DONVHT(LKY,LVX,ITPX)+ DONVHT(LKX,LVX,ITPX)  DNET 196
          C                                                               DNET 197
          C     CALCULATE DBHDAT-DLVDAT ARRAYS PER DAY AND ANNUAL           DNET 198
          C                                                               DNET 199
001256          DO 95 NDX =1,2                                            DNET 200
001261          DO 95 LVX =1,LV                                           DNET 201
001264          DBHDAT(NDX,LVX,3)= DBHDAT(NDX,LVX,1) + DBHDAT(NDX,LVX,2)  DNET 202
001272          DLVDAT(NDX,LVX,3)= DLVDAT(NDX,LVX,1) + DLVDAT(NDX,LVX,2)  DNET 203
001276          DBHDAT(NDX,LVX,4)= DBHDAT(NDX,LVX,3) * NUMDA              DNET 204
001300       95 DLVDAT(NDX,LVX,4)= DLVDAT(NDX,LVX,3) * NUMDA              DNET 205
          C                                                               DNET 206
          C     CALCULATE ON GDWY PERCENTS AND SET  SET UP TOTAL PMT-PHT-VMT-VHT  DNET 207
          C       ARRAYS                                                  DNET 208
          C                                                               DNET 209
001313          DO 100  ITPX =1,4                                         DNET 210
001316          DO 100  LVX = 1,LV                                        DNET 211
001321          DONPCl(1,LVX,ITPX)=(DONPMT(LKY,LVX,ITPX)/DONPMT(LKY,LVY,ITPX)) *  DNET 212
               X  100.                                                    DNET 213
001336          DONPCl(2,LVX,ITPX)= (DONPHT(LKY,LVX,ITPX)/DONPHT(LKY,LVY,ITPX)) * DNET 214
               X  100.                                                    DNET 215
001351          DONPCl(3,LVX,ITPX)= (DONVMT(LKY,LVX,ITPX)/DONVMT(LKY,LVY,ITPX)) * DNET 216
               X  100.                                                    DNET 217
001364          DONPCl(4,LVX,ITPX)= (DONVHT(LKY,LVX,ITPX)/DONVHT(LKY,LVY,ITPX)) * DNET 218
               X  100.                                                    DNET 219
001377      100  CONTINUE                                                 DNET 220
```

```
001404          DO 105 ITPX =1,4                                        DNET 221
001407          DO 105 LVX =1,LVY                                       DNET 222
001412          DTOPHT(ITPX,LVX) = DONPHT(LKY,LVX,ITPX)                 DNET 223
001421          DTOPMT(ITPX,LVX) = DONPMT(LKY,LVX,ITPX)                 DNET 224
001430          DTOVMT(ITPX,LVX) = DONVMT(LKY,LVX,ITPX)                 DNET 225
001436      105 DTOVHT(ITPX,LVX) = DONVHT(LKY,LVX,ITPX)                 DNET 226
          C                                                             DNET 227
          C     ZERO  OUT  DONDAT  ARRAY                                DNET 228
001452          DO 104 NDX=1,4                                          DNET 229
001455          DO 104  LVX=1,4                                         DNET 230
001460          DO 104  ITPX=1,4                                        DNET 231
001463      104 DONDAT(NDX,LVX,ITPX) = 0.0                              DNET 232
          C                                                             DNET 233
          C     COMPUTE ON-GOWY DATA ARRAY  DONDAT(ND,LV,ITP) - (4,3,4) DNET 234
          C        ND=1  PMT            ND=3  VMT    THIS IS SUMMATION  ON  DNET 235
          C        ND=2  PHT            ND=4  VHT            ALL LINKS  LK  DNET 236
001477          DO 106 ITPX= 1,4                                        DNET 237
001502          DO 106 LVX = 1,LV                                       DNET 238
001505          DONDAT(1,LVX,ITPX)  = DONPMT(LKY,LVX,ITPX)              DNET 239
001514          DONDAT(2,LVX,ITPX)  = DONPHT(LKY,LVX,ITPX)              DNET 240
001523          DONDAT(3,LVX,ITPX)  = DONVMT(LKY,LVX,ITPX)              DNET 241
001531      106 DONDAT(4,LVX,ITPX) = DONVHT(LKY,LVX,ITPX)               DNET 242
          C     SUM  DONDAT  ARRAY  FOR  ALL  VEHICLES  (LV = LVY)      DNET 243
001546          DO 107  NDX=1,4                                         DNET 244
001551          DO 107  ITPX=1,4                                        DNET 245
001554          DO 107  LVX=1,LV                                        DNET 246
001557      107 DONDAT(NDX,LVY,ITPX) = DONDAT(NDX,LVY,ITPX) + DONDAT(NDX,LVX,ITPX)DNET 247
          C       END ALL ON-GOWAY DATA CALCULATIONS                    DNET 248
          C                                                             DNET 249
          C                                                             DNET 250
          C                                                             DNET 251
001600          IF (JSPRNT (60) .EQ. 1)  GO TO   9001                   DNET 252
001602          PRINT 8501                                              DNET 253
001605          PRINT 8502 ,  NPKHRS,  NOPHRS,  NUMDA                   DNET 254
001617          PRINT 8503                                              DNET 255
001623          PRINT 8504 , (DVHCAP(LVX),LVX =1,LV)                    DNET 256
001636          PRINT 8505,  ((DMVLDF(LA,LB),LB=1,3),LA=1,2)            DNET 257
001654          PRINT 8506,  ((DVHOCC(LA,LB),LB=1,3),LA=1,2)            DNET 258
001672          PRINT 8507,  (PKPCT(LVX),LVX = 1,LV)                    DNET 259
001705     8507 FORMAT (1H0,4X,*PERCENT PERSONS/DAY IN PEAK PERIOD*,2X,3F10.3)  DNET 260
001705     8501 FORMAT(1H1,50X,28HDUAL-MODE  NETWORK  ANALYSIS,///)     DNET 261
001705     8502 FORMAT (1H0,10X,*PEAK HRS/DAY*,F8.1,6X,*OFF PEAK HRS/DAY*,F8.1,6X DNET 262
               X,*DAYS/YR.*,F8.1)                                       DNET 263
001705     8503 FORMAT (1H0,1X,*A. DUAL MODE  VEHICLE  CHARACTERISTICS*,6X,*VEH.1*DNET 264
               X, 5X,*VEH.2*,5X,*VEH.3*,5X,*VEH.4*,/)                   DNET 265
001705     8504 FORMAT (1H ,4X,*VEHICLE  CAPACITY*,19X,4F10.1)          DNET 266
001705     8505 FORMAT(1H0,4X,*VEHICLE  LOAD FACTOR*,6X,*PEAK*,6X,3F10.3,/,1H0,26 DNET 267
               XX,*OFF PEAK*,6X,3F10.3)                                 DNET 268
001705     8506 FORMAT (1H0,4X,*VEHICLE  OCCUPANCY*,8X,*PEAK*,6X,3F10.1,/,1H0,26X DNET 269
               X,*OFF PEAK*,6X,3F10.1)                                  DNET 270
001705     9001 CONTINUE                                                DNET 271
          C                                                             DNET 272
          C                                                             DNET 273
001705          IF (JSPRNT (61) .EQ. 1)  GO TO   9002                   DNET 274
001707          PRINT  8601                                             DNET 275
```

```
001713    8601 FORMAT (1H1,45X,*DM  OFF-GUIDEWAY  LINK  CHARACTERISTICS*,/)      DNET 276
001713         PRINT  8602                                                       DNET 277
001717    8602 FORMAT (1H ,46X,*(ONE-WAY TRAVEL TIME AND DISTANCE)*,/)           DNET 278
001717         DO  495  LVX = 1,LV                                               DNET 279
001723         PRINT  8603 , LVX                                                 DNET 280
001730    8603 FORMAT (1H0,10X,*DUAL-MODE VEHICLE *,I3,//)                       DNET 281
001730         PRINT  8604                                                       DNET 282
001734    8604 FORMAT (1H0,12X,*1*,11X,*2*,11X,*3*,11X,*4*,11X,*5*,11X,*6*,11X,*7DNET 283
              X*,11X,*8*,11X,*9*,11X,*10*)                                       DNET 284
001734         PRINT  8605                                                       DNET 285
001740    8605 FORMAT (1H0,7X,10(* MILES MIN. *),/)                              DNET 286
001740         NL =(JK/10)                                                       DNET 287
001743         NLC = 0                                                           DNET 288
001744         DO  493  NLX =1, NL                                               DNET 289
001750         JKX = NLC +1                                                      DNET 290
001752         JKXX = NLC +10                                                    DNET 291
001754         PRINT  8606 , NLC,((DOFLKC (J,LVX,LC),LC=1,2), J= JKX,JKXX)       DNET 292
001777     493 NLC = NLC + 10                                                    DNET 293
002004         JKX = NLC +1                                                      DNET 294
002006     495 PRINT  8606 , NLC, ((DOFLKC (J,LVX,LC),LC=1,2), J = JKX, JK)      DNET 295
002035    8606 FORMAT (1H ,1X,I3,3X,10(F5.1,F6.1,1X),/)                          DNET 296
        C      ADD  THIS  ABOVE  STATEMENT  499 ********************************DNET 297
002035    9002 CONTINUE                                                         DNET 298
002035         IF (LV .EQ. 1)  GO TO 498                                         DNET 299
002037         LVYY = LV + 1                                                     DNET 300
002040         GO TO  499                                                        DNET 301
002041     498 LVYY = 1                                                          DNET 302
        C      ************************************************************      DNET 303
        C                                                                        DNET 304
002042     499 IF (JSPRNT (62) .EQ. 1)  GO TO   9003                             DNET 305
        C                                                                        DNET 306
002044         PRINT 8701                                                        DNET 307
002050    8701 FORMAT (1H1,45X,*DUAL-MODE  ON-GUIDEWAY  LINK  DATA*,//)          DNET 308
002050         DO  510 LVX = 1,     LVYY                                         DNET 309
002054         IF (LVX .EQ. LVY)   GO TO 500                                     DNET 310
002056         PRINT  8703 , LVX                                                 DNET 311
002063    8703 FORMAT (1H0,10X,*DUAL-MODE VEHICLE*,I3,/)                         DNET 312
002063         GO TO 502                                                         DNET 313
002064     500 PRINT  8702                                                       DNET 314
002070    8702 FORMAT (1H0,10X,*ALL DUAL-MODE VEHICLES*,/)                       DNET 315
002070     502 PRINT  8704                                                       DNET 316
002074    8704 FORMAT (1H ,26X,*PEAK*,32X,*OFF-PEAK*,36X,*DAILY*)                DNET 317
002074         PRINT 8705                                                        DNET 318
002100    8705 FORMAT (1H0,1X,*LINK*,8X,*PERSONS*,8X,*PMT*,7X,*PHT*,9X,*PERSONS*,DNET 319
              X8X,*PMT*,7X,*PHT*,11X,*PERSONS*,12X,*PMT*,10X,*PHT*,/)            DNET 320
002100         DO  505  LKX =1,LK                                                DNET 321
002104     505 PRINT  8706, LKX,  (DONPFL (LKX,LVX,ITPX),DONPMT(LKX,LVX,ITPX),DONDNET 322
              XPHT (LKX,LVX,ITPX), ITPX =1,3)                                    DNET 323
002140    8706 FORMAT (1H ,2X,I3,2(4X,F11.0,F12.0,F10.0),4X,F14.0,F16.0,F12.0)   DNET 324
002140         PRINT  8707, (DONPMT (LKY,LVX,ITPX), DONPHT (LKY,LVX,ITPX),ITPX =1DNET 325
              X,3)                                                               DNET 326
002162    8707 FORMAT (1H0,1X,*TOTAL*,14X,F12.0,F10.0,15X,F12.0,F10.0,18X,F16.0, DNET 327
              XF12.0,/)                                                          DNET 328
002162     510 CONTINUE                                                          DNET 329
002165         PRINT  8701                                                       DNET 330
```

```
002170          DO 525  LVX =1,LVYY                                          DNET 331
002174          IF (LVX .EQ. LVY)  GO TO 515                                 DNET 332
002176          PRINT 8703 , LVX                                             DNET 333
002203          GO TO  516                                                   DNET 334
002204      515 PRINT  8702                                                  DNET 335
002210      516 PRINT  8704                                                  DNET 336
002214          PRINT  8708                                                  DNET 337
002220     8708 FORMAT (1H0,1X,*LINK*,7X,*VEHICLES*,8X,*VMT*,7X,*VHT*,8X,*VEHICLESDNET 338
            X*,8X,*VMT*,7X,*VHT*,10X,*VEHICLES*,12X,*VMT*,10X,*VHT*,/)       DNET 339
002220          DO 520  LKX = 1,LK                                           DNET 340
002224      520 PRINT 8706 ,LKX,   (DONVFL (LKX,LVX,ITPX), DONVMT (LKX,LVX,ITPX), DNET 341
            XDONVHT (LKX,LVX,ITPX) , ITPX = 1,3)                             DNET 342
002260      525 PRINT 8707, (DONVMT (LKY,LVX,ITPX) , DONVHT (LKY,LVX,ITPX) , ITPX=DNET 343
            X1,3)                                                            DNET 344
002306     9003 CONTINUE                                                     DNET 345
         C                                                                   DNET 346
         C                                                                   DNET 347
         C       BEGIN  OFF-GUIDEWAY  DATA  CALCULATIONS                     DNET 348
         C                                                                   DNET 349
         C                                                                   DNET 350
002306          DO 2  JKX = 1,150                                            DNET 351
002312          DO 2 LVX = 1,4                                               DNET 352
002315          DO 2 ITPX = 1,4                                              DNET 353
002320          DOFPHT(JKX,LVX,ITPX) = 0.0                                   DNET 354
002325          DOFPMT(JKX,LVX,ITPX) = 0.0                                   DNET 355
002332          DOFVFL(JKX,LVX,ITPX) = 0.0                                   DNET 356
002337          DOFVHT(JKX,LVX,ITPX) = 0.0                                   DNET 357
002344        2 DOFVMT(JKX,LVX,ITPX) = 0.0                                   DNET 358
         C     CALCULATE  OFF GDWAY 1-WAY TRAVEL YTIME IN HOURS               DNET 359
002360          DO 110  JKX = 1,JK                                           DNET 360
002363          DO 110  LVX = 1,LV                                           DNET 361
002366      110 DOFLKC (JKX,LVX,3) = DOFLKC(JKX,LVX,2) /  60.0               DNET 362
         C            CALCULATE  OFF GDWAY  PMT-PHT  PER  LINK               DNET 363
002405          DO 115 ITPX =1,2                                             DNET 364
002410          DO 115 LVX = 1,LV                                            DNET 365
002413          DO 115 JKX = 1,JK                                            DNET 366
002416          DOFPMT(JKX,LVX,ITPX)= DOFPFL(JKX,LVX,ITPX)* DOFLKC(JKX,LVX,1) DNET 367
002430      115 DOFPHT(JKX,LVX,ITPX)= DOFPFL(JKX,LVX,ITPX)* DOFLKC(JKX,LVX,3) DNET 368
         C      SUM  ARRAYS  FOR  DAILY  AND COMPUTE ANNUAL                  DNET 369
002452          DO 120 LVX =1,LV                                             DNET 370
002455          DO 120 JKX =1,JK                                             DNET 371
002460          DOFPFL(JKX,LVX,3)= DOFPFL(JKX,LVX,1) + DOFPFL(JKX,LVX,2)     DNET 372
002466          DOFPMT(JKX,LVX,3) = DOFPMT(JKX,LVX,1) + DOFPMT(JKX,LVX,2)    DNET 373
002472          DOFPHT(JKX,LVX,3)= DOFPHT(JKX,LVX,1) + DOFPHT(JKX,LVX,2)     DNET 374
002476          DOFPFL(JKX,LVX,4) = DOFPFL(JKX,LVX,3) * NUMDA                DNET 375
002501          DOFPMT(JKX,LVX,4)= DOFPMT(JKX,LVX,3) *  NUMDA                DNET 376
002504      120 DOFPHT(JKX,LVX,4)= DOFPHT(JKX,LVX,3) * NUMDA                 DNET 377
         C      SUM  ARRAYS FOR ALL JK AND ALL LV   ***************************** DNET 378
002520          JKY  = JK+1                                                  DNET 379
002522          LVY = LV+1                                                   DNET 380
002523          DO 125  ITPX =1,4                                            DNET 381
002526          DO 125  JKX = 1,JK                                           DNET 382
002531          DO 125  LVX = 1,LV                                           DNET 383
002534          DOFPFL(JKX,LVY,ITPX)= DOFPFL(JKX,LVY,ITPX) + DOFPFL(JKX,LVX,ITPX) DNET 384
002546          DOFPMT(JKX,LVY,ITPX)= DOFPMT(JKX,LVY,ITPX)+DOFPMT(JKX,LVX,ITPX)  DNET 385
```

```
002555      125 DOFPHT(JKX,LVY,ITPX)= DOFPHT(JKX,LVY,ITPX)+DOFPHT(JKX,LVX,ITPX)   DNET 386
002577          DO 126 ITPX=1,4                                                   DNET 387
002602          DO 126 LVX=1,LVY                                                  DNET 388
002605          DO 126 JKX=1,JK                                                   DNET 389
002610          DOFPMT (JKY,LVX,ITPX)= DOFPMT(JKY,LVX,ITPX)+ DOFPMT(JKX,LVX,ITPX) DNET 390
002622      126 DOFPHT(JKY,LVX,ITPX)= DOFPHT(JKY,LVX,ITPX)+ DOFPHT(JKX,LVX,ITPX)  DNET 391
          C     CALCULATE  NO. OF OFF-GDWY LOADED  VEHS. PER LINK JK              DNET 392
002644          DO 135 ITPX=1,2                                                   DNET 393
002647          DO 135 JKX =1,JK                                                  DNET 394
002652          DO 135 LVX =1,LV                                                  DNET 395
002655      135 DOFVFL(JKX,LVX,ITPX)= DOFPFL(JKX,LVX,ITPX) / DVHOCC(ITPX,LVX)     DNET 396
          C     CALCULATE BACKHAUL VMT-VHT FOR ALL OFF-GDWY  LINKS                DNET 397
002702          DO 140 ITPX=1,2                                                   DNET 398
002705          DO 140 LVX=1,LV                                                   DNET 399
002710          DO 140 JKX=1,JK                                                   DNET 400
002713          DBHDAT(3,LVX,ITPX)= DBHDAT(3,LVX,ITPX)+ (DOFVFL(JKX,LVX,ITPX) *   DNET 401
               X    BACKOF (JKX,LVX,ITPX) * DOFLKC(JKX,LVX,1) )                   DNET 402
002733          DBHDAT(4,LVX,ITPX)= DBHDAT(4,LVX,ITPX) + (DOFVFL(JKX,LVX,ITPX) *  DNET 403
               X    BACKOF (JKX,LVX,ITPX) * DOFLKC(JKX,LVX,3) )                   DNET 404
002747      140 CONTINUE                                                          DNET 405
          C     CALCULATE  LOADED  VEH. VMT-VHT FOR ALL OFF-GDWY  LINKS           DNET 406
002757          DO 145 ITPX =1,2                                                  DNET 407
002762          DO 145 LVX  =1,LV                                                 DNET 408
002765          DO 145 JKX  =1,JK                                                 DNET 409
002770          DLVDAT(3,LVX,ITPX) =DLVDAT(3,LVX,ITPX)+ (DOFVFL(JKX,LVX,ITPX) *   DNET 410
               X    DOFLKC(JKX,LVX,1) )                                           DNET 411
003004          DLVDAT(4,LVX,ITPX) = DLVDAT(4,LVX,ITPX) +(DOFVFL(JKX,LVX,ITPX) *  DNET 412
               X    DOFLKC(JKX,LVX,3) )                                           DNET 413
003016      145 CONTINUE                                                          DNET 414
          C     CALCULATE  LOADED +BACKHAUL VEHS-VMT-VHT PER OFF-GDWY  LINK       DNET 415
003026          DO 150  ITPX=1,2                                                  DNET 416
003031          DO 150  LVX =1,LV                                                 DNET 417
003034          DO 150  JKX =1,JK                                                 DNET 418
003037          DOFVFL(JKX,LVX,ITPX)= DOFVFL(JKX,LVX,ITPX)+ (DOFVFL(JKX,LVX,ITPX) DNET 419
               X     * BACKOF(JKX,LVX,ITPX)    )                                  DNET 420
003051          DOFVMT(JKX,LVX,ITPX) = DOFVFL(JKX,LVX,ITPX)* DOFLKC(JKX,LVX,1)    DNET 421
003061      150 DOFVHT(JKX,LVX,ITPX) = DOFVFL(JKX,LVX,ITPX)* DOFLKC(JKX,LVX,3)    DNET 422
          C     SUM OFF-GDWY VEH-VMT-VHT ARRAYS FOR DAILY AND COMPUTE  ANNUAL     DNET 423
003103          DO 155 LVX =1,LV                                                  DNET 424
003106          DO 155 JKX =1,JK                                                  DNET 425
003111          DOFVFL(JKX,LVX,3)= DOFVFL(JKX,LVX,1) + DOFVFL(JKX,LVX,2)          DNET 426
003117          DOFVMT(JKX,LVX,3)= DOFVMT(JKX,LVX,1) + DOFVMT(JKX,LVX,2)          DNET 427
003124          DOFVHT(JKX,LVX,3)= DOFVHT(JKX,LVX,1) + DOFVHT(JKX,LVX,2)          DNET 428
003127          DOFVFL(JKX,LVX,4)= DOFVFL(JKX,LVX,3) * NUMDA                      DNET 429
003132          DOFVMT(JKX,LVX,4)= DOFVMT(JKX,LVX,3) * NUMDA                      DNET 430
003136      155 DOFVHT(JKX,LVX,4)= DOFVHT(JKX,LVX,3) * NUMDA                      DNET 431
          C                                                                       DNET 432
          C     SUM ARRAYS FOR ALL JK AND ALL LV                                  DNET 433
          C                                                                       DNET 434
003152          DO 160 ITPX = 1,4                                                 DNET 435
003155          DO 160 JKX =1,JK                                                  DNET 436
003160          DO 160 LVX =1,LV                                                  DNET 437
003163          DOFVFL(JKX,LVY,ITPX)= DOFVFL(JKX,LVY,ITPX) + DOFVFL(JKX,LVX,ITPX) DNET 438
003175          DOFVMT(JKX,LVY,ITPX)= DOFVMT(JKX,LVY,ITPX) + DOFVMT(JKX,LVX,ITPX) DNET 439
003205      160 DOFVHT(JKX,LVY,ITPX)= DOFVHT(JKX,LVY,ITPX)+ DOFVHT(JKX,LVX,ITPX)  DNET 440
```

```
003227          DO 165 ITPX=1,4                                          DNET 441
003232          DO 165 LVX = 1,LVY                                       DNET 442
003235          DO 165 JKX = 1,JK                                        DNET 443
003240          DOFVMT(JKY,LVX,ITPX)=DOFVMT(JKY,LVX,ITPX)+ DOFVMT(JKX,LVX,ITPX)  DNET 444
003252      165 DOFVHT(JKY,LVX,ITPX)= DOFVHT(JKY,LVX,ITPX)+ DOFVHT(JKX,LVX,ITPX)  DNET 445
          C                                                              DNET 446
          C     CALCULATE DBHDAT - DLVDAT ARRAYS PER DAY AND COMPUTE  ANNUAL  DNET 447
          C                                                              DNET 448
003274          DO 170 NDX =3,4                                          DNET 449
003277          DO 170 LVX =1,LV                                         DNET 450
003302          DBHDAT(NDX,LVX,3) =DBHDAT(NDX,LVX,1) + DBHDAT(NDX,LVX,2)  DNET 451
003310          DLVDAT(NDX,LVX,3) =DLVDAT(NDX,LVX,1) + DLVDAT(NDX,LVX,2)  DNET 452
003314          DBHDAT(NDX,LVX,4) = DBHDAT(NDX,LVX,3) * NUMDA            DNET 453
003316      170 DLVDAT(NDX,LVX,4) = DLVDAT(NDX,LVX,3) * NUMDA            DNET 454
          C                                                              DNET 455
          C     CALCULATE OFF-GDWY PERCENTS  NAND  TOTAL PMT-PHT-VMT-VHT ARRAYS  DNET 456
          C                                                              DNET 457
          C                                                              DNET 458
003331          DO 175 ITPX=1,4                                          DNET 459
003334          DO 175 LVX =1,LV                                         DNET 460
003337          DOFPCl(1,LVX, ITPX) = (DOFPMT(JKY,LVX, ITPX)/ DOFPMT(JKY,LVY,ITPX)DNET 461
               X ) * 100.                                                DNET 462
003354          DOFPCl(2,LVX,ITPX) = (DOFPHT(JKY,LVX,ITPX)/DOFPHT(JKY,LVY,ITPX))  DNET 463
               X   * 100.                                                DNET 464
003367          DOFPCl(3,LVX,ITPX) = (DOFVMT(JKY,LVX,ITPX)/DOFVMT(JKY,LVY,ITPX))  DNET 465
               X   * 100.                                                DNET 466
003402          DOFPCl(4,LVX,ITPX) = (DOFVHT(JKY,LVX,ITPX)/DOFVHT(JKY,LVY,ITPX))  DNET 467
               X   * 100.                                                DNET 468
003415      175 CONTINUE                                                 DNET 469
003422          DO  180  ITPX=1,4                                        DNET 470
003425          DO  180  LVX =1,LVY                                      DNET 471
003430          DTOPHT(ITPX,LVX) = DTOPHT(ITPX,LVX) + DOFPHT(JKY,LVX,ITPX)  DNET 472
003440          DTOPMT(ITPX,LVX) = DTOPMT(ITPX,LVX) + DOFPMT(JKY,LVX,ITPX)  DNET 473
003447          DTOVMT(ITPX,LVX) = DTOVMT(ITPX,LVX) + DOFVMT(JKY,LVX,ITPX)  DNET 474
003456          DTOVHT(ITPX,LVX) = DTOVHT(ITPX,LVX) + DOFVHT(JKY,LVX,ITPX)  DNET 475
003465      180 CONTINUE                                                 DNET 476
          C                                                              DNET 477
          C     END  ALL OFF- GUIDEWAY CALCULATIONS                      DNET 478
          C                                                              DNET 479
          C     COMPUTE  ON+OFF GDWY VMT-VHT FOR ARRAYS DBHDAT - DLVDAT  DNET 480
          C                                                              DNET 481
003472          DO 200 ITPX =1,4                                         DNET 482
003475          DO 200 LVX  =1,LV                                        DNET 483
003500          DBHDAT(5,LVX,ITPX)= DBHDAT(1,LVX,ITPX) + DBHDAT(3,LVX,ITPX)  DNET 484
003511          DBHDAT(6,LVX,ITPX)= DBHDAT(2,LVX,ITPX) + DBHDAT(4,LVX,ITPX)  DNET 485
003515          DLVDAT(5,LVX,ITPX)= DLVDAT(1,LVX,ITPX) + DLVDAT(3,LVX,ITPX)  DNET 486
003521      200 DLVDAT(6,LVX,ITPX)= DLVDAT(2,LVX,ITPX) + DLVDAT(4,LVX,ITPX)  DNET 487
          C                                                              DNET 488
          C                                                              DNET 489
003537          DO 390 ITPX=1,3                                          DNET 490
003542          DO 390 LVX=1,LV                                          DNET 491
003545      390 DVTRIP(ITPX,LVX)=DPTRIP(ITPX,LVX)/DVHOCC(ITPX,LVX)       DNET 492
003563          DO 400 LVX=1,LV                                          DNET 493
003566          DPTRIP(3,LVX)=DPTRIP(1,LVX)+DPTRIP(2,LVX)                DNET 494
003573          DVTRIP(3,LVX)=DVTRIP(1,LVX)+DVTRIP(2,LVX)                DNET 495
```

```
003576          DPTRIP(4,LVX)=DPTRIP(3,LVX)*NUMDA                                DNET 496
003600      400 DVTRIP(4,LVX)=DVTRIP(3,LVX)*NUMDA                                DNET 497
003610          DO 410 ITPX=1,2                                                  DNET 498
003613          DO 410 LVX=1,LV                                                  DNET 499
003616          TOVDLY(ITPX,LVX,1)=DVTRIP(ITPX,LVX)*DVDLAY(ITPX,LVX)             DNET 500
003624      410 TOVDLY(ITPX,LVX,2)=TOVDLY(ITPX,LVX,1)/60.0                       DNET 501
003636          DO 420 LVX=1,LV                                                  DNET 502
003641          TOVDLY(3,LVX,1)=TOVDLY(1,LVX,1)+TOVDLY(2,LVX,1)                  DNET 503
003646          TOVDLY(4,LVX,1)=TOVDLY(3,LVX,1)*NUMDA                            DNET 504
003650          TOVDLY(3,LVX,2)=TOVDLY(3,LVX,1)/60.0                             DNET 505
003653      420 TOVDLY(4,LVX,2)=TOVDLY(4,LVX,1)/60.0                             DNET 506
003663          DO 430 ITPX=1,2                                                  DNET 507
003666          DO 430 LVX=1,LV                                                  DNET 508
003671      430 DPDLYI(ITPX,LVX)=DVDLAY(ITPX,LVX)/DVHOCC(ITPX,LVX)               DNET 509
003706          DO 440 ITPX=1,2                                                  DNET 510
003711          DO 440 LVX=1,LV                                                  DNET 511
003714          TOPDLY(ITPX,LVX,1)=DPTRIP(ITPX,LVX)*(DPDLYO(ITPX,LVX)+           DNET 512
               XDPDLYI(ITPX,LVX))                                                DNET 513
003724      440 TOPDLY(ITPX,LVX,2)=TOPDLY(ITPX,LVX,1)/60.0                       DNET 514
003736          DO 450 LVX=1,LV                                                  DNET 515
003741          TOPDLY(3,LVX,1)=TOPDLY(1,LVX,1)+ TOPDLY(2,LVX,1)                 DNET 516
003746          TOPDLY(4,LVX,1)=TOPDLY(3,LVX,1)*NUMDA                            DNET 517
003750          TOPDLY(3,LVX,2)=TOPDLY(3,LVX,1)/60.0                             DNET 518
003753      450 TOPDLY(4,LVX,2)=TOPDLY(4,LVX,1)/60.0                             DNET 519
003763          DO 460 ITPX=1,4                                                  DNET 520
003766          DO 460  LVX=1,LV                                                 DNET 521
003771          DADPHT(ITPX,LVX)=DTOPHT(ITPX,LVX)+TOPDLY(ITPX,LVX,2)             DNET 522
003777      460 DADVHT(ITPX,LVX)=DTOVHT(ITPX,LVX)+TOVDLY(ITPX,LVX,2)             DNET 523
004012          DO 470 ITPX=1,3                                                  DNET 524
004015          DO 470 LVX=1,LV                                                  DNET 525
004020          DAVPTL(ITPX,LVX)=DTOPMT(ITPX,LVX)/DPTRIP(ITPX,LVX)               DNET 526
004026          DAVVTL(ITPX,LVX)=DTOVMT(ITPX,LVX)/DVTRIP(ITPX,LVX)               DNET 527
004033          DAVPTT (ITPX,LVX)=DADPHT(ITPX,LVX)/DPTRIP(ITPX,LVX)              DNET 528
004040      470 DAVVTT (ITPX,LVX)=DADVHT(ITPX,LVX)/DVTRIP(ITPX,LVX)              DNET 529
004053          DO 480 ITPX =1,4                                                 DNET 530
004056          DO 480 LVX  =1,LVY                                               DNET 531
004061          DONPC2(1,LVX,ITPX)=  (1.0-(DOFPMT(JKY,LVX,ITPX)/DTOPMT(ITPX,LVX)))DNET 532
               X      *100.                                                      DNET 533
004075          DONPC2(2,LVX,ITPX)=  (1.0-(DOFPHT(JKY,LVX,ITPX)/DTOPHT(ITPX,LVX)))DNET 534
               X      *100.                                                      DNET 535
004107          DONPC2(3,LVX,ITPX)=  (1.0-(DOFVMT(JKY,LVX,ITPX)/DTOVMT(ITPX,LVX)))DNET 536
               X      *100.                                                      DNET 537
004121          DONPC2(4,LVX,ITPX)=  (1.0-(DOFVHT(JKY,LVX,ITPX)/DTOVHT(ITPX,LVX)))DNET 538
               X      *100.                                                      DNET 539
004133      480 CONTINUE                                                         DNET 540
004140          SMRYDT ( 82,MM) =  DADPHT(3,1)                                   DNET 541
004143          SMRYDT ( 84,MM) =  DADVHT(3,1)                                   DNET 542
004146          SMRYDT ( 86,MM) =  DADPHT(4,1)                                   DNET 543
004151          SMRYDT ( 88,MM) =  DADVHT(4,1)                                   DNET 544
004154          SMRYDT ( 90,MM) =  DADPHT(3,2)                                   DNET 545
004157          SMRYDT ( 92,MM) =  DADVHT(3,2)                                   DNET 546
004162          SMRYDT ( 94,MM) =  DADPHT(4,2)                                   DNET 547
004165          SMRYDT ( 96,MM) =  DADVHT(4,2)                                   DNET 548
          C     SECTION  D  CONTROLLED  BY PARCK3(4)                             DNET 549
          C                                                                      DNET 550
```

```
004170          IF (JSPRNT (63) .EQ. 1)  GO TO   9004                          DNET 551
004172          PRINT  8801                                                    DNET 552
004176     8801 FORMAT (1H1,45X,*DUAL-MODE  OFF-GUIDEWAY  LINK  DATA*,//)      DNET 553
004176          DO  540 LVX =  1, LVYY                                         DNET 554
004202          IF (LVX .EQ. LVY)  GO TO 530                                   DNET 555
004204          PRINT  8703 , LVX                                              DNET 556
004211          GO TO 532                                                      DNET 557
004212      530 PRINT  8702                                                    DNET 558
004216      532 PRINT  8704                                                    DNET 559
004222          PRINT  8705                                                    DNET 560
004226          DO  535  JKX = 1,JK                                            DNET 561
004232      535 PRINT  8706 , JKX,  (DOFPFL (JKX,LVX,ITPX), DOFPMT (JKX,LVX,ITPX),DNET 562
               XDOFPHT (JKX,LVX,ITPX), ITPX = 1,3)                             DNET 563
004266      540 PRINT  8707 , (DOFPMT (JKY,LVX,ITPX), DOFPHT (JKY,LVX,ITPX), ITPX=DNET 564
               X1,3)                                                           DNET 565
004314          PRINT  8801                                                    DNET 566
004317          DO 550  LVX = 1, LVYY                                          DNET 567
004323          IF (LVX .EQ. LVY)   GO TO 543                                  DNET 568
004325          PRINT 8703 , LVX                                               DNET 569
004332          GO TO 544                                                      DNET 570
004333      543 PRINT 8702                                                     DNET 571
004337      544 PRINT 8704                                                     DNET 572
004343          PRINT 8708                                                     DNET 573
004347          DO 548  JKX =1,JK                                              DNET 574
004353      548 PRINT  8706 , JKX ,(DOFVFL(JKX,LVX,ITPX), DOFVMT (JKX,LVX,ITPX),  DNET 575
               XDOFVHT (JKX,LVX,ITPX) ,ITPX = 1,3)                             DNET 576
004407      550 PRINT  8707,  (DOFVMT (JKY,LVX,ITPX), DOFVHT (JKY,LVX,ITPX), ITPX=DNET 577
               X1,3)                                                           DNET 578
004435     9004 CONTINUE                                                       DNET 579
            C                                                                  DNET 580
            C                                                                  DNET 581
004435          IF (JSPRNT (64) .EQ. 1)  GO TO   9005                          DNET 582
004437          PRINT  8805                                                    DNET 583
004443     8805 FORMAT (1H1,45X,*DUAL-MODE NETWORK SUMMARY DATA*,//)           DNET 584
004443          PRINT  8806                                                    DNET 585
004447     8806 FORMAT (1H0,54X,*ON-GUIDEWAY*,/)                               DNET 586
004447          PRINT  8807                                                    DNET 587
004453     8807 FORMAT (1H0,2X,*TEMPORAL   DM VEH.*,12X,*PMT*,6X,*PCT. OF*,10X,*PHDNET 588
               XT*,5X,*PCT. OF*,10X,*VMT*,5X,*PCT. OF*,10X,*VHT*,5X,*PCT. OF*)  DNET 589
004453          PRINT  8808                                                    DNET 590
004457     8808 FORMAT (1H ,3X,*PERIOD*,14X,4(17X,*ALL VEH.*),/)               DNET 591
004457          DO   570  ITPX = 1,4                                           DNET 592
004463          PRINT  8810 , HTTP (ITPX)                                      DNET 593
004470     8810 FORMAT (1H0,1A10)                                              DNET 594
004470          DO   560  LVX = 1,LV                                           DNET 595
004474      560 PRINT 8812 ,LVX, DONDAT(1,LVX,ITPX), DONPC1(1,LVX,ITPX) ,      DNET 596
               X  DONDAT(2,LVX,ITPX), DONPC1(2,LVX,ITPX), DONDAT(3,LVX,ITPX) ,  DNET 597
               X DONPC1(3,LVX,ITPX), DONDAT(4,LVX,ITPX), DONPC1(4,LVX,ITPX)    DNET 598
004540     8812 FORMAT (1H0,18X,I2,5X,4(F15.0,F8.2,2X) )                       DNET 599
004540          IF (LV .EQ. 1)  GO TO 570                                      DNET 600
004542          PRINT 8813 , (DONDAT(ND,LVY,ITPX),ND=1,4)                      DNET 601
004556     8813 FORMAT (1H0,14X,*TOTAL*,6X,4(F15.0,*  100.00  *))              DNET 602
004556      570 CONTINUE                                                       DNET 603
004561          PRINT  8805                                                    DNET 604
004564          PRINT  8815                                                    DNET 605
```

```
004570    8815 FORMAT (1H0,54X,*OFF-GUIDEWAY*,/)                                DNET 606
004570         PRINT  8807                                                      DNET 607
004574         PRINT  8808                                                      DNET 608
004600         DO  590  ITPX = 1,4                                              DNET 609
004604         PRINT  8810 , HITP (ITPX)                                        DNET 610
004611         DO  580  LVX  = 1,LV                                             DNET 611
004615     580 PRINT  8812 , LVX, DOFPMT (JKY,LVX,ITPX), DOFPC1 (1,LVX,ITPX), DOFDNET 612
           XPHT (JKY,LVX,ITPX), DOFPC1 (2,LVX,ITPX) , DOFVMT (JKY,LVX,ITPX) , DNET 613
           XDOFPC1 (3,LVX,ITPX), DOFVHT(JKY,LVX,ITPX),DOFPC1 (4,LVX,ITPX)     DNET 614
004671         IF (LV .EQ. 1)  GO  TO 590                                       DNET 615
004673         PRINT  8813 , DOFPMT (JKY,LVY,ITPX), DOFPHT (JKY,LVY,ITPX) , DOFVMDNET 616
           XT (JKY,LVY,ITPX) , DOFVHT (JKY,LVY,ITPX)                            DNET 617
004720     590 CONTINUE                                                         DNET 618
      C                                                                         DNET 619
004723         PRINT  8805                                                      DNET 620
004726         PRINT  8818                                                      DNET 621
004732    8818 FORMAT (1H0,50X,*ON AND OFF GUIDEWAY*,/)                         DNET 622
004732         PRINT  8819                                                      DNET 623
004736    8819 FORMAT (1H0,2X,*TEMPORAL   DM VEH.*,12X,*PMT*,6X,*PCT. ON*,10X,*PHDNET 624
           XT*,5X,*PCT. ON*,10X,*VMT*,5X,*PCT. ON*,10X,*VHT*,5X,*PCT. ON*)      DNET 625
004736         PRINT  8820                                                      DNET 626
004742    8820 FORMAT (1H ,3X,*PERIOD*,14X,4(17X,*GUIDEWAY*),/)                 DNET 627
004742         DO 600 ITPX = 1,4                                                DNET 628
004746         PRINT  8810 , HITP (ITPX)                                        DNET 629
004753         DO 595  LVX = 1,LV                                               DNET 630
004757     595 PRINT  8812 , LVX , DTOPMT (ITPX,LVX), DONPC2 (1,LVX,ITPX), DTOPHTDNET 631
           X(ITPX,LVX) , DONPC2 (2,LVX,ITPX) , DTOVMT (ITPX,LVX), DONPC2 (3,LVDNET 632
           XX,ITPX) , DTOVHT (ITPX,LVX) , DONPC2 (4,LVX,ITPX)                   DNET 633
005023         IF (LV .EQ. 1)  GO TO 600                                        DNET 634
005025         PRINT 8821, DTOPMT(ITPX,LVY),DONPC2(1,LVY,ITPX), DTOPHT(ITPX,LVY),DNET 635
           X  DONPC2(2,LVY,ITPX), DTOVMT(ITPX,LVY), DONPC2(3,LVY,ITPX),         DNET 636
           X  DTOVHT(ITPX,LVY) , DONPC2(4,LVY,ITPX)                             DNET 637
005061    8821 FORMAT (1H0,14X,*TOTAL*,6X, 4(F15.0,F8.2,2X))                    DNET 638
005061     600 CONTINUE                                                         DNET 639
005064         PRINT 8900                                                       DNET 640
005067    8900 FORMAT (1H1,50X,* DUAL-MODE  BACKHAUL  DATA *,//)                DNET 641
005067         PRINT 8901                                                       DNET 642
005073    8901 FORMAT (1H0,3X,*TEMPORAL   VEHICLE*,12X,*ON-GUIDEWAY*,20X,*OFF-GUIDNET 643
           XDEWAY*,26X,*TOTAL*/,1H,4X,*PERIOD*,9X,2(12X,*VMT*,12X,*VHT*),17X    DNET 644
           X,*VMT*,14X,*VHT*,/)                                                 DNET 645
005073         DO 8920 ITPX =1,4                                                DNET 646
005077         PRINT 8810, HITP(ITPX)                                           DNET 647
005104         DO 8920 LVX =1,LV                                                DNET 648
005110         PRINT 8915, LVX, (DBHDAT(NDX,LVX,ITPX),NDX=1,6)                  DNET 649
005126    8915 FORMAT (1H0,19X,I2,2(F17.0,F13.0),F20.0,F17.0)                   DNET 650
005126    8920 CONTINUE                                                         DNET 651
005133         PRINT 8925                                                       DNET 652
005136    8925 FORMAT (//,1H ,50X,* DUAL-MODE LOADED VEHICLE DATA*,//)          DNET 653
005136         PRINT 8901                                                       DNET 654
005142         DO 8930 ITPX =1,4                                                DNET 655
005146         PRINT 8810, HITP (ITPX)                                          DNET 656
005153         DO 8930 LVX =1,LV                                                DNET 657
005157         PRINT 8915 ,LVX,  (DLVDAT (NDX,LVX,ITPX),NDX =1,6)               DNET 658
005175    8930 CONTINUE                                                         DNET 659
005202    9005 CONTINUE                                                         DNET 660
```

```
          C                                                                    DNET 661
          C                                                                    DNET 662
005202          IF (JSPRNT (65) .EQ. 1)  GO TO   9006                          DNET 663
005204          PRINT  8822                                                    DNET 664
005210    8822 FORMAT (1H1,50X,*DUAL-MODE  TRIP  DATA *,//)                    DNET 665
005210          PRINT  8823                                                    DNET 666
005214    8823 FORMAT (1H0,3X,*DUAL-MODE TRIP DELAYS *)                        DNET 667
005214          PRINT  8824                                                    DNET 668
005220    8824 FORMAT (1H0,10X,*DM  VEHICLE*,9X,*PERSON DELAY*,18X,*PERSON DELAY*DNET 669
              X,18X,*VEHICLE DELAY*,/,1H ,30X,*OUT OF VEH.*,20X,*IN VEH.*,/)   DNET 670
005220          PRINT  8825                                                    DNET 671
005224    8825 FORMAT (1H ,28X,3(*PEAK*,7X,*OFF-PEAK*,11X) )                   DNET 672
005224          DO  610  LVX =1,LV                                             DNET 673
005230     610 PRINT  8826 , LVX, (DPDLYO (ITPX,LVX),ITPX=1,2),(DPDLYI (ITPX,LVX)DNET 674
              X,ITPX =1,2) , (DVDLAY (ITPX,LVX), ITPX =1,2)                    DNET 675
005271    8826 FORMAT (1H0,16X,I3,8X,3(F6.1,7X,F6.1,11X) )                     DNET 676
          C                                                                    DNET 677
          C                                                                    DNET 678
005271          PRINT  8827                                                    DNET 679
005274    8827 FORMAT (1H0,//,3X,*DUAL-MODE ADJUSTED  PHT - VHT  (INCLUDES DELAYSDNET 680
              X)*,//)                                                          DNET 681
005274          PRINT  8828                                                    DNET 682
005300    8828 FORMAT (1H ,5X,*DM VEHICLE*,15X,*PEAK*,20X,*OFF-PEAK*,17X,*DAILY*,DNET 683
              X23X,*ANNUAL* )                                                  DNET 684
005300          PRINT  8829                                                    DNET 685
005304    8829 FORMAT (1H0,15X,4(10X,*PHT*,10X,*VHT*),/)                       DNET 686
005304          DO 620  LVX =1,LV                                              DNET 687
005310     620 PRINT  8830 , LVX ,  (DADPHT (ITPX,LVX) ,DADVHT (ITPX,LVX) ,ITPX =DNET 688
              X1,4)                                                            DNET 689
005334    8830 FORMAT (1H0,7X,I3,7X,8F13.0)                                    DNET 690
          C                                                                    DNET 691
          C                                                                    DNET 692
005334          PRINT  8832                                                    DNET 693
005337    8832 FORMAT (1H1,3X,*DUAL-MODE PERSON - VEHICLE  TRIPS *,//)         DNET 694
005337          PRINT  8828                                                    DNET 695
005343          PRINT  8834                                                    DNET 696
005347    8834 FORMAT (1H ,16X,4(7X,*PERSON*,6X,*VEHICLE*),/)                  DNET 697
005347          DO  630  LVX =1,LV                                             DNET 698
005353     630 PRINT 8830 ,LVX , (DPTRIP(ITPX,LVX),DVTRIP(ITPX,LVX),ITPX =1,4) DNET 699
          C                                                                    DNET 700
          C                                                                    DNET 701
005377          PRINT  8836                                                    DNET 702
005402    8836 FORMAT (1H0,3X,*DUAL-MODE AVERAGE  TRIP LENGTH , TRIP TIME *,//) DNET 703
005402          PRINT  8837                                                    DNET 704
005406    8837 FORMAT (1H0,3X,*DM*,6X,*TEMPORAL*,11X,*TRIP LENGTH (MILES/TRIP)*, DNET 705
              X16X,*TRIP TIME (HOURS/TRIP)*,/,1H ,1X,*VEHICLE*,4X,*PERIOD*, )  DNET 706
005406          PRINT  8838                                                    DNET 707
005412    8838 FORMAT (1H ,32X,*PERSON*,8X,*VEHICLE*,20X,*PERSON*,7X,*VEHICLE*,/)DNET 708
005412          DO 650 LVX = 1,LV                                              DNET 709
005416          PRINT  8840 , LVX                                              DNET 710
005423    8840 FORMAT (1H0,3X,I3)                                              DNET 711
005423          DO 640 ITPX = 1,3                                              DNET 712
005427     640 PRINT  8842 , HITP(ITPX),  DAVPTL(ITPX,LVX), DAVVTL(ITPX,LVX) ,  DNET 713
              XDAVPTT (ITPX,LVX) , DAVVTT (ITPX,LVX)                           DNET 714
005455    8842 FORMAT (1H ,10X,1A10,10X,F10.2,4X,F10.2,16X,F10.2,4X,F10.2,/)   DNET 715
```



```
005455     650 CONTINUE                                                        DNET 716
005460    9006 CONTINUE                                                        DNET 717
005460          RETURN                                                         DNET 718
005462          END                                                            DNET 719
```

RUN VERSION 2.3 --PSR LEVEL 29A-- NTWORK 06/30/72
NTWORK

PROGRAM LENGTH INCLUDING I/O BUFFERS
034401

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

1	-	000045	2	-	002344	3	-	000111	5	-	000074
10	-	000130	15	-	000175	20	-	000225	30	-	000266
35	-	000157	40	-	000604	45	-	000631	50	-	000726
55	-	000776	60	-	001043	65	-	000326	70	-	000402
71	-	000426	75	-	000477	77	-	000544	80	-	001120
85	-	001167	90	-	001234	95	-	001300	100	-	001377
104	-	001463	105	-	001436	106	-	001531	107	-	001557
110	-	002366	115	-	002430	120	-	002504	125	-	002555
126	-	002622	135	-	002655	140	-	002747	145	-	003016
150	-	003061	155	-	003136	160	-	003205	165	-	003252
170	-	003316	175	-	003415	180	-	003465	200	-	003521
390	-	003545	400	-	003600	410	-	003624	420	-	003653
430	-	003671	440	-	003724	450	-	003753	460	-	003777
470	-	004040	480	-	004133	493	-	001777	495	-	002006
498	-	002041	499	-	002042	500	-	002064	502	-	002070
505	-	002104	510	-	002162	515	-	002204	516	-	002210
520	-	002224	525	-	002260	530	-	004212	532	-	004216
535	-	004232	540	-	004266	543	-	004333	544	-	004337
548	-	004353	550	-	004407	560	-	004474	570	-	004556
580	-	004615	590	-	004720	595	-	004757	600	-	005061
610	-	005230	620	-	005310	630	-	005353	640	-	005427
650	-	005455	8501	-	005503	8502	-	005511	8503	-	005522
8504	-	005535	8505	-	005543	8506	-	005555	8507	-	005474
8601	-	005567	8602	-	005576	8603	-	005604	8604	-	005611
8605	-	005623	8606	-	005630	8701	-	005635	8702	-	005651
8703	-	005644	8704	-	005656	8705	-	005664	8706	-	005701
8707	-	005710	8708	-	005720	8801	-	005744	8805	-	005753
8806	-	005761	8807	-	005765	8808	-	006003	8810	-	006011
8812	-	006014	8813	-	006021	8815	-	006027	8818	-	006033
8819	-	006040	8820	-	006056	8821	-	006064	8822	-	006133
8823	-	006140	8824	-	006145	8825	-	006163	8826	-	006170
8827	-	006175	8828	-	006205	8829	-	006216	8830	-	006223
8832	-	006227	8834	-	006235	8836	-	006243	8837	-	006252
8838	-	006270	8840	-	006300	8842	-	006303	8900	-	006071
8901	-	006077	8915	-	006117	8920	-	005126	8925	-	006125
8930	-	005175	9001	-	001705	9002	-	002035	9003	-	002306
9004	-	004435	9005	-	005202	9006	-	005460			

BLOCK NAMES AND LENGTHS

RDI	-	000307	RDR	-	034004	OV23	-	020631	SVR	-	003410
MM	-	000001									

VARIABLE ASSIGNMENTS

ASSRAT	-	000264C02	AUTOCH	-	021415C02	HACKOF	-	031470C02	HACKON	-	012575C02
DACRAT	-	021622C02	DADPHT	-	034062	DADVHT	-	034102	DAVPTL	-	034122

RUN VERSION 2.3 --PSR LEVEL 29A-- NTWORK 06/30/72

DAVPTT -	034134	DAVVTL -	034152	DAVVTT -	034166	DRMDAT -	000071C03
DPNCEN -	000076C02	DRANE -	033764C02	DGLKST -	010430C02	UGOPC -	033750C02
DGWDTH -	000050C01	DINT -	000150C01	DLIFE -	000129C01	ULMCAP -	000024C02
DLVDAT -	000291C03	DMAINT -	000211C02	DMVLDF -	021307C02	DMDISL -	000274C02
DMNMVH -	000257C02	DOFLKC -	026762C02	DOFPC1 -	033442	DOFPFL -	022222C02
DOFPHT -	015171	DOFPHT -	006353	DOFVFL -	024413	DOFVHT -	000311C03
DOFVHT -	005051C03	DONDAT -	034252	DONLKC -	010430C02	DONPC1 -	033542
DONPC2 -	033442	DONPFL -	001410C02	DONPHT -	015373	DONPHT -	006353
DONVFL -	024413	DONVHT -	000311C03	DONVHT -	011011C03	DPCVCH -	021434C02
DPDLYT -	033742	DWILYD -	021273C02	DRKDBF -	020011C02	DRKHRF -	010546C02
DPTRIP -	021257C02	DTUPHT -	034202	DTOPHT -	000021C03	DTOVHT -	034226
DTOVHT -	000044C03	DTWDPC -	033633C02	DVDLAY -	021301C02	DVHCAP -	000012C02
DVHLDF -	000017C02	DWHTCC -	033752	DVHSUC -	000223C02	DVMUC -	000253C02
DVMTPC -	021412C02	DWIPUC -	033383C02	DWTRIP -	000001C03	DWRYCC -	000103C02
DYARDS -	000119C02	ELUCAP -	033757C02	HACRAT -	021556C02	HITP -	033433
HLNDIR -	021412C02	HNCHAW -	021430C02	HOPDAT -	021030C02	HPCVCH -	021530C02
HVDRCK -	021444C02	HXTYPC -	021654C02	IDACCD -	000271C01	IPOLCO -	000241C01
ITPR -	034357	J -	034375	JK -	000226C01	JKR -	034361
JKRR -	034374	JKY -	034400	JSPHNT -	000000C01	LA -	034367
LR -	034376	LC -	034376	LK -	034354	LKR -	034355
LKY -	034343	LY -	034353	LYR -	034356	LYY -	034362
LYYY -	034377	MM -	034352	MMINDX -	000000C05	MO -	034364
NDIR -	034344	NL -	034371	NLC -	034372	NLM -	000216C01
NLR -	034373	NUPHRS -	000000C03	NPNHRS -	021254C02	NTP -	034365
NTR -	034340	NUMDA -	021255C02	NVC -	000274C01	NVM -	000215C01
PARCK2 -	000227C01	PARCT -	033437	POLRAT -	021544C02	RESNEW -	001354C02
RSAVHT -	000000C04	SCALFF -	000000C07	SHRYDT -	000410C04	TAXHAT -	008770C02
TOPDLY -	033742	TOVDLY -	034022	ULANDC -	000031C02	USTRUC -	000044C02
XACRAT -	021412C02	AVELAY -	021362C02	AOPCU -	021662C02	AOPICD -	022044C02
APCVCH -	021510C02	AWTRIP -	021342C02	ASYSCH -	021215C02	AVHTPC -	021406C02
Z -	033777C02						

START OF CONSTANTS
005444

START OF TEMPORARIES
006313

START OF INDIRECTS
006334

UNUSED COMPILER SPACE
104500

OVERLAY 3.00
(OTHR)

This overlay contains the following subroutines:

```
RUN VERSION 2.3 --PSR LEVEL 294--                                                    06/30/72

                OVERLAY (OTHR,3,0)                                                   OTHCNTL2
                FORTRAN VI PROGRAM ALLOTH                                            OTHCNTL3
000003          REAL  NPKHRS, NUMDA                                                  OTHCNTL4
000003          REAL  NOPHRS                                                         OTHCNTL5
000003          INTEGER  PARCK3                                                      OTHCNTL6
000003          COMMON/RD1/      JSPRNT(80)   , DLIFE(38)     , DINT(38)      ,      OTHCNTL7
                X   LECODE       , NVK        , WLM           , NSM           ,      OTHCNTL8
                X   MIM          , NTM        , NTSM          , NVM           ,      OTHCNTL9
                X   NOM          , IZ         , JK            , PARCK3(6)     ,      OTHCNTL10
                X   NM           , NE         , IPRINT        , IRV           ,      OTHCNTL11
                X   IPOLCD(4,2,3), IOACCD(3)  , NMV           , NOTM          ,      OTHCNTL12
                X   NVC(6)                                                           OTHCNTL13
000003          COMMON/RDR/      SCALEF(18)   , DVHCAP(5)     , DVHLDF(5)     ,      OTHCNTL14
                X DLNCAP(5)      , ULANDF(18)   , UTRASC(5)     , USTRUC(5)     ,      OTHCNTL15
                X NOWDTH(16)     , DCNCEN(11)   , DWATCC(5)     , DYARDS(5,13)  ,      OTHCNTL16
                X DWAINT(18)     , DVHBUF(4,6)  , DVHVC(4)                            OTHCNTL17
                X ASSMAT(6)      , TARRAT(4)    , DNOISE(20,4,2), DNUMVM(5)     ,      OTHCNTL18
                X DONPFL(225,4,4), DOLKST(225,5), BACKON(225,3,3), REMOEN(4,7)     ,  OTHCNTL19
                X NPKDRF(225,3)  ,                                                   OTHCNTL20
                X NPKHRS         , NUMDA        , DPKPCT        , DPTRIP(4,3)   ,      OTHCNTL21
                X DPDLYO(2,3)    , DVDLAY(2,3)  , DMVLDF(2,3)   , ASYSCH(4,5)   ,      OTHCNTL22
                X XPMPCT         , APTRIP(4,4)  , XDELAY(2,4,2) , NLMDIR(2,2)   ,      OTHCNTL23
                X XVMTPC(4)      , DVMTPC(3)    , AUTOCH(11)    , MMCMAR(6)     ,      OTHCNTL24
                X DPCVCM(3,6,2)  , APCVCU(4,4)  , MPCVCH(2,4)   , PDLRAT(3,3)   ,      OTHCNTL25
                X DMSTAC         , MACRAT(2,5,2), XACRAT(4,4)   , DACRAT(2,3)   ,      OTHCNTL26
                X MOPDAT(2,4)    , MVOPPC(2,6)  , MHYOPC(2,3)   , XOPCU(4,7,4)  ,      OTHCNTL27
                X KOPICD(4,7,4)  ,DOPPPL(150,4,4),MOPLKC(150,3,3),BACKOP(150,3,2)  ,  OTHCNTL28
                X DYUOPC         , DCCOPC       , DMOPC         , Z(4)          ,      OTHCNTL29
                X DVOPUC(6,9,4)  , DYMOPC(11,7) , DBOPC(7)      , FLOCAP(5)     ,      OTHCNTL30
                X DFARE(4,2,2)                                                       OTHCNTL31
000003          COMMON /OV13/ ASUMR(4,4), DTRTNI, DTLNMI, DTYDA                      OTHCNTL32
000003          COMMON /OV23/  NOPHRS  , DYTRIP(4,4)  , DTOPMT(4,6)  ,(DTOVMT(4,5),  OTHCNTL33
                X DUPVMT(150,6,4),  DONYMT(225,4,4)                                  OTHCNTL34
000003          COMMON /OV33/  AUTOMD(5,6,12)   ,  TRCKMD(5,6,12)  , XMDATA(5,6,12),  OTHCNTL35
                X  DONPWR(5,4,12) , DOPPWR(5,6,12) , XPWRC(5,6,12) , MPWRC(5,6,12),  OTHCNTL36
                X  DMVMT(5,6,12)                                                     OTHCNTL37
000003          COMMON/SVM/ RSAVDT(180,9), SMRYDT(320,5)                             OTHCNTL38
000003          COMMON /MM/ MMINDX                                                   OTHCNTL39
000003          DIMENSION  DONLKC(225,5)                                             OTHCNTL40
000003          EQUIVALENCE  (LV,NVM), (LK,WLM), (DONLKC(1),DOLKST(1))               OTHCNTL41
000003          CALL RNETWRK                                                         OTHCNTL42
000004          CALL MMYNET                                                          OTHCNTL43
000005          CALL POWER                                                           OTHCNTL44
000006          CALL AIRPOL                                                          OTHCNTL45
000007          CALL ACCDNT                                                          OTHCNTL46
000010          CALL MHYOP                                                           OTHCNTL47
000011          CALL TRNOP                                                           OTHCNTL48
000012          CALL DMOP                                                            OTHCNTL49
000013          RETURN                                                               OTHCNTL50
000015          END                                                                  OTHCNTL51
                C                                                                    OTHCNTL52
*FOLLOWING VARIABLES EQUIVALENCED BUT NOT REFERENCED                                 XNET  2
NVM
```

```
RUN VERSION 2.3 --PSR LEVEL 294--                                ALLOTH     06/30/72

ALLOTH

PROGRAM LENGTH INCLUDING I/O BUFFERS
000026

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS

BLOCK NAMES AND LENGTHS
RD1    =  000307    RDR    =  034804    OV13   =  000127    OV23   =  020631
OV33   =  00550#    SVM    =  003410    MM     =  000001

VARIABLE ASSIGNMENTS
ASSMAT =  000206C02 ASUMR  =  000000C04 AUTOCH =  021415C02 AUTOMD =  000000C05
BACKOP =  031470C02 BACKON =  012575C09 DACRAT =  021422C02 DBMDAT =  000071C04
DCNCEN =  000870C02 DFARE  =  033764C09 DOLKST =  010430C02 DBOPC  =  033750C02
DBWTH  =  000050C02 DMVMT  =  004730C04 DINT   =  000196C01 DLIFE  =  000120C01
DLNCAP =  000024C02 DLVDAT =  000201C04 DWAINT =  000411C02 DMVLDF =  021307C02
DNOISE =  000274C02 DNUMVM =  000057C09 DOPLKC =  026782C02 DOPPPL =  022222C02
DOPPWR =  002640C05 DOPVMT =  005311C04 DOPVMT =  005051C04 DONLKC =  010633C02
DONPFL =  031412C02 DONPWR =  002878C04 DOMVMT =  011011C04 DPCVCM =  021434C02
DPDLYO =  021293C02 DPKDRF =  022011C09 DPKHRF =  010546C02 DPTRIP =  021257C02
DTOPMT =  000021C04 DTOVMT =  000045C04 DTROPC =  033833C02 DVDLAY =  021301C02
DVMCAP =  000012C02 DVMLDF =  000017C09 DVMSUC =  000223C02 DVMUC  =  000253C02
DVMTPC =  021412C02 DVOPUC =  033303C09 DVTRIP =  000001C04 DWATCC =  000103C02
DYAMDS =  000110C02 FLOCAP =  033757C09 MICRAT =  021556C02 MLMDIR =  021402C02
MMCMAR =  021430C02 MOPDAT =  021430C09 MPCVCM =  021530C02 MPWRC  =  004100C05
MVOPCS =  021640C02 MHYOPC =  021054C09 IOACCD =  000271C01 IPOLCD =  000261C01
JSPRNT =  000000C01 NOPHRS =  000000C04 NPKHRS =  021254C02 NUMDA  =  021255C02
NVC    =  000276C01 PARCK3 =  000227C01 PDLRAT =  021544C02 RSAVDT =  001354C02
RSAVDT =  000000C08 SCALEF =  000000C09 SMRYDT =  000310C04 TARRAT =  000270C02
TRCKMD =  002550C05 ULANDC =  000031C09 USTRUC =  000044C02 XACRAT =  021602C02
XDELAY =  021362C02 XMDATA =  001320C04 XOPCU  =  021402C09 XOPICD =  022042C02
APCVCM =  021500C02 XPTRIP =  021342C09 XPWRC  =  003410C05 XSYSCH =  021315C02
XVMTPC =  021406C02 Z      =  033277C09

START OF CONSTANTS
000017

START OF TEMPORARIES
000024

START OF INDIRECTS
000026

UNUSED COMPILER SPACE
127380
```

```
      SUBROUTINE XNETWRK                                                XNET  3
C                                                                       XNET  4
      REAL  NPKHRS, NUMDA                                               XNET  5
      REAL  NOPHRS                                                      XNET  6
      INTEGER PARCK3                                                    XNET  7
C     DIMENSION STATEMENTS                                              XNET  8
      DIMENSION  XVTRIP(4,6), XAVTST(3,4,6),  XVHOCC(4,2) ,             XNET  9
     X  XNDPHR(5,2,2), APVDLY(4,4,3) , XNDVPC(5,7,6), XNDTPC(5,4,6)     XNET 10
      DIMENSION  MITP (4) , XNDT (6)                                    XNET 11
      DATA  MITP /10H  PEAK    ,10H  OFF-PEAK,10H  DAILY   ,10H  ANNUAL XNET 12
     X /                                                                XNET 13
      DATA  XNDT /10H  PHT     ,10H  VHT     ,10H  PHT     ,10H VHT    XNET 14
     X ,10H  A-PHT    ,10H  A-VHT   /                                   XNET 15
                                                                        XNET 16
C     COMMON STATEMENTS                                                 XNET 17
C                                                                       XNET 18
      COMMON/RDI/        JSPRNT(80)    , NLIFE(30)     , OJMT(30)     ,  XNET 19
     X   LECODE        , NVPK         , NLM           , MSM          ,  XNET 20
     X   NIM           , NTM          , NTSM          , MYM          ,  XNET 21
     X   NDM           , IZ           , JR            , PARCK3(6)    ,  XNET 22
     X   NM            , NA           , IPRINT        , IXV          ,  XNET 23
     X   IPOLCD(4,2,3), IDACCM(3)     , NMV           , MDTM         ,  XNET 24
     X   NVC(9)                                                         XNET 25
      COMMON/RDR/        SCALEF(18)    , DVHCAP(5)     , DVHLDF(5)    ,XNET 26
     X DLNCAP(5)       , ULANDF(18)    , UTRARC        , USTRUC(4)    ,XNET 27
     X DOWDTH(16)      , DCHCEN(13)    , DMATCC(5)     , DYARDS(5,13) ,XNET 28
     X DMAINT(18)      , DVMSUF(4,6)   , DVMUC(4)      , DMUMVM(5)    ,XNET 29
     X ASSRAT(4)       , TARRAY(4)     , DNOISE(20,14,2), RESDEN(4,7) ,XNET 30
     X DONPFL(225,4,4), DOLKST(225,5) , BACKON(225,3,3), DPKHRF(225,3) ,XNET 31
     X DPKDRF(225,3)                                                    XNET 32
     X NPKHRS          , NUMDA         , DPKPCT        , DPTRIP(4,3)  ,XNET 33
     X DPDLYD(2,3)     , DVDLAY(2,3)   , DMVLDF(2,3)   , XDYSCM(4,5)  ,XNET 34
     X XPKPCT          , XPTRIP(4,4)   , XDELAY(2,4,2) , MLNDIN(2,2)  ,XNET 35
     X XVMTPC(4)       , DVMTPP(3)     , AUTOCM(11)    , MMCHAR(4)    ,XNET 36
     X DPCVCM(3,6,2)   , XPCVCM(4,4)   , MPCVCM(2,4)   , PDLRAT(3,3)  ,XNET 37
     X O=STAC          , MACRAT(2,5,2) , ZACRAT(4,4)   , DACRAT(2,3)  ,XNET 38
     X -DPDAT(2,4)     , MVDPC(2,4)    , MMVDPC(2,3)   , XDPCU(4,7,4) ,XNET 39
     X XDPICD(4,7,4)   ,DOFPFL(150,4,4),DOFLKC(150,3,3) ,BACKOF(150,3,2),XNET 40
     X DYUDPC          , DCCOPF        , DMOPC         , Z(4)         ,XNET 41
     X DVDPUC(6,9,4)   , DTROPF(11,7)  , DBOPC(7)      , FLOCAP(9)    ,XNET 42
     X DFARE(4,2,2)                                                     XNET 43
      COMMON /DV23/  NOPHRS  , DVTRIP(4,4)  , DTOMHT(4,5)  ,DTOVMT(4,5),XNET 44
     X  OBMDAT(5,3,4) , DLVDAT(6,3,4)  ,   DOFVHT(150,4,4)  ,          XNET 45
     X DOFVHT(150,4,4),  DDNVHT(225,4,4)                                XNET 46
      COMMON /DV33/  AUTONO(5,6,12)  ,  TRCKMD(5,6,12) , XNDATA(5,6,12),XNET 47
     X  DONPHR(5,6,12) , DOFPWR(5,6,12) , XPHRC(5,6,12) , MPWRC(5,6,12),XNET 48
     X  DMVMT(5,6,12)                                                   XNET 49
      COMMON/SVA/ RSAVDT(100,9), SMRYDT(320,5)                          XNET 50
      COMMON /WW/ WWINDR                                                XNET 51
      EQUIVALENCE  (LV,NVM), (LK,NLM)= (DONLKC(1),BOLKST(1))            XNET 52
      MM = WWINDR                                                       XNET 53
C                                                                       XNET 54
C                                                                       XNET 55
C        APPLY  SCALE FACTOR  TO  TRANSIT   PHT/DAY                     XNET 56
```



```
                                                                        XNET 97
      DO 15  IXV = 1,IXV                                                XNET 98
C        APPLY  SCALE FACTOR TO TRANSIT PERSON TRIPS (PEAK-OFFPEAK)     XNET 99
      ASYSCM(IXV,5) = ASYSCM(IXV,5) * SCALEF(3)                         XNET 100
      DO 15  ITPR = 1,2                                                 XNET 101
   15 APTRIP(ITPR,IXV) = APTRIP(ITPR,IXV) * SCALEF(3)                   XNET 102
C                                                                       XNET 103
      DO 520 IXV=1,IXV                                                  XNET 104
      AVHOCC (IXV,1) = ASYSCM (IXV,1) * ASYSCM (IXV,3)                  XNET 105
  520 AVHOCC (IXV,2) = ASYSCM (IXV,1) * ASYSCM (IXV,4)                  XNET 106
      IXVT=IXV+1                                                        XNET 107
      ITPT=4                                                            XNET 108
      DO 530 IXV=1,IXVT                                                 XNET 109
      DO 530 ITPR=1,4                                                   XNET 110
      DO 530 NDTX=1,6                                                   XNET 111
  530 ANDATA(IXV,ITPR,NDTX)=0.0                                         XNET 112
      DO 540 IXV=1,IXV                                                  XNET 113
      ANDATA (IXV,1,1) = ASYSCM (IXV,4)  * DPKPCT                       XNET 114
  540 ANDATA (IXV,2,1) = ASYSCM (IXV,4) - ANDATA (IXV,1,1)              XNET 115
      DO 550 ITPR=1,2                                                   XNET 116
      DO 550 IXV=1,IXV                                                  XNET 117
      ANDATA(IXV,ITPR,2)=XNUDATA(IXV,1,ITPR,1)/XVHOCC(IXV,ITPR)         XNET 118
      ANDATA (IXV,ITPR,3) = ANDATA (IXV,ITPR,1) / ASYSCM (IXV,2)        XNET 119
  550 ANDATA (IXV,ITPR,4) = ANDATA (IXV,ITPR,2) / ASYSCM (IXV,2)        XNET 120
      CALL SUM(IXV,2,4,ANDATA)                                          XNET 121
      DO 560 IXV=1,IXV                                                  XNET 122
      DO 560  NDTX = 1,4                                                XNET 123
  560 ANDATA(IXV,4,NDTX)=XNUDATA(IXV,3,NDTX)*NUMDA                      XNET 124
      DO 570 NDTX=1,2                                                   XNET 125
      DO 570 IXV=1,IXV                                                  XNET 126
      ANDPHR(IXV,1,NDTX)=XNDATA(IXV,1,NDTX)/NPKHRS                      XNET 127
  570 ANDPHR(IXV,2,NDTX)=XNUDATA(IXV,2,NDTX)/NOPHRS                     XNET 128
      DO 580 ITPR=1,2                                                   XNET 129
      DO 580 IXV=1,IXV                                                  XNET 130
  580 AVTRIP(ITPR,IXV)=APTRIP(ITPR,IXV)/AVHOCC(IXV,ITPR)                XNET 131
      DO 590 IXV=1,IXV                                                  XNET 132
      AVTRIP(3,IXV)=APTRIP(1,IXV)+APTRIP(2,IXV)                         XNET 133
      AVTRIP(3,IXV)=AVTRIP(1,IXV)+AVTRIP(2,IXV)                         XNET 134
      AVTRIP(4,IXV)=APTRIP(3,IXV)*NUMDA                                 XNET 135
  590 AVTRIP(4,IXV)=AVTRIP(3,IXV)*NUMDA                                 XNET 136
      DO 600 ITPR=1,2                                                   XNET 137
      DO 600 IXV=1,IXV                                                  XNET 138
      APVDLY(ITPR,IXV,1)=(APTRIP(ITPR,IXV)*ADELAY(ITPR,IXV,1))/60.0     XNET 139
      APVDLY(ITPR,IXV,2)=(APTRIP(ITPR,IXV)*ADELAY(ITPR,IXV,2))/60.0     XNET 140
  600 APVDLY(ITPR,IXV,3)=(APTRIP(ITPR,IXV)*
     X  (ADELAY(ITPR,IXV,2)/AVHOCC(IXV,ITPR)))/60.0                     XNET 141
      DO 610 IXV=1,IXV                                                  XNET 142
      DO 610 IDTX=1,3                                                   XNET 143
      APVDLY(3,IXV,IDTX)=APVDLY(1,IXV,IDTX)+APVDLY(2,IXV,IDTX)          XNET 144
  610 APVDLY(4,IXV,IDTX)=APVDLY(3,IXV,IDTX)*NUMDA                       XNET 145
      DO 615 NDX = 5,9                                                  XNET 146
      DO 615 ITPR = 1,4                                                 XNET 147
      DO 615 IXV = 1,IXVT                                               XNET 148
      ANDATA (IXV,ITPR,NDX) = 0.0                                       XNET 149
  615 CONTINUE                                                          XNET 150
      DO 620 ITPR=1,4                                                   XNET 151
```



```
      DO 620 IXV=1,IXV                                                  XNET 112
      ANDATA(IXV,ITPR,5)=XNDATA(IXV,ITPR,3)*APVDLY(ITPR,IXV,1)          XNET 113
     X +APVDLY(ITPR,IXV,3)                                              XNET 114
  620 ANDATA(IXV,ITPR,6)=XNDATA(IXV,ITPR,4)*APVDLY(ITPR,IXV,2)          XNET 115
      DO 630 NDTX=5,6                                                   XNET 116
      DO 630 ITPR=1,4                                                   XNET 117
      DO 630 IXV=1,IXV                                                  XNET 118
  630 ANDATA(IXV,ITPR,NDTX)=ANDATA(IXV,ITPR,NDTX)+                      XNET 119
     X XNDATA(IXV,ITPR,NDTX)                                            XNET 120
      DO 640 NDTX=1,6                                                   XNET 121
      DO 640 ITPR=1,3                                                   XNET 122
      DO 640 IXV=1,IXVT                                                 XNET 123
      XNDVPC(IXV,ITPR,NDTX)=(XNDATA(IXV,ITPR,NDTX)/
     XXNDATA(IXVT,ITPR,NDTX))*100.0                                     XNET 124
  640 XNDTPC(IXV,ITPR,NDTX)=
     X (XNDATA(IXV,ITPR,NDTX)/ANDATA(IXV,3,NDTX))*100.0                 XNET 125
      DO 650 ITPR=1,3                                                   XNET 126
      DO 650 IXV=1,IXV                                                  XNET 127
      AAVTST(ITPR,IXV,1)=
     X XNDATA(IXV,ITPR,1)/APTRIP(ITPR,IXV)                              XNET 128
      AAVTST(ITPR,IXV,2)=
     X XNDATA(IXV,ITPR,2)/XVTRIP(ITPR,IXV)                              XNET 129
      AAVTST(ITPR,IXV,3)=XNUDATA(IXV,ITPR,3)/XPTRIP(ITPR,IXV)           XNET 130
      AAVTST(ITPR,IXV,4)=XNUDATA(IXV,ITPR,4)/XVTRIP(ITPR,IXV)           XNET 131
      AAVTST(ITPR,IXV,5)=XNDATA(IXV,ITPR,5)/XPTRIP(ITPR,IXV)            XNET 132
      AAVTST(ITPR,IXV,6)=XNUDATA(IXV,ITPR,6)/XVTRIP(ITPR,IXV)           XNET 133
  650 AAVTST(ITPR,IXV,6)=XNUDATA(IXV,ITPR,6)/XVTRIP(ITPR,IXV)           XNET 134
C     SUMMARY  DATA  STATEMENTS                                         XNET 135
      SMRYDT(11,MM)=0.0                                                 XNET 136
      SMRYDT(12,MM)=0.0                                                 XNET 137
      DO  999   IDB = 1,IXV                                             XNET 138
      SMRYDT(11,MM) = SMRYDT(11,MM) + APTRIP(1,IDB)                     XNET 139
      SMRYDT(12,MM) = SMRYDT(12,MM) + APTRIP(4,IDB)                     XNET 140
C                                                                       XNET 141
C                                                                       XNET 142
C                                                                       XNET 143
C                                                                       XNET 144
      IF (JSPRNT (30) .EQ. 1 ) GO TO   1                                XNET 145
      PRINT  488                                                        XNET 146
  488 FORMAT (1H1,50X,*TRANSIT  NETWORK  ANALYSIS  DATA *,///)          XNET 147
      PRINT  401                                                        XNET 148
  401 FORMAT (1H0,4X,*TRANSIT  VEHICLE  CHARACTERISTICS *,//)           XNET 149
      PRINT  402                                                        XNET 150
  402 FORMAT (1H0,10X,*TRANSIT SYSTEM*,4X,*VEHICLE   AVERAGE   VEH. PEAKXNET 151
     X    VEH. OFF PEAK*,6X,*PHT/DAY*,7X,*VEHICLE OCCUPANCY*,/,1H ,30X,XNET 152
     X*CAPACITY   SPEED      LOAD FACT.     LOAD  FACT.*,22X,*PEAK      OFFXNET 153
     X*PEAK*,/)                                                         XNET 154
      DO 405 IXVX =1,IXV                                                XNET 155
  405 PRINT 404 , IXVX, (ASYSCM(IXVX,IC),IC=1,5), XVHOCC(IXVX,1) ,XVHOCCXNET 156
     X (IXVX,2)                                                         XNET 157
  404 FORMAT (1H0,10X,I3,12X,F4.1,2X,F7.2,7X,F5.3,9X,F5.3,7X,F13.0,3X,  XNET 158
     XF7.1,4X,F7.1)                                                     XNET 159
      PRINT  406                                                        XNET 160
  406 FORMAT (1H0,//,4X,*TRANSIT  INPUT  TRIP  DELAYS *,///)            XNET 161
      PRINT  407                                                        XNET 162
  407 FORMAT (1H ,10X,*TRANSIT SYSTEM*,16X,*PERSON DELAY OUT OF VEH*,*)XNET 163
     XX,*VEHICLE  DELAY*,/,1H ,92X,*(MIN. / PERSON TRIP)*,19X,*(MIN./ VXNET 164
```

```
RUN VERSION 2.3 --PSR LEVEL 294--                                   XNETWRK                    06/30/72

              XW, TWIP1*,//,1H ,42X,*PEAK*,7X,*OFF-PEAK*,21X,*PEAK*,6X,*OFF-PEAK*XNET 167
              X,/)                                                                              XNET 168
000054        DO  418  IXVX =1,IXV                                                              XNET 169
000056    419 PRINT 409 , IXV4,  ((XUPLAY (IX,IXVX,I),I4=1,2),IT =1,2)                          XNET 170
000071    409 FORMAT (1H0,10X,I3,21X,*A*,6X,7X,*B*,6X,19X,*A*,6X,7X,*B*,6X)                     XNET 171
000101      1 CONTINUE                                                                          XNET 172
000101        IF (JSPRNT (31) .EQ. 1 ) GO TO   9                                                XNET 173
000103        PRINT 400                                                                         XNET 174
000107        DO  430  ITPX = 1,3                                                               XNET 175
000111        PRINT  412 , MITP (ITPX)                                                          XNET 176
000116    412 FORMAT (1H0,1X,1A10,/)                                                            XNET 177
000116        PRINT 415                                                                         XNET 178
000122    415 FORMAT (1H ,5X,*TRANSIT  SYS.*,11X,*PMT   PCT. OF*,22X,*PMT    PCTXNET 179
              X, OF*,20X,*A*PMT   PCT. OF *,/,1H ,38X,*ALL VEH.*,27X,*ALL VEH.*,2XNET 180
              X7X,*ALL VEH.*,2//)                                                               XNET 181
000022    416 FORMAT (1H ,5X,*TRANSIT  SYS.*,11X,*VMT   PCT. OF*,22X,*VMT    PCTXNET 182
              X, OF*,20X,*A*VMT   PCT. OF *,/,1H ,38X,*ALL VEH.*,27X,*ALL VEH.*,2XNET 183
000022        DO 420 IXVX =1,IXV                                                                XNET 184
000024    420 PRINT 422 , IXV4,  XNDATA(IXVX,ITPX,1), XNDVPC(IXVX,ITPX,1), XNDATXNET 185
              X(IXVX,ITPX,3), XNDVPC (IXVX,ITPX,3), XNDATA(IXVX,ITPX,5) , XNDVPCXNET 186
              X (IXVX,ITPX,5)                                                                   XNET 187
000060    422 FORMAT (1H ,13X,I3,4X, 9 (F15.0,3X,F5.1,12X),/)                                   XNET 188
000060        PRINT 416                                                                         XNET 189
000063        DO  425  IXVX =1, IXV                                                             XNET 190
000065    425 PRINT  422 , IXV4, XNDATA(IXVX,ITPX,2), XNDVPC(IXVX,ITPX,2), XNDATXNET 191
              X(IXVX,ITPX,4), XNDVPC (IXVX,ITPX,4), ANDATA (IXVX,ITPX,6) , XNDVPXNET 192
              XC (IXVX,ITPX,6)                                                                  XNET 193
000721    430 CONTINUE                                                                          XNET 194
000723        PRINT 412 ,*ITP (4)                                                               XNET 195
000730        PRINT  432                                                                        XNET 196
000734    432 FORMAT (1H0,3X,*TRANSIT*,/,1H ,4X,*SYSTEM*,11X,*PMT*,15X,*VMT*,19XXNET 197
              X*PMT*,15X,*VMT*,15X,*A*PMT*,13X,*A*VMT*,/)                                       XNET 198
000734        DO 435  IXVX =1, IXV                                                              XNET 199
000736    435 PRINT 434 , IXV4, IXNDATA (IXVX,4,IC),IC=1,4)                                     XNET 200
000755    434 FORMAT (1H0,4X,I3,6(F10.0) )                                                      XNET 201
000755      2 CONTINUE                                                                          XNET 202
C                                                                                               XNET 203
000755        IF (JSPRNT (32) .EQ. 1 ) GO TO   9                                                XNET 204
C                                                                                               XNET 205
000757        PRINT 400                                                                         XNET 206
000763        DO  438  IXVX = 1,IXV                                                             XNET 207
000765        IF (IXVX .EQ. IXV)   GO TO 438                                                    XNET 208
000767        PRINT 436 , IXV4                                                                  XNET 209
000774    436 FORMAT (1H ,3X,*TRANSIT SYSTEM    *,I3,/)                                         XNET 210
000774        GO TO 440                                                                         XNET 211
000775    438 PRINT 437                                                                         XNET 212
001001    437 FORMAT (1H ,3X,*ALL TRANSIT SYSTEMS*,/)                                           XNET 213
001001    440 PRINT 442                                                                         XNET 214
001005    442 FORMAT (1H ,37X,*PEAK     PCT. OF*,11X,*OFF-PEAK     PCT. OF*,12X, XNET 215
              X*DAILY*,/,1H ,40X,*DAY*,9X,*DAY*,/)                                              XNET 216
001005        DO 448 NDX =1,6                                                                   XNET 217
001007    448 PRINT 449 , XNDT(NDX), (XNDATA(IXVX,ITPX,NDX), XNDTPC (IXVX,ITPX, XNET 218
              XNDX),ITPX =1,2) ,  XNDATA (IXVX,3,NDX)                                           XNET 219
001041    449 FORMAT (1H ,10X,1A10,10X,F15.0,6X,F5.1,6X,F15.0,6X,F15.0) XNET 220
```

```
RUN VERSION 2.3 --PSR LEVEL 294--                                   XNETWRK                    06/30/72

XNETWRK

SUBPROGRAM LENGTH
002665

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
1     =  000481    2     =  000755    3     =  001044    4     =  001110
9     =  001216    15    =  000011    400   =  001231    401   =  001240
402   =  001248    484   =  001274    486   =  001305    407   =  001313
422   =  001411    432   =  001417    434   =  001433    436   =  001437
437   =  001444    438   =  000775    440   =  001001    442   =  001451
449   =  001464    455   =  001473    456   =  001502    459   =  001517
461   =  001112    462   =  001524    464   =  001532    468   =  001555
472   =  001562    475   =  001571    480   =  001611    484   =  001614
530   =  000837    500   =  000113    580   =  000151    638   =  000330

BLOCK NAMES AND LENGTHS
ADI   =  002387    ADA   =  034004    OV23  =  028031    OV33  =  005580

VARIABLE ASSIGNMENTS
ASSMAT =  000264C02 AUTOCM =  021615C09 AUTOND =  000080C04 BACKNF =  031478C02
BACKUN =  012575C02 DACDAT =  021022C09 DNUDAT =  000071C03 DCWCEN =  000070C02
DFANE  =  033764C02 DELKST =  010430C02 DEOPC  =  033756C09 DDWDTH =  000058C02
DHVMT  =  004330C04 DINT   =  000156C01 DLIFE  =  000120C01 DLNCAP =  000024C02
DLVUAT =  000201C03 DMAINT =  000211C09 DNVLDF =  021307C02 DNOISE =  000274C02
DNUMVH =  000257C02 DOFLKC =  036762C09 DOFPFL =  022222C09 DOFPWR =  002840C04
DOFVMT =  000311C03 DOFVHT =  005851C04 DONPFL =  001410C02 DONPWR =  002070C04
DONVMT =  011411C03 DPCVCM =  021434C09 DPDLYD =  021273C09 DPKDNF =  020011C02
DPKMNF =  016546C02 DPTRIP =  021257C09 DTDPMT =  000021C03 DTDVMT =  000045C03
DTRUPC =  033633C02 DVDLAY =  021301C09 DVMCAP =  000112C02 DVMLDF =  000017C02
DVMNUC =  000223C02 DVMUC  =  000253C09 DVMTPC =  021412C02 DVOPUC =  033383C02
DVTMIP =  000001C03 DWAYCC =  000103C09 DYAMDS =  000110C02 FLDCAP =  033757C02
HACMAT =  021558C02 HITP   =  002635    HLNDIR =  021442C02 HNCMAR =  021438C02
HOPUAT =  021430C02 HPCVCM =  021538C09 HRWRC  =  004460C04 HVOPCS =  021464C02
HRVUPC =  021454C02 IA     =  002661    IF     =  002660    IDACCD =  000271C01
IDG    =  002657    IDTX   =  002655    IPDLCD =  000241C01 IT    =  002642
ITPA   =  002651    ITPY   =  002653    ITS    =  002644    IXV   =  000240C01
IXVA   =  002650    IXVY   =  002652    JSPRNT =  000080C01 MH    =  002647
MHIMDX =  000080C06 MDT    =  002643    MDTX   =  002654    MDX   =  002656
MOPMHS =  000080C03 MPKMRS =  021254C09 NUMDA  =  021255C02 NVC   =  000270C01
PAMCK3 =  000227C01 PDLMAT =  021544C09 RESDEN =  001354C09 RSAVDT =  000080C05
SCALEF =  000080C02 SMMVDT =  000310C04 TAXRAT =  000470C09 TRCKWD =  000558C04
ULANUC =  000031C02 USTRUC =  000044C09 XGCRAT =  021082C02 XAVTST =  001677
XDELAY =  021362C09 XNDATA =  001328C04 XNOPWR =  002017    XNOT  =  002641
XMOTPC =  002445    XNDVPC =  002123    XOPCU  =  021862C09 XOPICD =  022642C02
XPCVCM =  021540C02 XPMPCT =  021341C09 XPTRIP =  021342C09 XPVULY =  002643
XPWMC  =  003410C04 XSYSCH =  021315C09 XVMOCC =  002007    XVMTPC =  021406C02
XVTMIP =  001657    Z     =  033277C09

START OF CONSTANTS
```

```
RUN VERSION 2.3 --PSR LEVEL 298--                                   XNETWRK                    06/30/72

001221

START OF TEMPORARIES
001020

START OF INDIRECTS
001033

UNUSED COMPILER-SPACE
122200
```

```
RUN VERSION 2.3 --PSR LEVEL 29H--                                                    08/30/72
000002          SUBROUTINE HWYNET                                                    HYNET  2
                REAL NUMDA,NPRHS,NOPHHS                                              HYNET  3
000002    C
000002          DIMENSION  KHVMT(5,6,12),  TWVMT(3,6),  AUTPTR(4,6),AUTVTR(4,6),     HYNET  5
                X     HCMS(2,2),  HVCRAT(2,3),  ABSPRT(2,2)  , ANDPC1 (2,4,4),       HYNET  6
                X     ANDPC2(4,6),HVHTPC(3,6,10),C(4)    ,                           HYNET  7
                X  AM(6), BM(4) , CM(2) ,  DM(4)                                     HYNET  8
                COMMON STATEMENTS                                                    HYNET  9
000002          COMMON/BD1/   JSPRNT(80)      , DLIFE(30)      , DINT(30)      ,     HYNET 10
                X     LECODE          , NVM           , WLM           , NSM           ,  HYNET 11
                X     NIM             , NTM           , NTSM          , NYM           ,  HYNET 12
                X     NUM             , IZ            , IR            , PARCK3(6)     ,  HYNET 13
                X     NM              , MA            , IPRINT        , IXV           ,  HYNET 14
                X     IPOLCD(4,2,3)   , IDACCD(3)     , NMV           , MNTM          ,  HYNET 15
                X     NVC(9)                                                         HYNET 16
000002          COMMON/BDR/           SCALEF(8)     , DVMCAP(5)     , DVMLDF(5)     ,HYNET 17
                X OLNCAP(5)           , VLANDF(18)    , UTRANC        , USTRUC(4)     ,HYNET 18
                X NOWUTM(16)          , DCNCEN(13)    , DWATCC(5)     , DYARDS(5,13)  ,HYNET 19
                X NWAINT(10)          , DXMSUP(4,6)   , DVMUC(6)      , DNUMV4(5)     ,HYNET 20
                X ASSRAT(4)           , TARRAT(4)     , DNOISE(20,14,2), RESDEN(4,7)  ,HYNET 21
                X NONPFL(225,6,4)     , DGLKST(225,5) , BACKUN(224,3,3), DPKHRF(225,3)HYNET 22
                X NPKDRF(225,3)       ,                                              HYNET 23
                X NPKHRS              , NUMDA         , DPKPCT        , DPTRIP(6,3)   ,HYNET 24
                X NPOLYD(2,3)         , DVOLAY(2,3)   , DMVLUF(2,3)   , ASYSC(4,5)    ,HYNET 25
                X NPKPCT              , AHTRIP(4,4)   , APELAY(2,4,2) , HLNDIR(2,2)   ,HYNET 26
                X XVMTPC(4)           , DVMTPP(3)     , AUTOCH(11)    , HMCHAR(4)     ,HYNET 27
                X NPCVCH(3,6,2)       , APCVCC(4,6)   , HPCVCH(2,4)   , POLRAT(3,3)   ,HYNET 28
                X DMSTAC              , MACRAT(2,5,2) , AACRAT(4,4)   , DACRAT(2,3)   ,HYNET 29
                X NUPUAT(2,4)         , HVOPCA(2,6)   , HMYOPC(2,3)   , XOPCU(4,7,4)  ,HYNET 30
                X KIPICD(4,7,4)       ,DOFPFL(158,4,4),NOFLKC(158,3,3) ,BACKOF(150,3,2)HYNET 31
                X NYUOPC              , DCCOPC        , DMOPC         , Z(4)          ,HYNET 32
                X NYOPUC(6,9,6)       , DTROPC(11,7)  , DGOPC(7)      , FLOCAP(5)     ,HYNET 33
                X NFARE(4,2,2)                                                       HYNET 34
000002          COMMON /DV23/  NOPHHS  , DVTRIP(4,4)  , DTOPMT(4,5) ,DTOVMT(4,5)HYNET 35
                X  OBMOAT(6,3,4) , DLVDAY(6,3,4)  ,  DOFVMT(150,4,4)  ,              HYNET 36
                X NOFVMT(150,4,4),  DONVMT(225,4,4)                                  HYNET 37
000002          COMMON /DV33/  AUTOND(5,6,12)  ,  TRCKND(5,6,12) , ANDATA(5,6,12),HYNET 38
                X  DONPHR(5,4,12) , DOFPHR(5,6,12) , XPHRC(5,6,12) , HPHRC(5,6,12)HYNET 39
                X  DMVMT(5,6,12)                                                     HYNET 40
000002          COMMON/SVM/ RSAVDT(100,3), SMRYDT(320,5)                             HYNET 41
000002          COMMON /MM/ MM(NDX                                                   HYNET 42
000002    C                                                                          HYNET 43
000002          DATA AM,BM,CM   /10H   BWT    ,10H   VWT    ,10H   PWT    ,          HYNET 44
                X                 10H   VWT    ,10H   A-PWT  ,10H   A-VWT  ,          HYNET 45
                X                 10H   PEAK   ,10H OFF PEAK ,10H DAILY  ,          HYNET 46
                X                 10H ANNUAL   ,10H FREEWAYS ,10H SURFACRT /         HYNET 47
000002          EQUIVALENCE(C(1),HVCRAT(1))                                          HYNET 48
000002          EQUIVALENCE  (LV,NVM), (LM,NLM), (DOMLKC(1),DGLKST(1))               HYNET 49
000002          JKT=JK+1                                                             HYNET 50
000004          IXMV = NM                                                            HYNET 51
000006          MM = MM(NDX                                                          HYNET 52
                C                                                                    HYNET 53
                C                                                                    HYNET 54
                C     APPLY SCALE FACTOR TO AUTO  PMT,DAY  AND  PERSON TRIPS         HYNET 55
```

```
RUN VERSION 2.3 --PSR LEVEL 29H--                                                    08/30/72
000323          AUTOND(MM,I,TPX,3)=AUTOND(MM,I,TPX,1)/MAVSPD(MM,I,TPX)               HYNET111
000331          AUTOND(MM,I,TPX,4)=AUTOND(MM,I,TPX,2)/MAVSPD(MM,I,TPX)               HYNET112
000335    100 TRCKND(MM,I,TPX,2)=TRCKND(MM,I,TPX,1)/MAVSPD(MM,I,TPX)                 HYNET113
000345        CALL SUM(2,2,4,AUTOND)                                                 HYNET114
000350        CALL SUM(2,2,2,TRCKND)                                                 HYNET115
000353        DO 110 MM=1,3                                                          HYNET116
000355    110 DO 110 LVX=1,LVY                                                       HYNET117
000356    110 DMVMT(MM,4,LVX)=DMVMT(MM,3,LVX)*NUMDA                                  HYNET118
000371        DO 120 MM=1,3                                                          HYNET119
000372    120 DO 120 IXM=1,IXMV                                                      HYNET120
000373    120 AMVMT(MM,4,IXM)=AMVMT(MM,3,IXM)*NUMDA                                  HYNET121
000400        DO 130 MM=1,3                                                          HYNET122
000407        DO 130 IYX=1,2                                                         HYNET123
000410    130 TRCKND(MM,4,IYX)=TRCKND(MM,3,IYX)*NUMDA                                HYNET124
000422        DO 132 ITPX=1,2                                                        HYNET125
000424    132 TWVMT(3,ITPX) = TWVMT(1,ITPX) + TWVMT (2,ITPX)                         HYNET126
000435        DO 140 MM=1,3                                                          HYNET127
000436        TWVMT(MM,3)=TWVMT(MM,1)+TWVMT(MM,2)                                    HYNET128
000441    140 TWVMT(MM,4)=TWVMT(MM,3)*NUMDA                                          HYNET129
000444        DO 150 ITPX=1,2                                                        HYNET130
000446    150 AUTPTR(ITPX,1)=AUTOCH(ITPX,2)                                          HYNET131
000453        AUTPTR(3,1)=AUTPTR(1,1)-AUTPTR(2,1)                                    HYNET132
000455        AUTPTR(4,1)=AUTPTR(3,1)*NUMDA                                          HYNET133
000457        DO 160 ITPX=1,4                                                        HYNET134
000460    160 AUTVTR(ITPX,1)=AUTPTR(ITPX,1)/AUTOCH(11)                               HYNET135
000465        DO 170 ITPX=1,4                                                        HYNET136
000467        AUTPTR(ITPX,2)=(AUTPTR(ITPX,1)*AUTOCH(9))/60.0                         HYNET137
000472        AUTVTR(ITPX,2)=(AUTVTR(ITPX,1)*AUTOCH(10))/60.0                        HYNET138
000475    170 AUTPTR(ITPX,3)=(AUTPTR(ITPX,1)*(AUTOCH(10)/AUTOCH(11)))/60.0           HYNET139
000502        DO 180 ITPX=1,3                                                        HYNET140
000504        AUTOND(3,ITPX,5)=AUTOND(3,ITPX,3)-AUTPTR(ITPX,2)-AUTPTR(ITPX,3)        HYNET141
000512    180 AUTOND(3,ITPX,6)=AUTOND(3,ITPX,4)-AUTVTR(ITPX,2)                       HYNET142
000517        DO 190 ND=1,4                                                          HYNET143
000521        DO 190 MM=1,3                                                          HYNET144
000522    190 AUTOND(MM,4,ND)=AUTOND(MM,3,ND)*NUMDA                                  HYNET145
000534        DO 195 NDX = 5,6                                                       HYNET146
000536    195 AUTOND(3,4,NDX) = AUTOND(3,3,NDX) * NUMDA                              HYNET147
000540        DO 200 ITPX=1,2                                                        HYNET148
000558        AUTPTH(ITPX,4)=AUTOND(3,ITPX,1)/AUTPTR(ITPX,1)                         HYNET149
000553        AUTPTH(ITPX,5)=AUTOND(3,ITPX,3)/AUTPTR(ITPX,1)                         HYNET150
000556        AUTPTR(ITPX,6)=AUTOND(3,ITPX,5)/AUTPTR(ITPX,1)                         HYNET151
000561        AUTVTR(ITPX,3)=AUTOND(3,ITPX,2)/AUTPTR(ITPX,1)                         HYNET152
000564        AUTVTR(ITPX,4)=AUTOND(3,ITPX,6)/AUTVTR(ITPX,1)                         HYNET153
000567    200 AUTVTR(ITPX,5)=AUTOND(3,ITPX,6)/AUTVTR(ITPX,1)                         HYNET154
000574        DO 211 ITPX=1,4                                                        HYNET155
000575        DO 211 MMX =1,2                                                        HYNET156
000576        DO 211 NDX =1,6                                                        HYNET157
000577    211 ANDPC1(MMX,ITPX,NDX)=(AUTOND(MMX,ITPX,NDX)/AUTOND(3,ITPX,NDX))*        HYNET158
                X 100.0                                                              HYNET159
000623        DO 226 ITPX=1,2                                                        HYNET160
000625        DO 226 NDX =1,6                                                        HYNET161
000626    226 ANDPC2(ITPX,NDX)=(AUTOND(3,ITPX,NDX)/AUTOND(3,3,NDX))*100.0            HYNET162
000645        DO 225 ITPX=3,4                                                        HYNET163
000647        DO 225 NDX=1,6                                                         HYNET164
000650    225 ANDPC2(ITPX,NDX)=100.00                                                HYNET165
```

```
RUN VERSION 2.3 --PSR LEVEL 29H--                                                    08/30/72
000697        AUTOCH(3)  =  AUTOCH(3) * SCALEF(4)                                    HYNET 56
000711        AUTOCH(4)  =  AUTOCH(4) * SCALEF(4)                                    HYNET 57
000712        AUTOCH(5)  =  AUTOCH(5) * SCALEF(4)                                    HYNET 58
000714        AUTOCH(6)  =  AUTOCH(6) * SCALEF(4)                                    HYNET 59
                C                                                                    HYNET 60
000715        AUTOCH(11)=AUTOCH(11)*AUTOCH(2)                                        HYNET 61
000717        AUTOND(1,3,1)=AUTOCH(5)*AUTOCH(7)                                      HYNET 62
000721        AUTOND(1,2,1)=AUTOCH(5)*(1.0-AUTOCH(7))                                HYNET 63
000724        AUTOND(2,1,1)=AUTOCH(6)* AUTOCH(7)                                     HYNET 64
000726        AUTOND(2,2,1)=AUTOCH(6)*(1.0-AUTOCH(7))                                HYNET 65
000731        DO 10 ITPX=1,2                                                         HYNET 66
000733        DO 10 MMX =1,2                                                         HYNET 67
000734    10  AUTOND(MMX,ITPX,2)=AUTOND(MMX,ITPX,1)/AUTOCH(11)                       HYNET 68
000746        DO 20 ITPX=1,2                                                         HYNET 69
000750        DO 20 MMX =1,2                                                         HYNET 70
000751    20  TRCKND(MMX,ITPX,1)=(AUTOND(MMX,ITPX,2)/AUTOCH(8))-                    HYNET 71
                X (AUTOND(MMX,ITPX,2))                                               HYNET 72
000764        DO 30 IXMV=1,IXMV                                                      HYNET 73
000766        DO 30 ITPX =1,2                                                        HYNET 74
000767        AMVMT(1,ITPX,IXMV)=ANDATA(IXMV,ITPX,6)*KVMTPC(IXMV)                    HYNET 75
000775    30  AMVMT(2,ITPX,IXMV)=ANDATA(IXMV,ITPX,2)*(1.0-KVMTPC(IXMV))            HYNET 76
000110        IXMV=IXMV+1                                                            HYNET 77
000112        CALL SUM(2,2,IXMV,AMVMT)                                               HYNET 78
000115        DO 40 LVX=1,LV                                                         HYNET 79
000117        DO 40 ITPX=1,2                                                         HYNET 80
000120        DMVMT(1,ITPX,LVX)=DOFVMT(JKT,LVX,ITPX)*DVMTPC(LVX)                     HYNET 81
000131    40  DMVMT(2,ITPX,LVX)=DOFVMT(JKT,LVX,ITPX)*(1.0-DVMTPC(LVX))             HYNET 82
000150        LVTPLV=1                                                               HYNET 83
000152        CALL SUM(2,2,LV,DMVMT)                                                 HYNET 84
000155        DO 50 MMX=1,2                                                          HYNET 85
000157        DO 50 ITPX=1,2                                                         HYNET 86
000160    50  TWVMT(MMX,ITPX)=AUTOND(MMX,ITPX,2)*TRCKND(MMX,ITPX,1)*               HYNET 87
                X AMVMT(MMX,ITPX,IXMV)*DMVMT(MMX,ITPX,LVV)                           HYNET 88
000205        DO  55  ITPX = 1,2                                                     HYNET 89
000207        HCMS (1,ITPX) = (HMCHAR (1) * HLNDIR (1,ITPX)) +  HMCHAR (3)           HYNET 90
000214    55  HCMS (2,ITPX) = (HMCHAR (2) * HLNDIR (2,ITPX)) +  HMCHAR (4)           HYNET 91
000222        DO  60  MMX =1,2                                                       HYNET 92
000223        HVCRAT (MMX,1) = (THVMT (MMX,1) / NPKHHS) / HCMS (MMX,1)               HYNET 93
000226    60  HVCRAT (MMX,2) = (THVMT (MMX,2) / NOPHHS) / HCMS (MMX,2)               HYNET 94
000234        DO 70 I=1,4                                                            HYNET 95
000235        IF (C(I).GT.65.65                                                      HYNET 96
000237    64  C(I)=-C(I)                                                            HYNET 97
000241    65  IF (C(I).LT.0.0.70.66.66                                               HYNET 98
000244    66  C(I)=1.0                                                               HYNET 99
000246    70  CONTINUE                                                               HYNET100
000250        DO 80 ITPX=1,2                                                         HYNET101
000252        IF (HVCRAT(1,ITPX).LE.(0.0)  GO TO 51                                  HYNET102
000255    52  HAVSPD(1,ITPX)= ((810.0*(1.0-HVCRAT(1,ITPX)))**0.5) + 30.0             HYNET103
000267        GO TO 80                                                               HYNET104
000267    51  HAVSPD(1,ITPX) = 54.0 + (0.0 * HVCRAT(1,ITPX)                         HYNET105
000275    80  CONTINUE                                                               HYNET106
000277        DO 90 ITPX=1,2                                                         HYNET107
000301    90  HAVSPD(2,ITPX)=((100.0 *(1.0 - HVCRAT(2,ITPX)))**0.5) + 12.0          HYNET108
000320        DO 100 MMX=1,2                                                         HYNET109
000322        DO 100 ITPX=1,2                                                        HYNET110
```

```
RUN VERSION 2.3 --PSR LEVEL 29H--                                                    08/30/72
000057        DO 230 ITPX=1,4                                                        HYNET166
000061        DO 230 MMX =1,3                                                        HYNET167
000062        HVMTPC(MMX,ITPX,1)=(AUTOND(MMX,ITPX,2)/THVMT(MMX,ITPX))*100.0          HYNET168
000072        HVMTPC(MMX,ITPX,2)=(TRCKND(MMX,ITPX,1)/THVMT(MMX,ITPX))*100.0          HYNET169
000077        IF(IXMV.EQ.0)  GO TO 241                                               HYNET170
000080    235 DO 240 IXMV=1,IXMV                                                     HYNET171
000082        IXZ=IXMV+2                                                             HYNET172
000084    240 HVMTPC(MMX,ITPX,IXZ)=(HVMMT(MMX,ITPX,IXMV)/THVMT(MMX,ITPX))*100.HYNET173
000720    241 CONTINUE                                                               HYNET174
000720        IF(LV.EQ.0)  GO TO 251                                                 HYNET175
000721    242 DO 250 LVX=1,LV                                                        HYNET176
000723        IZZ=IXMV+2+LVX                                                         HYNET177
000725    250 HVMTPC(MMX,ITPX,IZZ)=(HMVMT(MMX,ITPX,LVX)/THVMT(MMX,ITPX))*100.0 HYNET178
000742    251 CONTINUE                                                               HYNET179
000742    230 CONTINUE                                                               HYNET180
                C     SUMMARY  DATA  STATEMENTS                                      HYNET181
000746        SMRYDT (121,MM) =  HAVSPD(1,1)                                         HYNET182
000752        SMRYDT (122,MM) =  HAVSPD(1,2)                                         HYNET183
000755        SMRYDT (123,MM) =  HAVSPD(2,1)                                         HYNET184
000760        SMRYDT (124,MM) =  HAVSPD(2,2)                                         HYNET185
000763        SMRYDT (137,MM) =  XMVMT (3,3,IXMVT)                                   HYNET186
000770        SMRYDT (139,MM) =  THVMT (3,3)                                         HYNET187
000773        SMRYDT (142,MM) =  HMVMT (3,4,IXMVT)                                   HYNET188
001000        SMRYDT (144,MM) =  THVMT (3,4)                                         HYNET189
001003        SMRYDT (145,MM) =  AUTPTR(3,1)                                         HYNET190
001006        SMRYDT (146,MM) =  AUTPTR(4,1)                                         HYNET191
001011        SMRYDT (147,MM) =  AUTPTR(3,4)                                         HYNET192
001014        SMRYDT (148,MM) =  AUTPTR(3,6)  * 60.0                                 HYNET193
                C                                                                    HYNET194
                C                                                                    HYNET195
                C     OUTPUT STATEMENTS                                              HYNET196
                C                                                                    HYNET197
                C                                                                    HYNET198
001017        IF (JSPRNT (35) .EQ. 1 ) GO TO   A                                     HYNET199
001021        PRINT  280                                                             HYNET200
001025    280 FORMAT (1H1,50X,*HIGHWAY  NETWORK  ANALYSIS  DATA *,///)               HYNET201
001025        PRINT  292                                                             HYNET202
001031    292 FORMAT (1H ,10X,*VEHICLE    VEH. LOAD*,13X,*PERSON  TRIPS*,24X,*PMTHYNET203
              X / DAY*,18X,*PCT. PMT/DAY*,/,1H ,10X,*CAPACITY     FACTOR*,11X,*PEAHYNET204
              X*,8X,*OFF-PEAK*,12X,*FREEWAY*,18X,*SURFACE ART.     PEAK *,/)       HYNET205
001031        PRINT  293 , (AUTOCH (I),  IZ = 1,7)                                   HYNET206
001043    293 FORMAT (1H ,10X,F7.1,5X,F6.3,2X,2F10.0,2F20.0,5X,F5.3,/)               HYNET207
001043        PRINT  294                                                             HYNET208
001047    294 FORMAT (1H0,20X,*PCT.  AUTO     PERSON DELAY     VEHICLE DELAY         HYNET209
              XVEHICLE OCCUPANCY *,/,1H ,23X,*VMT*,8X,*OUT OF VEH *,/)               HYNET210
001047        PRINT  295 , (AUTOCH (I),  IZ =8,11)                                   HYNET211
001061    295 FORMAT (1H ,22X,F5.3,7X,F1[.0,8X,F12.0,12X,F8.1,/)                     HYNET212
                C                                                                    HYNET213
001061        PRINT  297                                                             HYNET214
001065    297 FORMAT (2/,1H ,3X,*HIGHWAY NETWORK CHARACTERISTICS*,//)                HYNET215
001065        PRINT  298                                                             HYNET216
001071    298 FORMAT (1H ,28X,*LANE*,12X,*AVERAGE*,9X,*DIRECTIONAL FACTORS*,15X,HYNET217
              X*CAPACITY MILES SUPPLIED*,/,1H ,10X,*MILES*,9X,*CAPACITY*,8X,*PEAHYNET218
              Xk*,8X,*OFF-PEAK*,12X,*PEAK*,13X,*OFF-PEAK*,/)                         HYNET219
001071        DO  300  MMX = 1,2                                                     HYNET220
```

```
RUN VERSION 2.3 --PSR LEVEL 294--                                   HWYNET          06/30/72

001173          IU = MMX + 2                                                        HYNET221
001175      294 FORMAT (1H0,4X,1A10,5X,2F15.0,6X,F6.3,9X,F6.3,3X,2F20.1)            HYNET222
001175      300 PRINT 200 ,CM(MMX), MMCHAR (MMX), MMCHAR (IU), (MLNDIR(MMX,ITPX),   HYNET223
                X1TPX =1,2) ,(MCMS (MMX,ITPX),ITPX =1,2)                            HYNET224
001126          PRINT  302                                                          HYNET225
001132      302 FORMAT  (1H0,35X,*TOTAL   VMT*,29X,*V/C   RATIO*,17X,*AVE.  SPEED*HYNET226
                X*/1H ,32X,*PEAK*,12X,*OFF-PEAK*,18X,*PEAK    OFF-PEAK*,14X,*PEAK HYNET227
                X   OFF-PEAK *,/)                                                   HYNET228
001132          DO 304  MMX = 1,2                                                   HYNET229
001134      304 FORMAT  (1H0,4X,1A10,5X,2F20.0,13X,F6.3,4X,F6.3,9X,2(4X,F7.2))      HYNET230
001134      305 PRINT  304, CM(MMX), (TMVMT(MMX,ITPX),ITPX=1,2) , (MVCRAT(MMX,ITPXHYNET231
                X1,ITPX=1,2), (MAVSPD(MMX,ITPX),ITPX =1,2)                          HYNET232
C                                                                                   HYNET233
001170          PRINT  307                                                          HYNET234
001174      307 FORMAT  (///,1H ,3X,*PERMEAN VMT ON FREEWAYS *,/)                   HYNET235
001174          PRINT  308, (IK , AVMTP(IK), IK=1,IXMV)                             HYNET236
001210          PRINT  309, (IK , OVMTP(IK), IK=1, LV)                              HYNET237
001224      308 FORMAT (1H0,9X, 4(*TRANSIT SYS.*,I3,6X,F6.3,5X) )                   HYNET238
001224      309 FORMAT (1H0,9X, 4(*DUAL MODE   *,I3,6X,F6.3,5X) )                   HYNET239
C                                                                                   HYNET240
001224          PRINT  312                                                          HYNET241
001230      312 FORMAT (///1H ,3X,*TRUCK  VEHICLE  HOURS OF  TRAVEL *,/)            HYNET242
001230          PRINT  313 , (BM (ITPX), ITPX = 1,4)                                HYNET243
001242      313 FORMAT (1H ,29X, 4(A10,1X))                                         HYNET244
001242          DO  315  MMX =1,2                                                   HYNET245
001244      314 FORMAT (1H0,9X,A10,4F20.6)                                          HYNET246
001244      315 PRINT  314 , CM(MMX), (TRCKHD (MMX,ITPX,2),ITPX=1,4)                HYNET247
001262        6 CONTINUE                                                            HYNET248
C                                                                                   HYNET249
001262          IF (JSPRNT (38) .EQ. 1 ) GO TO   7                                  HYNET250
C         PRINT  AUTO  NETWORK  DATA  ON  NEW  PAGE                                 HYNET251
C                                                                                   HYNET252
001264          DO 334 ITPX = 1,4                                                   HYNET253
001266          GO TO (320,322,320,322) , ITPX                                      HYNET254
001275      320 PRINT  319                                                          HYNET255
001301      319 FORMAT (1H1,51X,*AUTO  NETWORK  DATA*,///)                          HYNET256
001301      322 PRINT  324 , BM (ITPX)                                              HYNET257
001307      324 FORMAT (1H0,18X,A10,*TEMPORAL PERIOD *,///)                         HYNET258
001307          PRINT  325                                                          HYNET259
001313      325 FORMAT (1H0,31X,*FREEWAYS    PCT. OF*,14X,*SURFACE  ART.   PCT. OF*HYNET260
                X*,18X,*HIGHWAY     PCT. OF*/1H ,68X,*TOTAL*,31X,*TOTAL*,33X,*DAY*HYNET261
001313          DO  328  MDX = 1,4                                                  HYNET262
001315      328 PRINT 329 , AM(MDX), (AUTOND(MMX,ITPX,MDX), ANDPC(MMX,ITPX,MDX) ,  HYNET263
                XMMX =1,2) ,  AUTOND(3,ITPX,MDX) , ANDPC2 (ITPX,MDX)                HYNET264
001353      329 FORMAT (1H0,9X,A10,5X,2(F18.0,F10.1,7X), F2A.0 , F10.1)             HYNET265
001353          DO  330  MDX = 5,6                                                  HYNET266
001355      331 FORMAT (1H0,9X,A10,75X,F20.0,F10.1)                                 HYNET267
001355      330 PRINT 331 , AM(MDX), AUTOND(3,ITPX,MDX), ANDPC2 (ITPX,MDX)          HYNET268
001375      334 CONTINUE                                                            HYNET269
001377        7 CONTINUE                                                            HYNET270
C                                                                                   HYNET271
001377          IF (JSPRNT (37) .EQ. 1 ) GO TO   8                                  HYNET272
C                                                                                   HYNET273
001401          DO   355  ITPX = 1,4                                                HYNET274
                                                                                    HYNET275
```

```
RUN VERSION 2.3 --PSR LEVEL 294--                                   HWYNET          06/30/72

001403          IF  (ITPX .EQ. 3)   GO TO 335                                       HYNET276
001405          GO  TO   355                                                        HYNET277
001415      335 GO  TO  (334,334,336,334) ,ITPX                                     HYNET278
001415      336 PRINT  337                                                          HYNET279
001421      337 FORMAT (1H1,58X,*TOTAL  HIGHWAY  VMT  DATA *,///)                   HYNET280
001421      338 PRINT  339 , BM(ITPX)                                               HYNET281
001427      339 FORMAT (1H0,3X,A10,*TEMPORAL PERIOD*,/)                             HYNET282
001427          PRINT  340 , CM(1) , CM(2)                                          HYNET283
001437      340 FORMAT (1H0,9X,*VEHICLE*,12X,A10,*PCT. OF*,14X,A10,4X,*PCT. OF*HYNET284
                X1MX,*TOTAL      PCT. OF*/1H ,    43X,*TOTAL*,38X,*TOTAL*,28X,*TOT. HYNET285
001437          PRINT  342 , (AUTOND(MMX,ITPX,2), MVMTPC(MMX,ITPX,1),MMX=1,3)       HYNET286
001456      342 FORMAT (1H0, 9X,*AUTO*,4X,2(F20.0,3X,F5.1,7X), F20.0,3X,F5.1)       HYNET287
001456          PRINT  343, (TRCKHD(MMX,ITPX,1), MVMTPC(MMX,ITPX,2), MMX=1,3)       HYNET288
001475      343 FORMAT (1H0, 9X,*TRUCK*,4X,2(F20.0,3X,F5.1,7X), F20.0,3X,F5.1)      HYNET289
001475          DO   345  IXM = 1,IXMV                                              HYNET290
001477      344 FORMAT (1H0, 9X,*TRANSIT*,13X,2(F20.0,3X,F5.1,7X), F20.0,3X,F5.1)   HYNET291
001477      345 PRINT  344 , IXM, (XMVMT(MMX,ITPX,IXM), MVMTPC(MMX,ITPX,IXM+2), MMHYNET292
                XX =1,3)                                                            HYNET293
001526          IXM = IXMV +2                                                       HYNET294
001530          DO  348  LVX = 1,LV                                                 HYNET295
001531      347 FORMAT (1H0, 9X,*OW VEH.*,13,2(F20.0,3X,F5.1,7X), F20.0,3X,F5.1)    HYNET296
001531      348 PRINT 347 , LVX, (OMVMT(MMX,ITPX,LVX), MVMTPC(MMX,ITPX,IXM+LVX),   HYNET297
                XMMX = 1,3 )                                                        HYNET298
001561          PRINT  350, (TMVMT (MMX,ITPX) , MMX =1,3)                           HYNET299
001573      350 FORMAT (1H0,9X,*TOTAL*,4X,F20.0,3X,*100.0*,7X,F20.0,3X,*100.0*,7X,HYNET300
                XF20.0,3X,*100.0*,/)                                                HYNET301
001573      355 CONTINUE                                                            HYNET302
001575        8 CONTINUE                                                            HYNET303
C                                                                                   HYNET304
001575          IF (JSPRNT (38) .EQ. 1 ) GO TO   9                                  HYNET305
C         END  PRINTOUT OF  TOTAL  HWY VMT  DATA - PRINT AUTO TRIP DATA             HYNET306
C                                                                                   HYNET307
001577          PRINT  319                                                          HYNET308
001603          PRINT  360                                                          HYNET309
001607      360 FORMAT (1H ,3X,*AUTO  TRIP  STATISTICS *,///)                       HYNET310
001607          PRINT  362                                                          HYNET311
001613      362 FORMAT (1H0,10X,*PERSON  TRIPS*,/,1H ,27X,*NO. OF TRIPS*,6X,*PERSONHYNET312
                XN DELAY*,7X,*AVE. TRIP*,9X,*AVE. TRIP*,6X,*AVE. TRIP*/1H ,61X,*DURHYNET313
                XT VEH)   (IN VEH)      LENGTH*,10X,*TIME*,9X,*TIME (ADJ)*,/)       HYNET314
001613          DO  365  ITPX = 1,3                                                 HYNET315
001615      364 FORMAT (1H0,10X,A10,4X,F10.0,2F10.0,3F15.1)                         HYNET316
001615      365 PRINT  364 , BM(ITPX) , (AUTPTR (ITPX,IT), IT =1,6)                 HYNET317
001633          PRINT  364 , BM(4), (AUTPTR (4,IT), IT= 1,3)                        HYNET318
C                                                                                   HYNET319
001647          PRINT  370                                                          HYNET320
001653      370 FORMAT (4/,1H ,10X,*VEHICLE  TRIPS*,/,1H ,27X,*NO. OF TRIPS    VEHHYNET321
                X. DELAY      AVE. TRIP*,9X,*AVE. TRIP*,6X,*AVE. TRIP*/1H ,60X,*LENGHYNET322
                XTH*,12X,*TIME*,11X,*TIME (ADJ.)*,/)                                HYNET323
001653          DO  375  ITPX =1,3                                                  HYNET324
001655      374 FORMAT (1H0,10X,A10,4X,F10.0,F13.4,3F17.1)                          HYNET325
001655      375 PRINT  374 , BM(ITPX) , (AUTPTR (ITPX,IT), IT =1,5)                 HYNET326
001673          PRINT  374 , BM (4) ,   (AUTPTR (4, IT) , IT= 1,2)                  HYNET327
001707        9 CONTINUE                                                            HYNET328
001707          RETURN                                                              HYNET329
                                                                                    HYNET330
```

```
RUN VERSION 2.3 --PSR LEVEL 294--                                   HWYNET          06/30/72

HWYNET

SUBPROGRAM LENGTH
003035

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
6     =  001262    7     =  001377    8     =  001575    9     =  001707
10    =  000834    20    =  000851    50    =  000160    51    =  000267
52    =  000255    64    =  000237    65    =  000241    66    =  000244
70    =  000246    90    =  000275    98    =  000301    110   =  000356
120   =  000373    130   =  000410    132   =  000424    150   =  000446
160   =  000440    170   =  000522    195   =  000536    211   =  000577
220   =  000620    225   =  000650    230   =  000742    235   =  000780
241   =  000720    247   =  000721    251   =  000742    290   =  001734
292   =  001743    293   =  001747    294   =  001776    295   =  002013
297   =  002021    298   =  002027    299   =  002052    302   =  002061
304   =  002101    307   =  002110    308   =  002116    309   =  002124
320   =  001275    322   =  001301    324   =  002156    325   =  002163
329   =  002201    331   =  002207    335   =  001405    336   =  001415
337   =  002213    338   =  001421    339   =  002221    340   =  002226
342   =  002245    343   =  002254    344   =  002261    347   =  002272
350   =  002301    355   =  001573    360   =  002312    362   =  002317
364   =  002343    370   =  002350    374   =  002372

BLOCK NAMES AND LENGTHS
RDI   =  000307    RDR   =  034004    OV23  =  020031    OV33  =  005500
SVM   =  003410    MM    =  000001

VARIABLE ASSIGNMENTS
AM    =  003573    ANDPC1 =  003313   ANDPC2 =  003353   ASSRAT =  000264C02
AUTUCH =  021415C02 AUTOND =  000800C04 AUTPTR =  003222   AUTVTR =  003253
BACKOF =  031476C02 BACROW =  012575C09 BM    =  003001   C     =  002433
CM    =  003605    DACRAT =  021622C09 DAMDAT =  031371C03 DCNCEN =  000870C02
DFAME =  033784C02 DOLKST =  010438C09 DSOPC  =  033730C09 DSWDTH =  000050C02
DMVMT =  004730C04 DIMT   =  000150C01 DLIFE  =  000120C01 DLNCAP =  000824C02
DLVUAT =  002281C03 DM    =  003807    DMAINT =  000211C02 DMVLDF =  021307C02
DNOLSE =  002274C02 DNUMVM =  002257C09 DNPLKC =  026702C02 DOFFPL =  022222C02
DOFFSR =  002844C04 DOFVMT =  000311C09 DOFFVMT =  065051C03 DONPFL =  001410C02
DONPWR =  002078C04 DONVMT =  011811C09 DPCVCH =  021434C02 DPDLYO =  021273C02
DPKUNF =  029011C02 DPKWAF =  316544C09 DPTRIP =  021257C02 DTOPMT =  000821C03
DTOVMT =  000045C03 DTROPC =  033633C09 DVDLAY =  021381C02 DVMCAP =  000812C02
DVMLDF =  000817C02 DVMSUC =  000223C09 DVMUC  =  000453C02 DVMTPC =  021412C02
DVOVUC =  033383C02 DVTRIP =  000801C09 DWAYCC =  000103C02 DYARDS =  000110C02
FLOCAP =  033757C02 HACRAT =  021556C09 HAVSPD =  003387    HCMS  =  003363
HLNUIM =  021402C02 HMCMAR =  021438C09 HOPDAT =  021638C02 HPCVCH =  021530C02
HPRMC  =  004180C04 HVCRAT =  002433    HVMTPC =  003483    HVOPCS =  021440C02
HWYUPC =  021454C02 I     =  003824    IDACCD =  000471C01 IK    =  003631
IPOLCD =  000241C01 IT    =  003634    ITPX   =  003616    IU    =  003638
IXM   =  003632    IXMV   =  003814    IXMVX  =  003820    IXMVY =  003821
IXW   =  003633    IXZ    =  003827    IYX    =  003825    IZ    =  000225C01
```

```
RUN VERSION 2.3 --PSR LEVEL 294--                                   HWYNET          06/30/72

JM    =  000226C01 JKY   =  003613    JSPRNT =  000080C01 LV    =  000219C01
LVX   =  003822    LVY   =  003823    MMX    =  003617    MM    =  003615
MMMINDR =  000800C04 MDX   =  003620    MW     =  000235C01 NOPMWS =  000800C03
NPRMWS =  021256C02 NUMDA =  021255C09 NVC    =  000276C01 PARCK3 =  000227C01
POLMAT =  021544C02 RESDEN =  001354C09 RSAVDT =  000800C04 SCALFF =  000800C02
SMMTOT =  000310C05 TAIRAT =  000870C09 TMVMT  =  003287    TRCKHD =  000550C04
ULANUC =  000831C02 USTRUC =  000844C09 AACRAT =  021402C02 ADELAY =  021362C02
XMVMT =  002437    XNDATA =  001328C04 XOPCU  =  021462C02 XOPICD =  022842C02
XPCVCH =  021500002 XPTRIP =  021342C09 XPWRC  =  003410C04 XSYSCM =  021315C02
XWTPC =  021408C02 Z     =  033277C09

START OF CONSTANTS
001712

START OF TEMPORARIES
002377

START OF INDIRECTS
002411

UNUSED COMPILER SPACE
120000
```

```
      SUBROUTINE POWER                                                  PWR   2
                                                                        PWR   3
C*********************************************************************** PWR   4
C                                                                       PWR   5
C     POWER CONSUMPTION CALCULATIONS                                    PWR   6
C                                                                       PWR   7
C*********************************************************************** PWR   8
C                                                                       PWR   9
      REAL NPOWRS, NUMDA                                                PWR  10
      REAL NDPWRS                                                       PWR  11
      INTEGER PRNCR3                                                    PWR  12
      COMMON/ROOT/        JSPRNT(80)    , NLIFE(30)    , QINT(30)     ,  PWR  13
     X  LECODE          , NVM          , NLM          , NSM          ,  PWR  14
     X  NIM             , NTM          , NTSM         , NVM          ,  PWR  15
     X  NDM             , IZ           , IR           , PARCRA(6)    ,  PWR  16
     X  NM              , NA           , IPRINT       , IXV          ,  PWR  17
     X  (POLCD(4,2,3)), (UACCD(3)      , NMV          , NDTM         ,  PWR  18
     X  NVC(9)                                                          PWR  19
      COMMON/RDR/         SCALE(18)     , DVWCAP(5)    , DVWLOP(5)    ,PWR  20
     X DLNCAP(5)        , ULANDP(10)   , UTRANC       , USTRUC(4)    ,PWR  21
     X DLNUTH(10)       , DCNCEL(11)   , DWATCC(5)    , DYAROS(5,13) ,PWR  22
     X  MAINT(10)       , DVMSVC(4,6)  , DVHUC(4)     , DNUMVH(5)    ,PWR  23
     X ASSWAT(4)        , TARWAY(4)    , DNOISE(20,14,2), RESDEV(4,7) ,PWR  24
     X DINPFL(225,4,4)), DGLAST(225,5) , BACRUN(22K,3,3), DPKHRF(225,3),PWR  25
     X DKRDOF(225,3)    ,                                              PWR  26
     X NPKRS            , NUMDA        , DPKPCT       , DPTRIP(6,3)  ,PWR  27
     X PDLYDI(2,3)      , DVDLAY(2,3)  , DHYLOF(2,3)  , XSTSCH(4,5)  ,PWR  28
     X XPRPCT           , XPTWIN(4,4)  , XDELAT(2,4,2), HLNDIR(2,2)  ,PWR  29
     X XVNTPC(4)        , DVNTPC(3)    , AUTOCN(11)   , HNCWAR(4)    ,PWR  30
     X PPCVCM(3,6,2)    , APCVCM(4,6)  , MPCVCM(2,4)  , POLRAT(3,3)  ,PWR  31
     X DMSTAC           , MACRAT(2,5,2), XACRAT(4,4)  , DACRAT(2,3)  ,PWR  32
     X XOPDAT(2,4)      , MWOPC(2,6)   , MRYOPC(2,3)  , XOPCUT(4,T,4),PWR  33
     X XUPICD(4,7,4)    ,UDFPFL(150,4,4),NOFLKC(150,3,3),BACKOF(150,3,2)PWR  34
     X DYUOPC           , DCCOPC       , DWOPC        , ZIG(          ,PWR  35
     X DVOPUC(6,9,4)    , DTROPF(11,7) , DDOPCI(7)    , FLOCAP(5)    ,PWR  36
     X PFANE(4,2,2)                                                     PWR  37
      COMMON /OV23/  NPRHRS  , DVTRIP(4,4)  , DTOVNT(4,5) ,DTOVNT(4,5),PWR  38
     X  DRHDAT(6,1,6)   , DLYDAT(6,3,4)  ,  DOFVNT(150,4,4)  ,         PWR  39
     X DDFVNT(150,4,4),  DDNVNT(225,4,4)                                PWR  40
      COMMON /OV33/  ANTONDI(9,6,12)  ,  TRCKND(15,6,12) , XNDATA(5,6,12),PWR  41
     X  DDNPWR(5,6,12) , UDFPWR(5,6,12) , APWRC(5,6,12) , MPWRC(5,6,12),PWR  42
     X  DHVNT(5,6,12)                                                   PWR  43
      COMMON/SVAR/ RSAVNT(100+9), SWRYDT(320+5)                         PWR  44
      COMMON /WWW/ NMINDA                                               PWR  45
      DIMENSION  NONLRC(225+3)                                          PWR  46
      DIMENSION   A(6), B(220), TOPPWR(8,3,6), C(9,3)                   PWR  47
      EQUIVALENCE  (LVX,NVM), (LK,NLM), (DONLRC(1),DOLKST(1))           PWR  48
      NM = NMINDA                                                       PWR  49
                                                                        PWR  50
                                                                        PWR  51
      IF (JSPRNT (40) .EQ. 1 ) GO TO   1                                PWR  52
      PRINT VEHICLE  CHARACTERISTIC DATA                                PWR  53
                                                                        PWR  54
      PRINT 37                                                          PWR  55
```



```
   37 FORMAT(1H0,40X,* POWER  CONSUMPTION  *,///)                       PWR  56
      PRINT 38                                                          PWR  57
   38 FORMAT(1H0,30X,* ON-GROUNDWAY VEHICLE DATA *,////)                PWR  58
      PRINT 40                                                          PWR  59
   40 FORMAT (1H ,10X,7HVEHICLE,7X,13HWEIGHT,4X,9HTIME-OF-,4X,9HAERO-CO PWR  60
     1EF.,4X,6HHEIGHT,4X,5HFORM,4X,8HPOD CODE,///)                      PWR  61
C                                                                       PWR  62
      DO 42 LVX=1,LV                                                    PWR  63
C                                                                       PWR  64
   42 PRINT 44,LVX,(DPCVCM (LVX,IC+1),IC=1,6)                           PWR  65
   44 FORMAT (1H ,15X,I1,12X,F8.1,8X,F7.3,3X,F8.1,4X,F6.1,4X,5X,        PWR  66
     1 F3.0)                                                            PWR  67
C                                                                       PWR  68
      PRINT 45                                                          PWR  69
   45 FORMAT (1H0,////,30X,20HON OFF-GUIDEWAY VEHICLE DATA,///)         PWR  70
      PRINT 40                                                          PWR  71
      DO 47 LVX= 1,LV                                                   PWR  72
C                                                                       PWR  73
   47 PRINT 44,LVX,(DPCVCM (LVX,IC+2),IC=1,6)                           PWR  74
      PRINT 37                                                          PWR  75
      PRINT 49                                                          PWR  76
   49 FORMAT (1H0,30X,* TRANSIT VEHICLE DATA *,///)                     PWR  77
      PRINT 40                                                          PWR  78
      DO 50 IXV=1,IXV                                                   PWR  79
C                                                                       PWR  80
   50 PRINT 44,IXV,(APCVCM (IXV,IC+1),IC=1,6)                           PWR  81
      PRINT 52                                                          PWR  82
   52 FORMAT (1H0,////,30X,20HINDWAY VEHICLE DATA,///)                  PWR  83
      PRINT 40                                                          PWR  84
      DO 54 NHVX=1,NHV                                                  PWR  85
      HPCVCM(NHVX,6)=0                                                  PWR  86
   54 PRINT 44,NHVX,(HPCVCM (NHVX,IC+1),IC=1,6)                         PWR  87
    1 CONTINUE                                                          PWR  88
C                                                                       PWR  89
C     BEGIN PWR CONSUMPTION CALC.                                       PWR  90
C     INITIALIZE DONPWR AND DNFPWR WITH ZEROS                           PWR  91
      DO 60 ITA =1,4                                                    PWR  92
      DO 60 ITP =1,3                                                    PWR  93
      DO 60 LVX =1,5                                                    PWR  94
      DONPWR (LVX,ITP,ITA) =0.0                                         PWR  95
      DOFPWR (LVX,ITP,ITA) =0.0                                         PWR  96
   60 CONTINUE                                                          PWR  97
C     COMPUTE VEL. IN FPS AND SQUARE IT FOR EACH LINK LK                PWR  98
      DO 62 LKX =1,LK                                                   PWR  99
C     START CALC. OF DONPWR ARRAY                                       PWR 100
   62 W(LKX)= ((DONLKC(LKX+1)*1.47)**2)                                 PWR 101
      DO 68 ITPX= 1,2                                                   PWR 102
      DO 68 LVX = 1,LV                                                  PWR 103
      A(LVX) = DPCVCM (LVX+6+1) * DPCVCM (LVX+5+1)                      PWR 104
      IF (DPCVCM (LVX+6+1) .EQ. 0 )  GO TO 64                           PWR 105
C     EL.TYPE-CALC. KWHR/VMT AND KWHR PER LK AND SUM ON ALL LK          PWR 106
      DO 66 LKX =1,LK                                                   PWR 107
      C =((DPCVCM(LVX+1+1)* DPCVCM(LVX+7+1))+(.00134 * DPCVCM(LVX+3+1)* PWR 108
     X      A(LVX) *B(LKX))) * (.00311)                                 PWR 109
      DONPWR (LVX,ITPX,1) = DONPWR (LVX,ITPX+1)+(DONVMT(LKX+LVX+ITPX)*  PWR 110
```



```
                                                                        PWR 111
      A C)                                                              PWR 112
   66 CONTINUE                                                          PWR 113
      GO TO 68                                                          PWR 114
C     IF TYPE - CALC. LHS/VMT AND LBS PER LK AND SUM FOR ALL LK          PWR 115
   64 DO 65 LKX =1,LK                                                   PWR 116
      C =((DPCVCM (LVX+1+1)* DPCVCM(LVX+2+1))+ (.00134 * DPCVCM(LVX+3+1 PWR 117
     X) * A(LVX)) * W(LKX))) * (.00167)                                 PWR 118
      DONPWR (LVX+ITPX+2) = DONPWR (LVX+ITPX+2)+(DONVMT(LKX+LVX+ITPX)   PWR 119
     X  * C )                                                           PWR 120
   65 CONTINUE                                                          PWR 121
   68 CONTINUE                                                          PWR 122
C                                                                       PWR 123
C     BEGIN CALCULATION OF DOFPWR ARRAY                                 PWR 124
C     ARRAY A(LVX) IS VEH. FRONTAL AREA                                 PWR 125
      DO 70 ITPX =1,2                                                   PWR 126
      DO 70 LVX = 1,LV                                                  PWR 127
      A(LVX) = DPCVCM (LVX+6+2) * DPCVCM (LVX+5+2)                      PWR 128
      IF (DPCVCM (LVX+6+2) .EQ. 0 )  GO TO 73                           PWR 129
C     IF-TYPE  CALCULATE  LBS FUEL CONSUMED OFF GUIDEWAY                PWR 130
C     CALCULATE AV. SPEED SQUARED, LBS/VMT, AND LBS. FUEL FOR ALL SK    PWR 131
      DO 71 JKX = 1,JK                                                  PWR 132
      IF (DOFVMT(JKX,LVX,ITPX) .EQ. 0.0)  GO TO 71                      PWR 133
      E = (((DOFVMT(JKX+LVX+ITPX)/DOFVMT(JKX+LVX+ITPX))*1.47)**2)       PWR 134
      GO TO 72                                                          PWR 135
   71 E = 0.0                                                           PWR 136
   72 CONTINUE                                                          PWR 137
      F =((DPCVCM (LVX+1+2) * DPCVCM (LVX+2+2))+ (.00134 * DPCVCM (LVX+ PWR 138
     X 1+2) * A(LVX) * E)) * (.00167)                                   PWR 139
      DOFPWR (LVX,ITPX+2) = DOFPWR(LVX,ITPX+2)+ (DOFVMT(JKX+LVX,ITPX)*  PWR 140
     X F)                                                               PWR 141
   73 CONTINUE                                                          PWR 142
      GO TO 70                                                          PWR 143
C     EL.TYPE CALCULATE KWHR CONSUMED OFF GUIDEWAY                      PWR 144
C     CALC. AV.SPEED SQUARED ,KWHR/VMT ,AND KWHR FOR ALL JK             PWR 145
   75 DO 70 JKX = 1,JK                                                  PWR 146
      IF (DOFVMT(JKX,LVX,ITPX) .EQ. 0.0)  GO TO 76                      PWR 147
      E = (((DOFVMT(JKX,LVX,ITPX)/DOFVMT(JKX+LVX+ITPX))*1.47)**2)       PWR 148
      GO TO 77                                                          PWR 149
   76 E = 0.0                                                           PWR 150
   77 CONTINUE                                                          PWR 151
      F= ((DPCVCM (LVX+1+2) * DPCVCM (LVX+2+2))+ (.00134 * DPCVCM(LVX+  PWR 152
     X 3+2) * A(LVX)* E)) * (.00311)                                    PWR 153
   74 DOFPWR(LVX,ITPX,1)= DOFPWR(LVX,ITPX,1)+(DOFVMT(JKX,LVX,ITPX)* F)  PWR 154
   70 CONTINUE                                                          PWR 155
C     SUM ARRAYS DONPWR AND DNFPWR FOR TOTAL ON ALL LV AND PER DAY      PWR 156
      CALL SUM (LV+2,2,DONPWR)                                          PWR 157
      CALL SUM (LV+2,2+DOFPWR)                                          PWR 158
C     INITIALIZE APWRC ARRAY WITH ZEROS                                 PWR 159
      DO 84 ITPX =1,3                                                   PWR 160
      DO 84 IXVX =1,5                                                   PWR 161
      DO 84 ITX =1,4                                                    PWR 162
   84 APWRC (IXVX,ITPX,ITX) = 0.0                                       PWR 163
C                                                                       PWR 164
C     BEGIN CALCULATION OF APWRC ARRAY                                  PWR 165
C     ARRAY A(IXVX) IS VEH. FRONTAL AREA                                PWR 166
```

```
RUN VERSION 2.3 --PSR LEVEL 294--                                  POWER                    06/30/72

000445          DO 69 IXVX =1,IXV                                                          PWR  160
000446          A(IXVX) = XPCVCH (IXVX,4) * XPCVCH (IXVX,5)                                PWR  167
000450          DO 88 ITPX =1,2                                                            PWR  168
000452          E =((XNDATA (IXVX,ITPX,2)/ XNDATA (IXVX,ITPX,4)) * 1.47)**2                PWR  169
000460          IF (XPCVCH (IXVX,6) .EQ. 1) GO TO 86                                       PWR  170
C               IC=TYPE CALCULATE LBS/VHT THEN TOTAL LBS. PER IXVX AND ITPX                PWR  171
000463          C =((XPCVCH (IXVX,1)* APCVCH(IXVX,2))+ (.00134 * XPCVCH(IXVX,3)*           PWR  172
               X A(IXVX) * E)) * (.001167)                                                 PWR  173
000471          XPWRC (IXVX,ITPX,2) = XNDATA (IXVX,ITPX,2) * C                             PWR  174
000474          GO TO 88                                                                   PWR  175
C               EL.=TYPE  CALCULATE KWHR/VHT AND TOTAL KWHR PER IXV AND ITP                PWR  176
000475       86 C= ((XPCVCH(IXVX,1)* APCVCH(IXVX,2)) *(.00134 * APCVCH(IXVX,3) *            PWR  177
               A   A(IXVX) * E)) * (.001167)                                               PWR  178
000504          XPWRC (IXVX,ITPX,1) = XNDATA (IXVX,ITPX,2) * C                             PWR  179
000511       88 CONTINUE                                                                   PWR  180
000512       89 CONTINUE                                                                   PWR  181
C               SUM   XPWRC  ARRAY FOR ALL IXVX  AND TO GET  PER DAY DATA                  PWR  182
000516          CALL  SUM (IXV,2,2,XPWRF)                                                  PWR  183
C                                                                                          PWR  184
C               INITIALIZE  HPWRC ARRAY WITH ZEROS AND BEGIN HPWRC CALCULATIONS            PWR  185
C                                                                                          PWR  186
000521          DO 92 ITPX =1,3                                                            PWR  187
000523          DO 92 NHVX =1,3                                                            PWR  188
000524          DO 92 ISX =1,2                                                             PWR  189
000525       92 HPWRC (NHVX,ITPX,ISX) = 0.0                                                PWR  190
000540          DO 188  NHVX =1, NHV                                                       PWR  191
000541          A(NHVX) = HPCVC(NHVX,6) * HPCVCH (NHVX,5)                                  PWR  192
000543          IF (NHVX .EQ. 1 ) GO TO 96                                                 PWR  193
C               TRUCK = CALCULATE IC POWER RATE(LBS/VHT) AND POWER(LBS) PER ITP            PWR  194
C               CALCULATE AV. SPEED FOR TRUCKS ON HIGHWAYS IN FT/SEC (SQUARE IT)           PWR  195
000545          DO 94 ITPX =1,2                                                            PWR  196
000546          E =((TRCKWD (3,ITPX,1)/TRCKWD (3,ITPX,2)) *1.47) **2)                      PWR  197
000553          C =(( HPCVCH(NHVX,1) * HPCVCH(NHVX,2))* (.00134 * HPCVCH(NHVX,3)*          PWR  198
               A   A(NHVX)   * E)) *(.001167)                                              PWR  199
000562       94 HPWRC(NHVX,ITPX,1) =  TRCKWD (3,ITPX,1) * C                                PWR  200
000567          GO TO 188                                                                  PWR  201
C               AUTO =CALCULATE IC POWER RATE(LBS/VHT) AND POWER(LBS) PER ITP              PWR  202
C               CALCULATE AV. SPEED FOR AUTOS IN FT/SEC AND SQUARE IT                      PWR  203
000570       96 DO 99 ITPX =1,2                                                            PWR  204
000572          E= ((AUTOWD (3,ITPX,2)/AUTOWD(3,ITPX,4))*1.47)**2)                         PWR  205
000577          C= ((HPCVCH(NHVX,1) * HPCVCH(NHVX,2)) *(.00134 * HPCVCH(NHVX,3)*           PWR  206
               A   A(NHVX) * E)) * (.001167)                                               PWR  207
000606       99 HPWRC(NHVX,ITPX,1) = AUTOWD(3,ITPX,2) * C                                  PWR  208
000613      100 CONTINUE                                                                   PWR  209
C               SUM HPWRC  FOR ALL NHV  AND FOR  ITP=3 (DAY)                               PWR  210
000616          CALL  SUM (NHV,2,1,HPWRF)                                                  PWR  211
C               COMPUTE TOTAL BTUPS FOR NHV,XSIT,AND ON ON-OFF BUTDEWAY                    PWR  212
000621          NHVY = NHV+1                                                               PWR  213
000623          IXVY = IXV+1                                                               PWR  214
000625          LVY  = LV+1                                                                PWR  215
000626          DO 130 ITPX =1,3                                                           PWR  216
000630          DO 110 NHVX =1,NHVY                                                        PWR  217
000631      110 HPWRC(NHVX,ITPX,2) = HPWRC(NHVX,ITPX,1)* 2000.                             PWR  218
000642          DO 120 IXVX =1,IXVY                                                        PWR  219
000643          APWRC (IXVX,ITPX,3)= XPWRC(IXVX,ITPX,1) * 3413.                            PWR  220
```

```
RUN VERSION 2.3 --PSR LEVEL 294--                                  POWER                    06/30/72

000650      120 APWRC (IXVX,ITPX,4)= APWRC(IXVX,ITPX,2) * 2000.                            PWR  221
000655          DO 129 LVX = 1,LVY                                                         PWR  222
000657          DONPWR (LVX,ITPX,3)= DONPWR(LVX,ITPX,1) * 3413.                            PWR  223
000664          DONPWR (LVX,ITPX,4)= DONPWR(LVX,ITPX,2) * 2000.                            PWR  224
000667          DOFPWR (LVX,ITPX,3)= DOFPWR(LVX,ITPX,1) * 3413.                            PWR  225
000672          DOFPWR (LVX,ITPX,4)= DOFPWR(LVX,ITPX,2) * 2000.                            PWR  226
000677      130 CONTINUE                                                                   PWR  227
C               COMPUTE TOTAL ON PWR CONSUMPTION (ON + OFF HOWAY)                          PWR  228
000701          DO 135  ITPX =1,3                                                          PWR  229
000703          DO 135  LVX  =1,LVY                                                        PWR  230
000704          DO 135  ITX  =1,4                                                          PWR  231
000705      135 TONPWR (LVX,ITPX,ITX) = DONPWR(LVX,ITPX,ITX)+DOFPWR(LVX,ITPX,ITX)         PWR  232
C               COMPUTE TOTAL HTU FOR ON PER LVX,ITPX                                      PWR  233
000730          DO 165 ITPX =1,3                                                           PWR  234
000732          DO 165 LVX = 1,LVY                                                         PWR  235
000733      165 C(LVX,ITPX)= TOOPWR(LVX,ITPX,3) + TOOPWR(LVX,ITPX,4)                       PWR  236
000747          S=WRYDT (248,NH) =  TOOPWR (LVY+3,1)                                       PWR  237
000753          S=WRYDT (247,NH) =  TOOPWR (LVY+3,2)                                       PWR  238
000756          S=WRYDT (248,NH) =  C (LVY+3)                                              PWR  239
C                                                                                          PWR  240
C               END  ALL  CALCULATIONS                                                     PWR  241
000761          IF (JSPRNT (4) .EQ. 1 ) GO TO   2                                          PWR  242
C               OUTPUT  POWER   CONSUMPTION  ARRAYS FOR  ON, TRANSIT, AND HWY              PWR  243
000763          PRINT 140                                                                  PWR  244
000766      140 FORMAT (1H1,45X,33HON POWER CONSUMPTION -ON GUIDEWAY-///)                  PWR  245
000766          PRINT 141                                                                  PWR  246
000772      141 FORMAT (1H ,3X,7HVEHICLE,16X,4HPEAK,27X,8HOFF-PEAK,3X,3HDAY)               PWR  247
000772          PRINT 142                                                                  PWR  248
000776      142 FORMAT (1H ,14X,4HKWHR,4X,4HLBS.,4X,3HBTU,8X,4HKWHR,8X,4HLBS.,8X,          PWR  249
               X 3HBTU,18X,4HKWHR,18X,4HLBS.,8X,3HBTU//)                                   PWR  250
000774      143 FORMAT (1H ,8X,I1,3X,2F12.5,2F12.0,E12.5,2F14.0,E12.5)                     PWR  251
000774      144 FORMAT (1H0,3X,*TOTAL*,9X,2F12.0,F12.5,2F12.0,F12.5,2F14.0,E12.5)          PWR  252
C               PRINT  ON ON GUIDEWAY  ARRAY                                               PWR  253
000776          DO 150 LVX=1,LV                                                            PWR  254
001000          IF (DPCVCH (LVX,6,1).EQ. 1) GO TO 149                                      PWR  255
001003          ITX =4                                                                     PWR  256
001004          GO TO 150                                                                  PWR  257
001004      149 ITX=3                                                                      PWR  258
001005      150 PRINT 143, LVX,(DONPWR (LVX,1,IY),IY=1,2),DONPWR(LVX,1,ITX),(DONP          PWR  259
               XWR (LVX,2,IY),IY=1,2),DONPWR (LVX,2,ITX),(DONPWR(LVX,3,IY),IY=1,2          PWR  260
               X ),DONPWR (LVX,3,ITX)                                                      PWR  261
C               CALCULATE TOTAL HTU  FOR  LV= LVY                                          PWR  262
001062          LVY= LV+1                                                                  PWR  263
001064          DO 151 ITPX =1,3                                                           PWR  264
001065      151 A(ITPX) = DONPWR (LVY,ITPX,3) + DONPWR (LVY,ITPX,4)                        PWR  265
001077          PRINT 144,(DONPWR (LVY+1,IY),IY=1,2),A(1),(DONPWR (LVY+2,IY),IY=1          PWR  266
               X,2),A(2),(DONPWR (LVY+3,IY),IY=1,2),A(3)                                   PWR  267
C                                                                                          PWR  268
C               PRINT ON OFF GUIDEWAY ARRAY                                                PWR  269
C                                                                                          PWR  270
001135          PRINT 155                                                                  PWR  271
001141      155 FORMAT (1H0,45X,34HON POWER CONSUMPTION -OFF GUIDEWAY-///)                 PWR  272
001141          PRINT 141                                                                  PWR  273
001145          PRINT 142                                                                  PWR  274
001151          DO 168 LVX =1,LV                                                           PWR  275
```

```
RUN VERSION 2.3 --PSR LEVEL 294--                                  POWER                    06/30/72

001153          IF (DPCVCH (LVX,6,2) .EQ.1) GO TO 156                                      PWR  276
001156          ITX=4                                                                      PWR  277
001157          GO TO 160                                                                  PWR  278
001157      156 ITX=3                                                                      PWR  279
001160      160 PRINT 143, LVX,(DOFPWR (LVX,1,IY),IY=1,2),DOFPWR(LVX,1,ITX),(DOFP          PWR  280
               XWR (LVX,2,IY),IY=1,2),DOFPWR (LVX,2,ITX),(DOFPWR(LVX,3,IY),IY=1,2          PWR  281
               X),DOFPWR (LVX,3,ITX)                                                       PWR  282
C               CALCULATE  TOTAL BTU  FOR LV=LVY                                           PWR  283
001235          DO 161 ITPX =1,3                                                           PWR  284
001236      161 A(ITPX) = DOFPWR (LVY,ITPX,3) + DOFPWR (LVY,ITPX,4)                        PWR  285
001250          PRINT 144, (DOFPWR (LVY,1,IY),IY=1,2),A(1),(DOFPWR (LVY,2,IY),IY=          PWR  286
               X1,2),A(2),(DOFPWR (LVY+3,IY),IY=1,2),A(3)                                  PWR  287
C                                                                                          PWR  288
C               PRINT  TOTAL ON  POWER  CONSUMPTION  ARRAY                                 PWR  289
C                                                                                          PWR  290
001306          PRINT 163                                                                  PWR  291
001312      163 FORMAT (1H0,45X,26HTOTAL ON POWER CONSUMPTION-///)                         PWR  292
001312          PRINT 141                                                                  PWR  293
001316          PRINT 142                                                                  PWR  294
001322          DO 170 LVX = 1,LV                                                          PWR  295
001324      170 PRINT 143, LVX,(TOOPWR (LVX,1,IY),IY=1,2),C(LVX,1),(TOOPWR (LVX,2          PWR  296
               X,IY)=IY=1,2),C(LVX,2),(TOOPWR (LVX+3,IY),IY=1,2)+ C(LVX,3)                 PWR  297
001367          PRINT 144, (TOOPWR (LVY,1,IY),IY=1,2),C(LVY,1),(TOOPWR (LVY,2,IY)          PWR  298
               X,IY=1,2),C(LVY,2),(TOOPWR (LVY+3,IY),IY=1,2),C(LVY,3)                      PWR  299
001425        2 CONTINUE                                                                   PWR  300
001425          IXVY = IXV + 1                                                             PWR  301
C               SUM  TRANSIT  BTU PER IXV,PER ITP                                          PWR  302
001427          DO 175 ITPX = 1,3                                                          PWR  303
001431          DO 175 IXVX = 1,IXVY                                                       PWR  304
001432      175 C(IXVX,ITPX) = XPWRC(IXVX,ITPX,3) + XPWRC(IXVX,ITPX,4)                     PWR  305
001446          SWRYDT (241,NH) =  XPWRF  (IXVY,3,1)                                       PWR  306
001452          SWRYDT (242,NH) =  XPWRF  (IXVY,3,2)                                       PWR  307
001455          SWRYDT (243,NH) =  C (IXVY,3)                                              PWR  308
001460          IF (JSPRNT (42) .EQ. 1 ) GO TO   7                                         PWR  309
C                                                                                          PWR  310
C               PRINT  TRANSIT  POWER CONSUMPTION  ARRAY                                   PWR  311
001462          IXVY=IXV+1                                                                 PWR  312
001463          PRINT 173                                                                  PWR  313
001466      173 FORMAT (1H1,45X,25HTRANSIT POWER CONSUMPTION-///)                          PWR  314
001466          PRINT 141                                                                  PWR  315
001472          PRINT 142                                                                  PWR  316
001476          DO  177 IXVX =1,IXV                                                        PWR  317
001500      177 PRINT  143, IXVX,(XPWRC(IXVX,1,IY),IY=1,2),C(IXVX,1),(XPWRC (IXVX          PWR  318
               X,2,IY),IY=1,2),C(IXVX,2),(XPWRC (IXVX,3,IY),IY=1,2),C(IXVX,3)              PWR  319
001543          PRINT  144, (XPWRC (IXVY,1,IY),IY=1,2),C(IXVY,1),(XPWRC(IXVY,2,IY          PWR  320
               X),IY=1,2),C(IXVY,2),(XPWRC(IXVY,3,IY),IY=1,2),C(IXVY,3)                    PWR  321
001601        3 CONTINUE                                                                   PWR  322
001601          NHVY = NHV + 1                                                             PWR  323
001603          SWRYDT (244,NH) =  HPWRF  (NHVY+3,1)                                       PWR  324
001607          SWRYDT (245,NH) =  HPWRF  (NHVY+3,2)                                       PWR  325
001612          IF (JSPRNT (43) .EQ. 1 ) GO TO   4                                         PWR  326
C                                                                                          PWR  327
C               PRINT  HIGHWAY  POWER CONSUMPTION ARRAY                                    PWR  328
C                                                                                          PWR  329
001614          PRINT 180                                                                  PWR  330
```

```
RUN VERSION 2.3 --PSH LEVEL 204--                                   POWER                                   06/30/72

001017      180 FORMAT (1H0,45X,33HHIGHWAY VEHICLE POWER CONSUMPTION,///)                                  PWR  331
001017          PRINT 181                                                                                   PWR  332
001023          PRINT 182                                                                                   PWR  333
001027      181 FORMAT(1H ,6X,I1,15X,F12.0,E12.5,12X,F12.0,F12.5,14X,F14.0,E12.5)                           PWR  334
001027      182 FORMAT(1H0,3X,*TOTAL*,14X,F12.0,E12.5,12X,F12.0,E12.5,14X,F14.0,                            PWR  335
                  X  E12.5)
001027          DO 185 NHVX=1,NHV                                                                           PWR  336
001031          PRINT 181,NHVX,(HPWRC(NHVX,I),I=1,2),(HPWRC(NHVX,2,I),I=1,2) PWR  338
                  X1,(HPWRC(NHVX,3,I),I=1,2)                                                                PWR  339
001063      185 CONTINUE                                                                                    PWR  340
001066          NHVY =NHV +1                                                                                PWR  341
001070          PRINT 182,(HPWRC(NHVY,1,I),I=1,2),(HPWRC(NHVY,2,I),I=1,2),(HP PWR  342
                  XWRC(NHVY,3,I),I=1,2)                                                                     PWR  343
001720        4 CONTINUE                                                                                    PWR  344
001720          RETURN                                                                                      PWR  345
001721          END                                                                                         PWR  346
*FOLLOWING VARIABLES EQUIVALENCED BUT NOT REFERENCED
NVM
NLM
```

```
RUN VERSION 2.3 --PSH LEVEL 204--                                   POWER                                   06/30/72

APTHIP =  021342C02 APWRC  =  003410C0A ASYSCM =  021415C07 AVHTPC =  021406C02
Z      =  033277C02

START OF CONSTANTS
001723

START OF TEMPORARIES
002155

START OF INDIRECTS
002174

UNUSED COMPILER SPACE
121408
```

```
RUN VERSION 2.3 --PSR LEVEL 204--                                   POWER      06/30/72


POWER

SUBPROGRAM LENGTH
002702

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
1       =  000100    2       =  001425    3       =  001481    4       =  001720
37      =  001725    38      =  001732    48      =  001740    44      =  001754
54      =  001764    49      =  001773    52      =  002000    62      =  000203
64      =  000245    68      =  000272    71      =  000325    72      =  000326
75      =  000350    76      =  000378    77      =  000371    79      =  000413
84      =  000432    86      =  000475    88      =  000511    92      =  000525
96      =  000570    100     =  000613    110     =  000831    135     =  000765
140     =  002033    141     =  002042    142     =  002051    143     =  002065
144     =  002074    149     =  001004    150     =  001005    151     =  001065
155     =  002103    156     =  001157    160     =  001160    161     =  001236
163     =  002112    165     =  000733    173     =  002126    175     =  001432
180     =  002124    181     =  002135    182     =  002145

BLOCK NAMES AND LENGTHS
RDI     =  000307    RDR     =  034004    OV23    =  020431    OV33    =  005500
SVR     =  003410    MM      =  000001

VARIABLE ASSIGNMENTS
A       =  002285    ASSRAT  =  000224C07 AUTOCM  =  021415C07 AUTOND  =  000080C04
B       =  002211    BACKOF  =  031470C07 BACKON  =  012575C02 C       =  002641
DACMAT  =  021622C02 DBMDAT  =  000071C04 DENCEN  =  000070C02 DFAHF   =  033764C02
DGLAST  =  010438C02 DGOPC   =  333758C02 DGWDTH  =  000058C02 DHVMT   =  084738C04
DINT    =  000159C01 DLIFF   =  000120C01 DLWCAP  =  000024C07 DLVDAT  =  000201C03
DMAINT  =  000211C02 DMVLDF  =  021307C07 DNOISE  =  000274C07 DNUMVH  =  000257C02
DOFLKC  =  026762C02 DOFPFL  =  022222C07 DOFPWR  =  002448C04 DOFVHT  =  000311C03
DOFVHT  =  005651C03 DOMLKC  =  010430C07 DONPFL  =  001410C02 DONPWR  =  002070C04
DONVHT  =  011611C03 DPCVCM  =  021434C07 DRDLYD  =  021273C02 DRKDHF  =  020011C02
DRKMRF  =  016548C02 DPTRIP  =  021257C07 DTOPHT  =  000021C03 DTOVHT  =  000045C03
DTRUPC  =  033633C02 DVDLAT  =  021381C07 DVHCAP  =  000012C07 DVHLHF  =  000017C02
DVHSUC  =  000223C02 DVHUC   =  000253C07 DVHTPC  =  021412C07 DVOPUC  =  033303C02
DVTHIP  =  000801C03 DWAYCC  =  000103C07 DYARDS  =  000110C02 E       =  002672
F       =  002673    FLOCAP  =  033757C07 HACRAT  =  021556C07 HLNDIR  =  021402C02
HNCHAR  =  021438C02 HOPDAT  =  021438C07 HPCVCM  =  021538C07 HPWRC   =  004160C04
HVOPCS  =  021648C02 HWYOPC  =  021454C07 ICM     =  002682    IDACCO  =  000271C01
IPOLCO  =  000241C01 IS      =  002701    ISX     =  002674    ITP     =  002666
ITPX    =  002670    ITX     =  002665    IXV     =  000248C01 IXVX    =  002663
IXVY    =  002676    IY      =  002708    JK      =  000226C01 JKX     =  002671
JSPHWT  =  000008C01 LK      =  000218C01 LMX     =  002687    LV      =  000215C01
LVX     =  002661    LVY     =  002677    MM      =  002660    MWINDX  =  000080C06
MWV     =  000274C01 MWVX    =  002664    MWVY    =  002675    NOPHRS  =  000080C03
NPKMWS  =  021254C02 NUMDA   =  021255C07 NVC     =  000276C01 PARCW3  =  000227C01
POLMAT  =  021544C02 RESDEN  =  001354C07 RSAVDT  =  000080C05 SCALEF   =  000000C02
SWITCH  =  000310C05 TAXRAT  =  000270C07 TOPWR   =  002645    TRCAND  =  000550C04
ULANUC  =  000031C02 USTRUC  =  000044C07 XACRAT  =  021402C07 XDELAY  =  021362C02
WNDATA  =  001320C04 XOPCU   =  021662C07 XOPICD  =  022042C02 XPCVCM  =  021508C02
```

```
      SUBROUTINE AIRPOL                                                 POL   2
C                                                                       POL   3
C*********************************************************************POL   4
C                                                                       POL   5
C     AIR  POLLUTION  CALCULATIONS.                                     POL   6
C                                                                       POL   7
C*********************************************************************POL   8
C                                                                       POL   9
      REAL  NPKHRS, NUMDA                                               POL  10
      REAL  NOPHRS                                                      POL  11
      INTEGER  PARCK3                                                   POL  12
      DIMENSION  DONPOL(5,6,12)                                         POL  13
      DIMENSION  DOFPOL (5,6,12), DTOPOL (4,4),  XPOL(5,6,12)           POL  14
      DIMENSION  HWYPOL(5,6,12)  , HITP(4)                              POL  15
      DIMENSION  DOMLKC(225,5)                                          POL  16
      COMMON/RDI/        JSPRNT(80)     , NLIFE(30)     , DINT(30)     ,POL  17
     X   LECODE        , NVH           , WLM           , NSM          ,POL  18
     X   NIH           , NTH           , WTSH          , NYH          ,POL  19
     X   NDM           , II            , JK            , PARCK3(6)    ,POL  20
     X   NH            , NA            , IPRINT        , IXV          ,POL  21
     X   IPOLCD(4,2,3), IDACCD(3)      , NMV           , NOTH         ,POL  22
     X   NVC(9)                                                         POL  23
      COMMON/RDR/        SCALEF(10)     , DVHCAP(5)     , DVHLDF(5)    ,POL  24
     X DLMCAP(5)        , ULANDF(10)    , UTRAAC        , USTRUC(4)    ,POL  25
     X DGWDTH(16)       , DCNCEW(11)    , DWATCC(5)     , DYARDS(5,13) ,POL  26
     X DMAINT(10)       , DVHSUP(4,6)   , DVHUC(4)      , DNUMVH(5)    ,POL  27
     X ASSRAT(4)        , TARRAT(4)     , DNOISE(20,14,2), RESDEN(4,7) ,POL  28
     X DONPFL(225,4,4), DGLKST(225,5)  , BACKDN(225,3,3), DPKHRF(225,3) ,POL  29
     X DPKDRF(225,3)    ,                                               POL  30
     X NPKHRS           , NUMDA         , DPKPCT        , DPTRIP(4,3)  ,POL  31
     X DPOLYD(2,3)      , DVDLAY(2,3)   , DMVLDF(2,3)   , XSYSCH(4,5)  ,POL  32
     X XPKPCT           , XPTRIP(4,4)   , XDELAY(2,4,2) , MLNDIR(2,2)  ,POL  33
     X XVHTPC(4)        , DVHTPF(3)     , AUTOCH(11)    , HNCHAR(4)    ,POL  34
     X DPCVCH(3,4,2)    , XPCVCH(4,6)   , HPCVCH(2,4)   , POLRAT(3,3)  ,POL  35
     X DHSTAC           , HACRAT(2,5,2) , XACRAT(4,4)   , DACRAT(2,3)  ,POL  36
     X HOPDAT(2,4)      , HVOPCH(2,6)   , HWYOPC(2,3)   , XOPCUI(4,7,4),POL  37
     X XOPICD(4,7,4)    ,DOFPFL(158,4,4),DOFLKC(158,3,3) ,BACKOF(158,3,2),POL  38
     X DTUOPC           , DCCOPC        , DHDPC         , Z(4)         ,POL  39
     X DVOPUC(6,9,4)    , DTROPF(11,7)  , DODPC(7)      , FLOCAP(5)    ,POL  40
     X DFAME(4,2,2)                                                     POL  41
      COMMON /DV23/  NOPHRS  , DVTRIP(4,4)  , DTDPMT(4,5)  ,DTJVMT(4,5),POL  42
     X  DBHDAT(6,3,4) , DLVDAY(6,3,4)  ,  DOFVMT(158,4,4)  ,           POL  43
     X DDFVMT(158,4,4),  DONVMT(225,4,4)                                POL  44
      COMMON /DV33/  AUTOMD(5,6,12)  ,  TRCKMD(5,6,12)  , XMDATA(5,6,12),POL  45
     X  DONPWR(5,6,12)  , DOFPWR(5,6,12) , XPWRC(5,6,12) , HPWRC(5,6,12),POL  46
     X  DHVMT(5,6,12)                                                   POL  47
      COMMON/SVR/ RSAVMT(100,9), SMRYDT(320,5)                          POL  48
      COMMON /WW/ WWINDX                                                POL  49
      DATA HITP /10H  PEAK    ,10H OFF-PEAK ,10H DAILY    ,10H ANNUAL   POL  50
     X /                                                                POL  51
      EQUIVALENCE (DONPOL,XPOL) , (DOFPOL,HWYPOL)                       POL  52
      EQUIVALENCE  (LV,NVH), (LK,NLM) , (DOMLKC(1),DGLKST(1))           POL  53
      NW = WWINDX                                                       POL  54
      IF (JSPRNT (44) .EQ. 1 ) GO TO   9                                POL  55
```



```
      PRINT POLLUTION INPUT DATA                                        POL  56
C                                                                       POL  57
      PRINT 215                                                         POL  58
000011  215 FORMAT (1H1,50X,20HAIR  POLLUTION  DATA,///)                POL  59
000013      PRINT 216                                                   POL  60
000015  216 FORMAT (1H ,10X,23HINPUT  POLLUTION  CODES,/)               POL  61
000015      PRINT 217                                                   POL  62
000021  217 FORMAT (1H ,15X,7HVEHICLE,6X,11HON ON-DWAY,4X,12HON OFF-DWAY,13 POL  63
     X X,7HTRANSIT,13X,7HHIGHWAY)                                       POL  64
000021      DO 220 IVX=1,4                                              POL  65
000023  220 PRINT 221,(IVX,(IPOLCD(IVX,IY,1),IY=1,2),IPOLCD(IVX,1,2),IPOLCD(IV POL  66
     X X,1,3)                                                           POL  67
000045  221 FORMAT (1H ,10X,I1,15X,I1,3X,I1,9X,I1)                     POL  68
000045      PRINT 223                                                   POL  69
000051  223 FORMAT (1H ,//,1X,23HINPUT  POLLUTION  RATES,/)             POL  70
000051      PRINT 225                                                   POL  71
000055  225 FORMAT (1H ,20X,4HTYPE,10X,2HCO,10X,2HHC,10X,3HNOX)         POL  72
000055      PRINT 227,(POLRAT (I,1,I=1,3))                              POL  73
000067  227 FORMAT (1H ,20X,6HIC-GAS,6X,F10.6,10X,F10.6,10X,F10.6)      POL  74
000067      PRINT 228, (POLRAT (2,I,1), I=1,3)                          POL  75
000101  228 FORMAT (1H ,20X,9HIC-DIESEL,1X,F10.6,10X,F10.6,10X,F10.6)   POL  76
000101      PRINT 229                                                   POL  77
000105  229 FORMAT (1H0,20X,4HTYPE,4X,12HPARTICULATES,19X,3HSO2,17X,3HNOX) POL  78
000105      PRINT 230 ,(POLRAT (3,I,1),I=1,3)                           POL  79
000117  230 FORMAT (1H ,18X,8HELECTRIC,4X,F10.6,18X,F10.6,18X,F10.6)    POL  80
C                                                                       POL  81
C     END  PRINTOUT OF  ALL  INPUT  DATA                                POL  82
000117    5 CONTINUE                                                    POL  83
C     BEGIN CALCULATION OF DM  POLLUTION  DATA                          POL  84
C                                                                       POL  85
000117      JKYPJK=1                                                    POL  86
000121      LKYPLK=1                                                    POL  87
000123      DO 267 LVX=1,LV                                             POL  88
000124      JNV = IPOLCD (LVX,1,1)                                      POL  89
000126      JNF = IPOLCD (LVX,2,1)                                      POL  90
C     TEST TO SEE IF DM-GOWAY CODE IS IC-GAS,ICDIESEL,OR ELECTRIC       POL  91
000127      GO TO (235,240,245),JDO                                     POL  92
C     IC-GAS ***************************************                    POL  93
000136  235 DO 236 ITPX =1,2                                            POL  94
000140      DO 236 JTX  =1,3                                            POL  95
000141  236 DONPOL (LVX,JTX,ITPX)= POLRAT(1,JTX) * DONVMT (LKY,LVX,ITPX) POL  96
000142      GO TO 267                                                   POL  97
C     IC-DIESEL  ***************************************                POL  98
000142  240 DO 241 ITPX =1,2                                            POL  99
000144      DO 241  JTX  =1,3                                           POL 100
000145  241 DONPOL (LVX,JTX,ITPX)= POLRAT(2,JTX) * DONVMT (LKY,LVX,ITPX) POL 101
000206      GO TO 267                                                   POL 102
C     ELECTRIC ***************************************                  POL 103
000206  245 DO 246 ITPX =1,2                                            POL 104
000210      DO 246  JTX  =1,3                                           POL 105
000211  246 DONPOL (LVX,JTX,ITPX)= POLRAT(3,JTX) * DONPWR (LVX,ITPX,1)  POL 106
000230  247 GO TO (250,255,260),JOF                                     POL 107
C     TEST TO SEE IF  OFF-GOWAY CODE IS IC-GAS,IC-DIESEL,OR ELECTRIC    POL 108
C     IC-GAS ***************************************                    POL 109
000237  250 DO 251 ITPX =1,2                                            POL 110
```



```
000241      DO 251 JTX  =1,3                                            POL 111
000242  251 DOFPOL (LVX,JTX,ITPX) = POLRAT(1,JTX) * DOFVMT (JKY,LVX,ITPX) POL 112
000263      GO TO 267                                                   POL 113
C     IC-DIESEL  ***************************************                POL 114
000263  255 DO 256 ITPX =1,2                                            POL 115
000265      DO 256 JTX  =1,3                                            POL 116
000266  256 DOFPOL (LVX,JTX,ITPX) =POLRAT(2,JTX) *  DOFVMT (JKY,LVX,ITPX) POL 117
000307      GO TO 267                                                   POL 118
C     ELECTRIC ***************************************                  POL 119
000307  260 DO 261 ITPX =1,2                                            POL 120
000311      DO 261 JTX  =1,3                                            POL 121
000312  261 DOFPOL (LVX,JTX,ITPX) =POLRAT(3,JTX) *  DOFPWR (LVX,ITPX,1) POL 122
000331  267 CONTINUE                                                    POL 123
C     SUM  DONPOL AND DOFPOL  ARRAYS                                    POL 124
000334      CALL SUM  (LV*3*2*DONPOL)                                   POL 125
000337      CALL SUM  (LV*3*2*DOFPOL)                                   POL 126
C     COMPUTE  ANNUAL  POLLUTION VALUES FOR  DONPOL AND DOFPOL          POL 127
000342      LVY =LV*1                                                   POL 128
000344      DO 270 LVX =1,LVY                                           POL 129
000346      DO 270 JTX =1,4                                             POL 130
000347      DONPOL (LVX,JTX,4) = DONPOL (LVX,JTX,3) * NUMDA             POL 131
000354  270 DOFPOL (LVX,JTX,4) = DOFPOL (LVX,JTX,3) * NUMDA             POL 132
C                                                                       POL 133
000364      DO 271 ITPX=1,4                                             POL 134
000366      DO 271 LVX=1,4                                              POL 135
000367  271 DTOPOL(LVX,ITPX)=0.0                                        POL 136
C     COMPUTE  TOTAL  POLLUTION (ON AND OFF) PER LV, PER ITP            POL 137
C                                                                       POL 138
000376      DO 273  ITPX =1,4                                           POL 139
000377      DO 273  LVX  =1,LV                                          POL 140
000400  273 DTOPOL (LVX,ITPX) = DONPOL(LVX,4,ITPX)+ DOFPOL(LVX,4,ITPX) POL 141
000415      WRITE (249,MM) =  DTOPOL (1,3)                              POL 142
000421      WRITE (258,MM) =  DTOPOL (2,3)                              POL 143
000424      WRITE (256,MM) =  DTOPOL (1,4)                              POL 144
000427      WRITE (257,MM) =  DTOPOL (2,4)                              POL 145
C                                                                       POL 146
C     END ALL DM  POLLUTION  CALCULATIONS - PRINT  DM DATA OUT          POL 147
C                                                                       POL 148
000432      IF (JSPRNT (45) .EQ. 1 ) GO TO   4                          POL 149
000434      PRINT 274                                                   POL 150
000437  274 FORMAT (1H1,50X,25HGOWAY MODE  AIR  POLLUTION,//)           POL 151
000437  275 FORMAT (1H ,12X,*PART.*,9X,*SO2*,9X,*NOX*,1X,*TOTAL*,16X,*PART.*,* POL 152
     X 9X,*SO2*,8X,*NOX*,9X,*TOTAL*,10X,*TOTAL*)                        POL 153
000437  276 FORMAT (1H ,12X,*PART.*,9X,*SO2*,9X,*NOX*,1X,*TOTAL*,18X,*CO*,18XPOL 154
     X ,*HC*,8X,*NOX*,9X,*TOTAL*,10X,*TOTAL* )                          POL 155
000437  277 FORMAT (1H ,14X,*CO*,18X,*HC*,18X,*NOX*,18X,*TOTAL*,16X,*PART.*,* POL 156
     X ,*SO2*,8X,*NOX*,9X,*TOTAL*,10X,*TOTAL* )                         POL 157
000437  278 FORMAT (1H ,14X,*CO*,18X,*HC*,18X,*NOX*,18X,*TOTAL*,18X,*CO*,18X, POL 158
     X  *HC*,8X,*NOX*,9X,*TOTAL*,10X,*TOTAL*)                           POL 159
000437      DO 295 LVX=1,LV                                             POL 160
000441  280 PRINT 281, LVX                                              POL 161
000447  281 FORMAT (1H ,1X,7HVEHICLE,I2,14X,12HON  GUIDEWAY,51X,13HOFF  GUIDEWPOL 162
     XAY,/)                                                             POL 163
000447  284 JNO = IPOLCD (LVX,1,1)                                      POL 164
000451      JNF = IPOLCD (LVX,2,1)                                      POL 165
```



```
000453      GO TO (285,286),JOO                                         POL 166
000462  285 GO TO (287,288),JOF                                         POL 167
000471  286 GO TO (289,290),JOF                                         POL 168
000500  287 PRINT 278                                                   POL 169
000504      GO TO 291                                                   POL 170
000505  288 PRINT 277                                                   POL 171
000511      GO TO 291                                                   POL 172
000512  289 PRINT 276                                                   POL 173
000516      GO TO 291                                                   POL 174
000517  290 PRINT 275                                                   POL 175
000523  291 DO  293 ITPX =1,4                                           POL 176
000525      PRINT 206 , HITP(ITPX)                                      POL 177
000532  296 FORMAT (1H ,A10)                                            POL 178
000532  293 PRINT 292, (DONPOL (LVX,JT,ITPX),JT=1,4), (DOFPOL (LVX,JT,ITPX),J POL 179
     XT =1,4) , DTOPOL (LVX,ITPX)                                       POL 180
000570  292 FORMAT (1H ,10X,F10.0,2X,12.0,F14.0,7X,JF12.0,F14.0,F17.0)  POL 181
000570  295 CONTINUE                                                    POL 182
C                                                                       POL 183
C     END PRINTOUT OF DUAL MODE  POLLUTION  DATA  ******************    POL 184
C                                                                       POL 185
000573    4 CONTINUE                                                    POL 186
C                                                                       POL 187
C     BEGIN CALCULATION OF TRANSIT  POLLUTION DATA ******************   POL 188
000573      DO 310 IXV8 =1,IXV                                          POL 189
000575      JNO =IPOLCD (IXV8,1,2)                                      POL 190
000577      GO TO (297,300,305),JOO                                     POL 191
000606  297 DO 298 ITPX =1,2                                            POL 192
000610      DO 298 JTX  =1,3                                            POL 193
000611  298 APOL (IXV8,JTX,ITPX) = POLRAT (1,JTX) * XNDATA (IXV8,ITPX,2) POL 194
000630      GO TO 310                                                   POL 195
000638  300 DO  301  ITPX =1,2                                          POL 196
000642      DO  301  JTX  =1,3                                          POL 197
000643  301 APOL (IXV8,JTX,ITPX) = POLRAT (2,JTX) * XNDATA (IXV8,ITPX,2) POL 198
000652      GO TO 310                                                   POL 199
000652  305 DO 306 ITPX =1,2                                            POL 200
000654      DO 306 JTX  =1,3                                            POL 201
000655  306 APOL (IXV8,JTX,ITPX) = POLRAT (3,JTX) * XPWRC  (IXV8,ITPX,1) POL 202
000674  310 CONTINUE                                                    POL 203
C     SUM  APOL  ARRAY  ******************************                  POL 204
000677      CALL SUM (IXV*3,2,APOL)                                     POL 205
000702      IXVY =IXV *1                                                POL 206
C     CALCULATE  ANNUAL  POLLUTION VALUES FOR TRANSIT                   POL 207
000704      DO 312 IXV8 =1,IXVY                                         POL 208
000706      DO 312 JTX  =1,4                                            POL 209
000707  312 APOL (IXV8,JT,4) = APOL (IXV8,JT4,3) *  NUMDA               POL 210
C     END ALL TRANSIT  POLLUTION  CALCULATIONS  ******************      POL 211
C     BEGIN   HIGHWAY POLLUTION  CALCULATIONS  ******************       POL 212
C     ALL AUTOS ARE  ASSUMED TO BE  IC GAS                              POL 213
C                                                                       POL 214
000723      DO 315  ITPX =1,2                                           POL 215
000724      DO 315  JTX  =1,3                                           POL 216
000725  315 HWYPOL (1,JTX,ITPX) = POLRAT (1,JTX) * AUTOND (3,ITPX,2)    POL 217
000742      IFINHV .NE. 2) GO TO 325                                    POL 218
000744      JOO =  IPOLCD (2,1,3)                                       POL 219
000748      GO TO (318,318),JOO                                         POL 220
```

```
RUN VERSION 2.3 --PSR LEVEL 204--                                   AIRPOL          06/30/72

      C     IC-GAS  ***********************************************
000752  316 DO  317  ITPX=1,2                                                        POL  221
000754      DO  317  JTX =1,3                                                        POL  222
000755  317 MHYPOL (2,JTX,ITPX) = POLRAT (1,JTX) * TRCKND (3,ITPX,1)                POL  223
000772      GO TO 320                                                                POL  224
000772  318 DO 319  ITPX =1,2                                                        POL  225
000774      DO 319  JTX  =1,3                                                        POL  226
000775  319 MHYPOL (2,JTX,ITPX) = POLRAT (2,JTX) * TRCKND (3,ITPX,1)                POL  227
      C     SUM  MHYPOL ARRAY  ******************************************************POL  228
001012  320 CONTINUE                                                                 POL  229
001012      CALL SUM (MHV,3,2,MHYPOL)                                                POL  230
      C     COMPUTE  ANNUAL   MHY POLLUTION   VALUES                                 POL  231
001015      MHVY =MHV =1                                                             POL  232
001017      DO 323 MHVX =1,MHVY                                                      POL  233
001021      DO 323 JTX  =1,4                                                         POL  234
001022  323 MHYPOL (MHVX,JTX,4) =  MHYPOL (MHVX,JTX,3) * NUMDA                      POL  235
      C                                                                              POL  236
001036      SMRYDT (251,MM) =  XPOL   (1,4,3)                                        POL  237
001041      SMRYDT (252,MM) =  XPOL   (2,4,3)                                        POL  238
001044      SMRYDT (253,MM) =  XPOL   (3,4,3)                                        POL  239
001047      SMRYDT (254,MM) =  MHYPOL (1,4,3)                                        POL  240
001052      SMRYDT (255,MM) =  MHYPOL (2,4,3)                                        POL  241
001055      SMRYDT (258,MM) =  XPOL   (1,4,4)                                        POL  242
001060      SMRYDT (259,MM) =  XPOL   (2,4,4)                                        POL  243
001063      SMRYDT (260,MM) =  XPOL   (3,4,4)                                        POL  244
001066      SMRYDT (261,MM) =  MHYPOL (1,4,4)                                        POL  245
001071      SMRYDT (262,MM) =  MHYPOL (2,4,4)                                        POL  246
      C     END  ALL  HIGHWAY  POLLUTION   CALCULATIONS                              POL  247
      C     PRINT   MHY  AND  TRANSIT  POLLUTION DATA                                POL  248
      C                                                                              POL  249
001074      IF (JSPRNT (46) .EQ. 1 ) GO TO   7                                       POL  250
001076      PRINT 325                                                                POL  251
001102  325 FORMAT (1H1,50X,23HTRANSIT  AIR  POLLUTION,///)                          POL  252
      C     FOLLOWING IS FORMAT FOR MHY AND  TRANSIT                                 POL  253
001102  326 FORMAT (1H ,1X,*VEHICLE*,I2,7X,*CO*,12X,*HC*,10X,*NOX*,9X,*TOTAL*)POL  254
         X 14X,*PART.*,10X,*SO2*,8X,*NOX*,9X,*TOTAL*)                                POL  255
001102      DO 336 IXVX =1,IXV                                                       POL  256
001104      PRINT 326 ,  IXVX                                                        POL  257
001111      JDO =IPOLCD (IXVX,1,2)                                                   POL  258
001113      GO TO (330,330,3337,JDO                                                  POL  259
      C     IC GAS OR  DIESEL      ***********************************************POL  260
001121  330 DO 331 ITPX =1,4                                                         POL  261
001123      PRINT 332, HITP(ITPX), (XPOL (IXVX,JT,ITPX),JT=1,4)                      POL  262
001143  332 FORMAT (1H ,A10,3F12.8,F14.8)                                            POL  263
001143      GO TO 336                                                                POL  264
      C     ELECTRIC   ***********************************************************POL  265
001144  333 DO  334 ITPX =1,4                                                        POL  266
001146  334 PRINT 335, HITP(ITPX), (XPOL (IXVX,JT,ITPX),JT=1,4)                      POL  267
001146  335 FORMAT (1H ,A10,6X,3F15.8,F14.8)                                         POL  268
001146  336 CONTINUE                                                                 POL  269
001171    7 CONTINUE                                                                 POL  270
001171      IF (JSPRNT (47) .EQ. 1 ) GO TO   8                                       POL  271
      C     PRINT  HIGHWAY  POLLUTION DATA  ***************************************POL  272
001173      PRINT 340                                                                POL  273
001177  340 FORMAT (1H0,50X,* HIGHWAY  AIR  POLLUTION *,///)                         POL  274
                                                                                     POL  275
```

```
RUN VERSION 2.3 --PSR LEVEL 204--                                   AIRPOL          06/30/72

AIRPOL

SUBPROGRAM LENGTH
003121

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
5      = 000117   6      = 000573   7      = 001171   8      = 001233
215    = 001240   216    = 001245   217    = 001252   221    = 001264
223    = 001272   225    = 001300   227    = 001306   228    = 001314
229    = 001323   230    = 001332   235    = 000136   236    = 000141
246    = 000162   247    = 000185   245    = 000206   246    = 000211
247    = 000230   250    = 000237   251    = 000242   255    = 000263
250    = 000266   260    = 000307   261    = 000312   267    = 000331
271    = 000367   273    = 000400   274    = 001343   275    = 001351
276    = 001365   277    = 001401   278    = 001415   280    = 000441
281    = 001430   284    = 000447   285    = 000482   286    = 000671
287    = 000500   288    = 000505   289    = 000512   290    = 000517
291    = 000523   292    = 001443   296    = 001448   297    = 000606
298    = 000611   300    = 000630   301    = 000633   305    = 000652
306    = 000655   310    = 000674   312    = 000707   315    = 000725
316    = 000752   317    = 000755   318    = 000772   319    = 000775
320    = 001012   323    = 001022   325    = 001452   326    = 001460
330    = 001121   332    = 001474   333    = 001144   335    = 001500
336    = 001166   340    = 001504

BLOCK NAMES AND LENGTHS
ADI    = 000307   ADR    = 034404   OV23   = 020031   OV33   = 005500
SVN    = 003410   MM     = 000001

VARIABLE ASSIGNMENTS
ASSMAT = 000264C02 AUTOCH = 021415C0; AUTOND = 000080C04 BACKNF = 031470C02
BACKON = 012575C02 BACMAT = 021622C0; DAWDAT = 000071C03 DCWCEN = 000070C02
DFAME  = 033764C02 DGLKST = 010420C0; DNOPC  = 033756C02 DSWVTH = 000050C02
DHVMT  = 004730C04 DINT   = 000156C01 DLIFE  = 000128C01 DLWCAP = 000024C02
DLVMAT = 000281C03 DMAINT = 000211C0; DMVLDF = 021307C02 DNOISE = 000274C02
DNUMVH = 000257C02 DOFLKC = 024762C0; DOFPFL = 022222C0; DOFPOL = 002384
DOFPWR = 002648C04 DOFVMT = 000311C0; DOFVHT = 005051C03 DONLWC = 010630C02
DONPFL = 001410C02 DONPOL = 001534 DONPWR = 002078C04 DONVMT = 011611C03
DPCYCH = 021434C02 DPOLYD = 021273C0; DPMDRF = 029411C0; DPMMBF = 016546C02
DPTMIP = 021257C02 DTOPMT = 000021C0; DTOPOL = 003054 DTOVMT = 000045C03
DTNOPC = 033633C02 DVOLAT = 021301C0; DVMCAP = 000012C02 DVMLBF = 000017C02
DVHSUC = 000223C02 DVMUC  = 000253C0; DVMTPC = 021412C0; DVOPUC = 033353C02
DVTMIP = 000801C03 DWAYCC = 000103C0; DYARDS = 000110C02 FLOCAP = 033757C02
MACMAT = 021556C02 HITP   = 003074 MLNOIR = 021402C02 MMCMAR = 021430C02
NOPUAT = 021438C02 NPCYCH = 021530C0; NPMRC  = 021100C04 NVOPCS = 021640C02
NWVOPC = 021654C02 NWYPOL = 002304 IDACCD = 000271C01 IPOLCD = 000241C01
IT     = 003103    ITPX   = 003111    IVX    = 003101    IXV    = 000260C01
IXVX   = 003115    IXVY   = 003116    IY     = 003102    JDF    = 003110
JDO    = 003107    JR     = 000224C01 JKY    = 003104    JSPRNT = 000000C01
JT     = 003114    JTX    = 003112    LK     = 000216C01 LKY    = 003105
LV     = 000215C01 LVX    = 003106    LVY    = 003113    MM     = 003100
```

```
RUN VERSION 2.3 --PSR LEVEL 204--                                   AIRPOL          06/30/72

001177      DO 358 MHVX =1,MHV                                                       POL  276
001201      PRINT 326 , MHVX                                                         POL  277
001206      DO 345  ITPX =1,4                                                        POL  278
001210  345 PRINT 332 , HITP(ITPX) , (MHYPOL(MHVX,JT,ITPX),JT=1,4)                 POL  279
      C     FORMAT 332 IS IN  TRANSIT OUTPUT SECTION  ***********************************POL  280
001230  350 CONTINUE                                                                 POL  281
      C                                                                              POL  282
      C     END  PRINTOUT  OF ALL  POLLUTION  DATA                                   POL  283
001233    9 CONTINUE                                                                 POL  284
001233      RETURN                                                                   POL  285
001234      END                                                                      POL  286
*POLLO*IND VARIABLES EQUIVALENCED BUT NOT REFERENCED
NVH
NL=
```

```
RUN VERSION 2.3 --PSR LEVEL 204--                                   AIRPOL          06/30/72

MHIMHR = 000000C06 MHV    = 000274C01 MHVX   = 003120    MHVY   = 003117
NOPMHS = 000000C03 NPCMRS = 021254C0; NUMDA  = 021255C0; NVC    = 000276C01
PARCK3 = 000227C01 POLRAT = 021544C0; PSDEN  = 021354C0; RSAVDT = 000000C05
SCALEF = 000000C02 SMRYDT = 000310C04 TIXRAT = 000276C02 TRCKND = 000550C06
ULAMIC = 000311C02 USTRUC = 000044C0; XACRAT = 021602C02 XDELAY = 021362C02
XMDATA = 001320C04 XOPCU  = 021462C0; XOPICD = 022442C02 XPCYCH = 021500C02
XPOL   = 001534    XPTRIP = 021342C0; XPWRC  = 003410C04 XSYSCH = 021315C02
XVHTPC = 021480C02 Z      = 033277C0;

START OF CONSTANTS
001236

START OF TEMPORARIES
001512

START OF INDIRECTS
001520

UNUSED COMPILER SPACE
122480
```

```
RUN VERSION 2.3 --PSR LEVEL 298--                                                    06/30/72

          SUBROUTINE ACCDNT                                                          ACC   2
C                                                                                    ACC   3
C*****************************************************************************ACC   4
C                                                                                    ACC   5
C         DUAL-MODE  TRANSIT  HIGHWAY  ACCIDENT  CALCULATIONS ************          ACC   6
C                                                                                    ACC   7
C*****************************************************************************ACC   8
C                                                                                    ACC   9
000002    REAL  NPKHRS, NUMDA                                                        ACC  10
000002    REAL  NOPHRS                                                               ACC  11
000002    INTEGER  PARCK3                                                            ACC  12
000002    DIMENSION  ZZ(2,2,3)                                                       ACC  13
000002    DIMENSION   VEHACC  (5,4,12) ,  PERACC  (5,4,12)                           ACC  14
000002    DIMENSION  AH(4) , CH(3)                                                   ACC  15
C         SUMDAT (LVX,IAC)          CONTAINS  DATA  FOR  SUMMARY PRINTOUT            ACC  16
C            IAC=1  ACCIDENTS =ANNUAL                                                ACC  17
C            IAC=2  FATALITIES=ANNUAL                                                ACC  18
000002    DIMENSION  SUMDAT (2,2)                                                    ACC  19
000002     COMMON/BDI/        JSPRNT(80)      , NLIFE(30)      , DIMT(30)      ,      ACC  20
           X   LECODE         , NVM           , NLM            , MSM           ,      ACC  21
           X   NIM            , NTM           , NTSM           , NYM           ,      ACC  22
           X   NOM            , IZ            , JK             , PARCK3(6)     ,      ACC  23
           X   NH             , N4            , IPRINT         , IXV           ,      ACC  24
           X   IPOLCD(4,2,3)  , IDACCD(3)     , NMV            , MNTH          ,      ACC  25
           X   NVC(9)                                                                ACC  26
000002     COMMON/BDR/        SCALEF(15)      , DVHCAP(5)      , DVHLDF(5)     ,ACC  27
           X DLNCAP(5)        , DLANDF(15)    , UTRAAC         , USTRUC(4)     ,ACC  28
           X DDWDTM(16)       , DCMCEN(17)    , DWATCC(5)      , DYARDS(5,13)  ,ACC  29
           X DMAINT(10)       , DVHSUP(4,6)   , DVHUC(4)       , DMUMV(15)     ,ACC  30
           X ASSRAT(4)        , TAXRAT(4)     , DNDISE(20,14,2), RESDEV(4,7)   ,ACC  31
           X DONPPL(225,4,4), DOLKST(225,5)  , BACKUN(224,3,3), DPKHRF(225,3) ,ACC  32
           X DPKDRF(225,3)    ,                                                      ACC  33
           X NPKHRS           , NUMDA         , DPKPCT         , DPTRIP(4,3)   ,ACC  34
           X DPOLYD(2,3)      , DVDLAY(2,3)   , DMVLUF(2,3)    , RSTSCH(4,5)   ,ACC  35
           X XPKPCT           , XPTRIP(4,4)   , XDELAY(2,4,2)  , NLNDIR(2,2)   ,ACC  36
           X XVHTPC(4)        , DVHTPF(3)     , AUTOCH(11)     , HNCMAR(4)     ,ACC  37
           X DPCVCH(3,6,2)    , XPCVCU(4,6)   , HPCVCH(2,4)    , POLMAT(3,3)   ,ACC  38
           X DMSTAC           , MACRAT(2,5,2) , XACRAT(4,4)    , DACRAT(2,3)   ,ACC  39
           X HOPDAT(2,4)      , MVOPC(2,6)    , MHYDPC(2,3)    , XOPCU(4,7,4)  ,ACC  40
           X XOPICD(4,7,4)    ,DOFPPL(150,4,4),HDFLKC(150,3,3) ,BACKOF(150,3,2)ACC  41
           X DYUOPC           , DCCOPF        , DHOPC          , Z(4)          ,ACC  42
           X DVOPUC(6,9,4)    , DTROPF(17,7)  , DGOPC(7)       , PLOCAP(5)     ,ACC  43
           X DPARE(4,2,2)                                                            ACC  44
000002     COMMON /DV23/  NOPHRS  , DVTRIP(4,4)  , DTOPMT(4,5) ,DTOVHT(4,5)ACC  45
           X  DBMDAT(6,3,4)  , DLVDAT(6,3,4)  ,  DOFVHT(150,4,4)  ,                  ACC  46
           X DOFVHT(150,4,4),  DONVHT(225,4,4)                                       ACC  47
000002     COMMON /DV33/  AUTOMD(5,6,12)  ,  TRCKMD(5,6,12) , XMDATA(5,6,12)ACC  48
           X  DDNPWR(5,6,12)  , DOFPWR(5,6,12)  , XPWRC(5,6,12) , HPWRC(5,6,12)ACC  49
           X  DHVHT(5,6,12)                                                          ACC  50
000002     COMMON/SVR/ RSAVDT(100,9) , SMRYDT(320,5)                                 ACC  51
000002     COMMON /MM/ MMINDA                                                        ACC  52
000002     EQUIVALENCE  (LV,NVM), (LK,NLM), (DONLKC(1),DOLKST(1))                    ACC  53
000002     DATA BM,CM   /10H  PEAK    ,10H  OFF-PEAK,10H  DAILY  ,10H  ANNUAACC  54
           XL  ,10H  FREEWAYS,10HSUBP. ART.,10H  HIGHWAYS/                           ACC  55
```

```
RUN VERSION 2.3 --PSR LEVEL 298--                                    ACCDNT          06/30/72

000002          MM = MMINDA                                                          ACC  56
000004          IF (JSPRNT (40) .EQ. 1 ) GO TO   9                                   ACC  57
C                                                                                    ACC  58
C         PRINT  OUT  ACCIDENT  INPUT  DATA ***************************************ACC  59
C                                                                                    ACC  60
000006          PRINT 407                                                            ACC  61
000011    407 FORMAT (1H1,40X,*DUAL-MODE - TRANSIT - HIGHWAY  ACCIDENT  DATA*,//ACC  62
               X//)                                                                  ACC  63
000011          PRINT 406                                                            ACC  64
000015    406 FORMAT (1H0)                                                           ACC  65
000015          PRINT 408                                                            ACC  66
000021    408 FORMAT (1H0,18X,21HACCIDENT  INPUT  DATA,/)                            ACC  67
000021          PRINT 409                                                            ACC  68
000025    409 FORMAT (1H ,28X,21HDM  STATION ACC. RATE,22X,5HDM V1,5X,5HDM V2,5 ACC  69
               X,5HDM V3,5X,5HDM V4)                                                 ACC  70
000025          PRINT 410 ,  DMSTAC , (IDACCD (LVX),LVX =1,LV)                       ACC  71
000042    410 FORMAT (1H0,25X,F18.3,2GX,3(I3,7X),I3)                                 ACC  72
000042          PRINT 397                                                            ACC  73
000046    397 FORMAT (1H0,28X,*DM ON-HDWY ACCIDENT RATES      VEHICLE*,13X,*FATALACC  74
               X ACCIDENTS      PASS. FATALITIES *,/)                                ACC  75
000046          DO 395 LVX=1,LV                                                      ACC  76
000058    396 FORMAT (1H0,5X,I3,13X,F15.3,5X,F15.3)                                  ACC  77
000058    395 PRINT 396, LVX, (DACRAT (IAT,LVX),IAT =1,2)                            ACC  78
000067          PRINT 406                                                            ACC  79
000072          PRINT 412                                                            ACC  80
000076    412 FORMAT (1H0,2X,22HMIDHWAY ACCIDENT RATES,13X,6HPEAK,7X,9HOFF  PEAACC  81
               XK,/)                                                                 ACC  82
000076          PRINT 413                                                            ACC  83
000102    413 FORMAT (1H ,10X,10HFATAL ACC.,10H INJ. ACC.,3X,3HPDO,6X,6HFATAL.,ACC  84
               X5X,4HINJ.,13X,10HFATAL ACC.,10H INJ. ACC.,4X,3HPDO,5X,6HFATAL.,5X ACC  85
               X,4HINJ.,/)                                                           ACC  86
000102          DO  415 IHX =1,2                                                     ACC  87
000104    415 PRINT  418 ,CM(IHX), ((XACRAT(IHX,IA,ITP),IA=1,5),ITP=1,2)             ACC  88
000126    418 FORMAT (1H ,A10,5F10.2,10X,5F10.2,/)                                   ACC  89
000126          PRINT  488                                                           ACC  90
000132          PRINT 417                                                            ACC  91
000136    417 FORMAT (1H0,2X,24HTRANSIT  ACCIDENT  RATES,6X,10HTRAF. ACC.,5X,10 ACC  92
               XHPASS. ACC.,5X,12HPASS. FATAL.,3X,16HNON-PASS. FATAL.,/)                ACC  93
000136          DO 420  IXVX = 1,IXV                                                 ACC  94
000140    420 PRINT 419, IXVX, (XACRAT (IXVX,IB,IB =1,4)                             ACC  95
000157    419 FORMAT (1H0,3X,*TRANSIT SYS.*,5X,I3,12X,F10.2,5X,F10.2,5X,F12.3,3XACC  96
               XF10.3)                                                               ACC  97
C                                                                                    ACC  98
C         END PRINTOUT OF ALL INPUT DATA - BEGIN ACCIDENT CALCULATIONS               ACC  99
000157    9 CONTINUE                                                                 ACC 100
C         ********************  HIGHWAY ACCIDENTS ***********************************ACC 101
000157          AV = 100000000.                                                      ACC 102
000161          AU = 1000000.                                                        ACC 103
000162          DO  430  IHX = 1,2                                                   ACC 104
000164          DO  430  ITPX = 1,2                                                  ACC 105
000165          DO  425  IAX  = 1,3                                                  ACC 106
000166    425 VEHACC (IHX,IAX,ITPX) = (AUTOMD (IHX,ITPX,2) * MACRAT (IHX,IAX,IT ACC 107
               XPX)) /AV                                                             ACC 108
000205          DO  430  IHX = 1,2                                                   ACC 109
000206          IAX = IAX +3                                                         ACC 110
```

```
RUN VERSION 2.3 --PSR LEVEL 298--                                    ACCDNT          06/30/72

000210    430 PERACC (IHX,IAX,ITPX) = (AUTOMD (IHX,ITPX,2) * MACRAT (IHX,IAX,IT ACC 111
               XPX)) / AV                                                            ACC 112
C         SUM  VEHACC  AND  PERACC  ARRAYS  ***************************************ACC 113
000231          CALL SUM (Z,3,2,VEHACC)                                              ACC 114
000234          CALL SUM (Z,2,2,PERACC)                                              ACC 115
C         COMPUTE  ANNUAL  HIGHWAY  ACCIDENT  DATA  ***********************************ACC 116
000237          DO 435  IHX = 1,3                                                    ACC 117
000241          DO 433  IAX = 1,4                                                    ACC 118
000242    433 VEHACC (IHX,IAX,4) = VEHACC (IHX,IAX,3) * NUMDA                        ACC 119
000252          DO 435  IBX = 1,3                                                    ACC 120
000254    435 PERACC (IHX,IBX,4) = PERACC (IHX,IBX,3) * NUMDA                        ACC 121
000266          SMRYDT (263,MM) =  VEHACC (3,4,4)                                    ACC 122
000272          SMRYDT (264,MM) =  PERACC (3,1,4)                                    ACC 123
000275          SMRYDT (265,MM) =  PERACC (3,2,4)                                    ACC 124
000300          IF (JSPRNT (58) .EQ. 1 ) GO TO   1                                   ACC 125
C                                                                                    ACC 126
C         PRINT  HIGHWAY  ACCIDENT DATA  OUT  *************************************ACC 127
C                                                                                    ACC 128
000302          PRINT  437                                                           ACC 129
000305    437 FORMAT (1H1,58X,32HMIDHWAY  ACCIDENT  DATA  (AUTOS),//)                ACC 130
000305          PRINT  438                                                           ACC 131
000311    438 FORMAT (1H0,29X,5HFATAL,9X,6HINJURY,11X,3HPDO,13X,5HTOTAL,13X,10HFACC 132
               XATALITIES,6X,8HINJURIES)                                             ACC 133
000311          PRINT  439                                                           ACC 134
000315    439 FORMAT (1H ,23X,8HACCIDENTS,6X,9HACCIDENTS,6X,9HACCIDENTS,7X,9HACCACC 135
               XIDENTS,/)                                                            ACC 136
000315          DO 450  IHX = 1,3                                                    ACC 137
000317          PRINT  442 , CM(IHX)                                                 ACC 138
000324    442 FORMAT (1H0,3X,A10)                                                    ACC 139
000324          DO 450  ITPX = 1,4                                                   ACC 140
000326    450 PRINT 445,  AH(ITPX), (VEHACC (IHX,IAX,ITPX),IAX=1,4),(PERACC (IHXACC 141
               X,IBX,ITPX), IBX =1,2)                                                ACC 142
000363    445 FORMAT (1H ,3X,A10,7X,3F15.2,F18.2,5X,2F15.2)                          ACC 143
C                                                                                    ACC 144
C         END  PRINTOUT OF ALL HIGHWAY ACCIDENT DATA  *****************************ACC 145
000363    1 CONTINUE                                                                 ACC 146
C         ****************  TRANSIT ACCIDENTS ***********************************ACC 147
000363          DO 470 IXVX = 1,IXV                                                  ACC 148
000365          DO 470 ITPX = 1,2                                                    ACC 149
000366          VEHACC (IXVX,1,ITPX) = (XMDATA (IXVX,IIPX,2) * XACRAT(IXVX,1))/XU ACC 150
000374          VEHACC (IXVX,2,ITPX)= (XPTRIP (ITPX,IXVX) * XACRAT (IXVX,2)) /XU   ACC 151
000402          DO  470 IAX = 1,2                                                    ACC 152
000403          IVX = IAX + 2                                                        ACC 153
000405    470 PERACC (IXVX,IBX,ITPX)= (XPTRIP (ITPX,IXVX) * XACRAT (IXVX,IVX))/ ACC 154
               XXU                                                                   ACC 155
C         SUM  VEHACC  AND PERACC  ARRAYS ***************************************ACC 156
000424          CALL SUM (IXV,2,2,VEHACC)                                            ACC 157
000427          CALL SUM (IXV,2,2,PERACC)                                            ACC 158
C         COMPUTE  ANNUAL  TRANSIT  ACCIDENT  DATA ********************************ACC 159
000432          IXVY = IXV + 1                                                       ACC 160
000434          DO 475 IXVX = 1,IXVY                                                 ACC 161
000436          DO 475 IAX  = 1,3                                                    ACC 162
000437          VEHACC (IXVX,IAX,4) = VEHACC (IXVX,IAX,3) * NUMDA                    ACC 163
000444    475 PERACC (IXVX,IAX,4) = PERACC (IXVX,IAX,3) * NUMDA                      ACC 164
000453          IXVY = IXV + 1                                                       ACC 165
```

```
RUN VERSION 2.3 --PSR LEVEL 204--                                   ACCDMT                    06/30/72

000465          SHMYDT (266,MM) =  VEHACC (IKV,1,4)                                          ACC  166
000481          SHMYDT (267,MM) =  PERACC (IKV,3,4)                                          ACC  167
000464          IF (JSPRNT (5) .EQ. 1 ) GO TO   7                                            ACC  168
          C                                                                                  ACC  169
          C     PRINT  TRANSIT  ACCIDENT  DATA  *****************************                ACC  170
          C                                                                                  ACC  171
000465          PRINT  480                                                                   ACC  172
000471    480 FORMAT (1M1,58X,23HTRANSIT  ACCIDENT  DATA,//)                                 ACC  173
000471          PRINT  483                                                                   ACC  174
000475    483 FORMAT(1M ,34X,7HTRAFFIC,12X,9HPASSENGER,18X,9HPASSENGER,4X,13HNONACC         ACC  175
                X=PASSENGER+0X=SHTOTAL)                                                      ACC  176
000475          PRINT  486                                                                   ACC  177
000501    486 FORMAT (1M ,33X,9HACCIDENTS,11X,9HACCIDENTS,16X,10HFATALITIES,5X,              ACC  178
                X10HFATALITIES,7X,10HFATALITIES,//)                                          ACC  179
000501          DO 495 IKV4 = 1,IKV                                                          ACC  180
000503          PRINT 487 ,  IKV4                                                            ACC  181
000510    487 FORMAT (1M ,4X,13HTRANSIT SYS.,I3)                                             ACC  182
000510          DO 490 ITPX = 1,4                                                            ACC  183
000512    488 PRINT 489, 4HI(ITPX), (VEHACC (IKV,IAX,ITPX),IAX=1,2) , (PERACC                ACC  184
                X(IKV4,IBX,ITPX), IBX =1,3)                                                  ACC  185
000550    489 FORMAT (1M ,11X,A4,10(4X,F5.2,5X,F,5.2,10X,2F15.2,F20.2)                       ACC  186
000550          PRINT 491                                                                    ACC  187
000553    491 FORMAT (1HO,5X,5HTOTAL)                                                        ACC  188
000553          DO 492 ITPX = 1,4                                                            ACC  189
000555    492 PRINT  489, 8HI(ITPX), (VEHACC(IKV,IAX,ITPX),IAX=1,2) , (PERACC                ACC  190
                X(IKV,IBX,ITPX),  IBX=1,3)                                                   ACC  191
          C                                                                                  ACC  192
          C     END PRINTOUT OF ALL TRANSIT ACCIDENT  DATA  ******************               ACC  193
          C     ****************** DUAL-MODE  ACCIDENT  CALCULATIONS  ***************        ACC  194
000610      2 CONTINUE                                                                       ACC  195
000612        IF (JSPRNT (52) .EQ. 1 ) GO TO   7                                             ACC  196
000616        PRINT  530                                                                     ACC  197
          3 CONTINUE                                                                         ACC  198
          C                                                                                  ACC  199
          C     ZERO  OUT  SUMDAT  ARRAY                                                     ACC  200
000616        DO 10  LVX = 1,2                                                               ACC  201
000620        DO 10  IAC = 1,2                                                               ACC  202
000621     10 SUMDAT(LVX,IAC) = 0.0                                                          ACC  203
          C        ARRAY IZ  CONTAINS LOADED VEH.  VMT  PER IMX,LVX,ITPX                     ACC  204
000630        DO 493 IMX=1,2                                                                 ACC  205
000631        DO 493 ITPX=1,2                                                                ACC  206
000632        DO 493 LVX =1,LV                                                               ACC  207
000633        ZZ(IMX,ITPX,LVX) = DLVDAT(3,LVX,ITPX) *( DMVMT(IMX,ITPX,LVX) / (               ACC  208
                X DMVMT(1,ITPX,LVX) + DMVMT (2,ITPX,LVX)))                                   ACC  209
000660    493 CONTINUE                                                                       ACC  210
000667        DO 580 LVX = 1,LV                                                              ACC  211
000670        JDX = IDACCD (LVX)                                                             ACC  212
000672        IF (JDX .EQ. 5)  GO TO 495                                                     ACC  213
000674        GO TO 550                                                                      ACC  214
000674    495 DO  510 IMX  = 1,2                                                             ACC  215
000676        DO  510  ITPX = 1,2                                                            ACC  216
000677        DO  505 IAX  = 1,2                                                             ACC  217
000700    505 VEHACC (IMX,IAX,ITPX) = (DHVMT (IMX,ITPX,LVX) * MACRAT (IMX,IAX,I              ACC  218
                XPX)) / EV                                                                   ACC  219
000721        DO 510 IMX = 1,2                                                               ACC  220
```

```
RUN VERSION 2.3 --PSR LEVEL 204--                                   ACCDMT                    06/30/72

001217          SUMDAT (LVX,2)     =  PERACC (1,3,4)                                         ACC  276
001221    60 CONTINUE                                                                        ACC  277
001221          IF(JSPRNT(52) .EQ. 1)  GO TO 580                                             ACC  278
          C                                                                                  ACC  279
          C     PRINT DATA OUT FOR THIS ON VEHICLE ***************************               ACC  280
001223          PRINT 402                                                                    ACC  281
001227          PRINT 483                                                                    ACC  282
001233          PRINT 484                                                                    ACC  283
001237          PRINT 542, LVX                                                               ACC  284
001245          DO 585 ITPX = 1,4                                                            ACC  285
001247    585 PRINT 489, 8HI(ITPX) , (VEHACC (1,IAX,ITPX),IAX =1,2) ,(PERACC (1,             ACC  286
                X  IBX,ITPX),IBX =1,3)                                                       ACC  287
          C                                                                                  ACC  288
001300    580 CONTINUE                                                                       ACC  289
          C                                                                                  ACC  290
          C     CALCULATE  ON  ON-GUIDEWAY  ACCIDENTS  *********                             ACC  291
          C     ZERO  OUT  PERACC  AND  VEHICC  ARRAYS                                       ACC  292
001303        DO 11  LVX=1,5                                                                 ACC  293
001305        DO 11  IAC = 1,2                                                               ACC  294
001307        DO 11  ITPX = 1,12                                                             ACC  295
001313        PERACC(LVX,IAC,ITPX) = 0.0                                                     ACC  296
001324     11 VEHACC(LVX,IAC,ITPX) = 0.0                                                     ACC  297
001326        LKY = LK + 1                                                                   ACC  298
001327        DO 700  ITPX = 1,4                                                             ACC  299
001330        DO 700  LVX = 1,LV                                                             ACC  300
001341        VEHACC (LVX,1,ITPX) =(DACRAT(1,LVX) * DONVMT(LKY,LVX,ITPX))/XU                 ACC  301
          C                                                                                  ACC  302
001341    700 PERACC (LVX,1,ITPX) =(DACRAT(2,LVX) * DLVDAT(1,LVX,ITPX)) / XU                 ACC  303
          C     ***********************************                                          ACC  304
          C     CALCULATE  TOTAL DAILY AND ANNUAL ON PERSON TRIPS FOR ALL LV                 ACC  305
001354        Y = 0.0                                                                        ACC  306
001355        YZ = 0.0                                                                       ACC  307
001356        DO 585  LVX = 1,LV                                                             ACC  308
001357        Y = Y + OPTRIP (3,LVX)                                                         ACC  309
001362    585 YZ = YZ + OPTRIP (4,LVX)                                                       ACC  310
          C                                                                                  ACC  311
          C     COMPUTE  DAILY AND ANNUAL  ON  STATION  ACCIDENTS ************               ACC  312
001367        DHOSTA =  (DHSTAC * Y ) / XU                                                   ACC  313
001372        DMASTA = (DMSTAC * YZ) / XU                                                    ACC  314
001376        SHMYDT (268,MM) =  SUMDAT (1,1)                                                ACC  315
001377        SHMYDT (269,MM) =  SUMDAT (1,2)                                                ACC  316
001402        SHMYDT (270,MM) =  VEHACC (1,1,4)                                              ACC  317
001405        SHMYDT (271,MM) =  PERACC (1,1,4)                                              ACC  318
001410        SHMYDT (272,MM) =  SHRTNT (268,MM) + SHRYDT (270,MM)                           ACC  319
001416        SHMYDT (273,MM) =  SHRTNT (269,MM) + SHRYDT (271,MM)                           ACC  320
001421        SHMYDT (274,MM) =  SUMDAT (2,1)                                                ACC  321
001423        SHMYDT (275,MM) =  SUMDAT (2,2)                                                ACC  322
001424        SHMYDT (276,MM) =  VEHACC (2,1,4)                                              ACC  323
001426        SHMYDT (277,MM) =  PERACC (2,1,4)                                              ACC  324
001430        SHMYDT (278,MM) =  SHRTNT (274,MM) + SHRYDT (276,MM)                           ACC  325
001434        SHMYDT (279,MM) =  SHRTNT (275,MM) + SHRYDT (277,MM)                           ACC  326
001437        SHMYDT (280,MM) =  DMASTA                                                      ACC  327
          C                                                                                  ACC  328
          C     PRINT THIS DATA OUT ***************************************                  ACC  329
          C                                                                                  ACC  330
001441        IF (JSPRNT (52) .EQ. 1)   GO  TO 4
```

```
RUN VERSION 2.3 --PSR LEVEL 204--                                   ACCDMT                    06/30/72

000722          IAX  =  IBX + 3                                                              ACC  221
000724    510 PERACC  (IMX,IBX,ITPX) = (ZZ(IMX,ITPX,LVX) *  MACRAT (IMX,IAX,IT               ACC  222
                XPX)) / EV                                                                   ACC  223
          C     SUM  VEHACC  AND  PERACC  ARRAYS  ****************                           ACC  224
000746        CALL SUM  (2,3,2,VEHACC)                                                       ACC  225
000751        CALL  SUM  (2,2,2,PERACC)                                                      ACC  226
          C     COMPUTE  ANNUAL  ACCIDENT  DATA *****************                              ACC  227
000754        DO 527 IMX = 1,3                                                               ACC  228
000756        DO 525 IAX = 1,4                                                               ACC  229
000757    525 VEHACC (IMX,IAX,4) = VEHACC (IMX,IAX,3) * NUMDA                                ACC  230
000767        DO 527 IBX = 1,3                                                               ACC  231
000771    527 PERACC (IMX,IBX,4) = PERACC (IMX,IBX,3) * NUMDA                                ACC  232
001003        IF  (LVX  .GE. 3)  GO TO 58                                                    ACC  233
001006        SUMDAT (LVX,1)     =  VEHACC (3,4,4)                                           ACC  234
001007        SUMDAT (LVX,2)     =  PERACC (3,1,4)                                           ACC  235
001011     58 CONTINUE                                                                       ACC  236
          C                                                                                  ACC  237
          C     PRINT  DATA OUT  FOR  THIS  ON  VEHICLE ***********************              ACC  238
          C                                                                                  ACC  239
001011        IF (JSPRNT(52) .EQ. 1 )  GO TO 4                                               ACC  240
001013    538 FORMAT (1M1,58X,25HDUAL-MODE  ACCIDENT  DATA,//)                               ACC  241
001013        PRINT 402                                                                      ACC  242
001017    482 FORMAT (1HO,5X,4ON OFF-GUIDEWAY ACCIDENTS =,/)                                 ACC  243
001017        PRINT 438                                                                      ACC  244
001023        PRINT 439                                                                      ACC  245
001027        PRINT  542 , LVX                                                               ACC  246
001035    542 FORMAT (1M ,3X,14HDUAL-MODE VEH.,I3)                                           ACC  247
001035        DO 540 IMX = 1,3                                                               ACC  248
001037        PRINT 442, CM(IMX)                                                             ACC  249
001044        DO 540 ITPX =1,4                                                               ACC  250
001046    540 PRINT 445, 8HI(ITPX) , (VEHACC(IMX,IAX,ITPX),IAX=1,4) , (PERACC(IMX,ACC  251
                X IMX,ITPX), IBX =1,2)                                                       ACC  252
001103      4 CONTINUE                                                                       ACC  253
001103        GO TO 580                                                                      ACC  254
          C                                                                                  ACC  255
          C     ON VEH.  TRANSIT-LIKE  ACCIDENT  CALULATIONS ****************                 ACC  256
          C                                                                                  ACC  257
001104    550 JKY =JK + 1                                                                    ACC  258
001106        DO 555  ITPX = 1,2                                                             ACC  259
001110        VEHACC (1,1,ITPX) = (DOPVMT (JKY,LVX,ITPX) * XACRAT (JDX,1)) / XU              ACC  260
001121        VEHACC(1,2,ITPX) = ((DPTRIP(ITPX,LVX) * XACRAT(JDX,2))/XU) *                   ACC  261
                X  (DLVDAT(3,LVX,ITPX) / DLVDAT(5,LVX,ITPX) )                                ACC  262
001140        DO 555 IBX = 1,2                                                               ACC  263
001142        IVX = IBX + 2                                                                  ACC  264
001144    555 PERACC(1,IBX,ITPX) = ((DPTRIP(ITPX,LVX) * XACRAT(JDX,IVX))/ XU)                ACC  265
                X  * (DLVDAT(3,LVX,ITPX)  /  DLVDAT(5,LVX,ITPX))                             ACC  266
          C     SUM  VEHACC  AND PERACC  ARRAYS ****************                             ACC  267
001173        CALL SUM (1,2,2,VEHACC)                                                        ACC  268
001176        CALL SUM (1,2,2,PERACC)                                                        ACC  269
          C     COMPUTE  ANNUAL  ON  ACCIDENT  DATA ***************                             ACC  270
001201        DO 560 IAX = 1,3                                                               ACC  271
001203        VEHACC (1,IAX,4) = VEHACC (1,IAX,3) * NUMDA                                    ACC  272
001207    560 PERACC (1,IAX,4) = PERACC (1,IAX,3) * NUMDA                                    ACC  273
001213        IF  (LVX  .GE. 3)  GO TO  68                                                   ACC  274
001216        SUMDAT (LVX,1)     =  VEHACC (1,1,4)                                           ACC  275
```

```
RUN VERSION 2.3 --PSR LEVEL 298--                                   ACCDMT                    06/30/72

001443          PRINT 705                                                                     ACC  331
001446     705 FORMAT (1H0,5X,40H ON-HIGHWAY ACCIDENTS *,/)                                   ACC  332
001448          DO 710 LVX = 1,LV                                                             ACC  333
001450     706 FORMAT (1H0,2X,40H VEHICLE *,I4)                                               ACC  334
001458          PRINT 706 , LVX                                                               ACC  335
001455          PRINT 707                                                                     ACC  336
001461     707 FORMAT (1H ,28X,9HTEMPORAL PERIOD*,15X,9HFATAL ACCIDENTS      PASSENGACC  337
                RER FATALITIES *,/)                                                           ACC  338
001461          DO 710 ITPX =1,4                                                              ACC  339
001463     708 FORMAT (1H ,29X,4I8,15X,F15.3,5X,F20.3)                                        ACC  340
001463          PRINT 708 , BM(ITPX) , VEMACC(LVX,1,ITPX) , PERACC(LVX,1,ITPX)                ACC  341
001500     710 CONTINUE                                                                       ACC  342
001505          PRINT 400                                                                     ACC  343
001510          PRINT 500 , OMDSTA , DMASTA                                                   ACC  344
001520     500 FORMAT (1H0,10X,24HDUAL-MODE  STATION  ACCIDENTS,12X,5HDAILY,F17. ACC  345
               X2,5X,6HANNUAL,F17.2)                                                          ACC  346
001520       5    CONTINUE                                                                    ACC  347
001520          RETURN                                                                        ACC  348
001521          END                                                                           ACC  349
*FOLLOWING VARIABLES EQUIVALENCED BUT NOT REFERENCED
NVH
NLH
DDNLHC
```

```
RUN VERSION 2.3 --PSR LEVEL 298--                                   ACCDMT                    06/30/72

ACCDMT

SUBPROGRAM LENGTH
003524

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
1      =  000363    2      =  000610    3      =  000616    4      =  001183
5      =  001520    9      =  000157    10     =  000621    50     =  001311
40     =  001221    396    =  001576    397    =  001583    406    =  001535
407    =  001525    408    =  001537    409    =  001544    410    =  001550
412    =  001603    413    =  001613    416    =  001632    417    =  001637
419    =  001654    425    =  000166    433    =  000242    435    =  000254
437    =  001670    438    =  001676    439    =  001710    442    =  001720
445    =  001723    480    =  001730    483    =  001736    484    =  001780
487    =  001763    489    =  001767    491    =  001775    495    =  000674
505    =  000700    525    =  000757    527    =  000771    530    =  002000
542    =  002014    550    =  001104    580    =  001380    598    =  002051
602    =  002066    785    =  002022    786    =  002027    787    =  002033
708    =  002044

BLOCK NAMES AND LENGTHS
RDI    =  000387    RDR    =  034004    OV23   =  028031    OV33   =  005500
SVH    =  003410    MM     =  000001

VARIABLE ASSIGNMENTS
ASSMAT =  000284C02 AUTOCH =  021415C09 AUTOND =  000U88C04 BACKOF =  031470C02
BACKON =  012575C02 BM     =  003462    CM     =  003446    DACRAT =  021622C02
DBHDAT =  000871C03 DCNCEN =  000878C09 DFARE  =  033744C02 DBLRST =  010430C02
DDDPC  =  033775C02 DDWDTH =  000858C09 DHVMT  =  004732C04 DIMT   =  000156C01
DLIFE  =  000128C01 DLNCAP =  000824C09 DLYDAT =  000281C03 DMAINT =  000811C02
DMASTA =  003523    DMDSTA =  003522    DMSTAC =  021559C02 DMVLDF =  021367C02
DNDISE =  000274C02 DNUMVH =  000257C09 DOFLHC =  021672C09 DOFPFL =  022222C02
DOFPWR =  002648C04 DOFVHT =  000311C04 DOFVMT =  005051C03 DOMPFL =  021410C02
DOMPWR =  002078C04 DONVMT =  011411C04 DPCVCM =  021434C02 DPDLYD =  021273C02
DPKURF =  020811C02 DPKWRF =  016546C09 DPTRIP =  021257C09 DTOPWT =  000821C03
DTDVMT =  000845C03 DTRDPC =  033633C09 DVDLAY =  021301C02 DVMCAP =  000012C02
DVHLDF =  000017C02 DVHSUC =  000223C09 DVHUC  =  000253C02 DVMTPC =  021412C02
DVOPUC =  033303C02 DVTRIP =  000001C04 DWAYCC =  000103C09 DYARDS =  000110C02
FLOCAP =  033757C02 HACRAT =  021556C09 HLNOIR =  021402C02 HNCHAR =  021430C02
HOPUAT =  021630C02 HPCVCM =  021530C09 HPWRC  =  004168C04 HVOPCS =  021640C02
HWYUPC =  021854C02 IA     =  003301    IAC    =  003314    IAT    =  003477
IAX    =  003510    IB     =  003304    IBX    =  003511    IDACCD =  000271C01
IHX    =  003509    IPOLCD =  000241C02 ITP    =  003502    ITPX   =  003507
IVX    =  003512    IXV    =  000240C01 IXVX   =  003503    IXVT   =  003513
JDX    =  003515    JK     =  000229C01 JKY    =  003516    JSPRNT =  000000C01
LK     =  000216C01 LKY    =  003517    LV     =  000215C01 LVX    =  003476
MM     =  003475    MMINDX =  000000C04 NOPMRS =  000000C03 NPKMRS =  021294C02
NUMUA  =  021295C02 NVC    =  000276C01 PARCK3 =  000227C01 PERACC =  002712
POLHAT =  021544C02 RESDEN =  001384C09 RSAVDT =  000000C05 SCALEF =  000000C02
SMHYDT =  000310C05 SUMDAT =  003471    TIXRAT =  000279C02 TRCHWD =  000550C04
ULANDC =  000031C02 USTRUC =  000844C09 VEMACC =  002142    XACRAT =  021002C02
```

```
RUN VERSION 2.3 --PSR LEVEL 298--                                   ACCDMT                    06/30/72

XDELAY =  021362C02 XNDATA =  001320C04 XOPCU  =  021442C09 XOPICD =  022842C02
XPCVCM =  021500C02 XPTRIP =  021342C09 XPWRC  =  003418C04 XSYSCM =  021318C02
XU     =  003506    XV     =  003505    XVMTPC =  021446C02 Y      =  003520
YZ     =  003521    Z      =  033277C09 ZZ     =  002126

START OF CONSTANTS
001523

START OF TEMPORARIES
002063

START OF INDIRECTS
002077

UNUSED COMPILER SPACE
121280
```

```
      SUBROUTINE HWYOP                                                  HYOP   2
C                                                                       HYOP   3
C*********************************************************************HYOP   4
C                                                                       HYOP   5
C     HIGHWAY  OPERATING  COST  SECTION                                 HYOP   6
C                                                                       HYOP   7
C*********************************************************************HYOP   8
      REAL  NPRWRS, NUMUA                                               HYOP   9
      REAL  NOPWRS                                                      HYOP  10
      INTEGER  PARCR3                                                   HYOP  11
      DIMENSION   HVOPCT (3,7), HWYOCT (3,6) ,CM(3), VM(2)              HYOP  12
      COMMON/RO1/          JSPRNT(80)    , NLIFE(30)     , DINT(30)     ,HYOP  13
     X   LECODE          , NVH           , NLM           , NSM          ,HYOP  14
     X   NIM             , NTM           , NTSW          , NTM          ,HYOP  15
     X   NOM             , IZ            , JR            , PARCR3(8)    ,HYOP  16
     X   NH              , NA            , IPRINT        , IEV          ,HYOP  17
     X   IPOLCD(4,2,3), IUACCD(3)        , NHV           , NOTM         ,HYOP  18
     X   HVC(9)                                                         ,HYOP  19
      COMMON/RO8/          SCALEF(18)    , DVHCAP(5)     , DVHLDF(5)    ,HYOP  20
     X DLHCAP(5)         , ULANUF(18)    , UTRANC        , USTRUC(4)    ,HYOP  21
     X DGWDTH(16)        , DCNCEW(11)    , DWATCC(5)     , DYARDS(6,13) ,HYOP  22
     X DMAINT(18)        , DVHSUF(4,8)   , DVHUC(4)      , DNUMVH(5)    ,HYOP  23
     X ASSHAT(4)         , TANHAF(4)     , DNOISE(20,16,2), RESDEV(6,7) ,HYOP  24
     X DUNPFL(225,4,4), DGLRST(225,5)    , BACKUN(225,3,3), DPKHRF(225,3) ,HYOP  25
     X IPRDRF(225,3)   ,                                                ,HYOP  26
     X NPRHWS            , NUMUA         , DPKPCT        , DPTRIP(4,3)  ,HYOP  27
     X IPOLYO(2,3)       , DVOLAV(2,3)   , DHVLUF(2,3)   , ASYSC4(4,5)  ,HYOP  28
     X IPRVCT            , APTRIP(4,4)   , ADELAY(2,4,2) , HLNDIR(2,2)  ,HYOP  29
     X AVHTPC(4)         , DVHTPF(3)     , AUTOCH(11)    , HNCHAR(4)    ,HYOP  30
     X IPCVCH(3,6,2)     , APCVCW(4,8)   , HPCVCH(2,4)   , POLRAT(3,3)  ,HYOP  31
     X HNSTAC            , MACHAT(2,5,2) , RACRAT(4,4)   , DACRAT(2,3)  ,HYOP  32
     X HIPJAT(2,4)       , HVOPC4(2,6)   , HVYOPC(2,3)   , AOPCU(4,7,4) ,HYOP  33
     X KIPIC(4,7,4)      ,DOPPRL(15A,4,4),DOPLRC(150,3,3) ,BACKOF(150,3,2),HYOP  34
     X DTYUOPC           , DCCOPF        , DHOPC         , Z(4)         ,HYOP  35
     X HWOPUC(6,9,4)     , DTHOPF(11,7)  , DOOPC(7)      , FLOCAP(5)    ,HYOP  36
     X IFAHE(4,2,2)                                                     ,HYOP  37
      COMMON /DV21/  NOPHWS  , DVTRIP(4,4)  , DTOPNT(4,4)  ,DTJVNT(4,5),HYOP  38
     X  UPHMAT(6,7,4) , DLVDAT(9,3,4)  ,  DOFVNT(150,4,4)  ,            HYOP  39
     X IOFVNT(150,4,4),  DONVNT(225,4,4)                                HYOP  40
      COMMON /DV33/  AUTOND(9,6,12)  ,  TRCKND(5,4,12)  , RNDATS(9,6,12),HYOP  41
     X  IONPWR(5,4,12) , DOPPWR(5,6,12)  , RPWRC(5,6,12) , HPWRC(5,6,12),HYOP  42
     X  DNVNT(5,6,12)                                                   ,HYOP  43
      COMMON/SVR/ RSAVNT(100,7), SNRYDT(320,5)                          HYOP  44
      COMMON /NHF/ NHINDX                                               HYOP  45
      DATA CM,VM /18H  FREEWAYS,18HSURF. ART.,10H  HIGHWAYS,10H   AUTOS HYOP  46
     X ,10H  TRUCKS  /                                                  HYOP  47
      NH = NHINDX                                                       HYOP  48
      NHVT = NHV + 1                                                    HYOP  49
      IF (JSPRNT (53) .EQ. 1 ) GO TO   4                                HYOP  50
C                                                                       HYOP  51
C     PRINT HIGHWAY OPERATING  COST  INPUT  DATA                        HYOP  52
C                                                                       HYOP  53
  808 FORMAT  (1H0)                                                     HYOP  54
```



```
      PRINT 809                                                         HYOP  56
  809 FORMAT (1H1,58X,27HHIGHWAY   OPERATING   COSTS,///)               HYOP  57
      PRINT 810                                                         HYOP  58
  810 FORMAT (1H ,10X,27HOPERATING COST  INPUT  DATA,/)                 HYOP  59
      PRINT 811                                                         HYOP  60
  811 FORMAT  (1H ,5X,*FREE*,5X,22X,*SURFACE ART.*,/)                   HYOP  61
      PRINT 812, (HOPDAT (MHX+1),MHX=1,2)                               HYOP  62
  812 FORMAT (1H ,20X,*AVE. NO. OF LANES*,13A,F10.1,17A,F10.1)          HYOP  63
      PRINT 813, (HOPDAT (MHX+3),MHX=1,2)                               HYOP  64
  813 FORMAT (1H ,20X,*AVE. NO. HYR/RT. MILES*,8X,F10.2,8X,F10.2,/)     HYOP  65
  815 FORMAT (1H0,5X,7HVEHICLP,12X,7HDEMREC.,8X,6HMAINT.,8X,6HFUEL,10X, HYOP  66
     X6HINSUR.,7X,5HPARK.,10X,5HOTHER,14X,5HTOTAL,/)                    HYOP  67
      PRINT 886                                                         HYOP  68
      PRINT 815                                                         HYOP  69
      DO 816 NHVX =1,NHV                                                HYOP  70
  816 PRINT  817, VM(NHVX),(HVOPCS(NHVX,ICT),ICT=1,6)                   HYOP  71
  817 FORMAT (1H0,3X,A10,5X,6F10.2)                                     HYOP  72
  819 FORMAT (1H0,5X,*MODE*,21X,*POWER*,15X,*MANPOWER*,13X,*MAINTENENCE*HYOP  73
     X,27X,*TOTAL*,/)                                                   HYOP  74
      PRINT  808                                                        HYOP  75
      PRINT 819                                                         HYOP  76
      DO  821  MHX =1,2                                                 HYOP  77
  821 PRINT   822 , CM(MHX), (WHYOPC (MHX,IT),IT=1,3)                   HYOP  78
C                                                                       HYOP  79
C     END  PRINTOUT OF  INPUT  DATA  - BEGIN  COST  CALCULATIONS        HYOP  80
C     ******************************************************************HYOP  81
    4 CONTINUE                                                          HYOP  82
C                                                                       HYOP  83
C     COMPUTE  NUMBER OF HOUR MILES PER  FY  AND  SA =HOPDAT(MHX,3)     HYOP  84
C     COMPUTE  NUMBER OF  MANYEARS   PER  FRY AND  SA =HOPDAT(MHX,4)    HYOP  85
      DO 825 MHX =1,2                                                   HYOP  86
      HOPDAT (MHX,3) =  MHCHAR (MHX) / HOPDAT (MHX,1)                   HYOP  87
  825 HOPDAT (MHX,4) = HOPDAT (MHX,3) *  HOPDAT (MHX,2)                 HYOP  88
C     CALCULATE  ANNUAL  VEHICLE  OPERATING  COSTS                      HYOP  89
C     INITIALIZE HVOPCT  ARRAY WITH ZERNS FOR  SUMMING                  HYOP  90
      DO 827 NHVX =1,3                                                  HYOP  91
      DO 827 ICTX =1,7                                                  HYOP  92
  827 HVOPCT (NHVX,ICTX) = 0.0                                          HYOP  93
      DO 830  ICTX =1,6                                                 HYOP  94
  830 HVOPCT (1,ICTX) = HVOPCS (1,ICTX) * AUTOND (3,6,2)                HYOP  95
      IF (NHV .EQ. 1)  GO TO 835                                        HYOP  96
C     CALCULATE TRUCK  OPERATING  COSTS                                 HYOP  97
      DO  832  ICTX =1,6                                                HYOP  98
  832 HVOPCT (2,ICTX) = HVOPCS (2,ICTX) * TRCKND (3,6,1)                HYOP  99
C     SUM ARRAY  HVOPCT  ***************************************        HYOP 100
  835 NHVX =NHV +1                                                      HYOP 101
      DO 837 NHVX =1,NHV                                                HYOP 102
      DO 837 ICTX =1,6                                                  HYOP 103
  837 HVOPCT (NHVX,7) = HVOPCT (NHVX,7) + HVOPCT (NHVX,ICTX)            HYOP 104
      DO 838 ICTX =1,7                                                  HYOP 105
      DO 838 NHVX =1,NHV                                                HYOP 106
  838 HVOPCT (NHVY,ICTX) = HVOPCT (NHVY,ICTX) + HVOPCT (NHVX,ICTX)      HYOP 107
C     END CALCULATION  OF VEH. OPERATING  COSTS ******************      HYOP 108
C     BEGIN  CALCULATION  OF HWY OPERATING COSTS ******************     HYOP 109
C     INITIALIZE  ARRAY  HWYOPT WITH ZEROS  FOR  SUMMING ******************HYOP 110
```

```
000227          DO 840 MWX =1,3                                          HYOP 111
000231          DO 840 ITA =1,4                                          HYOP 112
000232    840   MWYOCT (MWX,ITA) = 0.0                                   HYOP 113
000241          DO 842 MWX =1,2                                          HYOP 114
000242          MWYOCT (MWX,1) =  MWYOP-(MWX+1) + MOPDAT (MWX+3)         HYOP 115
000244          MWYOCT (MWX,2) =  MWYOP-(MWX+2) + MOPDAT (MWX+4)         HYOP 116
000246    842   MWYOCT (MWX,3) =  MWYOP-(MWX+3) + MOPDAT (MWX+3)         HYOP 117
C               SUM  MWYOCT  ARRAY  ***********************************HYOP 118
000252          DO 845 MWX =1,2                                          HYOP 119
000253          DO 845 ITA =1,3                                          HYOP 120
000254    845   MWYOCT (MWX,4) = MWYOCT (MWX,4) + MWYUCT (MWX,ITA)      HYOP 121
000265          DO 847 MWX =1,2                                          HYOP 122
000266          DO 847 MWX =1,2                                          HYOP 123
000267    847   MWYOCT (3,ITA) = MWYOCT (3,ITA) + MWYOCT (MWX,ITA)      HYOP 124
C               END CALCULATION  OF WAY OPERATING COSTS  ***************HYOP 125
C               COMPUTE TOTAL WAY  OPERATING  COST  ********************HYOP 126
000301          TWWYOC = WVOPCT (MWYY,7) + MWYOCT (3,4)                  HYOP 127
000303          SWWYUT (161,MM) =  WVOPCT (1,7)                          HYOP 128
000307          SWWYUT (162,MM) =  WVOPCT (MWYY,7) = WVOPCT (1,7)        HYOP 129
000312          SWWYUT (163,MM) =  WVOPCT (MWYY,7)                       HYOP 130
000315          SWWYUT (164,MM) =  WWYOPT (1,4)                          HYOP 131
000320          SWWYUT (165,MM) =  WWYOPT (2,4)                          HYOP 132
000323          SWWYUT (166,MM) =  WWYOPT (3,4)                          HYOP 133
000326          SWWYUT (167,MM) =  TWWYOC                                HYOP 134
000331          IF (TWWYOC .EQ. 0.0)  GO TO 6                            HYOP 135
000332          SWWYUT (168,MM) =  WVOPCT (MWYY,7) / TWWYOC   *  100.    HYOP 136
000337          SWWYUT (169,MM) =  WWYOPT (3,4)    / TWWYOC   *  100.    HYOP 137
000343          GO TO 7                                                  HYOP 138
000343        6 SWWYUT(168,MM) = 0.0                                     HYOP 139
000346          SWWYUT(169,MM) = 0.0                                     HYOP 140
000351        7 SWWYUT(170,MM) = 0.0                                     HYOP 141
000354          SWWYUT (171,MM) =  0.0                                   HYOP 142
C                                                                        HYOP 143
C               END ALL  WWY OPERATING  COST  CALCULATIONS - PRINT DATA OUT HYOP 144
C                                                                        HYOP 145
000357          IF (JSPRNT (54) .EQ. 1 ) GO TO   9                       HYOP 146
000361          PRINT  809                                               HYOP 147
000364          PRINT 850                                                HYOP 148
000370    850   FORMAT (1H0,10X,*VEHICLE  OPERATING  COSTS *,///)        HYOP 149
000370          PRINT 815                                                HYOP 150
000374          DO  855  MWVE = 1, MWV                                   HYOP 151
000376    855   PRINT 852 , VM(MWVE), (WVOPCT (MWVE,ICT), ICT=1,7)       HYOP 152
000415    852   FORMAT (1H0,3X,A10,5X,6F14.0,F20.0)                      HYOP 153
000415          PRINT  808                                               HYOP 154
000420          PRINT 857                                                HYOP 155
000424    857   FORMAT  (1H0,10X,*WAY  OPERATING  COSTS*,///)            HYOP 156
000424          PRINT 811                                                HYOP 157
000438          PRINT 860, WOPDAT (1,3) , WOPDAT (2,3)                   HYOP 158
000440    860   FORMAT (1H0,20X,15HTOTAL  ROUTE  =ILE*,9X,F12.0,18X,F12.0///) HYOP 159
000448          PRINT 861, WOPDAT (1,4) , WOPDAT (2,4)                   HYOP 160
000458    861   FORMAT (1H0,20X,15HTOTAL  WAYEARS*,13X,F12.0,18X,F13.0///) HYOP 161
000458          PRINT 819                                                HYOP 162
000454          DO 865 MWX =1,3                                          HYOP 163
000456    865   PRINT 866 , CM(MWX), (MWYOCT (MWX,ITA),ITA =1,4)         HYOP 164
000474    866   FORMAT (1H0,2A,A10,6X,3(F20.0,4X),F30.0)                 HYOP 165
```



```
000474          PRINT  808                                               HYOP 166
C               PRINT TOTAL WWY OPERATING COST  ************************ HYOP 167
000500          PRINT 870, TWWYOC                                        HYOP 168
000506    870   FORMAT (1H0,///,7HA,*TOTAL  HIGHWAY  OPERATING COST*,3X,F21.0) HYOP 169
000506        9 CONTINUE                                                 HYOP 170
000506          RETURN                                                   HYOP 171
000507          END                                                      HYOP 172
```



```
HWYOP

SUBPROGRAM LENGTH
000766

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
4     =  000127    5     =  000506    6     =  000343    7     =  000351
808   =  000512    809   =  000515    810   =  000523    811   =  000531
812   =  000537    813   =  000544    815   =  000555    817   =  000571
819   =  000575    822   =  000646    827   =  000141    830   =  000151
832   =  000164    835   =  000173    837   =  000200    838   =  000214
840   =  000232    845   =  000254    847   =  000267    850   =  000613
852   =  000621    857   =  000625    860   =  000632    861   =  000641
866   =  000650    870   =  000655

BLOCK NAMES AND LENGTHS
RDI   =  000307    RDR   =  034004    OV23  =  020031    OV33  =  009500
SVH   =  003410    HM    =  000001

VARIABLE ASSIGNMENTS
ASSMAT =  000284C02 AUTOCH =  021415C0> AUTONO =  000008C04 BACKOF =  031476C02
BACKON =  012575C02 CM     =  0007A7    UICRAT =  021622C02 DBHUAT =  000071C03
DCWCEN =  000070C02 DFARE  =  033764C0> DGLRST =  010438C02 DGOPC  =  033750C02
DGWUTM =  000050C02 DHVMT  =  004730C04 DINT   =  000154C01 DLIFE  =  000126C01
DLNCAP =  000024C02 DLVDAT =  000201C0> DMAINT =  000211C0> DMVLDF =  021307C02
DMOISE =  000274C02 DNUMVM =  000257C0> DOFLNC =  024782C02 DOFPFL =  022222C02
DOFWHR =  002644C04 DOFVMT =  000311C0> DONFVMT =  025051C03 DONPFL =  001410C02
DONWHR =  022670C04 DONVMT =  011611C0> DPCVCH =  021434C02 DPOLYD =  021273C02
DPRURF =  020011F02 DPRWHF =  015946C0> DPTRIP =  021257C02 DTOPWT =  000021C03
DTOVMT =  000045C03 DTAOPC =  033633C0> DVDLAY =  021381C02 DVHCAP =  000012C02
DVHLDF =  000017C02 DVHSUC =  000223C0> DVHUC  =  000253C02 DVHTPC =  021412C02
DVOPUC =  033383C02 DVTRIP =  000001C01 DVATCC =  000103C02 DVARDS =  000110C02
FLOCAP =  033757C02 HACRAT =  021556C0> HLNOIR =  021482C02 HNCHWR =  021430C02
HOPUAT =  021630C02 HPCVCH =  021520C0> HPWRC  =  004160C04 HVOPCS =  021440C02
HVOPCT =  000706    HWYOCT =  000733    HWYOPC =  021454C02 ICT    =  000746
IGTA   =  000743    IDACCD =  000271C01 IPOLCD =  000241C01 IT     =  000762
ITA    =  000744    JSPRNT =  000808C01 MM     =  000756    MWX    =  000761
MM     =  000756    MMIMDX =  000000C04 MWV    =  000274C01 MWVA   =  000757
MWVT   =  000755    NOPWRS =  000000C04 NPXHRS =  012254C02 NUMDA  =  021255C02
NVC    =  000276C01 PARCK3 =  000227C0> POLRAT =  021544C02 RESUPN =  001354C02
RSAVDT =  000080C05 SCALEF =  000000C0> SWRYDT =  000310C05 TARRAT =  000370C02
TWWYOC =  000765    TRCKWO =  000550C04 UNMDC  =  000031C02 USTRUC =  000844C02
VM     =  000752    XACRAT =  021402C0> XDELAY =  021382C02 XWOATA =  001120C04
XOPCU  =  021442C02 XOPICD =  0208A2C0> XPCVCH =  021500C02 XRTRIP =  021342C02
XPWHC  =  003410C04 XSYSCH =  021315C0> XVMTPC =  021486C02 Z      =  033277C02

START OF CONSTANTS
000511

START OF TEMPORARIES
000844
```

```
                SUBROUTINE TRNOP                                        XOP   2
          C                                                             XOP   3
          C*************************************************************XOP   4
          C                                                             XOP   5
          C     TRANSIT   OPERATING  SECTION                            XOP   6
          C                                                             XOP   7
          C*************************************************************XOP   8
          C                                                             XOP   9
000002          REAL  MPKHRS, NUMDA                                     XOP  10
000002          REAL  NOPHRS                                            XOP  11
000002          INTEGER  PARCK3                                         XOP  12
000002          DIMENSION  XOPCD(5,8,5)                                 XOP  13
000002          COMMON/RD1/          JSPRNT(80)    , NLIFE(30)     , DINT(30)      ,  XOP  14
                X    LECODE          , NVM         , NLM           , NSM           ,  XOP  15
                X    NIM             , NTM         , NTSM          , NYM           ,  XOP  16
                X    NOM             , IZ          , JK            , PARCK3(8)     ,  XOP  17
                X    NM              , NX          , IPRINT        , IXV           ,  XOP  18
                X    IPOLCD(4,2,3)   , IDACCM(3)   , NHV           , MDTM          ,  XOP  19
000002          NVC(8)                                                  XOP  20
                COMMON/RDR/          SCALEF(18)    , DVHCAP(5)     , DVHLDF(5)     ,XOP  21
                X DLNCAP(5)          , VLANDF(18)  , UTRANC        , USTRUC(4)     ,XOP  22
                X DOWDTH(18)         , DCNCEW(11)  , DWATCC(5)     , DYAROS(5,13)  ,XOP  23
                X DMAINT(18)         , DVMSUP(4,6) , DVMUC(4)      , DMUNWM(5)     ,XOP  24
                X ASSRAT(4)          , TARRAY(4)   , DNDISE(20,14,2), RESDEN(4,7)  ,XOP  25
                X DDNPFL(225,4,4)    , DGLKST(225,5) , BACKON(225,3,3), DPKHRF(225,3) ,XOP  26
                X DPKDRF(225,3)      ,                                                XOP  27
                X MPKHRS             , NUMDA       , DPKPCT        , DPTRIP(4,3)   ,XOP  28
                X DPOLTO(2,3)        , DVDLAV(2,3) , DMVLUF(2,3)   , RSYSCM(4,5)   ,XOP  29
                X APKPCT             , APTRIP(4,4) , ADELAY(2,4,2) , MLNDIR(2,2)   ,XOP  30
                X AVMTPC(4)          , DVMTPF(3)   , AUTOCH(3)     , MMCHAR(4)     ,XOP  31
                X DPCYCM(3,6,2)      , APCYCW(4,6) , MPCYCM(2,6)   , POLRAT(3,3)   ,XOP  32
                X DMSTAC             , MACRAT(2,5,2) , AACRAT(4,4) , DACRAT(2,3)   ,XOP  33
                X MOPDAT(2,4)        , MVDPCM(2,6) , MMYOPC(2,3)   , XOPCU(4,7,4)  ,XOP  34
                X XOPICD(4,7,4)      ,DOPPFL(158,4,4),DOFLKC(158,3,3) ,BACKOF(158,3,2),XOP  35
                X DYTUDPC            , DCCOPF      , DMOPC         , Z(4)          ,XOP  36
                X DVOPUC(6,9,4)      , DTMOPF(11,7)  , DGOPC(7)    , FLOCAP(5)     ,XOP  37
                X DFARE(4,2,2)                                                     XOP  38
000002          COMMON /OV23/  NOPHMS  , DVTRIP(4,4)  , DTDWHT(4,4)  ,DTDWHT(4,5),XOP  39
                X  DRMDAT(6,3,4)  , DLVDAT(6,3,4)  ,  DOFVHT(158,4,4)  ,          XOP  40
000002          X DOFVHT(158,4,4)  ,  DDNWHT(225,4,4)                             XOP  41
000002          COMMON /OV33/  AUTONO(5,4,2)  ,  TRCKND(5,6,2) , KNDATA(5,6,12),XOP  42
                X  UDNPWR(5,4,12)  , DOPPWR(5,6,12)  , RPWRC(5,6,12)  , MPWRC(5,6,12),XOP  43
                X  DMVMT(5,6,12)                                                   XOP  44
000002          COMMON/SVM/ RSAVDT(100,9), SWRYDT(320,5)                           XOP  45
000002          COMMON /WM/  MMINDA                                                XOP  46
000002          MM = MMINDA                                                        XOP  47
000004          IF (JSPRNT (55) .EQ. 1 ) GO TO   4                                 XOP  48
          C     ************************************************************XOP  49
          C     ALL INPUT DATA IS READ IN  - PRINT IT OUT                   XOP  50
          C     ************************************************************XOP  51
000006          PRINT 1000                                                         XOP  52
000011    1000  FORMAT (1H1,50X,27HTRANSIT   OPERATING   COSTS.///)                XOP  53
000011          PRINT 1007                                                         XOP  54
000015    1007  FORMAT (1H0,10X,25HOPERATING COST INPUT DATA.//)                   XOP  55
```



```
          C                                                             XOP  111
000327          DO 1040 IXV = 1,5                                       XOP  112
000331          DO 1040 JX  = 1,8                                       XOP  113
000332          DO 1040 KX  = 1,5                                       XOP  114
000333    1040  XOPCU (IXV,JX,KX) = 0.0                                 XOP  115
          C     ************************************************************XOP  116
          C     BEGIN  TRANSIT  OPERATING  COST   CALCULATIONS          XOP  117
          C     ************************************************************XOP  118
000346          DO  1045  IXV = 1,IXV                                   XOP  119
000347          XOPCD (IXV,1,1) =   XOPICD (IXV,1,1) *  XNDATA (IXV,4,2)XOP  120
000351          XOPCD (IXV,1,2) =   XOPICD (IXV,1,2) *  XNPCU (IXV,1,1) XOP  121
000353          XOPCD (IXV,1,3) =   XOPICD (IXV,1,3) *  XNPCU (IXV,1,1) XOP  122
000355          XOPCD (IXV,2,1) =   XOPICD (IXV,2,1) *  XNDATA (IXV,4,1)XOP  123
000357          XOPCD (IXV,2,2) =   XOPICD (IXV,2,2) *  XNDATA (IXV,4,2)XOP  124
000361          XOPCD (IXV,3,1) =   XOPICD (IXV,3,1) *  XNDATA (IXV,4,2)XOP  125
000363          XOPCD (IXV,4,1) =   XOPICD (IXV,4,1) *  XOPCU  (IXV,2,1)XOP  126
000365    1045  XOPCD (IXV,5,1) =   XOPICD (IXV,5,1) *  XOPCU  (IXV,3,1)XOP  127
000371          DO  1050  IXV = 1,IXV                                   XOP  128
000373          DO  1050  KX  = 1,5                                     XOP  129
000374          XOPCD (IXV,6,KX) =  XOPICD (IXV,6,KX) * XOPCU  (IXV,4,KX)XOP  130
000404    1050  XOPCD (IXV,7,KX) =  XOPICD (IXV,7,KX) * XOPCU  (IXV,5,KX)XOP  131
          C     SUM    XOPCD     ARRAY ***************************************XOP  132
000410          IXV = IXV +1                                            XOP  133
000420          JAY = 4                                                 XOP  134
000421          KAY = 5                                                 XOP  135
000422          CALL  SUM3  (6,7,6, XOPCD)                              XOP  136
000425          SWRYDT (172,MM) =  XOPCD    (1,1,5)                     XOP  137
000431          SWRYDT (173,MM) =  XOPCD    (1,2,5)                     XOP  138
000436          SWRYDT (174,MM) =  XOPCD    (1,3,1)                     XOP  139
000437          SWRYDT (175,MM) =  XOPCD    (1,4,1)                     XOP  140
000442          SWRYDT (176,MM) =  XOPCD    (1,5,1)                     XOP  141
000445          SWRYDT (177,MM) =  XOPCD    (1,6,5)                     XOP  142
000450          SWRYDT (178,MM) =  XOPCD    (1,7,5)                     XOP  143
000453          SWRYDT (179,MM) =  XOPCD    (1,8,5)                     XOP  144
000456          SWRYDT (180,MM) =  XOPCD    (2,1,5)                     XOP  145
000461          SWRYDT (181,MM) =  XOPCD    (2,2,5)                     XOP  146
000464          SWRYDT (182,MM) =  XOPCD    (2,3,1)                     XOP  147
000467          SWRYDT (183,MM) =  XOPCD    (2,4,1)                     XOP  148
000472          SWRYDT (184,MM) =  XOPCD    (2,5,1)                     XOP  149
000475          SWRYDT (185,MM) =  XOPCD    (2,6,5)                     XOP  150
000500          SWRYDT (186,MM) =  XOPCD    (2,7,5)                     XOP  151
000503          SWRYDT (187,MM) =  XOPCD    (2,8,5)                     XOP  152
000506          SWRYDT (188,MM) =  XOPCD    (3,1,5)                     XOP  153
000511          SWRYDT (189,MM) =  XOPCD    (3,2,5)                     XOP  154
000514          SWRYDT (190,MM) =  XOPCD    (3,3,1)                     XOP  155
000517          SWRYDT (191,MM) =  XOPCD    (3,4,1)                     XOP  156
000522          SWRYDT (192,MM) =  XOPCD    (3,5,1)                     XOP  157
000525          SWRYDT (193,MM) =  XOPCD    (3,6,5)                     XOP  158
000528          SWRYDT (194,MM) =  XOPCD    (3,7,5)                     XOP  159
000533          SWRYDT (195,MM) =  XOPCD    (3,8,5)                     XOP  160
000536          SWRYDT (196,MM) =  XOPCD    (5,8,5)                     XOP  161
000541          IF (SWRYDT(196,MM) .EQ. 0.0)  GO TO 5                   XOP  162
000543          SWRYDT (197,MM) =  SWRYDT (179,MM)/ SWRYDT (196,MM)  * 100.XOP  163
000552          SWRYDT (198,MM) =  SWRYDT (187,MM)/ SWRYDT (196,MM)  * 100.XOP  164
000555          SWRYDT (199,MM) =  SWRYDT (195,MM)/ SWRYDT (196,MM)  * 100.XOP  165
```



```
000015          PRINT 1008                                              XOP  56
000021    1008  FORMAT (1H0,15X,12HTRANSIT SYS.,1X,1HMRSUTF MILES,5X,9HSTOR. VS XOP  57
                AP,5X,10HMAINT- VSP)                                      XOP  58
000021          DO 1010  IXV =1,IXV                                     XOP  59
000023    1010  PRINT 1009, IXV, (XOPCD (IXV,NX,1),NX =1,3)              XOP  60
000042    1009  FORMAT (1H ,20X,I2,10X,F11.0,5X,F10.0,5X,F10.0)         XOP  61
000042          PRINT 1012                                              XOP  62
000045    1012  FORMAT (1H0,40X,9HTERMINALS,6X,9H1.,10X,9H2.,10X,9H3.,10X,9H4.)XOP  63
000045          DO 1014  IXV = 1,IXV                                    XOP  64
000047    1014  PRINT 1013 , IXV ,  (AOPCU (IXV,4,NX),NX =1,4)          XOP  65
000047    1013  FORMAT (1H ,20X,I2,20X,4F10.0)                          XOP  66
000067          PRINT 1015                                              XOP  67
000072    1015  FORMAT (1H0,40X,9HPARK. VSP,6X,9H1.,10X,9H2.,10X,9H3.,10X,9H4.)XOP  68
000072          DO 1016  IXV =1,IXV                                     XOP  69
          C     ************************************************************XOP  71
          C     FORMAT  STATEMENTS                                      XOP  72
          C     ************************************************************XOP  73
000076    1016  PRINT 1013 , IXV ,  (AOPCU (IXV,5,NX),NX =1,4)          XOP  70
000114    1020  FORMAT (1H0,8X,14HTRANSIT  SYS.  ,15X,9H1.,19X,9H2.,19X,9H3.,19X,9H4.)XOP  74
                X1X,5HTOTAL,/)                                            XOP  75
000114    1021  FORMAT(1H0,4X,9HDAY  POWER    =,4F2n.3)                   XOP  76
000114    1022  FORMAT(1H ,4X,9HMANPOWER=,4F2n.3)                        XOP  77
000114    1023  FORMAT(1H ,4X,9HMAINT.  =,4F2n.3)                        XOP  78
000114    1024  FORMAT(1H0,4X,9HEMP. DEPREC. =,4F2n.3)                   XOP  79
000114    1025  FORMAT(1H ,6X,9HOTHER    =,4F2n.3)                       XOP  80
000114    1026  FORMAT(1H0,6X,9HMANPOWER    =,4F2n.3)                   XOP  81
000114    1027  FORMAT(1H0,4X,9HSTORAGE    =,4F2n.3)                   XOP  82
000114    1028  FORMAT(1H0,4X,9HMAINTENANCE =,4F2n.3)                   XOP  83
000114    1065  FORMAT (1H )                                            XOP  84
000114    1029  FORMAT (1H ,8X,5HTENN.,T6,4X,4F20.0,10X,F10.0)          XOP  85
000114    1030  FORMAT (1H ,6X,9HPARK FAC.,T3,2X,4F20.0,10X,F10.0)      XOP  86
000114    1031  FORMAT (1H ,6X,1HM  TOTAL          ,4F20.0,10X,F10.0)    XOP  87
          C     ************************************************************XOP  88
000114          PRINT 999                                               XOP  89
000117    999   FORMAT(1H0)                                              XOP  90
000117          PRINT 1020                                              XOP  91
000123          PRINT 1021 , (XOPICD (IXV,1,1), IXV =1,IXV)             XOP  92
000136          PRINT 1022 , (XOPICD (IXV,1,2), IXV =1,IXV)             XOP  93
000151          PRINT 1023 , (XOPICD (IXV,1,3), IXV =1,IXV)             XOP  94
000164          PRINT 1024 , (XOPICD (IXV,2,1), IXV =1,IXV)             XOP  95
000177          PRINT 1025 , (XOPICD (IXV,2,2), IXV =1,IXV)             XOP  96
000212          PRINT 1026 , (XOPICD (IXV,3,1), IXV =1,IXV)             XOP  97
000225          PRINT 1027 , (XOPICD (IXV,4,1), IXV =1,IXV)             XOP  98
000240          PRINT 1028 , (XOPICD (IXV,5,1), IXV =1,IXV)             XOP  99
000253          PRINT 1065                                              XOP 100
000257          DO  1035  KX = 1,4                                      XOP 101
000261    1035  PRINT 1029 , KX , (XOPICD (IXV,6,KX),IXV =1,IXV)        XOP 102
000301          PRINT 1065                                              XOP 103
000305          DO  1036  KX = 1,4                                      XOP 104
000307    1036  PRINT 1030 , KX , (XOPICD (IXV,7,KX),IXV =1,IXV)        XOP 105
          C     ************************************************************XOP 106
          C     END  PRINTOUT  OF  ALL   INPUT  DATA                    XOP 107
          C     ************************************************************XOP 108
000327    6 CONTINUE                                                    XOP 109
          C     ZERO OUT  XOPCD  ARRAY   FOR   SUMMING                  XOP 110
```

```
000560          GO TO 4                                                         XOP 164
000560        5 SWRYOT(197,WW) = 0.0                                            XOP 167
000563          SWRYOT(198,WW) = 0.0                                            XOP 168
000566          SWRYOT(199,WW) = 0.0                                            XOP 169
000570        4 SWRYOT(200,WW) = 0.0                                            XOP 170
            C   ******************************************************************XOP 171
            C   END CALCULATION   OF OPTIMATIMA   COST  DATA  - PRINT  DATA  OUT  XOP 172
            C   ******************************************************************XOP 173
000573          IF (JSPRNT (56) .EQ. 1 ) GO TO   7                              XOP 174
000575          PRINT 1000                                                      XOP 175
000601          PRINT 1020                                                      XOP 176
000605          PRINT 1121 , (XOPCD (IAVX,I,1), IXVX =1,5)                      XOP 177
000617          PRINT 1122 , (XOPCD (IAVX,I,2), IXVX =1,5)                      XOP 178
000631          PRINT 1123 , (XOPCD (IAVX,I,3), IXVX =1,5)                      XOP 179
000643          PRINT 1031 , (XOPCD (IAVX,I,KXY) , IXVX =1,4)                   XOP 180
000657          PRINT 1124 , (XOPCD (IAVX,9,1) ,IXVX = 1,5)                     XOP 181
000671          PRINT 1125 , (XOPCD (IAVX,9,2) ,IXVX = 1,5)                     XOP 182
000703          PRINT 1031 , (XOPCD (IAVX,9,KXY) , IXVX =1,4)                   XOP 183
000717          PRINT 1126 , (XOPCD (IAVX,3,1) ,IXVX = 1,5)                     XOP 184
000731          PRINT 1127 , (XOPCD (IAVX,4,1) ,IXVX = 1,5)                     XOP 185
000743          PRINT 1128 , (XOPCD (IAVX,5,1) ,IXVX = 1,5)                     XOP 186
000755   1121   FORMAT (1H0,6X,*WAT  POWER   *,4F2.0,10X,F1.0)                  XOP 187
000755   1122   FORMAT (1H ,6X,*     HAMDOWER*,4F2.0,10X,F1.0)                  XOP 188
000755   1123   FORMAT (1H ,6X,*     MAINT.  *,4F2.0,10X,F1.0)                  XOP 189
000755   1124   FORMAT (1H0,6X,*VEH. DEMREC.*,4F2.0,10X,F1.0)                  XOP 190
000755   1125   FORMAT (1H ,6X,*     OTWER   *,4F2.0,10X,F1.0)                  XOP 191
000755   1126   FORMAT (1H0,6X,* HANPOWER   *,4F2.0,10X,F1.0)                  XOP 192
000755   1127   FORMAT (1H0,6X,* STORAGE     *,4F2.0,10X,F1.0)                  XOP 193
000755   1128   FORMAT (1H0,6X,* MAINTENANCE*,4F2.0,10X,F1.0)                  XOP 194
000755          PRINT 1045                                                      XOP 195
000761          DO  1055  KX =1,4                                               XOP 196
000763   1055   PRINT 1030 , KX , (XOPCD (IAVX,6,KX), IXVX = 1,5)               XOP 197
001002          PRINT 1031 , (XOPCD (IAVX,6,KXY) ,  IXVX =1,5)                  XOP 198
001016          PRINT 1045                                                      XOP 199
001022          DO  1060  KX =1,4                                               XOP 200
001024   1060   PRINT 1030 , KX , (XOPCD (IAVX,7,KX),IAVX = 1,5)               XOP 201
001043          PRINT 1031 , (XOPCD (IAVX,7,KXY) , IAVX = 1,5)                  XOP 202
001057          PRINT 1045                                                      XOP 203
001063          PRINT 1031 , (XOPCD (IAVX,JXY,KXY) ,IAVX =1,5)                  XOP 204
001100        7 CONTINUE                                                        XOP 205
001100          RETURN                                                          XOP 206
001101          END                                                             XOP 207
```



```
TRNOP

SUBPROGRAM LENGTH
001753

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
4     =  000570   5     =  000560   6     =  000327   7     =  001100
999   =  001267   1000  =  001105   1007  =  001113   1008  =  001121
1009  =  001132   1012  =  001146   1013  =  001147   1015  =  001153
1020  =  001182   1021  =  001173   1022  =  001200   1023  =  001205
1024  =  001212   1025  =  001217   1026  =  001224   1027  =  001231
1028  =  001236   1029  =  001245   1030  =  001253   1031  =  001261
1040  =  000333   1045  =  001243   1121  =  001274   1122  =  001302
1123  =  001310   1124  =  001316   1125  =  001324   1126  =  001332
1127  =  001340   1128  =  001346

BLOCK NAMES AND LENGTHS
RDI   =  000367   RDR   =  034884   DV23  =  020431   DV33  =  005580
SVH   =  003410   WW    =  000001

VARIABLE ASSIGNMENTS
ASSMAT =  000264C02 AUTOCM =  021415C02 AUTONO =  000000C04 BACKOF =  031478C02
BACKON =  012573C02 DACRAT =  021422C02 DAMDAT =  000071C03 DCMCEN =  000078C02
DFAME  =  033784C02 DGLKST =  010430C02 DHOPC  =  033750C02 DHWDTH =  000050C02
DHYMT  =  004730C04 DINT   =  000158C01 DLIFE  =  000120C01 DLNCAP =  000024C02
DLYUAT =  000201C03 DMAINT =  000211C02 DMYLOF =  021387C02 DNOISE =  000274C02
DNUMVM =  000257C02 DOFLKC =  026742C02 DOFPFL =  022222C02 DOFPWR =  002548C04
DOFYMT =  000311C03 DOFVMT =  005851C02 DONPFL =  001910C02 DONPWR =  002278C04
DONYMT =  011611C03 DPCVCM =  021434C02 DPOLYO =  021273C02 DPKORF =  020811C02
DPKMRF =  016540C02 DPTRIP =  021257C02 DTOPMT =  000021C03 DTOVMT =  000045C03
DTRUPC =  033633C02 DVOLAT =  021381C02 DVMCAP =  000012C02 DVMLOF =  000017C02
DVMBUC =  000223C02 DYMUC  =  000253C02 DYNTPC =  021412C02 DYOPUC =  033383C02
DYTMIP =  000001C03 DWAYCC =  000103C02 DYAROS =  000110C02 FLOCAP =  033757C02
HACMAT =  021554C02 HLNDIR =  021402C02 HNCMAR =  021430C02 HOPDAT =  021630C02
HPCYCM =  021538C02 HPWRC  =  004160C04 HVOPCS =  021440C02 HWYOPC =  021554C02
IDACCO =  000271C01 IPOLCO =  000241C01 IXV   =  000240C01 IXVA   =  001744
IXVT   =  001750    JSPRNT =  000000C01 JX    =  001747    JXY    =  001751
KX     =  001746    KXY    =  001752    MM    =  001743    MMINDX =  000000C04
MR     =  001745    NOPMRS =  000000C01 NPKMRS =  021254C02 NUMDA  =  021255C02
NVC    =  000274C01 NX     =  000230C01 PANCK3 =  000227C01 POLRAT =  021544C02
RESVEN =  001354C02 RSAVDT =  000000C04 SCALEF =  000000C02 SWRYNT =  000318C05
TAXMAT =  000270C02 TRCKWD =  000550C04 ULAMOC =  000031C02 USTRUC =  000044C02
XACMAT =  021442C02 XDELAT =  021302C02 XNDATA =  001320C04 XOPCD  =  001433
XOPCU  =  021442C02 XOPICO =  022042C02 XPCVCM =  021500C02 XPTRIP =  021342C02
XPWMC  =  003410C04 XSYSCM =  021315C02 XVMTPC =  021406C02 Z      =  032277C02

START OF CONSTANTS
001103

START OF TEMPORARIES
001254
```

```
      SUBROUTINE DMOP
      REAL  NPRMAS, NUMDA
      REAL  NOPMAS
      INTEGER PARCK3
C
C*****************************************************************
C
C     BEGIN DM OPERATING COST SECTION
C
C*****************************************************************
C
      DIMENSION DVRMOF(4), DVMOPC(6,9,4)
      DIMENSION DEMDM(4,2,3), REVENU(5,9)
      DIMENSION  DANLE(8),  DANLPR(5,9)
      DIMENSION  DEFARE(4,2,9)
      COMMON/MDI/          JSPRNT(80)     , NLIFE(30)      , DINT(30)      ,
     X   LECODE           , NVM          , NLM            , NSM           ,
     X   MIN              , NTH          , NTSH           , NYM           ,
     X   NDM              , IZ           , JK             , PARCK3(6)     ,
     X   NM               , NX           , IPRINT         , IXV           ,
     X   IPOLCD(4,2,3)    , IDACCN(3)    , NMV            , NDTM          ,
     X   NVC(5)
      COMMON/RDR/          SCALEF(10)     , DVMCAP(5)      , DVMLDF(5)     ,
     X NLNCAP(5)           , VLANDF(10)   , UTRARC         , USTRUC(4)     ,
     X NUMDTH(16)          , DCMCEW(11)   , DNATCC(5)      , DTARGS(5,13)  ,
     X NMAINT(10)          , DVMSUP(4,8)  , DVMUC(4)       , DNUMVM(5)     ,
     X ASSNAT(4)           , TARRAY(4)    , DNDISE(20,14,2), RESDEN(4,7)   ,
     X NUMPFL(225,4,4)     , DGLKST(225,5), BACKON(224,3,3), DPRMRF(225,3)  ,
     X DMRDRF(225,3)       ,
     X NPRMAS              , NUMDA        , DPRPCT         , DPTRIP(4,3)   ,
     X RPDLYD(2,3)         , DVDLAY(2,3)  , DMVLUF(2,4)    , XBYSCM(4,5)   ,
     X RPRPCT              , APTRIP(4,4)  , RDELAY(2,4,2)  , MLNDIR(2,2)   ,
     X AVRTPC(4)           , DVRTPP(3)    , AUTOCH(11)     , MNCHAR(4)     ,
     X RPCVCM(3,8,2)       , RPCVCM(4,4)  , MPCVCM(2,4)    , POLRAT(3,3)   ,
     X RMSTAC              , MACRAT(2,5,2), RACRAT(4,4)    , DACRAT(2,3)   ,
     X MOPDAT(2,4)         , MXOPCK(2,8)  , MAXOPC(2,3)    , RDPCU(4,7,4)  ,
     X RDPICD(4,7,4)       ,DOFPFL(150,4,4),DOFLKC(150,3,3) ,BACKOF(150,3,2),
     X DYUOPC              , DCCOPC       , DMOPC          , Z(4)          ,
     X DVOPUC(6,9,4)       , DTROPF(11,7) , DBOPC(7)       , FLOCAP(8)     ,
     X DFARE(4,2,2)
      COMMON /DV13/ ASUMR(14,4), DTRTMI, DTLNMI, DTYDA
      COMMON /DV23/  NOPMAS  , DVTRIP(4,4)  , DTOPMT(4,5)  ,DTDVMT(4,5),
     X  DBMDAT(6,3,4)  , DLVDAT(6,3,4)  ,  DOFVMT(150,4,4)  ,
     X DOFVMT(150,4,4),  DONVUT(225,4,4)
      COMMON/SVR/ RSAYDT(180,9), SMRYDT,320,9)
      COMMON /MM/  MMINDX
      MM = MMINDX
C
C*****************************************************************
C
C     DM VEHICLE OPERATING COSTS
C
C*****************************************************************
```



```
C
C     COMPUTE DM VEHICLE OPERATING COSTS DVMOPC ($/YR)
C
C        NV =  VEHICLE TYPES
C        MVD = 1 = DEPRECIATION
C              2 = MAINTENANCE , TIRES
C              3 = POWER (FUEL)
C              4 = GARAGE , PARKING
C              5 = INSURANCE
C              6 = TAXES
C              7 = DRIVERS
C              8 = ALL
C        JOF = 1 = DM GUIDEWAY
C              2 = OFF GUIDEWAY
C              3 = TOTAL ON + OFF
C
C*****************************************************************
C
      CALL SUM2(NVM,7,7,DVOPUP)
      LK = NLM
C     INITIALIZE  DVMOPC
      DO 610  NV = 1,6
      DO 610  NVD = 1,9
      DO 610  JOF = 1,4
  610 DVMOPC(NV,NVD,JOF) = 0.0
      LKT = LK + 1
      JKT = JK + 1
      DO 615  NV = 1,NVM
      DO 615  NVD = 1,7
      DVMOPC(NV,NVD,1) = DVOPUC(NV,NVD,1) * DONVMT(LKT,NV,4)
  615 DVMOPC(NV,NVD,2) = DVOPUC(NV,NVD,2) * DOFVMT(JKT,NV,4)
      CALL SUM2(NVM,7,2,DVMOPP)
      IND = 205
      DO  10  NV = 1,NVM
      INDA = IDD + NV
   10 SMRYDT(INDA,MM) = DVMOPP(NV,8,3)
      IF (NVM .GE. 3)  GO TO 25
      NVT = NVM + 1
      DO 15  IXX = NVT,3
      INDX =   IDD + IXX
   15 SMRYDT(INDA,MM) = 0.0
   25 SMRYDT(200,MM)  = DVMOPP(NVT,8,3)
      PRINT DM VEHICLE UNIT COSTS ON + OFF THE GUIDEWAY
      DO 620  JF = 1,2
      GO TO (622,624),JF
  622 PRINT 620
      PRINT 631
  631 FORMAT(1H , 2(4X,*VEH.*),  *     DEPRE-    MAINT-     POWER    GARAGE
     X    RE    INSUR-     TAXES    DRIVER     TOTAL* ,/ 1H , *   INDER   CO
     X   RDE    CIATION    ENANCE*, 13X, *. PARK.*, 8X, *ANCE*,///)
      GO TO 626
  624 PRINT 630
      PRINT 631
  626 DO 620  NV = 1,NVM
  620 PRINT 632, *V*,NVC(NV),(DVOPUC(NV,NVD,JF), NVD = 1,8)
  628 FORMAT(///,1H1,43HDM ON GUIDEWAY VEHICLE UNIT OPERATING COSTS,////
```



```
  630 FORMAT(18(/),1H ,44HDM OFF GUIDEWAY VEHICLE UNIT OPERATING COSTS,
     X///)
  632 FORMAT(2I8,4F10.3)
      NVT = NVM + 1
C     PRINT DM YEARLY VEH OPER. COSTS ON+ OFF, AND ON + OFF GUIDEWAY
      DO 650  JOF = 1,3
      GO TO (636,638,640), JOF
  636 PRINT 652
      PRINT 657
  657 FORMAT(1H , 2(2X,*VEH.*), *     DEPRE-     MAINT-     POWER
     X   GARAGE    INSUR-     TAXES    DRIVER     TOTAL*,/ 1H ,
     X *INDEX   CODE     CIATION     ENANCE*, 17X, *. PARK.*, 8X,
     X *ANCE*, //)
      GO TO 642
  638 PRINT 654
      PRINT 657
      GO TO 642
  640 PRINT 656
      PRINT 657
  642 DO 650  NV = 1,NVM
  650 PRINT 658, *V*,NVC(NV),(DVMOPC(NV,NVD,JOF), NVD = 1,8)
      PRINT 660,      (DVMOPC(NVT,NVD,JOF), NVD = 1,8)
  652 FORMAT(//,1H1,44HDM YEARLY VEHICLE OPERATING COST ON GUIDEWAY////
  654 FORMAT(//////////,1H ,38HDM YEARLY VEH. OPER. COST OFF GUIDEWAY,////
  656 FORMAT(//////////,1H ,41HDM YEARLY VEH. OPER. COST ON+OFF GUIDEWAY
     X///)
  658 FORMAT(1H ,2I6,4F12.0)
  660 FORMAT(/,1H ,6HTOTALS,6X,4F12.0)
C
C*****************************************************************
C
C     DM TERMINAL OPERATING COST SECTION
C
C*****************************************************************
C
C     COMPUTE DM TERM. OPER. COST
      DO 680  NDT = 1,NDTM
      DTMOPC(NDT,5) = DTROPC(NDT,2) + DTROPC(NDT,4)
      DTMOPC(NDT,6) = DTROPC(NDT,3) + DTROPC(NDT,4)
      DTMOPC(NDT,7) = DTROPC(NDT,5) + DTROPC(NDT,6)
C     SUM DATA ON ALL TERMINALS
      NTTT = NDTM + 1
C     INITIALIZE DTMOPC
      DO 681 NDTD = 1,7
      DTMOPC(NTTT,NDTD)=0.0
      DO 685  NDTA = 4,7
      DO 685  NDT = 1,NDTM
  685 DTMOPC(NTTT,NDTD) = DTMOPC(NDTT,NDTD) + DTMOPC(NDT,NDTD)
      SMRYDT (210,MM) =  DTMOPC (NDTT,7)
C     PRINT TERMINAL OPERATING COSTS
      PRINT 690
  690 FORMAT(//,1H1,36HDM YEARLY TERMINAL OPERATING COSTS,////)
      PRINT 691
  691 FORMAT(1H , * TERM TERM*, 4X,*MAINTENANCE     OPERATION*, 9X,
     X *NO. OP*, 4X, *MAINTENANCE     OPERATION*, 10X, *TOTAL*,/ 1H ,
```



```
     X *INDEX TYPE*, 2(4X,*($/YR/TERM)*), 4X, *TERMINALS*, 3(8X,*($/YR.
     X*),///)
      DO 692  NDT = 1,NDTM
  692 PRINT 694, NDT, (DTMOPC(NDT,NDTD), NDTD =1,7)
  694 FORMAT(1H ,I5,F5.0,6F15.0)
      PRINT 696,  (DTMOPC(NDTT,NDTD), NDTD = 4,7)
  696 FORMAT(/,1H ,6HTOTALS,3X,4F15.0)
C
C*****************************************************************
C
C     DM GUIDEWAY OPERATING COST SECTION
C
C*****************************************************************
C
      DGOPC(4) = DGOPC(1) + DTLNMI
      DGOPC(5) = DGOPC(2) + DTLNMI
      DGOPC(6) = DGOPC(3) + DTLNMI
      DGOPC(7) = DGOPC(4) + DAOPC(5) + DGOPC(6)
      SMRYDT (211,MM) =  DGOPC  (7)
C     PRINT DM GUIDEWAY OPER. COST
      PRINT 705
  705 FORMAT(//,1H1,33HDM GUIDEWAY YEARLY OPERATING COST,////)
      PRINT 707
  707 FORMAT(1H , *      TOTAL     MAINTENANCE      OPERATION     OPERATION     COMM.* CO
     ANT     MAINTENANCE      OPERATION     COMM.* CONT*, 10X, *TOTAL*,/
     X1H , *LANE MILES*, 31X,*($/YR/LN.MI.)*,4(8X,*($/YR)*), //)
      PRINT 710, DTLNMI,(DGOPC(JG), JG = 1,7)
  710 FORMAT(1H ,F10.2,7F15.0)
C
C*****************************************************************
C
C     OTHER DM OPERATING COSTS
C
C*****************************************************************
C
C     COMPUTE DM YARD COST
      UYOPC = DYUOPC + UTYDA
      SMRYDT (212,MM) =  DTOPC
C     PRINT YARD, CONTROL CENTER, AND MAIN. FACIL. OPER. COSTS
      PRINT 720, DYOPC
  720 FORMAT(//////////,1H ,31HDM YEARLY YARD OPERATING COST =,10X,F10.0)
      SMRYDT (213,MM) =  UCCOPC
      PRINT 725, DCCOPC
  725 FORMAT(//////////,1H ,41HDM YEARLY CONTROL CENTER OPERATING COST =,
     XF10.0)
      SMRYDT (214,MM) =  DMOPP
      PRINT 730, DMOPC
  730 FORMAT(//////////,1H ,41HDM YEARLY MAIN. FACIL. OPERATING  COST = ,
     XF10.0)
C     COMPUTE TOTAL DM YEARLY OPERATING COST
      UTOOPC = DY+OPC(NVT+8+3) + DTROPC(NDTT+7) + DBOPC(7) + DYOPC
     X+ UCCOPC + DMOPC
      SMRYDT (201,MM) =  DTOOPC
      PRINT 735,DTOOPC
  735 FORMAT(//////////,1H ,32HDM TOTAL YEARLY OPERATING COST = ,9X,
```

```
          X F10.0)
C         COMPUTE TOTAL PUBLIC VEHICLE DEPRECIATION PER YEAR
000030    DPVDEP = 0.0
000031    DO 760 NV = 1,NVM
000033    IF (NVC(NV).GT. 1 .AND. NVC(NV) .LT. 7) GO TO 750
000044    GO TO 760
000044  750 DPVDEP = DPVDEP + DVMOPC(NV,1,3)
000047  760 CONTINUE
C
C
C*********************************************************************
C
C         BEGIN BREAKEVEN FARE CALCULATIONS
C
C*********************************************************************
C
C         COMPUTE  DENOM(NV,IVP,JF)    IVP = 2 =  PERSON
C                                                 1 =  VEHICLES
C              (NV,2,1) = PER THIS      JF  =  1 =  FLAT FARE
C              (NV,1,1) = VEH THIS              2 =  FLAT MILEAGE RATE
C              (NV,2,2) = PMT
C              (NV,1,2) = VMT
C
C*********************************************************************
C
C         DTOOPC = TOTAL YEARLY DM OPERATING COST
C         DPRVOC = TOTAL PRIVATE VEH YEARLY OPERATING COST OFF THE GUIDE
C         DTABOC = TOTAL YEARLY AGENCY DM OPERATING COST
C         DTCAPC = TOTAL ANNUAL DM CAPITAL COST(W/INT.) LESS ALL VEH CAP
C         DTABCC = TOTAL ANNUAL AGENCY DM CAPITAL COST (W/INTEREST)
C         DTAVCC = TOTAL AGENCY ANNUAL VEHICLE CAPITAL COST (W/INTEREST)
C
C*********************************************************************
C
C         COMPUTE TOTAL AGENCY ANNUAL VEH. CAPITAL COST (W/INTEREST)
000052    DTAVCC = 0.0
C         COMPUTE TOTAL ANNUAL ALL VEHICLE CAPITAL COST
000053    DTVHCC = 0.0
000054    IVM = NVM + 4
000056    DO 765  IV = 5,IVM
000057    NV = IV - 4
000061    IF (NVC(NV) .GT. 2)  DTAVCC = DTAVCC + ASUMR(IV,3)
000066  765 DTVHCC = DTVHCC + ASUMR(IV,3)
000073    DO 1488 NV = 1,NVM
000074    LVA = NV
000075    DENOM(NV,2,1) = DPTRIP(4,LVA)
000080    DENOM(NV,1,1) = DVTRIP(4,LVA)
000083    DENOM(NV,1,3)=DLVDAT(1+LVA,4)
000089    DENOM(NV,2,3)=0.0
000087    DENOM(NV,2,2) = DTOPMT(4,LVA)
000091  1488 DENOM(NV,1,2) = DLVDAT(4,LVA+4)
C         INITIALIZE  BEFARE
000096    DO 1482  NV = 1,NVM
000020    DO 1482  JF = 1,2
000021    DO 1482  IVP = 1,2
```



```
000022  1482 BEFARE(NV,IVP,JF) = 0.0
C
C         COMPUTE DENOMINATOR OF OVRHDF EQUATION
000035    NLT = LK + 1
000037    ADENOM = 0.0
000040    DO 1485  NV = 1,NVM
C
C         COMPUTE OVERHEAD ALLOCATION FACTOR
000041  1485 ADENOM = ADENOM +    Z(NV) * DOMVMT(NLT,NV,4)
000052    DO 1470  NV = 1,NVM
000053  1470 OVRHDF(NV) = ( Z(NV) * DOMVMT(NLT,NV,4)) / ADENOM
000066    DMVOPC = DTOOPC - DVMOPC(NVT,0,3)
000070    PRINT 1410
000074  1410 FORMAT(1H1)
000074    DO 1475  JF = 1,2
000076    DO 1475  JLC = 1,5
000077    DO 1438  NV = 1,NVM
000100    IF ( NVC(NV).LE.3)  GO TO 1438
000103    JNF = 3
000104    GO TO 1435
000104  1438 JNF = 1
000105  1435 CONTINUE
C         COMPUTE TOTAL ANNUAL CAPITAL COST LESS ALL VEHICLE CAPITAL COST
000106    IT = NVM + 4
000107    DTCAPC = ASUMR(IT,3) - DTVHCC
000111    IT = NV + 4
000113    IF(NVC(NV).LE.  2)  GO TO 774
C         COMPUTE PUBLIC VEHICLE OPERATING COST LESS DEPRECIATION
000116    ATT = DVMOPC(NV,0,3) - DVMOPC(NV,1,3)
000120    GO TO 776
000120  774 ASUMR(IT,3) = 0.0
000122    IF (NVC(NV).EQ.2) GO TO 775
C         COMPUTE PRIVATE VEHICLE DM-GUIDEWAY OPERATING COST INCURRED BY
C         AGENCY  (POWER  + PARKING.)
000124    ATT = DVMOPC(NV,3,1) + DVMOPC(NV,4,1)
000126    GO TO 776
C         OPERATING COST TO AGENCY IF PRIVATE VEHICLE DM-GUIDEWAY USES ITS
C         OWN POWER
000127  775 ATT=DVMOPC(NV,4,1)
000131  776 CONTINUE
C         OPERATING COST PALLETS ON GUIDEWAY ONLY
000131    IF (NVC(NV).EQ.3) ATT=DVMOPC(NV,0,1)+DVMOPC(NV,1,1)
C         COMPUTE ANNUAL COST TO LOCAL AGENCY PER VEHICLE FLEET TYPE
000136    REVENU(NV,JF) = ((DTCAPC * FLOCAP(JLC)) + DMVOPC) + OVRHDF(NV))
        X + ASUMR(IT,3) * FLOCAP(JLC) + ATT
000150    NVT = NVM + 1
C         COMPUTE TOTAL ANNUAL COST TO LOCAL AGENCY
000152    REVENU(NVT,JF) = 0.0
000155    DO 1438  NB = 1, NVM
000156  1438 REVENU(NVT,JF) = REVENU(NVT,JF) + REVENU(NB,JF)
000167    IF (NVC(NV).LE.4)  GO TO 1426
000172    IVP = 2
000173    GO TO 1428
000174  1426 IVP = 1
000175  1428 CONTINUE
```



```
000775          IF(NVC(NV).LE.3.AND.JF.EQ.2) GO TO 1438
001004          JFF=JF
001007          GO TO 1437
001007  1438 JFF=3
001010  1437 CONTINUE
001010          BEFARE(NV,IVP,JF)=REVENU(NV,JF)/DENOM(NV,IVP,JFF)
001021  1428 CONTINUE
C
C         PRINT  BREAK EVEN FARE TABLES
001023          PRINT 1488, FLOCAP(JLC)
001031  1488 FORMAT(////,1H ,19HOM BREAK EVEN FARES,15X,22HLOCAL CAPITAL FACTO
        RM =,F5.2)
001031          IF(JF .EQ.1)  GO TO 1444
001033          PRINT 1492
001037  1492 FORMAT(/,1H ,32HFLAT MILEAGE RATE FARE STRUCTURE,/)
001037          PRINT 1494
001043  1494 FORMAT(1H ,18H VEH. VEH.,2(5X,5HS PER),18H  OVERHEAD,5X,18HTOTAL A
        RNN.,/1H ,18HINDER CODE,9AH        VMT        PMT,4X,6HFACTOR,8X,
        A7HREVENUE,/)
001043          GO TO 1442
001044  1444 PRINT 1492
001050  1482 FORMAT(/,1H ,19HFLAT FARE STRUCTURE,/)
001050          PRINT 1484
001054  1484 FORMAT(1H ,18H VEH. VEH.,2(5X,5HS PER),18H  OVERHEAD,5X,18HTOTAL A
        RNN.,/1H ,18HINDER CODE,9AH  VEH.TRIP  PER.TRIP,4X,6HFACTOR,8X,
        A7HREVENUE,/)
001054  1442 CONTINUE
001054          DO 1445  NV =1,NVM
001056  1445 PRINT 1485, NV, NVC(NV), BEFARE(NV, 1 ,JF), BEFARE(NV,2,JF),
        X OVRHDF(NV), REVENU(NV,JF)
001105  1485 FORMAT(1H ,2I5,4F18.3,F18.3,F15.0)
001105          PRINT 1450, REVENU(NVT,JF)
001114  1450 FORMAT(/,1H ,29X,15HTOTAL REVENUE =,F15.0)
001114  1475 CONTINUE
C
C*********************************************************************
C
C         BEGIN DUAL MODE FARE CALCULATIONS
C
C*********************************************************************
C
C         PRINT UNIT FARES
C         PRINT 1510
001120  1510 FORMAT(1H1,22H DM VEHICLE FARE RATES,///,16X,9HFLAT FARE,7X,17HFLA
        AT MILEAGE RATE,/,18H VEH. VEH.,6(4X,5HS PER),
        A8X,7HVEHICLE,9X,4HPERSON,/
        A                                                  18HINDER CODE,6H
        AVR.,6HTRIP  PER.TRIP       VMT        PMT,2(10X,5HTRIP,2X,3HVMT,
        A12X,3HPMT,//)
001124          DO 1515  NV = 1, NVM
001126  1515 PRINT 1520, NV, NVC(NV), DFARE(NV,1,1), DFARE(NV,2,1),
        X DFARE(NV,1,2),  DFARE(NV,2,2),((DENOM(NV,IVP,JF),IVP=1,2),JF=1,2)
001143  1520 FORMAT(2I5,4F18.3,6F15.0)
001143          NVT = NVM + 1
C
```



```
C         INITIALIZE REVENUE ARRAY
001165          DO 1522  NV = 1,NVT
001166          DO 1522  JF = 1,2
001167  1522 REVENU(NV,JF) = 0.0
C         COMPUTE REVENUE FROM EARN VEH TYPE  FOR BOTH FARE STRUCTURES
001174          DO 1525  NV = 1,NVM
001200          DO 1525  JF = 1,2
001201          IF(NVC(NV).LE.3.AND.JF.EQ.2) GO TO 1523
001212          JFF=JF
001213          GO TO 1524
001213  1523 JFF=3
001214  1524 CONTINUE
001214          REVENU(NV,JF) = 0.0
001217          DO 1525  IVP = 1,2
001221  1525 REVENU(NV,JF)=REVENU(NV,JF)+(DFARE(NV,IVP,JF)*DENOM(NV,IVP,JFF))
C         COMPUTE TOTAL REVENUE FOR EACH FARE STRUCTURE
001244          DO 1530  JF = 1,2
001246          REVENU(NVT,JF) =0.0
001251          DO 1530  NV = 1,NVM
001252  1530 REVENU(NVT,JF) = REVENU(NVT,JF) + REVENU(NV,JF)
C
C         PRINT REVENUE TABLE
001265          PRINT 1535
001271  1535 FORMAT(/,1H ,24HTOTAL DUAL MODE REVENUES,///,1H ,18H VEH. VEH.,
        A11X,9HFLAT FARE,3X,17HFLAT MILEAGE RATE,/1H ,18HINDER CODE,///)
001271          DO 1540  NV = 1,NVM
001273  1540 PRINT 1545, NV, NVC(NV), (REVENU(NV,JF), JF =1,2)
001314  1545 FORMAT(1H ,2I5,2F20.0)
001314          PRINT 1550, (REVENU(NVT,JF), JF = 1,2)
001326  1550 FORMAT(/,1H ,6HTOTAL5,4X,2F20.0)
C
C*********************************************************************
C
C         COMPUTE ANNUAL AGENCY COST AND PROFIT
C
C*********************************************************************
C
001326          DO 1560  NV =1,NVM
C         COMPUTE TOTAL PRIVATE VEHICLE OPERATING COST OFF THE GUIDEWAY
001338          NV=NVM
001331          IF(NVC(NV).LE.3)GO TO 1485
001334  1560 CONTINUE
001336          DMVOC = 0.0
001337          GO TO 1568
001337  1565 DMVOC=DVMOPC(NVV,0,2)
001341  1568 DTABOC = DTOOPC - DPRVOC
001343          DTABCC = DTCAPC + DTAVCC
001345          DO 1570  JLC = 1,5
001347  1570 DABLER(JLC) = (DTABCC*FLOCAP(JLC)) + DTABOC
001354          DO 1575 JLC = 1,5
001356          DO 1575 JF = 1,2
001357  1575 DABLPR(JLC,JF) = REVENU(NVT,JF) - DABLER(JLC)
001372          ITO = NVM + 9
001376          PRINT 1580, ASUMR(ITO,3)
001402  1580 FORMAT(1H1,5X,40  TOTAL ANNUAL DM CAPITAL COST (WITH INTEREST)
```

```
RUN VERSION 2.3 --PSR LEVEL 294--                                   DMOP          06/30/72

          X          = *,F15.0,/)
001402        PRINT 1591, DTAGCC                                                  DOPER441
001410   1591 FORMAT(1H ,* TOTAL ANNUAL PUBLIC DM CAPITAL COST (WITH INTEREST)    DOPER442
          X          = *,F15.0,/)                                                DOPER443
001410        DPRICC = ASUMR(ITG+3) - DTAGCC                                      DOPER444
001413        PRINT 1592, DPRICC                                                  DOPER445
001420   1592 FORMAT(1H ,* TOTAL ANNUAL PRIVATE DM CAPITAL COST (WITH INTEREST)  DOPER446
          X          = *,F15.0,/)                                                DOPER447
001420        PRINT 1593, DTOOPC                                                  DOPER448
001426   1593 FORMAT(1H ,* TOTAL ANNUAL DM OPERATING COST                         DOPER449
          X          = *,F15.0,/)                                                DOPER450
001426        SWRYDT (202,MM) = DTAGDC                                            DOPER451
001432        PRINT 1594, DTAGOC                                                  DOPER452
001437   1594 FORMAT(1H ,* TOTAL ANNUAL PUBLIC DM OPERATING COST                  DOPER453
          X          = *,F15.0,/)                                                DOPER454
001437        DPRIOC = DTOOPC - DTAGDC                                            DOPER455
001441        SWRYDT (203,MM) = DPRIDC                                            DOPER456
001445        PRINT 1595, DPRIOC                                                  DOPER457
001452   1595 FORMAT(1H ,* TOTAL ANNUAL PRIVATE DM OPERATING COST                 DOPER458
          X          = *,F15.0,/)                                                DOPER459
001452        PRINT 1596, DTCAPC                                                  DOPER460
001460   1596 FORMAT(1H ,* TOTAL ANNUAL DM CAPITAL COST (W/INT.-EXCLUDING VEHICLDOPER461
          XEST)      = *,F15.0,/)                                                DOPER462
001460        PRINT 1597, DTVCCC                                                  DOPER463
001466   1597 FORMAT(1H ,* TOTAL ANNUAL DM VEHICLE CAPITAL COST (WITH INTEREST)  DOPER464
          X          = *,F15.0,/)                                                DOPER465
001466        PRINT 1598, DTAVCC                                                  DOPER466
001474   1598 FORMAT(1H ,* TOTAL ANNUAL PUBLIC VEHICLE CAPITAL COST (WITH INTEREDOPER467
          XST)       = *,F15.0,/)                                                DOPER468
001474        DTAVCC = DTVHCC - DTAVCP                                            DOPER469
001476        PRINT 1599, DTAVCC                                                  DOPER470
001504   1599 FORMAT(1H ,* TOTAL ANNUAL PRIVATE VEHICLE CAPITAL COST (WITH INTERDOPER471
          XEST)      = *,F15.0,/)                                                DOPER472
001504        SWRYDT (204,MM) = DNVOPC                                            DOPER473
001510        PRINT 1611, DNVUPC                                                  DOPER474
001515   1611 FORMAT(1H ,* TOTAL ANNUAL DM NON-VEHICLE OPERATING COST            DOPER475
          X          = *,F15.0,/)                                                DOPER476
001515        PRINT 1612, DPVUEP                                                  DOPER477
001523   1612 FORMAT(1H ,* TOTAL ANNUAL PUBLIC VEHICLE DEPRECIATION              DOPER478
          X          = *,F15.0,/)                                                DOPER479
001523        SWRYDT (205,MM) = DPRVNC                                            DOPER480
001527        PRINT 1613, DPRVOC                                                  DOPER481
001534   1613 FORMAT(1H ,* TOTAL ANNUAL PRIVATE VEHICLE OPERATING COST (OFF-BUIDOPER482
          XEAT) = *,F15.0,/)                                                     DOPER483
C         PRINT ANNUAL AGENCY COST AND PROFIT PER LOCAL CAPITAL FACTOR           DOPER484
001534        PRINT 1586                                                          DOPER485
001540   1586 FORMAT(8X,*ANNUAL AGENCY DM EXPENDITURE AND PROFIT PER LOCALDOPER486
          X CAPITAL FACTOR,//)                                                   DOPER487
001540        SWRYDT (228,MM) = DANLPR (1)                                        DOPER488
001544        SWRYDT (227,MM) = DANLPR (2)                                        DOPER489
001547        SWRYDT (228,MM) = DANLPR (3)                                        DOPER490
001552        SWRYDT (229,MM) = DANLPR (4)                                        DOPER491
001555        SWRYDT (230,MM) = DANLPR (5)                                        DOPER492
001560        SWRYDT (231,MM) = DANLPR (1+1)                                      DOPER493
001563        SWRYDT (232,MM) = DANLPR (2+1)                                      DOPER494
```

```
RUN VERSION 2.3 --PSR LEVEL 294--                                   DMOP          06/30/72

DMOP

SUBPROGRAM LENGTH
003327

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
25    = 000111   615   = 000014   622   = 000124   624   = 000135
626   = 000145   628   = 001677   630   = 001707   631   = 001653
632   = 001717   634   = 002205   638   = 000216   640   = 000227
642   = 000237   652   = 001750   654   = 001760   656   = 001770
657   = 001722   658   = 002000   660   = 002003   681   = 000320
685   = 002330   690   = 002007   691   = 002016   694   = 002043
696   = 002047   705   = 002053   707   = 002062   710   = 002106
720   = 002111   725   = 002121   730   = 002131   735   = 002141
750   = 000544   760   = 000567   765   = 000586   774   = 000720
775   = 000727   776   = 000731   1482  = 000022   1485  = 000641
1410  = 002151   1426  = 000734   1428  = 000775   1430  = 000704
1435  = 000705   1434  = 001007   1437  = 001010   1438  = 000796
1440  = 001044   1442  = 001054   1450  = 002241   1470  = 000653
1480  = 002153   1482  = 002211   1484  = 002216   1485  = 002235
1494  = 002164   1494  = 002172   1518  = 002246   1520  = 002311
1524  = 001167   1523  = 001213   1524  = 001214   1525  = 001221
1530  = 001252   1535  = 002315   1545  = 002333   1550  = 002330
1565  = 001337   1568  = 001341   1570  = 001347   1575  = 001357
1580  = 002530   1581  = 002542   1582  = 002558   1588  = 002566
1590  = 002362   1591  = 002364   1592  = 002365   1593  = 002376
1594  = 002407   1595  = 002420   1596  = 002431   1597  = 002442
1540  = 002453   1598  = 002464   1611  = 002475   1612  = 002508
1613  = 002517

BLOCK NAMES AND LENGTHS
RDI   = 000307   RDM   = 034004   DV13  = 000127   DV23  = 000631
SVM   = 003410   MM    = 000001

VARIABLE ASSIGNMENTS
ASSMAT = 000264C02 ASUMR  = 000000C04 AUTOCM = 021415C02 BACMOF = 031470C02
BACMON = 012575C02 BEFARE = 003235    DACMAT = 021022C02 DANLFX = 003216
DANLMR = 003223    DBMOAT = 000071C04 DCCOPC = 033275C02 DCMCEN = 000070C02
DENOM  = 003154    DFARE  = 033766C02 DOLKST = 010438C02 DOOPC  = 033760C02
DOUTM  = 000050C02 DINT   = 000196C01 DLTFE  = 000120C01 DLNCAP = 000024C02
DLVMAT = 000231C04 DMAINT = 000211C02 DMOPC  = 033276C02 DNVLOF = 021387C02
DNUISE = 000276C02 DNUMVM = 000257C04 DNVOPC = 003110    DOPLEC = 026762C02
DOFPFL = 022222C02 DOFVMT = 000311C04 DOFVMT = 005051C04 DONPFL = 001410C02
DONVMT = 011411C04 DPCVCM = 021434C02 DMOLYD = 021273C02 DPROMF = 020011C02
DPRMMF = 016466C02 DPRICC = 003324    DPRIOC = 003325    DPRVNC = 003320
DPTRIP = 021257C02 DPVDEP = 003277    DTAGCC = 003322    DTAGDC = 003321
DTATCC = 003300    DTCAPC = 003313    DTLMMI = 000125C03 DTOOPC = 003276
DTOMMT = 000021C04 DTOVMT = 000465C04 DTROPC = 033633C02 DTRTCC = 003326
DTVMCC = 003301    DTVDA  = 000126C04 DVDLAY = 021301C02 DVMCAP = 000012C02
DVMLOF = 000017C02 DVMOPC = 002624    DVMSUC = 000423C02 DVMUC  = 000253C02
DVMTPC = 021412C02 DVDPUC = 033303C03 DVTRIP = 000081C04 DWATCC = 000103C02
```

```
RUN VERSION 2.3 --PSR LEVEL 294--                                   DMOP          06/30/72

001566        SWRYDT (233,MM) = DANLMR (3+1)                                      DOPER496
001571        SWRYDT (234,MM) = DANLMR (4+1)                                      DOPER497
001574        SWRYDT (235,MM) = DANLMR (5+1)                                      DOPER498
001577        SWRYDT (236,MM) = DANLMR (1+2)                                      DOPER499
001602        SWRYDT (237,MM) = DANLMR (2+2)                                      DOPER500
001605        SWRYDT (238,MM) = DANLMR (3+2)                                      DOPER501
001610        SWRYDT (239,MM) = DANLMR (4+2)                                      DOPER502
001613        SWRYDT (240,MM) = DANLMR (5+2)                                      DOPER503
001616        PRINT 1581                                                          DOPER504
001621   1581 FORMAT(1H ,*14,*WMFLAT FARE,11X, WMFLAT RATE,/)                     DOPER505
001621        PRINT 1582                                                          DOPER506
001625   1582 FORMAT(1H ,10H   LOC CAP,3(8X,12MLOCAL AGENCY),/1H ,10H    FACTOR,  DOPER507
          X   8X,12HANNU. EXPEND.,8X,12HANNU.  PROFIT,12HANNU.  PROFIT,//)        DOPER508
001625        DO 1584  JLC =1,*                                                   DOPER509
001627   1584 PRINT 1586, FLOCAP(JLC), DANLER(JLC)  , DANLPR(JLC,1)               DOPER510
          X  , DANLPR(JLC,2)                                                     DOPER511
001644   1586 FORMAT(1H ,F18.2,3F20.0)                                           DOPER512
001644        RETURN                                                              DOPER513
001645        END                                                                 DOPER514
```

```
RUN VERSION 2.3 --PSR LEVEL 294--                                   DMOP          06/30/72

DYANDS = 000110C02 DYOOPC  = 003275   DYUOPC  = 033274C02 FLOCAP  = 033957C02
HACMAT = 021556C02 HLNDTR  = 021402C03 HMCMAR  = 021430C02 HOPDAT  = 021630C02
HPCVCM = 021530C02 HVOPCS  = 021046C03 HWYOPC  = 021054C02 IDACCD  = 000271C01
IDG    = 003244    IDGX    = 003265    IPOLCD  = 000241C01 IT     = 003312
ITO    = 003323    IV      = 003303    IWM     = 003302    IWP    = 003315
IX4    = 003267    JF      = 003278    JFF     = 003316    JG     = 003274
JK     = 000234C01 JKT     = 003263    JLC     = 003311    JOF    = 003261
JSPMNT = 000000C01 LR      = 003256    LST     = 003262    LVX    = 003304
MM     = 003255    MMINDR  = 000000C04 MB      = 003315    MDT    = 003271
MDTU   = 003273    MDTM    = 000275C01 MDTT    = 003272    MLM    = 000216C01
MLT    = 003306    NOPWMS  = 000000C04 NPKMMS  = 021434C03 NUMDA  = 021355C02
MV     = 003257    MVC     = 000276C01 MYD     = 003266    MYM    = 000215C01
MVT    = 003264    MVV     = 003317    DVMMOF  = 002628    PARCK3 = 000227C01
POLMAT = 021544C02 RESDEN  = 001354C03 RPVENU  = 003204    RSAVDT = 000000C05
SCALEF = 000005C02 SMMMDT  = 000310C03 TIMRAT  = 000270C02 VLANDC = 000031C02
USTMUC = 000044C02 XACMAT  = 021002C03 XDELAY  = 021302C02 XDENOM = 003307
XDPCU  = 021462C02 XDPICO  = 022062C03 XPCVCM  = 021000C02 XPTRIP = 021342C02
XSYSCM = 021315C02 XTT     = 003316    XVMTPC  = 021406C02 Z      = 033277C02

START OF CONSTANTS
001647

START OF TEMPORARIES
002571

START OF INDIRECTS
002001

UNUSED COMPILER SPACE
117500
```

```
                SUBROUTINE SUM(I,J,M,UNIT)                                SUM    2
          C     SUBROUTINE TO EVALUATE THE SUMMATIONS.                    SUM    3
          C                                                               SUM    4
          C                                                               SUM    5
          C                                                               SUM    6
000007          DIMENSION UNIT(5,6,12)                                    SUM    7
          C                                                               SUM    8
          C     SUMMATION OF ROWS.                                        SUM    9
000007          DO 45 KM=1,M                                              SUM   10
000010          DO 45 KI=1,I                                              SUM   11
000011          UNIT(KI,J+1,KM)=0.0                                       SUM   12
000016          DO 45 KJ=1,J                                              SUM   13
000017       45 UNIT(KI,J+1,KM)=UNIT(KI,J+1,KM)+UNIT(KI,KJ,KM)            SUM   14
          C                                                               SUM   15
          C     SUMMATION OF COLUMNS.                                     SUM   16
000037          DO 46 KM=1,M                                              SUM   17
000040          DO 46 KJ=1,J                                              SUM   18
000041          UNIT(I+1,KJ,KM)=0.0                                       SUM   19
000046          DO 46 KI=1,I                                              SUM   20
000047       46 UNIT(I+1,KJ,KM)=UNIT(I+1,KJ,KM)+UNIT(KI,KJ,KM)            SUM   21
          C                                                               SUM   22
          C     SUMMATION TOWARDS REAR.                                   SUM   23
          C                                                               SUM   24
000067          DO 47 KJ=1,J                                              SUM   25
000070          DO 47 KI=1,I                                              SUM   26
000071          UNIT(KI,KJ,M+1)=0.0                                       SUM   27
000076          DO 47 KM=1,M                                              SUM   28
000077       47 UNIT(KI,KJ,M+1)=UNIT(KI,KJ,M+1)+UNIT(KI,KJ,KM)            SUM   29
          C                                                               SUM   30
          C     SUMMATION OF ROWS AND COLUMNS                             SUM   31
          C                                                               SUM   32
000117          DO 51 KM=1,M                                              SUM   33
000120          UNIT(I+1,J+1,KM)=0.0                                      SUM   34
000124          DO 51 KI=1,I                                              SUM   35
000126       51 UNIT(I+1,J+1,KM)=UNIT(I+1,J+1,KM)+UNIT(KI,J+1,KM)         SUM   36
          C                                                               SUM   37
          C     SUMMATION OF ROWS AND REAR ELEMENTS.                      SUM   38
          C                                                               SUM   39
000144          DO 54  KI=1,I                                             SUM   40
000145          UNIT(KI,J+1,M+1)=0.0                                      SUM   41
000151          DO 54  KM=1,M                                             SUM   42
000153       54 UNIT(KI,J+1,M+1)=UNIT(KI,J+1,M+1)+UNIT(KI,J+1,KM)         SUM   43
          C                                                               SUM   44
          C     SUMMATION OF COLUMNS AND THE REAR ELEMENTS.               SUM   45
          C                                                               SUM   46
000171          DO 56  KJ=1,J                                             SUM   47
000172          UNIT(I+1,KJ,M+1)=0.0                                      SUM   48
000176          DO 56  KM=1,M                                             SUM   49
000200       56 UNIT(I+1,KJ,M+1)=UNIT(I+1,KJ,M+1)+UNIT(I+1,KJ,KM)         SUM   50
          C                                                               SUM   51
          C     GRAND TOTAL                                               SUM   52
          C                                                               SUM   53
000216          UNIT(I+1,J+1,M+1)=0.0                                     SUM   54
000222          DO 57 KM=1,M                                              SUM   55
```



```
SUM

SUMPROGRAM LENGTH
000253

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
45     =  000017    46     =  000047    47     =  000077    51     =  000126
54     =  000153    56     =  000200    57     =  000223

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS
KI     =  000251    KJ     =  000252    KM     =  000250

START OF CONSTANTS
000240

START OF TEMPORARIES
000241

START OF INDIRECTS
000245

UNUSED COMPILER SPACE
127500
```



```
000223       57 UNIT(I+1,J+1,M+1)=UNIT(I+1,J+1,M+1)+UNIT(I+1,J+1,KM)      SUM   56
          C                                                               SUM   57
000230          RETURN                                                    SUM   58
000230          END                                                       SUM   59
```

```
                SUBROUTINE SUM2(I,J,M,UNIT)                                SUM2   2
          C                                                                SUM2   3
          C     SUBROUTINE TO EVALUATE THE SUMMATIONS.                     SUM2   4
          C                                                                SUM2   5
          C                                                                SUM2   6
000007          DIMENSION UNIT(6,9,6)                                      SUM2   7
          C                                                                SUM2   8
          C     SUMMATION OF ROWS.                                         SUM2   9
000007          DO 45 KM=1,M                                               SUM2  10
000010          DO 45 KI=1,I                                               SUM2  11
000011          UNIT(KI,J+1,KM)=0.0                                        SUM2  12
000016          DO 45 KJ=1,J                                               SUM2  13
000020       45 UNIT(KI,J+1,KM)=UNIT(KI,J+1,KM)+UNIT(KI,KJ,KM)             SUM2  14
          C                                                                SUM2  15
          C     SUMMATION OF COLUMNS.                                      SUM2  16
000041          DO 46 KM=1,M                                               SUM2  17
000042          DO 46 KJ=1,J                                               SUM2  18
000043          UNIT(I+1,KJ,KM)=0.0                                        SUM2  19
000050          DO 46 KI=1,I                                               SUM2  20
000052       46  UNIT(I+1,KJ,KM)=UNIT(I+1,KJ,KM)+UNIT(KI,KJ,KM)           SUM2  21
          C                                                                SUM2  22
          C     SUMMATION TOWARDS REAR.                                    SUM2  23
          C                                                                SUM2  24
000073          DO 47 KJ=1,J                                               SUM2  25
000076          DO 47 KI=1,I                                               SUM2  26
000075          UNIT(KI,KJ,M+1)=0.0                                        SUM2  27
000102          DO 47 KM=1,M                                               SUM2  28
000104       47 UNIT(KI,KJ,M+1)=UNIT(KI,KJ,M+1)+UNIT(KI,KJ,KM)             SUM2  29
          C                                                                SUM2  30
          C     SUMMATION OF ROWS AND COLUMNS                              SUM2  31
          C                                                                SUM2  32
000125          DO 51 KM=1,M                                               SUM2  33
000126          UNIT(I+1,J+1,KM)=0.0                                       SUM2  34
000133          DO 51 KI=1,I                                               SUM2  35
000134       51 UNIT(I+1,J+1,KM)=UNIT(I+1,J+1,KM)+UNIT(KI,J+1,KM)          SUM2  36
          C                                                                SUM2  37
          C     SUMMATION OF ROWS AND REAR ELEMENTS.                       SUM2  38
          C                                                                SUM2  39
000152          DO 54  KI=1,I                                              SUM2  40
000153          UNIT(KI,J+1,M+1)=0.0                                       SUM2  41
000160          DO 54  KM=1,M                                              SUM2  42
000161       54 UNIT(KI,J+1,M+1)=UNIT(KI,J+1,M+1)+UNIT(KI,J+1,KM)          SUM2  43
          C                                                                SUM2  44
          C     SUMMATION OF COLUMNS AND THE REAR ELEMENTS.                SUM2  45
          C                                                                SUM2  46
000200          DO 56  KJ=1,J                                              SUM2  47
000201          UNIT(I+1,KJ,M+1)=0.0                                       SUM2  48
000206          DO 56  KM=1,M                                              SUM2  49
000207       56 UNIT(I+1,KJ,M+1)=UNIT(I+1,KJ,M+1)+UNIT(I+1,KJ,KM)          SUM2  50
          C                                                                SUM2  51
          C     GRAND TOTAL                                                SUM2  52
          C                                                                SUM2  53
000226          UNIT(I+1,J+1,M+1)=0.0                                      SUM2  54
000232          DO 57 KM=1,M                                               SUM2  55
```



```
SUM2

SUMPROGRAM LENGTH
000265

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
45    -  000026    46    -  000052    47    -  000104    51    -  000134
54    -  000161    56    -  000207    57    -  000236

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS
KI    -  000203    KJ    -  000264    KM    -  000462

START OF CONSTANTS
000252

START OF TEMPORARIES
000253

START OF INDIRECTS
000257

UNUSED COMPILER SPACE
127500
```



```
000234       57 UNIT(I+1,J+1,M+1)=UNIT(I+1,J+1,M+1)+UNIT(I+1,J+1,KM)       SUM2  56
          C                                                                SUM2  57
000240          RETURN                                                     SUM2  58
000240          END                                                        SUM2  59
```

```
                SUBROUTINE SUM3(I,J,M,UNIT)                                     SUM3   2
          C                                                                     SUM3   3
          C*********************************************************************SUM3   4
          C                                                                     SUM3   5
          C     SUBROUTINE TO EVALUATE THE SUMMATIONS.                          SUM3   6
          C                                                                     SUM3   7
          C*********************************************************************SUM3   8
          C                                                                     SUM3   9
000007          DIMENSION UNIT (5,8,5)                                          SUM3  10
          C                                                                     SUM3  11
          C     SUMMATION OF ROWS.                                              SUM3  12
000007          DO 45 KM=1,M                                                    SUM3  13
000010          DO 45 KI=1,I                                                    SUM3  14
000011          UNIT(KI,J+1,KM)=0.0                                             SUM3  15
000016        1 DO 45 KJ=1,J                                                    SUM3  16
000017       45 UNIT(KI,J+1,KM)=UNIT(KI,J+1,KM)+UNIT(KI,KJ,KM)                  SUM3  17
          C                                                                     SUM3  18
          C     SUMMATION OF COLUMNS.                                           SUM3  19
000037          DO 46 KM=1,M                                                    SUM3  20
000040          DO 46 KJ=1,J                                                    SUM3  21
000041          UNIT(I+1,KJ,KM)=0.0                                             SUM3  22
000046          DO 46 KI=1,I                                                    SUM3  23
000047       46  UNIT(I+1,KJ,KM)=UNIT(I+1,KJ,KM)+UNIT(KI,KJ,KM)                SUM3  24
          C                                                                     SUM3  25
          C     SUMMATION TOWARDS REAR.                                         SUM3  26
          C                                                                     SUM3  27
000070          DO 47 KJ=1,J                                                    SUM3  28
000071          DO 47 KI=1,I                                                    SUM3  29
000072          UNIT(KI,KJ,M+1)=0.0                                             SUM3  30
000077          DO 47 KM=1,M                                                    SUM3  31
000101       47 UNIT(KI,KJ,M+1)=UNIT(KI,KJ,M+1)+UNIT(KI,KJ,KM)                  SUM3  32
          C                                                                     SUM3  33
          C     SUMMATION OF ROWS AND COLUMNS                                   SUM3  34
          C                                                                     SUM3  35
000122          DO 51 KM=1,M                                                    SUM3  36
000123          UNIT(I+1,J+1,KM)=0.0                                            SUM3  37
000130          DO 51 KI=1,I                                                    SUM3  38
000131       51 UNIT(I+1,J+1,KM)=UNIT(I+1,J+1,KM)+UNIT(KI,J+1,KM)               SUM3  39
          C                                                                     SUM3  40
          C     SUMMATION OF ROWS AND REAR ELEMENTS.                            SUM3  41
          C                                                                     SUM3  42
000147          DO 54  KI=1,I                                                   SUM3  43
000150          UNIT(KI,J+1,M+1)=0.0                                            SUM3  44
000155          DO 54  KM=1,M                                                   SUM3  45
000156       54 UNIT(KI,J+1,M+1)=UNIT(KI,J+1,M+1)+UNIT(KI,J+1,KM)               SUM3  46
          C                                                                     SUM3  47
          C     SUMMATION OF COLUMNS AND THE REAR ELEMENTS.                     SUM3  48
          C                                                                     SUM3  49
000174          DO 56  KJ=1,J                                                   SUM3  50
000175          UNIT(I+1,KJ,M+1)=0.0                                            SUM3  51
000202          DO 56  KM=1,M                                                   SUM3  52
000203       56 UNIT(I+1,KJ,M+1)=UNIT(I+1,KJ,M+1)+UNIT(I+1,KJ,KM)               SUM3  53
          C                                                                     SUM3  54
          C     GRAND TOTAL                                                     SUM3  55
```

```
        C
000221        UNIT(I+1,J+1,M+1)=0.0                                    SUM3  56
000225        DO 57 KM=1,4                                             SUM3  57
000226     57 UNIT(I+1,J+1,M+1)=UNIT(I+1,J+1,M+1)+UNIT(I+1,J+1,KM)     SUM3  58
        C                                                              SUM3  59
000242        RETURN                                                   SUM3  60
000242        END                                                      SUM3  61
                                                                       SUM3  62
```



```
SUM3

SUBPROGRAM LENGTH
000257

FUNCTION ASSIGNMENTS

STATEMENT ASSIGNMENTS
45     =  000017    46     =  000047    47     =  000101    51     =  000131
54     =  000156    56     =  000203    57     =  000226

BLOCK NAMES AND LENGTHS

VARIABLE ASSIGNMENTS
KI     =  000255    KJ     =  000256    KM     =  000254

START OF CONSTANTS
000244

START OF TEMPORARIES
000255

START OF INDIRECTS
000251

UNUSED COMPILER SPACE
127500
```