COMPUTERIZED TRAFFIC DATA ACQUISITION SYSTEM — UPDATED

by

Jerry L. Korf
Highway Research Scientist

(The opinions, findings, and conclusions expressed in this report are those of the author and not necessarily those of the sponsoring agencies.)

Virginia Highway & Transportation Research Council
(A Cooperative Organization Sponsored Jointly by the Virginia Department of Highways & Transportation and the University of Virginia)

Charlottesville, Virginia

January 1980
VHTRC 80-R26

1756

TABLE OF CONTENTS

1758

ABSTRACT

Although the parameters that characterize traffic flow have been established nationally for several years, it is only recently that technology has made accurate measurement of them economically feasible. This report describes a system that provides accurate measurement of great quantities of traffic flow data at a low cost by employing digital electronics for measurement and computer processes for analysis.

The system described is an improved version of one placed in operation in 1977. For traffic flow data acquisition this revised system requires a minimum of two and a maximum of three axle detectors per lane and is capable of recording data from as many as 30 axle detectors pseudo-simultaneously. A maximum of five sites can be specified with up to four lanes per site. Special circuitry was added to allow each of the detectors to be monitored during data collection.

Based on experience with this system and technological advances made since this system was designed, a recommendation for future improvements is made.

1760

COMPUTERIZED TRAFFIC DATA ACQUISITION SYSTEM — UPDATED

by

Jerry L. Korf
Highway Research Scientist

INTRODUCTION

The gathering of data that provide an accurate description of the flow of traffic on Virginia's highways is fundamental to the goal of many of the Virginia Highway and Transportation Research Council's research projects. The quantity of data necessary for confident statements with respect to traffic flow parameters such as the 85th percentile speed, traffic volumes, and traffic mix, and similar data applied to platoons (queues) of vehicles, often requires many hours of data gathering with several thousand vehicles being gauged. With advances in technology, the gathering of these data has first moved from the era of being impractically difficult and inaccurate to warrant examination, to an era of radar speed indication and pneumatic counting devices. These devices were then replaced by a system of tape switches and a chart recorder which provided an increase in accuracy while requiring a great deal of hand processing and, more recently, by tape switches and a digital electronic system which has virtually eliminated human intervention.[1]

Others have used automated procedures similar to those described here; each representing a "state-of-the-art" design that has in some way become obsolete.[2,3,4] The primary objectives in the automated traffic data acquisition system evolutionary design process are to minimize the time interval from data collection to completion of the data reduction process and to maximize the equipment portability. Computer compatible tape and advanced software have combined in satisfying the first objective while technological breakthroughs in the electronics field have made briefcase sized systems a reality.

PURPOSE AND SCOPE

Recent field experience with the "Computerized Traffic Data Analysis System", reported on in 1975, has suggested improvements to this system. Changes in hardware and software have been combined to provide a system that offers greater accuracy and yet requires fewer axle detectors per site. Additionally, the procedures for setting up the system and the interpretation of the analysis reports have been simplified. This report serves to update the 1975 report and to provide some insight as to the next logical step in the technological evolution of traffic data acquisition systems.

SYSTEM OVERVIEW

The Virginia Highway and Transportation Research Council traffic data acquisition system (TDAS) consists of a master controller, three remote stations, a magnetic tape recorder, and a software system of programs to interpret the data. This system is capable of monitoring three physically separate sites with up to four lanes per site and can record the detector activations for as many as 50,000 two-axle vehicles on a single reel of tape. The software system processes the data one site at a time, interpreting data for up to four lanes simultaneously.

The master controller, which was specially designed and constructed for this system, is housed with a standard low speed computer tape recorder as shown in Figure 1. The remote boxes (a total of three), also of special design and construction, are housed as shown in Figure 2. The detectors used are of the pressure switch, simple closure type produced by Tapeswitch Corporation of America. The master controller and remote stations require a 12V D.C. source, while the tape deck uses approximately 175 watts of 115V A.C. power. A description of the hardware installation procedure is provided in Appendix A.

The software system consists of four programs: RDTAPE to read the tape and reformat the data; PROCESS to translate switch closure events into vehicle types, speeds, headways, and lateral placements; CORECT to assist the running of PROCESS by editing incomplete data; and REPORT to summarize the output of PROCESS into average standard deviation and 85th percentile statistics by site and lane.

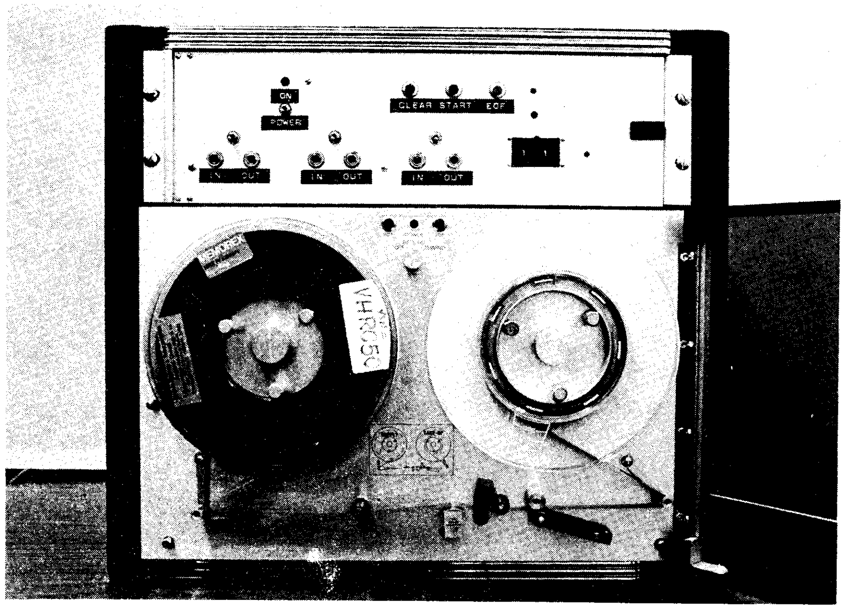


Figure 1. Central processor and tape recorder.

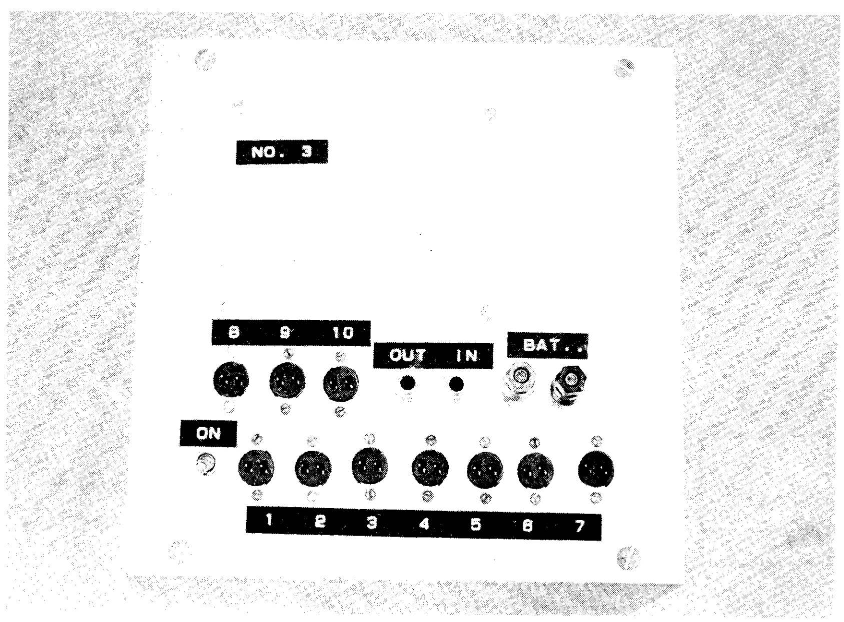


Figure. 2. Typical remote station.

1764

DATA ACQUISITION SYSTEM HARDWARE

Over the past 2 years several improvements have been made to the initial design of the system hardware. The major one is the addition of circuitry in the central processor (CP) to allow the operator to examine the condition of any tape switch connected to the system. This feature can be used before or during the data-taking process. To incorporate this addition, the fourth remote station communication connectors were removed from the front panel of the CP and replaced by a rotary switch and a light-emitting diode (LED) indicator.

The present system consists of up to three remote stations, each capable of time-multiplexing signals from up to ten tape switches, and a central processor (CP) which performs three functions: (a) sends clock pulses to the remote stations to synchronize their operation, (b) receives event (data) pulses from the remote stations and time-multiplexes these, and (c) issues commands and presents data to the digital magnetic tape unit (MTU), (see Figure 3).

The CP communicates with the remote stations and vice versa by means of differential line signaling over shielded twisted-pair cables not exceeding .80 km (.5 mile) in length. Each remote station uses one cable for receiving from the CP and another cable for transmitting data to the CP.

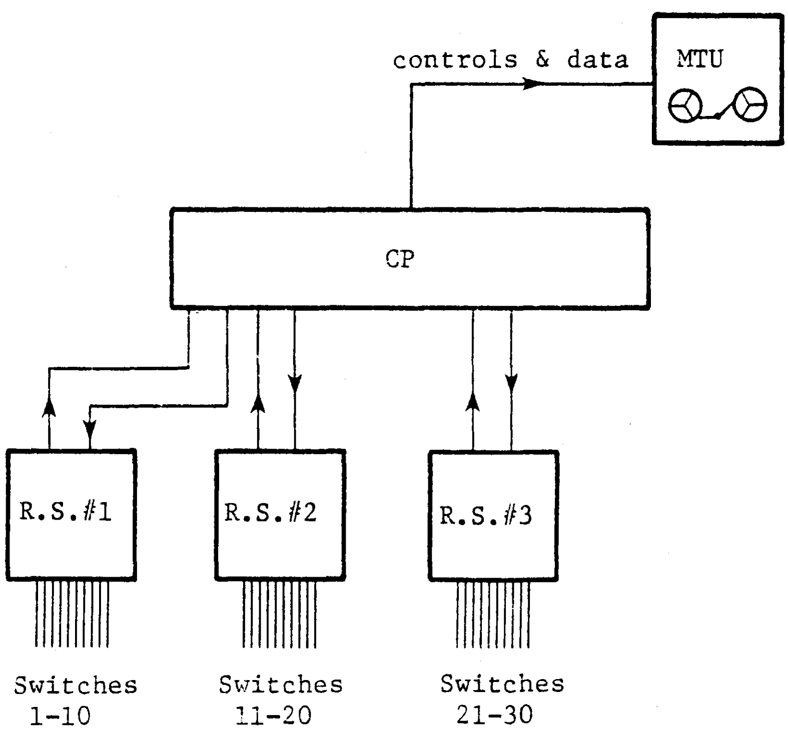


Figure 3. Overall functional diagram.

Operation

A. Synchronization of Remote Stations

The CP sends a pulse train with a 200-μsec pulse repetition time (PRT) and a 2/5 duty cycle, as depicted in Figure 4.

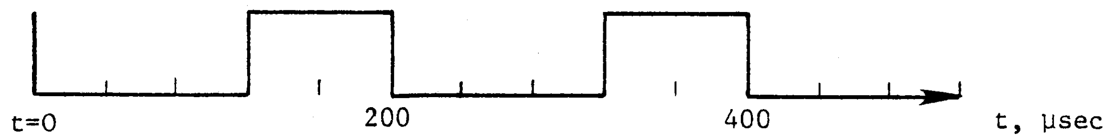


Figure 4. Central processor normal control signal.

Every tenth pulse is modified, however, to be 3/5 duty cycle, as depicted in Figure 5. In fact, this modification is made on the first pulse sent from the CP and then on every succeeding tenth pulse. The exceptional "sync pulse" is detected and used to force the decade counters of the remote stations to a zero count. Each such counter points in turn to each of the ten tape switches serviced by the remote station. These would ordinarily rezero themselves at the proper time by counting on the clock pulses, but momentary noise in the system or in the transmission line might cause an extra or missed count. Without the sync pulse detection and rezeroing mechanism, an offset between the remote station and CP decade counters would continue over time and invalidate the data recorded. With this mechanism, momentary noise can lead at worst to a few invalid events recorded on tape.

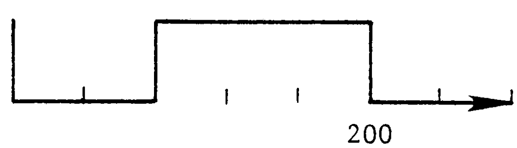


Figure 5. Central processor control signal, tenth pulse.

B. Treatment of Tape Switch Signals

It is desirable to poll every switch for a possible tripping every 2 msec, and yet a vehicle tire is capable of holding a switch closed for as long as 20 msec. Detecting only switch closers every 2 msec would, in general, lead to multiple data pulses being sent out for a single tripping. Instead, the system uses a dual flip-flop (FF) circuit wired so as to produce "1" output upon receiving the input sequence "switch open, switch closed". Events, closures of the tape switches by vehicle tires, occur randomly in time and could occur at the moment the associated FF is being examined and result in a missing event or a double event. These potential errors are eliminated by connecting a second FF to the output of the first. This second FF is activated only when both the first FF has been set and an end of a clock pulse is detected. It is the second FF that is examined to determine if an event has occurred. The second flip-flop is sampled, and if "1" it is reset. The input from the tape switch may, of course, still be low (meaning "switch closed"), but the second FF remains at a zero level until the tape switch has been open and then closed.

A logic diagram of this dual FF configuration for the processing of each tape switch input by a remote station is presented in Figure 6. The FF reset line will go low when all of the following

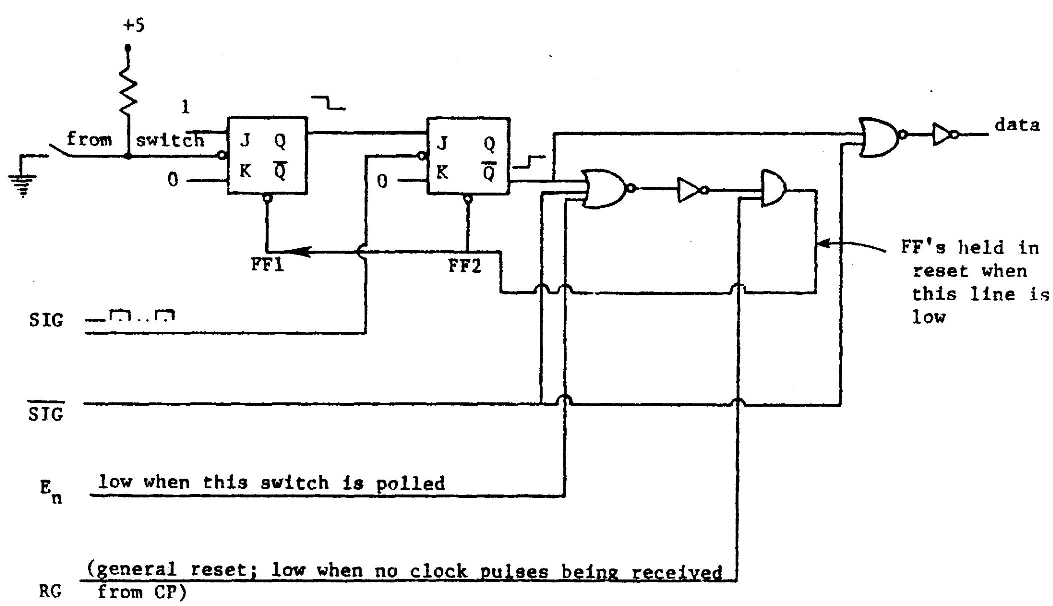


Figure 6. Control logic for tape switches, remote station.

three conditions occur (assuming the RG line is high):

(a) $\overline{Q}$ of FF2 is low, i.e., an event has been registered and synchronized;

(b) $\overline{SIG}$ is low, i.e., SIG is high; and

(c) this switch is being polled, i.e., E_n is low.

Note that (a) will occur only after falling edges of SIG, and (b) will then not be satisfied until at least 2/5 of a SIG period later (since a "sync pulse" is down for only 2/5 period). Thus $\overline{Q}$ remains low at least 2/5 · 200 μsec(or 80 μsec) and this is the minimum duration of the data pulses sent back to the CP (most are 3/5 · 200 μsec or 120 μsec). The data line is normally high; pulses are low periods.

The logic shown above is performed for either four or two tape switch inputs on the circuit boards designated "REMOTE FLIP-FLOP" boards.

C. Other Remote Station Functions

As has already been indicated, the lines E1 - E10 "polling" the various switches must be enabled (i.e., made low) in turn for 200 msec periods. This is done by a decade counter which counts on falling edges of SIG, and a "4-to-10 decoder" which makes one of ten lines low according to the binary count on the decade counter outputs. The advisability of using special sync pulses to rezero the decade counters was discussed in Section A. The circuitry utilized in detecting the sync pulses is illustrated in Figure 7.

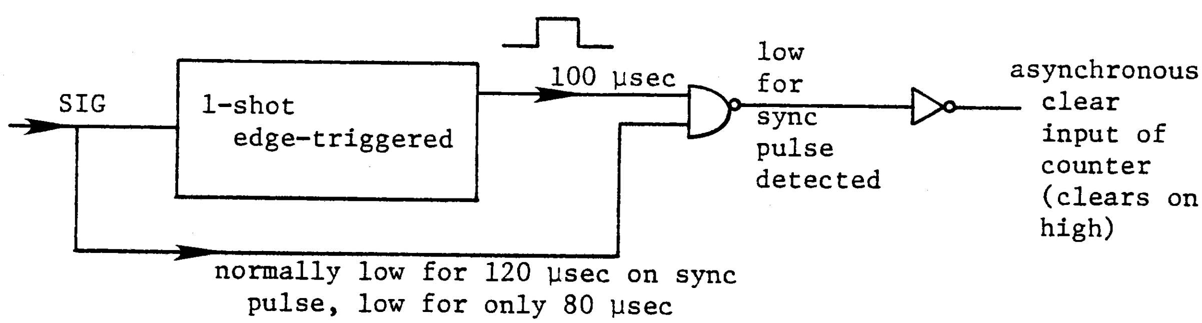


Figure 7. Sync pulse detector.

It is desirable to hold the switch input FF's in the reset mode while no signals are being received from the CP; otherwise all of these would be tripped by the time the CP was started, with the result that a burst of spurious data would be recorded. The line RG is used for this purpose; it is simply the output of a retriggerable 1-shot having the SIG line for input. RG remains high so long as SIG does not remain low for more than 5 normal SIG periods of 2 msec. If RG should go low, however, the FF's and the decade counter are held in the reset mode and zero count, respectively.

The functions described above are performed on the circuit board designated "MAIN REMOTE BOARD".

D. Central Processor: Internal Timing

The internal clock of the CP generates a square wave clock signal (CLK) of period 10 μsec, which is subdivided into CLK/2 and CLK/4 signals of period 20 and 40 μsec, respectively. The signal CLK/4 is used to generate the signals shown in Figure 8, but only after the CP has been activated by the START pushbutton.

During any interval of 200 μsec, the signals DS2 and DS1 take on four pairs of values and are used to select one of the four data lines from the remote stations (only three of which are currently being used) for a period of 40 μsec each. Recall that the data pulses last for only 80 μsec and are sent only when SIG is low, i.e., during the first 80 or 120 μsec of every 200 μsec period. It would thus be impossible to see a data pulse on lines 3 and 4, which would be selected by DS2, DS1 during the interval 120-200 μsec. To overcome this, FF's are used as temporary memory devices to hold data pulses for 200 μsec after the period begins; these are cleared by the signal DR during the first 40 μsec of the next period. Note that the signal SIG sent to the remote stations is logically DS2 + Φ DSI, where Φ is logical 1 only when the CP's main decade counter reads zero.

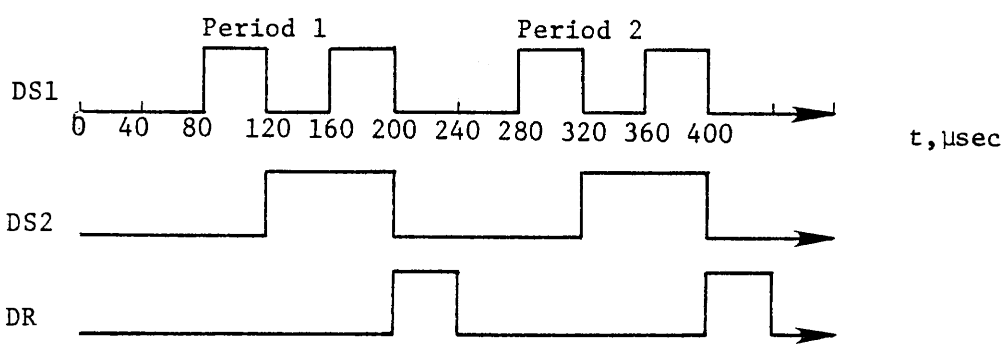


Figure 8. Typical pulse train to select remote data lines.

The 10 μsec oscillator, frequency dividers, and logic circuitry associated with the development of DS1, DS2, DR, and SIG are to be found on the circuit board designated "CLOCK + SIG GEN." The main decade counter referred to above, the data receivers and associated FF's, and the remote station selector are located on the board designated "RECEIVER."

E. CP: Data Compilation and Transfer

The system is designed to record to within 2 msec accuracy the time of passage of vehicle axles over any of forty switches. This time is taken to be an 18-bit binary integer variable which increments every 2 msec and overflows back to zero roughly every 7.5 minutes. A switch closure event will be recorded by indicating the switch number using the first 6 bits, plus the 18-bit time value for a total of 24 bits. The magnetic tape unit stores 6 bits in parallel (one frame on tape); therefore, it is necessary for the CPU to present 4 groups of 6 bits each successively on the 6 data lines to the MTU.

The 6 bits specifying the switch number are organized as follows. The most significant bit (MSB) and second MSB are used to select the remote stations in turn, as described in Section D; they (signals DS2 and DS1) encode the station to which a switch belongs. The third through sixth MSB's are the outputs of the main decade counter. Their codings change every 200 μsec and indicate a particular switch of the remote station selected by DS2, DS1. The switch-designator bits point to the switches in the order SW1, SW11, SW21, SW31, SW2, SW12, SW22, SW32, etc., changing every 40 μsec. When an event is detected at a remote station during a 2-msec cycle, the CP input FF for that station will be set during the 200-μsec interval (within the 2-msec cycle) corresponding to the switch number (1 through 10) at the remote station. The lines DS2, DS1, and the four main decade counter outputs of the CP will specify the switch, as described above, during a 40-μsec subinterval of the 200-μsec interval. Thus all of the data specifying the event must be stored in the buffer of the MTU within 40 μsec. Since there are four 6-bit words used for each event to store, six selector devices capable of choosing one of four inputs are used, and the select code is switched every 10 μsec. (The select code is given by the lines S2 (MSB) and S1, which are identical to CLK/4 and CLK/2, respectively.) This is done continuously, whether or not an event was detected. The six selector outputs are the data lines to the MTU buffer. Event detection results in sending four 10-μsec pulses along the record command line (REC) to the MTU; the data line values are buffered in, according to the MTU specifications, when the REC line goes high. Since square pulses are used (the system CLK is simply gated into REC), REC stays high for 5 μsec, and the data lines remain fixed as well.

The functions described above are largely performed by the "18-bit counter and output selector" circuit board.

F. CP: System Clear, Start, Inter-record Gap and Pseudo-end-of-file

The system CLEAR and START pushbuttons may be thought of as inputs to a simple FF whose outputs are the system state lines R (is high when system is cleared and is low when active) and S = $\overline{R}$. Actually, the situation is a little more complicated in that the CP is capable of clearing itself after the pseudo-EOF (PEOF) sequence has been executed. Thus, the logic has the form illustrated in Figure 9.

The lines S and R are used to clear counters or free them and to enable and disable signals to the remote stations and to the MTU.

The dual buffers of the MTU can store a maximum of 512 6-bit characters each. Issuance of an inter-record gap pulse (IRG) to the MTU causes subsequent data to be loaded into the other buffer while the first buffer is dumped to tape and a record gap made.

The CP uses a programmable counter to issue an IRG pulse simultaneously (to within the propagation delay of an 8-bit counter + 3 TTL gates) with the nth REC pulse after the last IRG is issued. The value of n here is wired to be any multiple of 16 up to 256 (the CP is being delivered with n wired to 256). The simultaneity of the IRG with a REC pulse is in conformity with the MTU spec's.

An IRG is issued when all outputs of an 8-bit counter become zero after counting on a REC pulse just initiated. The 8-bit counter consists of a low-order hex (4-bit) counter cascaded with a programmable hex counter. During system clear or after completion of an IRG pulse, the programmable counter displays its "programmed" (actually hardwired) initial count, which would be equal to κ ($0<\kappa<16$) to achieve a fixed record length $n = 16\ (16-\kappa)$. A 1-shot triggering off the falling edge of the IRG pulse is responsible for reloading the counter with the fixed initial count (see functional diagram of "TAPE INTERFACE" circuit board).

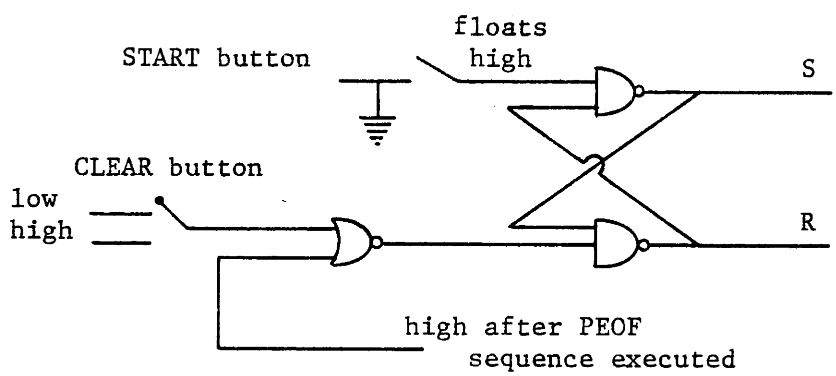


Figure 9. CLEAR and START logic.

The CP separates various blocks of records of data on tape by writing a program-recognizable mark referred to as a "pseudo-EOF" (PEOF). Depression of the "EOF" pushbutton causes the line EOF to go high, but only after the current 80 used cycle so as not to interrupt any event data store operation already in progress. The high state of EOF produces high states on the six lines leading to the output selector, which specify the switch number. In the meantime, the REC command line to the MTU is strobed until the current record is filled. The above mentioned 1-shot delivers a short pulse following the end of the IRG pulse, which (only if the EOF line had been high) is gated into the system state FF and clears the system. The net result is that a series of events on "switch" #63 = 111111_2 (which is not an actual switch) are recorded to fill out the current tape record.

G. CP: Tape Switch Monitor Circuitry

The tape switch test feature includes two rotary decade switches, an LED indicator, and associated switch number decoding circuitry. One rotary switch supplies the remote station number and the other supplies the tape switch number. The LED will remain lighted for approximately 1/2 second after the closing of the selected tape switch. The outputs of the rotary switches are compared to the current 6-bit switch number to determine if the current input is from the selected switch; if so, the LED is lighted.

TRAFFIC DATA ACQUISITION SYSTEM SOFTWARE

The purpose of the TDAS software is to process data recorded by the TDAS hardware by constructing the characterization of vehicles and summarizing these into traffic flow characteristics by lane. All software is programmed in FORTRAN for the CDC CYBER 172.

The software system consists of four programs; three core programs with a processing sequence as illustrated in Figure 10 and a utility program with a processing flow as illustrated in Figure 11. The first program, designated "RDTAPE", is a preprocessing program that reads the field data tape and generates a disk file with detector activation times and detector numbers reformatted to facilitate subsequent processing. The program "PROCESS" reads the disk file generated by "RDTAPE" and translates detector activations into vehicle speeds and axle spacings — which are used to establish headways, lateral placements, and to conjecture vehicle types. This program optionally lists individual vehicle characteristics, writes these characteristics to a disk file for further processing, or both writes a detailed vehicle report and creates a disk file. The program "REPORT" translates the individual vehicle characterizations into traffic flow parameters such as traffic volume, speed statistics, headway distributions, and queueing information (i.e., vehicle platooning data).[5]

1772

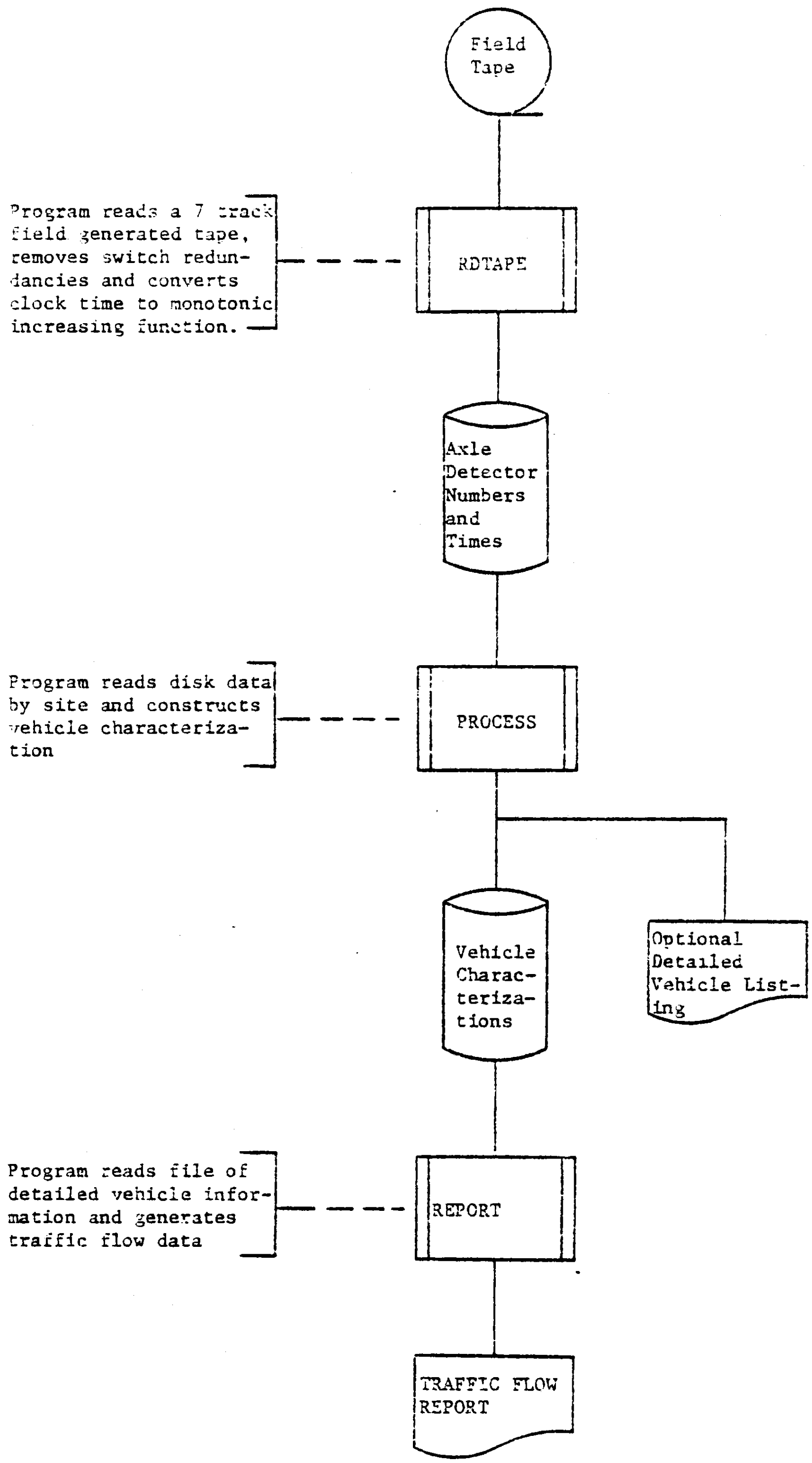


Figure 10. TDAS program execution sequence.

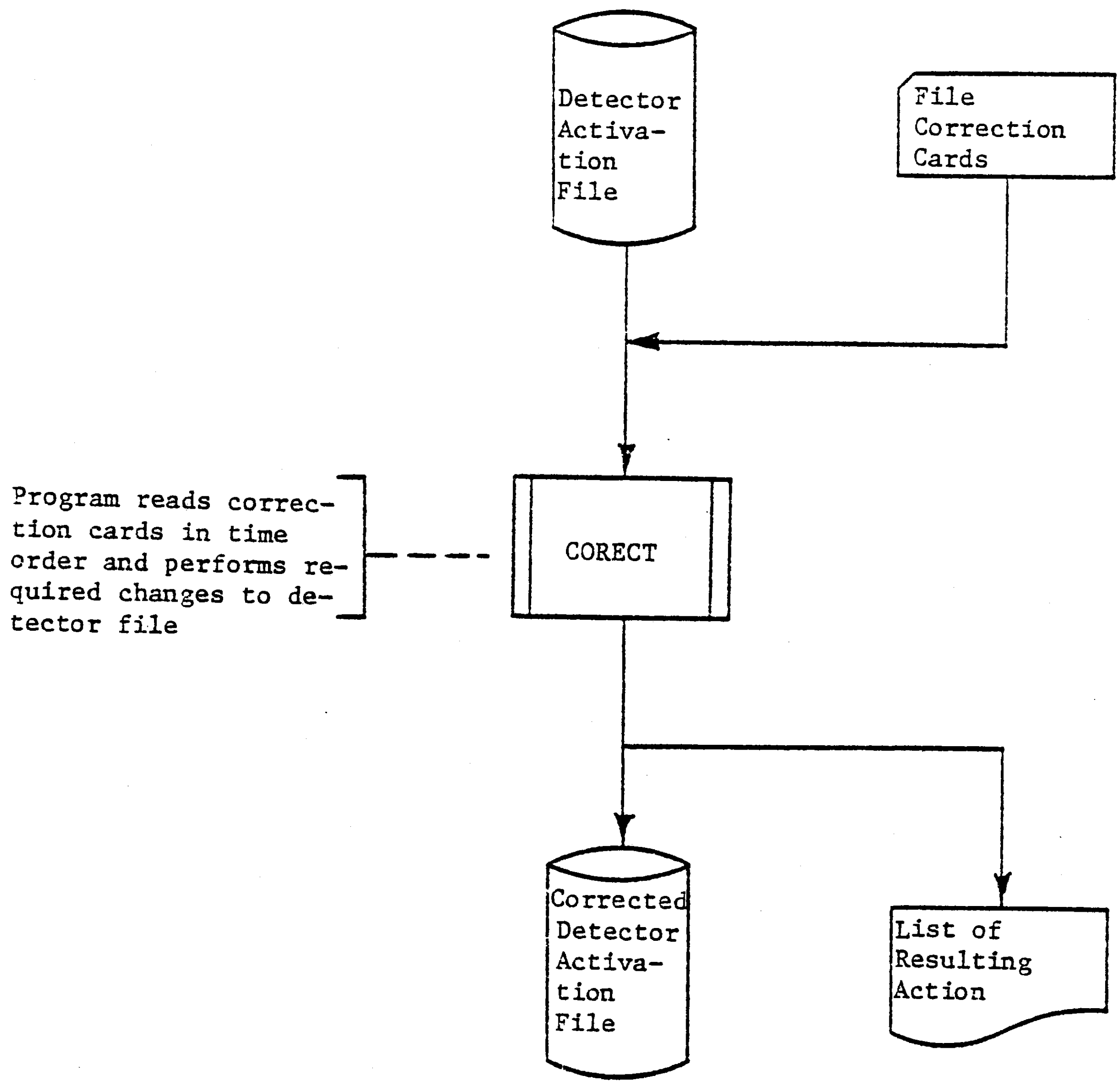


Figure 11. TDAS file correction procedure.

A utility program "CORECT" is provided to add or delete data from the disk file created by RDTAPE. This utility is used in conjunction with program PROCESS to detect and correct data errors resulting from lane changes and detector failures.

The function, control, input and output of these programs are described in this section, while program listings are provided in Appendix B.

RDTAPE Program

This program reads and reformats the field data recorded on a 7-track tape by the TDAS hardware. The format of this tape is four characters per record (a character contains six bits) and 60 records per block. The first character of each record identifies the detector that has been actuated and the last three characters contain the time of detector actuation. The detector (switch) identification number is coded by remote terminal (the two high order bits) and detector number within that terminal (the four low order bits). Although this coding scheme allows up to four remote terminals (00, 01, 10, 11) and up to sixteen detectors at each terminal (0000, ..., 1111), hardware facilities limit the system to three remote terminals and ten detectors per terminal. The program checks for valid remote terminal codes (00, 01, 10) and detector codes (0000, ..., 1010), tabulates the errors, and eliminates the erroneous records. The otherwise erroneous detector identification code consisting of all 1's (site four and detector sixteen) is used to indicate "end of data". The time (18 bits) indicates the number of 2-msec time increments that have transpired since the last time the clock counter recycled (this occurs approximately every 7.5 minutes). For this reason it is important that an event be recorded within 7.5 minutes of the previous event to ensure the integrity of the clock. A continuous test of the detector actuation time is made to ensure that the clock time is maintained in increasing order.

In an effort to provide a system exhibiting flexibility of site and detector configuration, program control parameters are made available to the user through a FORTRAN feature designated "NAMELIST".

This FORTRAN input feature is employed throughout the TDAS software system for program control. The convention for using NAMELIST includes the following four rules:

1. The word "&PARAM" must appear (with the & in column 2) at the beginning of a NAMELIST keyword list.

2. Data are assigned by placing the keyword, an equals sign, and then the data value terminated with a comma.

3. Keywords that accept multiple values (arrays) are assigned values by listing the values after the equals sign separated by commas.

4. The keyword list is terminated by the word "&END".

The first parameter keyword "RTLMT" is an integer variable used to set the upper limit on the number of remote terminals used during the data gathering. Tape records containing a remote terminal number exceeding this value will be considered in error and discarded. The default value for this variable is the system maximum of three. The second control parameter is keyword "SITEID". This variable is a thirty-element array that allows the user to assign a site identification number to each of the thirty possible detectors. The default values for this variable assigns the first ten detectors to site number one, the next ten to site number two, and the last ten to site number three. The third control parameter keyword, "DELTAT", is used to establish the time threshold for the elimination of detector activation redundancies. Repeated detector activations, within the specified DELTAT, 2-msec clock counts, are considered to be a result of spurious electronic noise (e.g., switch contact bounce), and only the first one is transferred to the disk file. Since a vehicle advances 8.94 mm (.352 inch) per clock count for every 16.1 km/h (10 mph) of velocity, the correct value for DELTAT should be based on the maximum speed and the minimum wheelbase of the vehicles to be monitored. Although vehicle speeds vary with location, wheelbase minimums are relatively constant at approximately 1.2 m (4 ft.) for motorcycles and dual-axle trucks. If the location selected experiences vehicle speeds that seldom exceed 113 km/h (70 mph), a DELTAT of 18 would allow a wheelbase as short as 1.13 m (3.7 ft.) to be detected. Experience has shown that 18 is an effective number for most situations.

The characteristics of these parameters are summarized in Table 1. Figure 12 illustrates the display format employed by the programmer to provide these data for user review.

The processing of each data record is completed by writing the site number, switch number and clock time onto an output disk corresponding to the site number. This process is repeated until the input data are exhausted.

Table 1

RDTAPE Control Parameters

Keyword	Dimension	Type	Default	Maximum	Purpose
RTLMT	1	Integer	3	3	Number of remote terminals
SITEID	30	Integer	1 for 1 — 10 2 for 11 — 20 3 for 21 — 30	5	Matches detector to appropriate site number
DELTAT	1	Integer	18	N.A.	Defines detector redundancy period

```
CONTROL PARAMETERS

$PARAM

RTLMT     =  3

DELTAT    = 18

SITE      =  1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

             2, 2, 2, 2, 2, 2, 2, 2, 2, 2,

             3, 3, 3, 3, 3, 3, 3, 3, 3, 3,

$END

0 INVALID SWITCH NUMBERS WERE ENCOUNTERED

0 INVALID REMOTE TERMINAL NUMBERS WERE ENCOUNTERED
```

Figure 12. Normal output for program "RDTAPE".

PROCESS Program

Program PROCESS converts a series of axle detector epochs read from a disk file created by RDTAPE into a characterization of a vehicle passing a point on the roadway. The system is designed to allow only one site to be processed at a time, where a site may contain up to four traffic lanes. This program performs four functions: it translates detector actuation into vehicle axle speeds, determines the number of axles and the distance between adjacent pairs, deduces the vehicle classification, and determines the vehicles' lateral distance from the edge of the pavement in tenths of feet.

The program reads four title cards and as many cards as required to contain the optional control parameters that describe the test conditions and control the processing being performed. The sequence and format of the title cards are shown in Table 2, while the control parameter keyword definitions are given in Table 3. The program displays these input values for user confirmation as shown in Figure 13. The program is designed with complete flexibility for matching physical switch configurations to the proper logical site and lane designations. The actual switch and remote box inputs used are noted at TDAS installation time on forms similar to that shown in Appendix C. Referring to the field forms, the program user determines the appropriate values for control parameters SITE, LANE, and SWTYPE. Each of these parameters has thirty storage locations corresponding to the thirty inputs of the TDAS hardware. The first ten positions, for example, correspond to the ten switch inputs of a remote box connected to position one of the master controller. Even though the installation used one remote box connected to the second master controller input (the program user would only be concerned with parameter positions 11 through 20) and monitored a single lane using switch inputs 8, 9, and 10, the software is capable of calling this site number one (site positions 18, 19, and 20 would be assigned a value of two), and the three switches would be assigned according to their function (SWTYPE for positions 18, 19, and 20 might be given the respective values of 1, 2, and 3).

Table 2

Heading Data Input Format for Program "PROCESS"

Data Name	Dimension	Format	No. of Cards	Default	Definition
Run Name	8	8A10	1	Required	General information for the run
Site Name	8	8A10	1	Required	Test site identification
Condition	8	8A10	1	Required	Condition of test site (weather, road surface)
Collection Date	8	8A10	1	Required	Date of collection of data

1778

Table 3

Control Parameter Data Input for Program "PROCESS"

Keyword	Dimension	Type	Default	Maximum	Definition
SITENO	1	Integer	1	3	Indicates site to be processed
PRCSLN	4	Logical	T,T,T,T	N.A.	Determines lanes to be processed
PRNTLN	4	Logical	T,T,T,T	N.A.	Determines lanes to be printed
LIST	1	Logical	F	N.A.	Listing of each vehicle
BEGLIST	1	Integer	000000	235959	Starting time of data collection in hours, minutes, seconds.
ENDLIST	1	Integer	000000	235959	Ending time of data collection in hours, minutes, seconds
SITE	30	Integer	first 10 are 1's second 10 are 2's third 10 are 3's	3	Assignment of detectors to logical site numbers
LANE	30	Integer	1,1,1,2,2,2,3,3,4,4 1,1,1,2,2,2,3,3,4,4 1,1,1,2,2,2,3,3,4,4	4	Location of each switch
SWTYPE	30	Integer	1,2,3,1,2,3,1,2,1,2 1,2,3,1,2,3,1,2,1,2 1,2,3,1,2,3,1,2,1,2	3	Switch type for each switch
STARTTM	1	Integer	000000	235959	Starting time of data collection in hours, minutes, seconds
DEBUG	1	Logical	F	N.A.	Detailed diagnostics used to identify software malfunctions
VHCLNO	1	Integer	0	No limit	Used to determine the starting point for DEBUG testing
METRIC	1	Logical	F	N.A.	Speed and lateral placement in metric units if true
SPDMIN	1	Real	10.0	No limit	Program terminates if vehicle speed is below SPDMIN or greater than SPDMAX
SPDMAX	1	Real	100.0	No limit	

HEADING INFORMATION

```
RUN NAME     - RURAL PAVEMENT MARKING STUDY  -  BEFORE DATA
SITE NAME    - ROUTE *220 IN ALLEGHANY COUNTY  -  LANE #1 NORTHBOUND
CONDITION    - CLEAR, HOT, SUNNY        95 DEGREES FARENHEIT
DATE         - JULY 17, 1979
```

CONTROL PARAMETERS

```
$PARAM

SITENO    =  1

STARTTM   =  153500

LIST      =  T     BEGLIST  =  60000
                   ENDLIST  = 240000

PRCSLN    =  T, F, F, F

PRNTLN    =  T. T, T, T

SWTYPE    =  1.  2,  3,  2,  1,  3,  1,  2,  1,  2,
             1,  2,  3,  1,  2,  3,  1,  2,  1,  2,
             1,  2,  3,  1,  2,  3,  1,  2,  1,  2,

LANE      =  1,  1.  1,  2,  2,  2,  3,  3,  4,  4,
             1,  1,  1,  2,  2,  2,  3.  3.  4,  4,
             1,  1,  1,  2,  2,  2,  3,  3,  4,  4,

SITE      =  1,  1.  1,  1,  1,  1,  1,  1.  1,  1,
             2.  2.  2,  2.  2,  2,  2,  2,  2,  2,
             3.  3.  3,  3,  3,  3,  3,  3,  3,  3.

DBUG      =  F     VHCLNO    =     0

METRIC    =  F

SPDMIN    =  10.0

SPDMAX    = 100.0

$END
```

Figure 13. Heading and control parameter output from program "PROCESS"

The program reads the edited field data prepared by program RDTAPE, identifies site and lane, determines whether the detector actuated is an entry, exit, or zone detector, and directs flow of control accordingly. For each entry detected, the program generates an axle entry in an event queue. Due to faulty detectors or close spacing between successive axles, two consecutive entry activations may occur without an intermediate exit activation. Similarly, two exit activations may occur without an intermediate entry activation occurring. The program handles these situations by continuing to process data until a clear definition of a vehicle occurs and then determines which data are erroneous or superfluous.

For each exit detected, the program adds an axle to the axle position and velocity tables corresponding to previous axle entry activation. Successive activations of the trap entry detector are used to calculate the wheelbase for the vehicle. If the wheelbase is found to be greater than the maximum allowable wheelbase (10.7 m [35 ft.]), the program then assumes a new vehicle has been detected and begins the process of relating groups of axles to specific vehicles to establish the characteristics of the previous vehicle. The vehicle is assigned to one of thirty-four recognizable classes or to one of six unrecognizable vehicle classes based on axle number and spacing. Then, the program calculates the lateral placement of the vehicle. (Lateral placement is defined as the distance from right edge of the lane to the right edge of the right wheel of the first axle of the vehicle.)

A summary of the vehicle characterization is optionally written to the printer, the disk, or both. This characterization includes the items below.

Site: The number of the sites through which the vehicle passed as defined by the parameter cards.

Lane: The number of the lane in which the vehicle was traveling as defined by the parameter cards.

Lateral Placement: The distance from right edge of the lane to the right wheel of the first axle or, for passing vehicles, the distance from left edge of the lane to the left wheel of the first axle of the vehicle measured in feet or metres.

Entry time: The clock time at which the vehicle enters the site.

Speed: An average of the individual axle speeds of the vehicle in miles per hour or kilometers per hour.

Back Headway: The time interval measured from the last axle of the front-going vehicle to the first axle of the present vehicle. This measurement is made at the entry detector of each lane of the site.

Front Headway: The time interval measured from the first axle of the front-going vehicle to the first axle of the present vehicle; measured at the entry detector.

Vehicle Type: Type of vehicle is determined from the number of axles and their wheelbase by referencing the vehicle configuration prototypes as shown in Appendix D.

The disk output format for these data is shown in Table 4 and the printer output form is illustrated in Figure 14. The summary listing, Figure 15, displays the total number of vehicles by type and lane.

Because of the absence of proper headway information for the first vehicle and because of program termination considerations for the last vehicle, the first vehicle and the last vehicle in each lane of each site are omitted from the reference disk file of vehicle characterizations.

Table 4

Vehicle Information File Output by Program "PROCESS"

Data Name	Description	Format
Site	Coded positive integer denoting the location of the trap	I1
Lane	Coded positive integer lane number counted from the shoulder of the road	I1
Lateral Placement	Distance in feet or metres from the edge of road to right front wheel	F4.1
Vehicle Type	Coded positive integer one of 41 possible vehicle types	I2
Entry Time	24 hr. clock time — hour: minute: sec.: tenths of second	F8.1
Vehicle Speed	Average of individual axle speeds through the trip in tenths of miles per hour or kilometers per hour	F4.1
Front Headway	Time between front axles of successive vehicles in tenths of seconds	F5.1
Back Headway	Time between back axle of front vehicle and front axle of following vehicle in tenths of seconds	F5.1

1782

RUN NAME - RURAL PAVEMENT MARKING STUDY - BEFORE DATA
SITE NAME - ROUTE #220 IN ALLEGHANY COUNTY - LANE #1 NORTHBOUND
CONDITION - CLEAR, HOT, SUNNY 95 DEGREES FARENHEIT
DATE - JULY 17, 1979
START TIME - 15:35:00

VEHICLE NUMBER	LOCATION SITE	LANE	ZONE(FT)	ENTRY TIME HR MIN SEC	SPEED MPH	HEADWAY(SEC) BACK	FRONT	TYPE OF VEHICLE	NO. OF AXLES	DISTANCE BETWEEN AXLES 1--2	2--3	3--4	4--5	5--6
1	1	1	4.7	15: 35: 27	45.5	0.00	0.00	CAR	2	8.5				
2	1	1	4.4	15: 36: 26	43.8	58.72	58.85	CAR	2	7.9				
3	1	1	3.9	15: 37: 48	39.2	81.94	82.07	CAR	2	9.6				
4	1	1	2.4	15: 37: 50	38.8	1.69	1.85	CAR	2	8.9				
5	1	1	2.2	15: 38: 27	37.1	36.43	36.59	CAR	2	9.5				
6	1	1	5.0	15: 40: 39	43.3	132.61	132.78	CAR	2	6.9				
7	1	1	3.9	15: 41: 03	47.4	23.05	23.16	CAR	2	10.3				
8	1	1	2.3	15: 45: 00	48.5	237.35	237.50	CAR	2	9.0				
9	1	1	6.6	15: 48: 36	-47.0	215.97	216.09	CAR	2	9.7				
10	1	1	3.2	15: 50: 19	33.7	102.94	103.08	CAR	2	10.4				
11	1	1	3.3	15: 51: 22	34.3	62.47	62.68	CAR	2	10.5				
12	1	1	3.9	15: 53: 06	38.3	103.87	104.08	CAR	2	10.0				
13	1	1	4.1	15: 53: 21	42.8	14.63	14.81	CAR	2	9.8				
14	1	1	3.6	15: 53: 38	38.3	17.25	17.39	PU OR VAN	2	11.0				
15	1	1	4.0	15: 54: 24	44.9	45.80	46.00	CAR	2	8.0				
16	1	1	3.3	15: 57: 13	36.4	169.00	169.13	CAR	2	8.6				
17	1	1	2.8	15: 57: 16	32.5	2.49	2.65	CAR	2	7.7				
18	1	1	4.3	15: 58: 22	38.7	65.72	65.88	CAR	2	9.9				
19	1	1	2.3	15: 58: 23	37.5	1.39	1.57	CAR	2	8.8				
20	1	1	3.3	15: 58: 25	42.8	1.75	1.91	CAR	2	10.3				
21	1	1	4.-	15: 59: 04	44.3	38.32	38.48	PU OR VAN	2	11.2				
22	1	1	3.5	16: 00: 14	39.0	69.91	70.08	CAR	2	9.6				
23	1	1	3.8	16: 00: 16	35.0	2.08	2.25	PU OR VAN	2	11.0				
24	1	1	3.2	16: 01: 08	39.7	51.32	51.53	CAR	2	8.0				
25	1	1	3.9	16: 01: 56	38.5	48.38	48.52	CAR	2	10.6				
26	1	1	2.8	16: 02: 42	45.1	45.69	45.87	CAR	2	10.3				
27	1	1	3.6	16: 04: 44	43.3	121.38	121.53	CAR	2	7.7				
28	1	1	3.9	16: 06: 06	39.8	82.74	82.86	CAR	2	7.5				
29	1	1	4.8	16: 06: 41	47.0	34.46	34.59	CAR	2	9.6				
30	1	1	3.5	16: 07: 42	42.3	61.30	61.43	CAR	2	10.2				
31	1	1	4.0	16: 07: 44	42.3	1.08	1.25	CAR	2	10.0				
32	1	1	3.1	16: 07: 45	40.9	1.22	1.38	TRUCK	2	13.8				
33	1	1	4.4	16: 08: 55	34.1	69.24	69.47	CAR	2	9.9				
34	1	1	3.1	16: 10: 31	42.6	96.72	96.92	CAR	2	7.9				
35	1	1	5.8	16: 13: 59	50.3	207.46	207.59	CAR	2	7.9				
36	1	1	2.0	16: 15: 16	37.9	76.76	76.87	CAR	2	9.2				
37	1	1	3.8	16: 25: 09	44.2	593.12	593.28	CAR	2	9.2				
38	1	1	2.6	16: 25: 22	41.5	12.73	12.88	CAR	2	9.2				
39	1	1	3.2	16: 25: 29	46.4	7.01	7.16	CAR	2	9.8				
40	1	1	2.9	16: 28: 25	40.1	175.89	176.03	CAR	2	7.6				
41	1	1	2.2	16: 30: 18	26.4	112.81	112.94	T.T. 5X T2	5	11.6	4.2	20.8	4.2	
42	1	1	3.0	16: 30: 23	23.0	3.61	4.66	ST TRK 2AX	2	21.4				
43	1	1	3.7	16: 32: 18	49.8	114.73	115.36	CAR	2	8.6				
44	1	1	4.1	16: 33: 54	40.7	95.56	95.68	CAR	2	9.9				
45	1	1	2.7	16: 34: 30	37.2	35.72	35.88	CAR	2	8.3				
46	1	1	4.0	16: 34: 32	41.0	1.98	2.13	CAR	2	9.9				
47	1	1	2.7	16: 38: 00	39.5	207.99	208.15	CAR	2	10.4				
48	1	1	3.6	16: 38: 30	22.4	29.48	29.66	CAR	2	7.7				
49	1	1	2.7	16: 39: 01	40.7	31.18	31.42	CAR-TRLR2	4	9.6	15.6	2.4		
50	1	1	4.4	16: 39: 41	50.5	39.44	39.90	CAR	2	9.6				

Figure 14. Optional output from program "PROCESS".

RUN NAME	- RURAL PAVEMENT MARKING STUDY - BEFORE DATA
SITE NAME	- ROUTE #220 IN ALLEGHANY COUNTY - LANE #1 NORTHBOUND
CONDITION	- CLEAR, HOT, SUNNY 95 DEGREES FARENHEIT
DATE	- JULY 17, 1979
START TIME	- 15:35:00

SUMMARY(VEHICLE TYPE BY LANE)
****

VEHICLE_TYPE	LANE1	LANE2	LANE3	LANE4	TOTAL
MOTORCYCLE	0	0	0	0	0
MC & TRLR	0	0	0	0	0
CAR	93	0	0	0	93
CAR-TRLR1	0	0	0	0	0
CAR-TRLR2	1	0	0	0	1
PU OR VAN	7	0	0	0	7
PU.VAN-TL1	0	0	0	0	0
PU.VAN-TL2	0	0	0	0	0
TRUCK	4	0	0	0	4
TRUCK 3 AX	0	0	0	0	0
TRUCK 4 AX	0	0	0	0	0
TRK 2AX T1	0	0	0	0	0
TRK 2AX T2	0	0	0	0	0
TRK 2AX T3	0	0	0	0	0
TRK 2AX T4	0	0	0	0	0
TRK 3AX T1	0	0	0	0	0
TRK 3AX T2	0	0	0	0	0
TRK 3AX T3	0	0	0	0	0
TRK 4AX T1	0	0	0	0	0
TRK 4AX T2	0	0	0	0	0

VEHICLE_TYPE	LANE1	LANE2	LANE3	LANE4	TOTAL
TRK 4AX T3	0	0	0	0	0
ST TRK 2AX	1	0	0	0	1
ST TRK 3AX	0	0	0	0	0
STRK 2X T1	0	0	0	0	0
STRK 3X T1	0	0	0	0	0
T.T. 3X T1	0	0	0	0	0
T.T. 4X T2	0	0	0	0	0
T.T. 4X T1	0	0	0	0	0
T.T. 5X T2	3	0	0	0	3
T.T. 6X T3	0	0	0	0	0
HOUSE TL3	0	0	0	0	0
HOUSE TL4	0	0	0	0	0
BUS 2 AX	0	0	0	0	0
BUS 3 AX	0	0	0	0	0
UNKNWN 2AX	0	0	0	0	0
UNKNWN 3AX	0	0	0	0	0
UNKNWN 4AX	0	0	0	0	0
UNKNWN 5AX	0	0	0	0	0
UNKNWN 6AX	0	0	0	0	0
UNKNWN 7AX	0	0	0	0	0
TOTAL	109	0	0	0	109

Figure 15. Summary output from program "PROCESS".

1783

CORECT Program

Program CORECT provides the TDAS system user with the tools necessary to modify the disk file created by RDTAPE. This correction routine is most frequently used to solve problems related to missing detector activations due to vehicles changing lanes while passing through the axle detector area. Whenever program PROCESS loses track of the proper matching of detector activations (manifested as unusually high or low axle velocities or an axle queue overflow), the program prints the reason processing was terminated and displays the present contents of the detector activation queue as shown in Figure 16. Careful evaluation of the queue contents is usually adequate to determine if extra detector activations should be deleted or if a detector activation needs to be inserted. The formats for the correction commands are shown in Table 5.

The error output shown in Figure 16 can be analyzed by first noting that the last usable switch time was 2150387. This value is located in the third column in the row marked Queue Position 2. This informs the user that all event times equal to or less than this value have been successfully processed. It can be further assumed that the event time in column 5 of this same row is also valid. This fact is easily verified by calculating the time difference between queue positions 1 and 2 for both column 3 and column 5 and between column 3 and column 5. This procedure should be continued in increasing event time order until a discrepancy is detected. This activity is illustrated in Figure 17, where the absence of a matching event time in column 5 for queue position 4 has been observed. An insertion card is prepared and the correction program is run resulting in the output of Figure 18 and the creation of a corrected data file. Program PROCESS is run using the corrected file as data; if an error dump reoccurs the correction process is repeated with the original file as input to program CORECT and with the output replacing the old corrected file.

```
*** FATAL ERROR *** -- QUEUE INSERTION OVERFLOW ENCOUNTERED AFTER VEHICLE NUMBER   59

                    ENTRY TIME FOR LAST IDENTIFIABLE VEHICLE IS 16:46:41

                          QUEUE DUMP OF SWITCH TIMES FOR LANE 1

   PASS = F     QUEUE POINTER = 36     LAST USEABLE SWITCH TIME =   2150387
```

QUEUE POSITION	SWITCH TYPE 1		SWITCH TYPE 2	
34	1	2700167	2	2700428
35	1	2700273	2	2128422
36	1	2726715	2	2128516
37	1	2130998	2	2131138
38	1	2131067	2	2131208
39	1	2146114	2	2146252
40	1	2146209	2	2146347
1	1	2150316	2	2150442
2	1	2150387	2	2150513
3	1	2168566	2	2168693
4	1	2168638	2	2210755
5	1	2210613	2	2210849
6	1	2210706	2	2231736
7	1	2231611	2	2231810
8	1	2231686	2	2343427
9	1	2343299	2	2343501
10	1	2343373	2	2387586
11	1	2387461	2	2387669
12	1	2387542	2	2398288
13	1	2398156	2	2398363
14	1	2398232	2	2425488
15	1	2425370	2	2425557
16	1	2425438	2	2464824
17	1	2464706	2	2464891
18	1	2464773	2	2567148
19	1	2566898	2	2567307
20	1	2567055	2	2567375
21	1	2567122	2	2567673
22	1	2567417	2	2567741
23	1	2567482	2	2640151
24	1	2640030	2	2640230
25	1	2640108	2	2641706
26	1	2641578	2	2641783
27	1	2641655	2	2649220
28	1	2649097	2	2649296
29	1	2649171	2	2660359
30	1	2660229	2	2660428
31	1	2660298	2	2695101
32	1	2694973	2	2695189
33	1	2695061	2	2700323

Figure 16. Error output from program PROCESS.

1786

Table 5

Command Format for Program CORECT

Command	Entry	Columns	
Delete	Switch no.	1-2	right justified
	Event time	3-20	right justified
Insert	99	1-2	
	Event time	3-20	right justified
	Switch no.	31-32	right justified

*** FATAL ERROR *** -- QUEUE INSERTION OVERFLOW ENCOUNTERED AFTER VEHICLE NUMBER 59

ENTRY TIME FOR LAST IDENTIFIABLE VEHICLE IS 16:46:41

QUEUE DUMP OF SWITCH TIMES FOR LANE 1

PASS = F   QUEUE POINTER = 36   LAST USEABLE SWITCH TIME = 2150387

QUEUE POSITION	SWITCH TYPE 1		SWITCH TYPE 2	
34	1	2700167	2	2700428
35	1	2700273	2	2128422
36	1	2726715	2	2128516
37	1	2130998	2	2131138
38	1	2131067	2	2131208
39	1	2146114	2	2146252
40	1	2146209	2	2146347
1	1	2150316	2 126	2150442
2	1	2150387 71	2	2150513 71
3	1	2168566	2 127	2168693
4	1	2168638 72	2 2168765	2210755
5	1	2210613	2	2210849
6	1	2210706	2	2231736
7	1	2231611	2	2231810
8	1	2231686	2	2343427
9	1	2343299	2	2343501
10	1	2343373	2	2387586
11	1	2387461	2	2387669
12	1	2387542	2	2398288
13	1	2398156	2	2398363
14	1	2398232	2	2425488
15	1	2425370	2	2425557
16	1	2425438	2	2464824
17	1	2464706	2	2464891
18	1	2464773	2	2567148
19	1	2566898	2	2567307
20	1	2567055	2	2567375
21	1	2567122	2	2567673
22	1	2567417	2	2567741
23	1	2567482	2	2640151
24	1	2640030	2	2640230
25	1	2640108	2	2641706
26	1	2641578	2	2641783
27	1	2641655	2	2649220
28	1	2649097	2	2649296
29	1	2649171	2	2660359
30	1	2660229	2	2660428
31	1	2660298	2	2695101
32	1	2694973	2	2695189
33	1	2695061	2	2700323

Figure 17. Analysis of error output from program PROCESS.

Figure 18. Output from program CORECT.

REPORT Program

Program REPORT reads and analyzes the disk file generated by program PROCESS. It provides the statistical analysis and comprehensive display of the most popular traffic flow parameters.[5] The program reads the header records from the disk file prepared by program PROCESS and the control parameters from the card reader. The sequence and format of the header and control data to be read from the card reader are in Table 6. The program displays the header information and control parameters for user verification as shown in Figure 19. The program reads the vehicle data stored on disk file by program PROCESS (Table 4) and adds each data item onto the appropriate table for subsequent analysis.

When the program detects the end of data, it constructs the following traffic information tables.

Traffic Volume Table (Figure 20): This table indicates traffic volume counts and percentage volume by lane and zone for each vehicle type and for the sum of all vehicle types. Vehicles are classed into cars, car-trailer combinations, trucks, tractor-trailers, and others.

Vehicle Speed Table (Figure 21): This table indicates the average, standard deviation, and 85th percentile for the speeds of each vehicle type by lane and zone.

Headway Information Tables (Figure 22): The first table indicates the average, standard deviation, and the median of the headway distribution of each lane. The second table indicates the headway distribution by time intervals in seconds. The last time category in this table includes all headways greater than 59 seconds.

Queue Information Tables (Figure 23): The first table indicates the number of queues encountered, the average and standard deviation of the number of vehicles in the queues, and the average and standard deviation of the queue speed distribution. The second table shows the frequency of vehicle type in each of the first five queue positions.

Table 6

Input Data Format for Program REPORT

Heading Information

Data Name	Dimension	Format	No. of Cards	Default	Definition
Run Name	8	8A10	1	Required	General information for the run

Control Parameters

Keyword	Dimension	Type	Default	Maximum	Definition
STARTTM	1	Integer	000000	235959	Starting time of analysis in 24-hour clock time
ENDTM	1	Integer	235959	235959	Ending time of analysis in 24-hour clock time
QCUTOFF	5	Real	6.0		5 different queue cutoff times
METRIC	1	Logical	F	N.A.	Speed and lateral placement values presented in metric units
ZNSIZE*	1	Real	N.A.	ZNWIDTH	Size of each zone Segment in in. or cm
ZNWIDTH*	1	Real	N.A.	No limit	Width of zone lane is a function of detector length in in. or cm
BKHDWY	1	Logical	F	N.A.	Determines if front headway or back headway is to be used for platoon calculations

*NOTE: These values should be provided in metric units if process was run with the control parameter keyword METRIC set to TRUE.

1790

HEADING INFORMATION

```
RUN NAME  - RURAL PAVEMENT MARKING STUDY  -  BEFORE DATA
SITE NAME - ROUTE #220 IN ALLEGHANY COUNTY  -  LANE #1 NORTHBOUND
CONDITION - CLEAR, HOT, SUNNY        95 DEGREES FARENHEIT
DATE      - JULY 17, 1979
```

CONTROL PARAMETERS

```
$PARAM
STARTTM   = 153500
ENDTM     = 181501
QCUTOFF   =   6.0,  0.0,  0.0,  0.0,  0.0,
METRIC    = F
ZNSIZE    =   8.0
ZNWIDTH   =  84.0
BKHDWY    = F
FRSTLN    =  1      LASTLN    =  1
$END
```

Figure 19. Parameter output from program REPORT.

```
RUN NAME          - RURAL PAVEMENT MARKING STUDY  -  BEFORE DATA
SITE NAME         - ROUTE #220 IN ALLEGHANY COUNTY  -  LANE #1 NORTHBOUND
CONDITION         - CLEAR, HOT, SUNNY        95 DEGREES FARENHEIT
DATE              - JULY 17, 1979
PERIOD ANALYZED   - 15:35:00   TO   18:15:01
ZONE SIZE         -   8.0 INCHES
ZONE LANE WIDTH   -  84.0 INCHES
```

T R A F F I C   V O L U M E

LANE	ZONE	CARS VOLUME	CARS PERCENT	CAR-TRAILERS VOLUME	CAR-TRAILERS PERCENT	TRUCKS VOLUME	TRUCKS PERCENT	TRACTOR-TRAILERS VOLUME	TRACTOR-TRAILERS PERCENT	OTHERS VOLUME	OTHERS PERCENT	ALL VEHICLES VOLUME	ALL VEHICLES PERCENT
1		100	91.7	1	.9	5	4.6	3	2.8	0	0.0	109	100.0
	1	1	1.0	0	0.0	0	0.0	0	0.0	0	0.0	1	.9
	2	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
	3	3	3.0	0	0.0	1	20.0	1	33.3	0	0.0	5	4.6
	4	12	12.0	0	0.0	0	0.0	2	66.7	0	0.0	14	12.8
	5	25	25.0	1	100.0	2	40.0	0	0.0	0	0.0	28	25.7
	6	28	28.0	0	0.0	0	0.0	0	0.0	0	0.0	28	25.7
	7	21	21.0	0	0.0	2	40.0	0	0.0	0	0.0	23	21.1
	8	6	6.0	0	0.0	0	0.0	0	0.0	0	0.0	6	5.5
	9	4	4.0	0	0.0	0	0.0	0	0.0	0	0.0	4	3.7
	10	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0

1792

RUN NAME - RURAL PAVEMENT MARKING STUDY - BEFORE DATA
SITE NAME - ROUTE #220 IN ALLEGHANY COUNTY - LANE #1 NORTHBOUND
CONDITION - CLEAR, HOT, SUNNY 95 DEGREES FARENHEIT
DATE - JULY 17, 1979
PERIOD ANALYZED - 15:35:00 TO 18:15:01
ZONE SIZE - 8.0 INCHES
ZONE LANE WIDTH - 84.0 INCHES

V E H I C L E S P E E D
(MPH)

		CARS			CAR-TRAILERS			TRUCKS			TRACTOR-TRAILERS			OTHERS			ALL VEHICLES		
LANE	ZONE	AVG.	S.D.	85%	AVG.	S.D.	85%	AVG.	S.D.	85%	AVG.	S.D.	85%	AVG.	S.D.	85%	AVG.	S.D.	85%
1		40.3	5.6	46	40.7	0.0	0	36.1	7.8	43	28.8	8.7	39	0.0	0.0	0	39.8	6.1	46
	1	38.3	0.0	0	0.0	0.0	0	0.0	0.0	0	0.0	0.0	0	0.0	0.0	0	38.3	0.0	0
	2	0.0	0.0	0	0.0	0.0	0	0.0	0.0	0	0.0	0.0	0	0.0	0.0	0	0.0	0.0	0
	3	34.6	12.3	43	0.0	0.0	0	37.5	0.0	0	38.4	0.0	0	0.0	0.0	0	35.9	8.9	43
	4	39.4	3.5	43	0.0	0.0	0	0.0	0.0	0	24.0	3.5	27	0.0	0.0	0	37.2	6.5	42
	5	39.6	5.1	47	40.7	0.0	0	32.0	12.7	41	0.0	0.0	0	0.0	0.0	0	39.1	5.7	46
	6	41.0	6.2	47	0.0	0.0	0	0.0	0.0	0	0.0	0.0	0	0.0	0.0	0	41.0	6.2	47
	7	39.8	5.6	45	0.0	0.0	0	39.5	4.9	43	0.0	0.0	0	0.0	0.0	0	39.8	5.4	45
	8	43.7	3.4	48	0.0	0.0	0	0.0	0.0	0	0.0	0.0	0	0.0	0.0	0	43.7	3.4	48
	9	45.0	3.6	51	0.0	0.0	0	0.0	0.0	0	0.0	0.0	0	0.0	0.0	0	45.0	3.6	51
	10	0.0	0.0	0	0.0	0.0	0	0.0	0.0	0	0.0	0.0	0	0.0	0.0	0	0.0	0.0	0

RUN NAME - RURAL PAVEMENT MARKING STUDY - BEFORE DATA
SITE NAME - ROUTE #220 IN ALLEGHANY COUNTY - LANE #1 NORTHBOUND
CONDITION - CLEAR, HOT, SUNNY 95 DEGREES FARENHEIT
DATE - JULY 17, 1979
PERIOD ANALYZED - 15:35:00 TO 18:15:01
ZONE SIZE - 8.0 INCHES
ZONE LANE WIDTH - 84.0 INCHES

H E A D W A Y   I N F O R M A T I O N

HEADWAY CUTOFF TIME 60.0 SECONDS USING FRONT HEADWAY DATA

		H E A D W A Y S		
LANE	VOLUME	AVG.	S.D.	MEDIAN
1	109	68.2	78.2	49.0

DISTRIBUTION OF HEADWAYS BY TIME INTERVALS IN SECONDS

TIME INTERVAL	VEHICLE NUMBERS	TIME INTERVAL	VEHICLE NUMBERS	TIME INTERVAL	VEHICLE NUMBERS	TIME INTERVAL	VEHICLE NUMBERS
0 - 1	0	15 - 16	1	30 - 31	1	45 - 46	1
1 - 2	9	16 - 17	1	31 - 32	1	46 - 47	1
2 - 3	5	17 - 18	2	32 - 33	1	47 - 48	0
3 - 4	2	18 - 19	1	33 - 34	0	48 - 49	1
4 - 5	2	19 - 20	0	34 - 35	1	49 - 50	0
5 - 6	1	20 - 21	0	35 - 36	1	50 - 51	0
6 - 7	0	21 - 22	1	36 - 37	2	51 - 52	1
7 - 8	1	22 - 23	2	37 - 38	0	52 - 53	1
8 - 9	1	23 - 24	2	38 - 39	3	53 - 54	1
9 - 10	0	24 - 25	0	39 - 40	1	54 - 55	1
10 - 11	2	25 - 26	1	40 - 41	0	55 - 56	0
11 - 12	1	26 - 27	0	41 - 42	0	56 - 57	2
12 - 13	1	27 - 28	0	42 - 43	1	57 - 58	0
13 - 14	0	28 - 29	0	43 - 44	1	58 - 59	1
14 - 15	1	29 - 30	1	44 - 45	1	59 - 60	47

1793

1794

RUN NAME - RURAL PAVEMENT MARKING STUDY - BEFORE DATA
SITE NAME - ROUTE #220 IN ALLEGHANY COUNTY - LANE #1 NORTHBOUND
CONDITION - CLEAR, HOT, SUNNY 95 DEGREES FARENHEIT
DATE - JULY 17, 1979
PERIOD ANALYZED - 15:35:00 TO 18:15:01
ZONE SIZE - 8.0 INCHES
ZONE LANE WIDTH - 84.0 INCHES

Q U E U E   I N F O R M A T I O N

QUEUE CUTOFF TIME 6.0 SECONDS

LANE	NO. OF QUEUES	QUEUE LENGTH AVG.	QUEUE LENGTH S.D.	QUEUE SPEED AVG.	QUEUE SPEED S.D.
1	16	2.2	.4	37.3	5.9

LANE	POSITION	CAR	CAR-TRL	TRUCK	TRACTOR TRAILER	OTHERS
1						
	1	15	0	0	1	0
	2	14	0	2	0	0
	3	1	0	2	0	0
	4	0	0	0	0	0
	5	0	0	0	0	0

ERROR ANALYSIS

The purpose of this section of the report is to show the extent to which elements of the traffic data acquisition system influence the accuracy of the system. Each element of the system is examined while the remainder of the system is held constant at its nominal values. The system assumes a tape switch configuration as shown in Figure 24.

Switch Placement

Speed switches are to be placed perpendicular to the edge of the roadway 4.88 m ± 5 cm (16 ft. ± 2 in.) apart. Assuming a 5-cm (2-in.) error, there would be an error of 1% in speed. Similarly, a 5-cm (2-in.) error in parallelism would induce a variable error (a function of the lateral position of the vehicle) with a maximum error of approximately 1%.

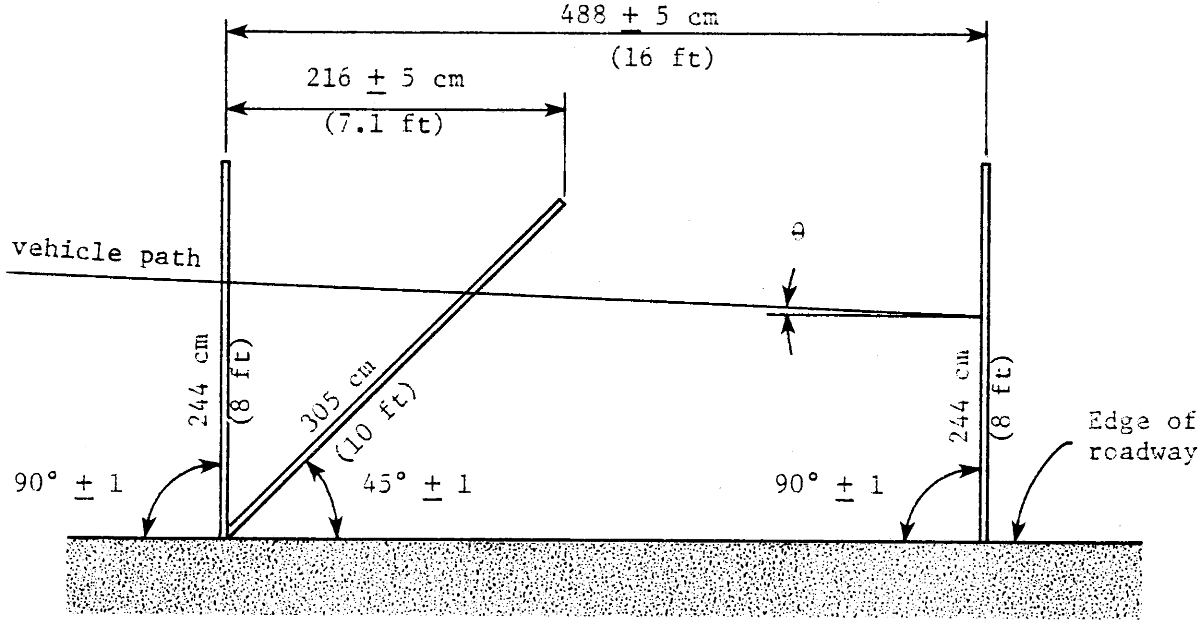


Figure 24. Switch placement tolerances.

The lateral placement switch is positioned from the point where the first speed switch intersects the edge of the roadway to a point that is equal distance from the edge of the roadway and the first speed switch. The distance used to locate this second point is a function of the length of the switch (i.e., the switch is the hypotenuse of a right-isosceles triangle). For a 3.05-m (10-ft.) switch, this distance would be 2.16 m (7.07 ft.). Assuming this point can be located within 5 cm (2 in.), the maximum error would be experienced at a lateral placement of 2.13 m (7 ft.), where the distance to the reference switch is a minimum of 2.74 m (9 ft.), and would be approximately 1.9%.

Polling Error

The switches are examined once every 0.002 second for evidence of previous activation; a switch event can be recorded only on time or up to 0.002 second late.

The magnitude of the vehicle speed error is a function of the velocity of the vehicle. For example, a vehicle traveling at 97 km/h (60 mph) that strikes the second speed switch immediately after it is polled will result in a speed registration that is 1.1% slower than actual. The lateral placement error for this vehicle will vary from 1.1% at the edge of the pavement to 2.0% at the 2.13-m (7-ft.) position.

The system error is best determined by simulating switch event times that represent extreme conditions and examining program results using these data. Table 7 shows the results of program execution using data representing the following three conditions: nominal values of switch placement and polling times; used for reference purposes. Error case 1; with the second switch placed 4.93 m (16 ft., 2 in.) from the first switch and the second switch is polled one count late. Error case 2; with the second switch placed 4.83 m (15 ft.,10 in.) from the first switch and the first switch is polled one count late.

Speed errors are in every case less than 2%, and although distance-related errors for high velocity vehicles in the 30.5 cm (1 ft.) lateral placement range are as high as 20%, none represent deviations of greater than 6 cm (2.5 in.).

Table 7

Error Analysis

Target Values	Program Results					
	Nominal		Case 1		Case 2	
	m	(ft.)	m	(ft.)	m	(ft.)
Vehicle speed — 32.19 km/h (20 mph)	32.15	(19.98)	31.75	(19.73)	32.57	(20.24)
Wheelbase — 3.048 m (10.0 ft.)	3.04	(9.96)	2.99	(9.82)	3.06	(10.05)
Lateral placement — .305 m (1.0 ft.)	.30	(1.00)	.30	(0.98)	.30	(0.98)
1.219 m (4.0 ft.)	1.21	(3.99)	1.20	(3.93)	1.22	(4.01)
2.134 m (7.0 ft.)	2.13	(6.97)	2.10	(6.87)	2.14	(7.04)
Vehicle speed — 80.47 km/h (50 mph)	80.53	(50.04)	79.08	(49.14)	81.66	(50.74)
Wheelbase — 3.048 m (10.0 ft.)	3.04	(9.98)	2.99	(9.80)	3.05	(10.02)
Lateral placement — .305 m (1.0 ft.)	.31	(1.03)	.31	(1.01)	.29	(0.97)
1.219 m (4.0 ft.)	1.21	(3.96)	1.19	(3.89)	1.20	(3.94)
2.134 m (7.0 ft.)	2.15	(7.05)	2.11	(6.92)	2.15	(7.07)
Vehicle speed — 128.75 km/h (80 mph)	129.09	(80.21)	126.32	(78.49)	131.05	(81.43)
Wheelbase — 3.048 m (10.0 ft.)	3.08	(10.12)	3.00	(9.83)	3.10	(10.18)
Lateral placement — .305 m (1.0 ft.)	.29	(0.94)	.25	(0.80)	.29	(0.95)
1.219 m (4.0 ft.)	1.22	(4.00)	1.16	(3.80)	1.24	(4.06)
2.134 m (7.0 ft.)	2.15	(7.06)	2.07	(6.79)	2.18	(7.16)

1797

1798

CONCLUSIONS AND RECOMMENDATIONS

The traffic data and acquisition system is a blend of two very powerful technologies and provides the traffic researcher with a tool that can produce timely answers to complex traffic flow questions heretofore requiring several man-months of manual data reduction.

Even though this system represents a major step toward simplified traffic data gathering, because of the rate of technological advance in the electronics field, it is clear that the hardware portion of this system can be greatly improved. A state-of-the-art implementation would embody a microprocessor for data handling, display, and self-check functions; a compact, low power consumption, wide temperature range cassette tape recorder for data storage; a sealed, rechargeable power source capable of 24 hours of data collection; and a watertight, compact packaging container approximately the size of a large briefcase. These components are available now and are relatively inexpensive (the total parts cost should be about one-half the cost of the presently used tape recorder alone). The present system will continue to serve a valuable function while a new generation of hardware is assembled; the software, however, will be as valid for the new generation of hardware as it is for the present system.

ACKNOWLEDGMENTS

The author expresses appreciation to all of those who contributed to the evolution of the TDAS system. Special thanks go to Dr. James Aylor and Wesley McDonald, who engineered the recent design changes and did much to increase the reliability of the TDAS hardware. Thanks go to Woon-Ho Song for his programming assistance early in the development of the TDAS software.

For assistance in the preparation of this document, thanks go to Jan Kennedy for her tireless typing of the draft version; Jennifer Ward for her help in preparing the flowcharts; and the report section personnel, Harry Craft, Allen Baker, Jean Vanderberry, and Jerry Garrison, for their respective roles of editing, graphics preparation, typing, and duplication of the final report.

1800

REFERENCES

1. Korf, J. L. and F. D. Shepard, "Computerized Traffic Data Analysis System", VHTRC 76-R21, Virginia Highway & Transportation Research Council, November 1975.

2. Taragin, Asriel, and R. C. Hopkins, "A Traffic Analyzer: Its Development and Application", Public Roads, Volume 31, No. 5, December 1960.

3. Ronbech, Jens, "Digital Magnetic Tape Recorder for Road Traffic Analysis", Traffic Engineering & Control, November 1969.

4. Lenz, K. H. and Ruediger Hotop, "Fully-Automated Acquisition and Processing of Traffic Data", Traffic Engineering and Control, August/September 1974.

5. Drew, Donald R., Traffic Flow Theory and Control, McGraw-Hill, 1968.

1802

APPENDIX A

1803

INSTALLATION PROCEDURES FOR TDAS HARDWARE

A. Preliminaries

1. The system requires two cables, a control cable and a data cable, from the central processor (CP) to each remote station (RS). These should be carefully distinguished at both ends. Each cable must be fitted at both ends with a 3-conductor jack. The jack conductors are designated as shown in Figure A-1. Using 18 AWG, 2-conductor, shielded wire such as Belden #8760, connect the black lead to #3, connect the white lead to #2, and connect the shield to #1. After so connecting both ends of a cable, assure that there is continuity (< 20 ohms) between corresponding connector parts. Next assure than an open circuit (> 10 megaohms) exists between 1-2, 2-3, and 1-3. Improper connection will cause an equipment malfunction, and potentially result in damage.

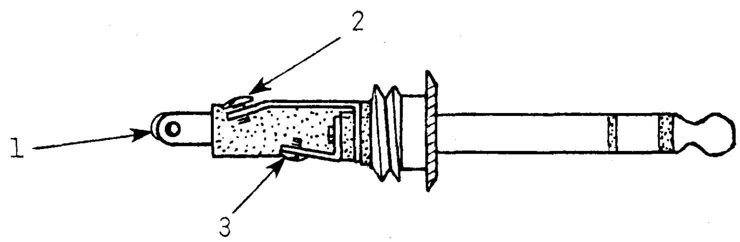


Figure A-1. Control cable and data cable connector.

2. Install RS electronic boxes into wooden containers (Figure A-2). With master switches off, wire the output terminal of the outer switch to the + (red) input of the RS box. Wire the negative battery terminal to the - (black) input of the RS box.

   <u>It is most important to do this correctly, as the electronics are not protected against reverse voltage</u>.

B. On-Site Installation

1. Install tape switches in each lane of interest in accordance with Figure 24. Note: Switches are fastened to the pavement with an underlayer of double sided tape and an overlayer of duct tape.

2. Check that power switch of CP is "off".

3. For each remote station to be used, connect a pair of cables to the "in" and "out" ports of a channel of the CP. At the RS, insert the cable that is "in" at the CP to the port designated "out" at the RS and vice versa. Only after both cables are fully inserted at both ends may power be applied to the RS. (Tape switch inputs may be inserted or removed at liberty, whether or not RS is powered.) Using a voltmeter, check that battery voltage is greater than 12 volts with the power switch "ON". Note: It is good practice to charge the batteries before each usage.

4. Make any final adjustments of tape switch input configurations. Note: Make diagrams of the tape switch configurations and the ports used at every RS site, including the location of the attached switches. It will be helpful when analyzing the data.

5. Activate those CP channels connected to RS's by throwing the corresponding channel switches in the "up" position. Channel switches of unused channels should be "down".

6. After the RS's to be used have been readied as described in (3), position power switch of CP to "ON" and press the "CLEAR" button.

7. Turn power on at the battery box (Figure A-3) and ready the magnetic tape unit (MTU) to receive data by mounting a reel of magnetic tape, setting the MTU to "RECORD", and pressing the BOT button to advance the tape to the load point.

8. To begin data acquisition, press "START" on CP.

9. To terminate acquisition, press "EOF" on CP. Acquisition may be reinitiated without loss of data already written by pressing "START". Alternatively, data may be overwritten by rewinding tape to load point before proceeding. (Note: Momentarily switching to "REWIND" and returning to "RECORD" prevents the tape from being unloaded.)

C. De-installation

1. Power down CP. Note: Do not remove CP cables until the respective RS has been powered down.

2. For each RS, power down and remove cables. Tape switches can be removed at any time.

1806

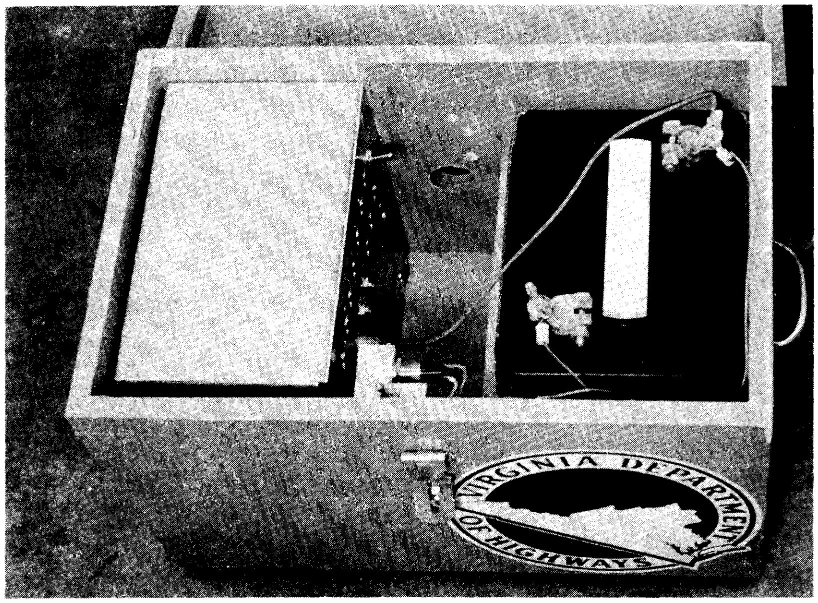


Figure A-2. Typical remote station with power source.

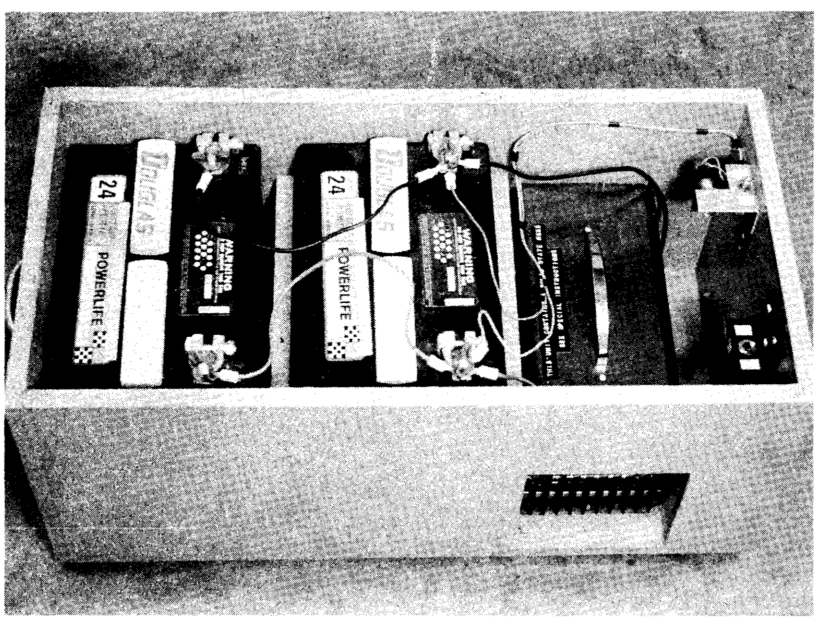


Figure A-3. Power supply for tape recorder and central processor.

APPENDIX B

1807

PROGRAM LISTINGS

1808

```
      PROGRAM RDTAPE(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
C
C    FIELD DATA TAPE                  EDITED OUTPUT DATA FILES (5 MAXIMUM)
C
     -   TAPE1,                       TAPE11,TAPE12,TAPE13,TAPE14,TAPE15)
C
C
C**********************************************************************
C
C                TRAFFIC DATA PROGRAM TO READ FIELD DATA TAPE
C
C                              WRITTEN FOR
C
C             THE VIRGINIA HIGHWAY AND TRANSPORTATION RESEARCH COUNCIL
C
C                                  BY
C                             JERRY L. KORF
C                           RESEARCH SCIENTIST
C                                9/1/76
C
C                      REVISED 11/15/77 BY J. L. KORF
C
C**********************************************************************
C
C
C        THE PURPOSE OF THIS PROGRAM IS TO READ A 7-TRACK MAGNETIC TAPE
C    CREATED BY THE TRAFFIC DATA ACQUISITION SYSTEM HARDWARE AND TO EDIT
C    THE DATA AS FOLLOWS
C
C        1)  DECODES SWITCH IDENTIFICATION DATA INTO REMOTE TERMINAL
C            NUMBER AND SWITCH NUMBER THEN VALIDATES EACH.
C
C        2)  CONVERTS VALID SWITCH IDENTIFICATION DATA INTO A COMBINATION
C            OF SITE NUMBER AND SWITCH NUMBER ACCORDING TO THE PARAMETERS
C            PROVIDED BY THE USER OR ACCORDING TO THE DEFAULT VALUES.
C
C        3)  ELIMINATES DUPLICATE EVENTS (WITHIN DELTAT * .002 SECONDS OF
C            A PREVIOUS SIMILAR EVENT) DUE TO CONTACT BOUNCE.
C
C        4)  CONVERTS THE CYCLIC EPOCH VALUES INTO A MONOTONIC INCREASING
C            FUNCTION.
C
C        5)  CREATES UP TO FIVE OUTPUT DATA FILES (TAPE11 TO TAPE15)
C            CORRESPONDING TO SITE NUMBERS ONE THROUGH FIVE.
C
C        THE PROGRAM IS LIMITED TO FIVE POSSIBLE SITES WITH A MAXIMUM OF
C    FOUR LANES PER SITE. THE HARDWARE IS LIMITED TO THREE REMOTE TERM-
C    INALS WITH TEN SWITCHES PER TERMINAL.
C
C
      DIMENSION  INBUF(24)
```

1809

```
C
      INTEGER    SITE(30), NOSWCH(60), TIMEPK(60), SVTIME(30),
     -           SWTCHN(30), OUTFILE(5),
     -           SWN, TIMER, PREVTM, TEMPTM, OUTIME, DELTAT,
     -           READER, PRINTER, INTAPE, DISKFL, RTLMT
C
C*********************************************************************
C
C          THE FOLLOWING ARE THE USER CONTROLLED PARAMETERS WITH THEIR
C     DEFAULT VALUES. "RTLMT" SETS A LIMIT ON THE NUMBER OF REMOTE
C     TERMINALS USED WHILE "SITEID" ALLOWS THE USER TO DEFINE A CORRES-
C     PONDENCE BETWEEN THE FIVE POSSIBLE SITE NUMBERS AND THE REMOTE
C     TERMINAL SWITCH CONNECTIONS USED.
C
      NAMELIST /PARAM/ RTLMT, DELTAT, SITE
C
      DATA SITE /10*1,10*2,10*3/, DELTAT /0/, RTLMT /3/
C
C*********************************************************************
C
      DATA    SWTCHN /3*(1,2,3,4,5,6,7,8,9,10)/, SVTIME /30*0/,
     -        OUTFILE/11,12,13,14,15/,
     -        READER, PRINTER, INTAPE /5,6,1/,
     -        TIMER, PREVTM, INVLDSN, INVLDRT, TEMPTM /5*0/
C
      WRITE (PRINTER,2000)
 2000 FORMAT (1H1)
C
C        READ USER PARAMETERS AND ECHO THOSE USED FOR USER VERIFICATION.
C
      READ (READER,PARAM)
      IF (EOF(READER)) 1,2
C
    1 WRITE (PRINTER,2001)
 2001 FORMAT (30X,"NO USER CONTROL PARAMETERS WERE PROVIDED - THE "
     -            "DEFAULT VALUES ARE AS FOLLOWS:"//)
C
    2 WRITE (PRINTER,2002) RTLMT, DELTAT, SITE
 2002 FORMAT (30X,"CONTROL PARAMETERS"/
     -        1H+,29X,"__________________"///
     -             30X,"$PARAM"//
     -             30X,"RTLMT      = ",I2//
     -             30X,"DELTAT     = ",I2//
     -             30X,"SITE       = ",10(I2,","),//,42X,10(I2,",")
     -                                        ,//,42X,10(I2,",")//
     -             30X,"$END",//)
C
C     READ A PHYSICAL RECORD FROM THE FIELD DATA TAPE.
C
    3 BUFFER IN (INTAPE,1) (INBUF(1),INBUF(24))
      IF (UNIT(INTAPE)) 4,98,88
C
```

```
C         PARITY ERROR FOUND; REPORT PRESENT CLOCK TIME AND CONTINUE.
C
   88 OUTIME = (TEMPTM * .002)/60
      WRITE (6,8800) OUTIME
 8800 FORMAT (10X,"READ PARITY ERROR - CLOCK INTEGRITY QUESTIONABLE"
     -             " AFTER ",I8," MINUTES")
C
C     BREAK UP PHYSICAL RECORD INTO LOGICAL RECORDS
C
    4 DECODE (240,1000,INBUF(1)) ((NOSWCH(I),TIMEPK(I)),I=1,60)
 1000 FORMAT (60(R1,R3))
C
C**********************************************************************
C
C         DECODE REMOTE TERMINAL NUMBER AND SWITCH NUMBER. NOTE THAT THE
C     TWO MOST SIGNIFICANT BITS BECOME THE REMOTE TERMINAL NUMBER WHILE
C     THE FOUR LEAST SIGNIFICANT BITS BECOME THE SWITCH NUMBER.
C
      DO 100 I=1,60
         N = NOSWCH(I)
         N10 = N/16
         N10S = N10*10
         N1 = N - (N10*16) + 1
         SWN = N10S + N1
         IF (SWN .EQ. 46) GO TO 99
C
C**********************************************************************
C
C
C         ADVANCE THE CLOCK ADJUSTING AS NECESSARY TO PRODUCE A MONOTONIC
C     INCREASING FUNCTION.
C
         TEMPTM = TIMEPK(I)
         IF (PREVTM .GT. TEMPTM) TIMER = TIMER + 262143
         PREVTM = TEMPTM
         TEMPTM = TEMPTM + TIMER
C
C        CHECK FOR BOTH INVALID SWITCH NUMBER AND INVALID REMOTE TERMINAL
C     NUMBER.
C
         IF (N10 .LT. RTLMT) GO TO 5
         INVLDRT = INVLDRT + 1
         GO TO 100
    5    IF (N1 .LE. 10) GO TO 10
         INVLDSN = INVLDSN + 1
         GO TO 100
C
C      CHECK FOR DUPLICATE SWITCH CLOSURES.
C
   10    IF (SVTIME(SWN) + DELTAT .GE. TEMPTM) GO TO 100
C
C         CONVERT SWITCH NUMBER INTO SITE NUMBER AND LOCAL SWITCH NUMBER
```

```
C     AND WRITE THEM TOGETHER WITH THE EPOCH TIME TO THE APPROPRIATE FILE.
C
         SITE = SITE(SWN)
         DISKFL = OUTFILE(SITE)
         SVTIME(SWN)=TEMPTM
         WRITE (DISKFL,2200) SWTCHN(SWN), TEMPTM
 2200    FORMAT (I2,I18)
  100 CONTINUE
C
C*****************************************************************************
C
C        READ NEXT 20 RECORD BUFFER
C
      GO TO 3
C
C        PRINT APPROPRIATE ERROR MESSAGES.
C
   98 WRITE (6,9800)
 9800 FORMAT(10X,"END OF FILE ENCOUNTERED ON TAPE BEFORE END OF DATA",
     - " MARK")
C
C
C        END OF DATA ENCOUNTERED - SUMMARIZE ERRORS.
C
   99 WRITE (PRINTER,9900) INVLDSN, INVLDRT
 9900 FORMAT (23X,I8," INVALID SWITCH NUMBERS WERE ENCOUNTERED",//,
     -        23X,I8," INVALID REMOTE TERMINAL NUMBERS WERE ENCOUNTERED")
      STOP
      END
```

1812

```
      PROGRAM PROCESS (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
C
C        INPUT DISK FILE                          OUTPUT DISK FILE
C
     -        TAPE1,                                     TAPE11              )
C
C***********************************************************************
C
C              TRAFFIC DATA PROGRAM TO PROCESS EDITED FIELD DATA
C
C                                 WRITTEN FOR
C
C            THE VIRGINIA HIGHWAY AND TRANSPORTATION RESEARCH COUNCIL
C
C                                      BY
C                                JERRY L. KORF
C                              RESEARCH SCIENTIST
C
C                                     AND
C
C                                 WOON-HO SONG
C                              GRADUATE ASSISTANT
C                                    3/1/77
C
C                       REVISED 11/15/77 BY J. L. KORF
C                       REVISED  5/17/79 BY J. L. KORF
C                       REVISED 10/1/79  BY J. L. KORF
C
C***********************************************************************
C
C          'PROCESS' IDENTIFIES VEHICLES FROM THE SITE FILE AND ESTABLISHES
C      THE FOLLOWING RELATED INFORMATION;
C
C          LANE: NUMBER OF THE LANE IN WHICH VEHICLE IS TRAVELLING,
C
C          LATERAL PLACEMENT: DISTANCE FROM RIGHT EDGE OF THE LANE
C                TO THE RIGHT WHEEL OF THE FIRST AXLE OF THE VEHICLE
C                MEASURED IN FEET,
C
C          ENTRY TIME: THE CLOCK TIME AT WHICH THE VEHICLE ENTERS THE SITE,
C
C          SPEED: AN AVERAGE OF INDIVIDUAL AXLE SPEEDS OF THE VEHICLE
C                IN MILE PER HOUR,
C
C          BACK HEADWAY: THE TIME INTERVAL MEASURED FROM THE LAST AXLE
C                OF THE LEADING VEHICLE TO THE FIRST AXLE OF THE
C                FOLLOWING VEHICLE. THIS MEASUREMENT IS MADE AT SWITCH
C                NUMBER 1 OF EACH LANE OF THE SITE,
C
C          FRONT HEADWAY: THE TIME INTERVAL MEASURED FROM THE FIRST AXLE
C                OF THE LEADING VEHICLE TO THE FIRST AXLE OF THE
```

```
C                 FOLLOWING VEHICLE AS IN BACK HEADWAY,
C
C           VEHICLE TYPE: TYPE OF THE VEHICLE IS DETERMINED FROM THE NUMBER
C                 OF AXLES AND THEIR WHEEL BASE BY REFERENCING THE VEHICLE
C                 CONFIGURATION PROTOTYPES. PRESENTLY THERE ARE 39 VEHICLE
C                 TYPES. ADDITIONALLY THE PROGRAM ALLOWS FOR 6 UNKNOWN
C                 VEHICLE TYPES.
C
C          NOTE! THIS PROGRAM DISCARDS THE LAST VEHICLE TO PASS THROUGH
C      THE SITE DUE TO PROGRAM TERMINATION CONSDERATIONS.
C
C        IT SHOULD BE FURTHER NOTED THAT PRESENTLY THE HARDWARE IS LIMITED
C      TO THIRTY TAPE SWITCHES. THE PROGRAM PROCESSES ONE SITE AT A TIME
C      WITH A LIMIT OF FOUR LANES PER SITE.
C
C          'PROCESS'  ASSUMES THE INPUT FILE IS IN SWITCH EVENT-TIME ORDER
C      BY SITE. THIS IS ACCOMPLISHED BY THE NATURAL ORDER OF DATA RECORDING
C      AND BY PROGRAM 'RDTAPE' WHICH SEPARATES THE FIELD DATA TAPE INTO FILES
C      BY SITE NUMBER.
C
C*********************************************************************
C
       IMPLICIT INTEGER(A-Z)
C
C          INPUT PARAMETER VARIABLES FOR HEADING INFORMATION
C
       INTEGER    RUNNAME(8),SITENM(8),CONDITN(8),DATE(8)
C
C          INPUT PARAMETER VARIABLES FOR PROCESSING CONTROL
C
       INTEGER   SWTYPE(30)
      -          ,LANE(30)
      -          ,SITE(30)
      -          ,SUMMARY(4,41)
      -          ,STARTTM
      -          ,FRSTVH
      -          ,AXLUZD
      -          ,OLDAX1
      -          ,VHCLNO
      -          ,SVLAXTM
      -          ,SWTIME
C
       INTEGER   TRIPTM(4,3,40)
      -          ,DIAGNZ(4,4,40)
      -          ,ISTRTM(3)
      -          ,POINTER(4,3,2)
      -          ,LNSUM(4)
      -          ,AXLCNT(4)
      -          ,AX1PTR(4)
      -          ,VNOAXLE(4)
      -          ,VTYPE(4)
      -          ,VTPNAME(40)
```

1814

```
     -          ,ENTHR(4),ENTMINT(4)
     -          ,FRSTSW,SCNDSW,ZONESW
     -          ,FRSWTM,SNSWTM,ZNSWTM,ZNSWTM2
     -          ,AXLE1
     -          ,AXFRNT
     -          ,READER,PRINTER,INTAPE,OUTTAPE
     -          ,FRONT,REAR
     -          ,FRNTPTR, REARPTR
     -          ,QLMT
C
      INTEGER    SITENO,NOLANE,NOSWID,NOSWTYP,AXLLMT,NOPTRTP,NOVTYPE
     -          ,RDSPEED,NOAXLES
C
      INTEGER    MCYCLE, MCTRLR, TRCK113, CAR, CAR111, CAR112,
     -           TTRL111, TTRL112, TTRL113, HSTL113, HSTL114,
     -           TRCK123, TTRL121, TTRL122, TRUCK11, TRUCK12, TRUCK13,
     -           TRCK131, TRCK132, TRCK121, TRCK122, TRCK111, TRCK112,
     -           TRCK114, TRCK133,
     -           BUS11, BUS12, PUVAN, PUVN111, PUVN112, PUVN113,
     -           STRCK11, STRCK12, STRK111, STRK121, TTRL123
C
      LOGICAL    PRCSLN(4)
     -          ,PRNTLN(4)
     -          ,PASS(4)
     -          ,DELFLAG(4,3)
     -          ,VPASS
     -          ,DBUG
     -          ,DEBUG
     -          ,LIST
     -          ,PGCNTRL
     -          ,SPDERR
     -          ,METRIC
C
      REAL       AXBS(4,40)
     -          ,AXSPEED(4,40)
     -          ,TAXBS(8)
     -          ,TAXSPD
     -          ,AXBASE
     -          ,AXSDTM,AXBSTM
     -          ,BHEDWAY(4),FHEDWAY(4)
     -          ,ZONE(4)
C
      REAL       VSPEED(4)
     -          ,SWDIST,POLTIME,SPDFCT,MPHFCT
     -          ,ENTSEC(4)
     -          ,REGTIME,SEC
     -          ,ENTRYT(4)
     -          ,SPEED
     -          ,ZNWDTH
     -          ,DUALS
     -          ,MAXAXBS
     -          ,MINAXBS
```

```
     -          ,SPDMIN
     -          ,SPDMAX
     -          ,SVAXSPD
C
C
      COMMON /JOINT/ REAR,AXLLMT,TRIPTM,POINTER
     -       /DELETEQ/ FRONT,DELFLAG,PRINTER
C
C*****************************************************************************
C
C         PROGRAM CONTROL PARAMETER VARIABLES ARE AS FOLLOWS;
C
      NAMELIST /PARAM/  DBUG, LIST, SWTYPE, LANE, STARTTM, SITENO,
     -                  BEGLIST, ENDLIST, PRCSLN, PRNTLN, SITE,
     -                  SPDMIN, SPDMAX, VHCLNO, METRIC
C
C         DEFAULT INPUT PARAMETER VALUES FOR PROCESSING CONTROL
C
      DATA      LIST/.FALSE./, DBUG/.FALSE./
     -          ,BEGLIST, ENDLIST /0,0/
     -          ,SWTYPE /1,2,3,1,2,3,1,2,1,2
     -                  ,1,2,3,1,2.3,1,2,1,2
     -                  ,1,2,3,1,2,3,1,2,1,2/
     -          ,LANE /1,1,1.2,2,2.3,3,4,4
     -                ,1,1,1.2,2,2.3,3,4,4
     -                ,1.1,1.2,2,2.3,3,4,4/
     -          ,SITE /1,1,1,1,1,1,1,1,1,1
     -                ,2,2,2,2,2,2,2,2,2,2
     -                ,3,3,3,3,3,3,3,3,3,3/
     -          ,PRCSLN/4*.TRUE./
     -          ,PRNTLN/4*.TRUE./
     -          ,SITENO/1/
     -          ,STARTTM/000000/
     -          ,VHCLNO/0/
     -          ,SPDMIN/10.0/
     -          ,SPDMAX/100.0/
     -          ,METRIC/.FALSE./
C
C*****************************************************************************
C
C         SET CONSTANTS TO REFLECT THE DISTANCE BETWEEN THE PARALLEL
C     MEMBERS OF THE SWITCH TRAP AND TO REFLECT THE TIME BETWEEN SUCESSIVE
C     SWITCH EXAMINATIONS (I.E. THE TIME VALUE GIVEN EACH COUNT).
C
      DATA      SWDIST/16.0/
     -          ,POLTIME/0.002/
     -          ,ZNWDTH/7.07/
C
C
      DATA      NOVTYPE/40/
     -          ,DELFLAG /12*.FALSE./
     -          ,PASS /4*.FALSE./
```

1816

```
-          ,PGCNTRL /.FALSE./
-          ,DEBUG /.FALSE./
-          ,SPDERR/.FALSE./
-          ,RDSPEED/40/,MAXAXBS,MINAXBS/35.0,3.5/
-          ,INTAPE /1/
-          ,OUTTAPE /11/
C
      DATA     SEQ/1/
-          ,FRSTSW,SCNDSW,ZONESW /1,2,3/
-          ,FRONT,REAR/1,2/
-          ,READER,PRINTER /5,6/
-          ,AXLE1/1/
-          ,NOSWTYP/3/
-          ,AXLLMT/40/
-          ,NOLANE/4/
-          ,NOPTRTP/2/
-          ,DUALS/4.8/
-          ,TAXBS/8*0.0/
-          ,QLMT/36/
C
      DATA     MCYCLE, MCTRLR, CAR, CAR111, CAR112, PUVAN, PUVN111,
-              PUVN112,            TRUCK11, TRUCK12, TRUCK13, TRCK111,
-              TRCK112, TRCK113, TRCK114, TRCK121, TRCK122, TRCK123,
-              TRCK131, TRCK132, TRCK133, STRCK11, STRCK12, STRK111,
-              STRK121, TTRL111, TTRL112,          TTRL121, TTRL122,
-              TTRL123, HSTL113, HSTL114, BUS11, BUS12
-              /1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,
-               22,23,24,25,26,27,28,29,30,31,32,33,34/
C
      DATA     VTPNAME /"MOTORCYCLE","MC & TRLR","CAR","CAR-TRLR1",
-              "CAR-TRLR2","PU OR VAN","PU.VAN-TL1","PU.VAN-TL2",
-                          "TRUCK","TRUCK 3 AX","TRUCK 4 AX",
-              "TRK 2AX T1","TRK 2AX T2",
-              "TRK 2AX T3","TRK 2AX T4","TRK 3AX T1","TRK 3AX T2",
-              "TRK 3AX T3",
-              "TRK 4AX T1","TRK 4AX T2","TRK 4AX T3","ST TRK 2AX",
-              "ST TRK 3AX","STRK 2X T1","STRK 3X T1",
-              "T.T. 3X T1","T.T. 4X T2",              "T.T. 4X T1",
-              "T.T. 5X T2","T.T. 6X T3","HOUSE TL3","HOUSE TL4",
-                                   "BUS 2 AX","BUS 3 AX",
-                          "UNKNWN 2AX","UNKNWN 3AX","UNKNWN 4AX",
-              "UNKNWN 5AX","UNKNWN 6AX","UNKNWN 7AX"/
C
C*******************************************************************
C
C        BEGIN PROCESSING.
C
C*******************************************************************
C
C        START PRINT-OUT ON NEW PAGE.
C
      WRITE(PRINTER,5900)
```

```
 5900 FORMAT(1H1)
C
C         READ HEADING INFORMATION FROM CARD READER THEN PRINT PAGE HEADER.
C
      READ(READER,1010) RUNNAME,SITENM.CONDITN,DATE
 1010 FORMAT(8A10/8A10/8A10/8A10)
C
      WRITE(PRINTER,5915) RUNNAME,SITENM,CONDITN,DATE
 5915 FORMAT(30X,"HEADING INFORMATION"/
     -   1H+,29X,"___________________"///
     -        30X,"RUN NAME  - ",8A10//
     -        30X,"SITE NAME - ",8A10//
     -        30X,"CONDITION - ",8A10//
     -        30X,"DATE      - ",8A10//////)
C
C         READ CONTROL PARAMETERS FROM CARD READER AND PRINT THE OPTIONS
C     IN EFFECT FOR THIS RUN.
C
      READ(READER,PARAM)
      IF(EOF(READER)) 600,700
C
  600 WRITE(PRINTER,*) "INADEQUATE INPUT DATA FOUND"
      STOP
C
  700 IF (BEGLIST .EQ. 0) BEGLIST = STARTTM
      IF (ENDLIST .EQ. 0) ENDLIST = BEGLIST + 1000
      WRITE (PRINTER,5917) SITENO, STARTTM, LIST, BEGLIST, ENDLIST,
     -                     PRCSLN, PRNTLN, SWTYPE, LANE, SITE, DBUG,
     -                     VHCLNO, METRIC, SPDMIN, SPDMAX
 5917 FORMAT(30X,"CONTROL PARAMETERS"/,1H+,
     -        29X,"___________________"///
     -        30X,"$PARAM"//
     -        30X,"SITENO     = ",I2//
     -        30X,"STARTTM    =  ",I6.5//
     -        30X,"LIST       =  ",L1,4X,"BEGLIST  =",I7.5,/,
     -                             48X,"ENDLIST  =",I7.5,//.
     -        30X,"PRCSLN     = ",3(1X,L1,","),1X,L1,//
     -        30X,"PRNTLN     = ",3(1X,L1,","),1X.L1,//
     -        30X,"SWTYPE     =",10(I3,","),/,
     -                    41X,10(I3,","),/,41X,10(I3,",")//
     -        30X,"LANE       =",10(I3,","),/,
     -                    41X,10(I3,","),/,41X,10(I3,",")//
     -        30X,"SITE       =",10(I3,","),/,
     -                    41X,10(I3,","),/,41X,10(I3,",")//
     -        30X,"DBUG       =  ",L1,4X,"VHCLNO   =",I6,//
     -        30X,"METRIC     =  ",L1,//
     -        30X,"SPDMIN     =",F6.1,//
     -        30X,"SPDMAX     =",F6.1,//
     -        30X,"$END")
C
C        CLEAR DATA STORAGE ARRAYS.
C
```

1818

```
      DO 510 I=1,NOLANE
      DO 510 J=1,NOSWTYP
      DO 510 K=1,AXLLMT
      DIAGNZ(I,J,K) = 0
  510 TRIPTM(I,J,K) = 0
C
      DO 515 I=1,NOLANE
      LNSUM(I) = 0
      DO 515 J=1,NOSWTYP
      DO 515 K=1,NOPTRTP
  515 POINTER(I,J,K) = 1
C
      DO 525 I=1,NOLANE
      DO 525 J=1,AXLLMT
      AXSPEED(I,J)=0.0
  525 AXBS(I,J)=0.0
C
      DO 526 I=1,NOLANE
      DO 526 J=1,NOVTYPE
  526 SUMMARY(I,J)=0
C
      DO 527 I=1,NOLANE
      ZONE(I)=0
      AXLCNT(I)=0
      ENTHR(I)=0
      ENTMINT(I)=0
      ENTSEC(I)=0.0
      ENTRYT(I)=0.0
      VSPEED(I)=0.0
      BHEDWAY(I)=0.0
      FHEDWAY(I)=0.0
  527 VTYPE(I)=0
C
C        CONVERT STARTING TIME INTO THE DESIGNATED HOUR, MINUTE, SECOND.
C
      ISTRTM(1) = STARTTM/10000
      ISTRTM(2) = STARTTM/100 - ISTRTM(1)*100
      ISTRTM(3) = STARTTM - (STARTTM/100)*100
C
C        FORMULATE SPEED RELATED CONSTANTS ACCORDING TO THE FOLLOWING
C    RELATIONSHIPS

C
C        SPEED(FT/SEC) = SWDIST / (COUNTERTIME*POLTIME)
C        SPEED(MILE/HOUR) = SPEED(FT/SEC) * (3600/5280)
C
      MPHFCT = 3600.0/(POLTIME * 5280.0)
      SPDFCT = SWDIST * MPHFCT
C
C        WRITE HEADER INFORMATION TO OUTPUT TAPE.
C
      WRITE (OUTTAPE,5910) RUNNAME,SITENM,CONDITN,DATE,METRIC
```

```
 5910 FORMAT (8A10/8A10/8A10/8A10,L1)
C
C*********************************************************************
C
C         READ SWITCH DATA FROM SITE FILE ACCUMULATING AXLE INFORMATION
C     UNTIL AN AXLE DISTANCE IS FOUND TO SUGGEST A SEPARATION OF VEHICLES.
C
C*********************************************************************
C
    1 READ (INTAPE,1300) SWID, TIME
 1300 FORMAT (I2,I18)
      IF (EOF(INTAPE)) 90,2
C
C         VERIFY THAT THIS IS A VALID SWITCH NUMBER TO PROCESS, THEN
C         SET PRESENT LANE AND PRESENT SWITCH TYPE
C
    2 IF (SITE(SWID) .NE. SITENO) GO TO 1
      PLANE = LANE(SWID)
      IF (.NOT. PRCSLN(PLANE)) GO TO 1
      IF (SEQ .GT. VHCLNO .AND. DBUG) DEBUG = .TRUE.
      IF (PLANE .LT. 1 .OR. PLANE .GT. NOLANE) GO TO 880
      PSWTYPE = SWTYPE(SWID)
      IF (PSWTYPE .LT. 1 .OR. PSWTYPE .GT. 3) GO TO 890
      DELFLAG(PLANE,PSWTYPE) = .FALSE.
C
C         CHECK TO SEE IF THIS VEHICLE IS TRAVELING IN THE WRONG DIRECTION
C
      IF (PASS(PLANE)) CALL PASSING (PSWTYPE)
      IF (PSWTYPE .NE. SCNDSW) GO TO 3
      IF (POINTER(PLANE,SCNDSW,REAR) .NE. POINTER(PLANE,FRSTSW,REAR))
     -                                                        GO TO 3
      PASS(PLANE) = .NOT. PASS(PLANE)
      CALL PASSING (PSWTYPE)
C
C         INSERT THE SWITCH ACTIVATION TIME INTO THE AXLE QUEUE.
C
    3 REARPTR = POINTER(PLANE,PSWTYPE,REAR)
      IF (REARPTR .EQ. AXLLMT) POINTER(PLANE,PSWTYPE,REAR) = 1
      IF (REARPTR .NE. AXLLMT)
     -     POINTER(PLANE,PSWTYPE,REAR) = POINTER(PLANE,PSWTYPE,REAR) + 1
      REARPTR = POINTER(PLANE,PSWTYPE,REAR)
      FRONPTR = POINTER(PLANE,PSWTYPE,FRONT)
C
      IF (DEBUG)
     -  WRITE(PRINTER,7100) SWID, REARPTR, FRONPTR, PSWTYPE, TIME
 7100 FORMAT (30X,"SWID = ",I2,5X,"REARPTR = ",I2,5X,"FRONPTR = ",I2,5X,
     -        "PSWTYPE = ",I1,5X,"TIME = ",I10)
C
      IF (REARPTR .EQ. FRONPTR) GO TO 865
C
      TRIPTM(PLANE,PSWTYPE,REARPTR) = TIME
C
```

```
      IF (PSWTYPE .EQ. ZONESW) GO TO 1
      DIAGNZ(PLANE,PSWTYPE,REARPTR) = SWID
      DIAGNZ(PLANE,PSWTYPE+2,REARPTR) = TIME
C
      IF (PSWTYPE .NE. SCNDSW) GO TO 1
C
C        INCREMENT AXLE COUNTER AND SAVE QUEUE POINTER FOR FIRST AXLE
C
      AXLCNT(PLANE) = AXLCNT(PLANE) + 1
C
      IF (AXLCNT(PLANE) .EQ. 1)
     -                          AX1PTR(PLANE) = POINTER(PLANE,SCNDSW,REAR)
C
      CALL DELETEQ(FRSTSW,PLANE)
      CALL DELETEQ(SCNDSW,PLANE)
C
C        CALCULATE AXLE SPEED
C
      PAXPTR = POINTER(PLANE,SCNDSW,FRONT)
      LAXPTR = PAXPTR - 1
      IF (PAXPTR .EQ. 1) LAXPTR = AXLLMT
C
      AXSDTM = TRIPTM(PLANE,SCNDSW,PAXPTR) - TRIPTM(PLANE,FRSTSW,PAXPTR)
      AXSPEED(PLANE,AXLCNT(PLANE)) = SPDFCT / AXSDTM
      IF (.NOT. SPDERR) SVAXSPD = AXSPEED(PLANE,AXLCNT(PLANE))
      IF (AXSPEED(PLANE,AXLCNT(PLANE)) .LT. SPDMIN) SPDERR = .TRUE.
      IF (AXSPEED(PLANE,AXLCNT(PLANE)) .GT. SPDMAX) SPDERR = .TRUE.
      IF (PASS(PLANE))
     -     AXSPEED(PLANE,AXLCNT(PLANE)) = - AXSPEED(PLANE,AXLCNT(PLANE))
C
      IF (DEBUG)
     -  WRITE (PRINTER,9000) AXLCNT(PLANE), AXSPEED(PLANE,AXLCNT(PLANE))
     -                       ,SPDFCT, TRIPTM(PLANE,SCNDSW,PAXPTR)
     -                       ,TRIPTM(PLANE,FRSTSW,PAXPTR)
 9000 FORMAT(30X,"AXSPEED",I1," :",F7.2," = ",F6.1,"(SPDFCT) / ( ",
     -          ,I8," - ",I8," )")
C
C        IF THIS IS THE FIRST AXLE GET NEXT AXLE ELSE COMPUTE WHEELBASE.
C
      IF (AXLCNT(PLANE) .EQ. 1) GO TO 1
C
      AXBSTM = TRIPTM(PLANE,FRSTSW,PAXPTR) -
     -                  TRIPTM(PLANE,FRSTSW,LAXPTR)
      AXBSIDX = AXLCNT(PLANE) - 1
      IF (AXBSIDX .GE. QLMT) GO TO 870
      AXBS(PLANE,AXBSIDX) = AXBSTM * ABS(AXSPEED(PLANE,AXBSIDX))/ MPHFCT
      AXBASE = AXBS(PLANE,AXBSIDX)
C
      IF (DEBUG)
     -     WRITE (PRINTER,9010) AXBSIDX, AXBASE
     -          ,TRIPTM(PLANE,FRSTSW,PAXPTR),TRIPTM(PLANE,FRSTSW,LAXPTR)
     -          ,AXSPEED(PLANE,AXBSIDX),AXBSIDX,MPHFCT
```

```
 9010 FORMAT(30X,"AXBS",I1," : ",F9.2," = ( ",I8," - ",I8," )"
     -           ," * ",F7.2,"(AXSPEED",I1,") / ",F5.1,"(MPHFCT)")
C
C          IF AXLE-BASE IS GREATER THAN MAXIMUM ALLOWABLE AXLE-BASE,
C     THEN ASSUME THAT A NEW VEHICLE HAS BEEN DETECTED.
C
      IF (AXBASE .LT. MAXAXBS) GO TO 1
      AXLCNT(PLANE) = AXLCNT(PLANE) - 1
      IF (SPDERR) GO TO 875
      SVLAXTM = TRIPTM(PLANE,FRSTSW,LAXPTR)
      AXFRNT = 1
C
C***********************************************************************
C
C          A DEFINITE BREAK IN THE FLOW OF VEHICLES HAS BEEN DETECTED
C          NOW IDENTIFY INDIVIDUAL VEHICLES.
C
C***********************************************************************
C
C
C          DEDUCE VEHICLE TYPE BASED ON NUMBER OF AXLES AND THE DISTANCE
C     BETWEEN TANDEM AXLES.
C_______________________________________________________________________
C
    9 NOAXLES = AXLCNT(PLANE)
      NOAXBS = NOAXLES - 1
      IF (NOAXBS .GT. 8) NOAXBS = 8
      DO 100 I=1,NOAXBS
         J = AXFRNT + I - 1
         TAXBS(I) = AXBS(PLANE,J)
  100 CONTINUE
C
C     CHECK FOR SINGLE AXLE VEHICLE
C
      IF (NOAXLES .GT. 1) GO TO 102
      GO TO 860
C
C
  102 IF (TAXBS(1) .GT. 3.5) GO TO 103
      AXLUZD = 2
      IF (NOAXLES .LT. 4) AXLUZD = NOAXLES
      GO TO 888
C
  103 IF (TAXBS(1) .GT. 5.5) GO TO 107
      VTYPE(PLANE) = MCYCLE
      AXLUZD = 2
      IF (NOAXLES .LT. 3) GO TO 11
      IF (TAXBS(2) .GT. 10.0) GO TO 11
      VTYPE(PLANE) = MCTRLR
      AXLUZD = 3
      GO TO 11
C
```

```
  107 IF (TAXBS(1) .GT. 10.9) GO TO 117
      IF (NOAXLES .GT. 2) GO TO 108
  157 VTYPE(PLANE) = CAR
      AXLUZD = 2
      GO TO 11
C
  108 IF (TAXBS(2) .GT. DUALS) GO TO 111
      IF (NOAXLES .GT. 3) GO TO 109
  158 VTYPE(PLANE) = TRUCK12
      AXLUZD = 3
      GO TO 11
C
  109 IF (TAXBS(3) .GT. DUALS) GO TO 139
  149 IF (NOAXLES .GT. 4) GO TO 110
  159 VTYPE(PLANE) = TRUCK13
      AXLUZD = 4
      GO TO 11
C
  110 IF (TAXBS(4) .GT. 18.0) GO TO 159
      IF (NOAXLES .EQ. 5) GO TO 150
      IF (TAXBS(5) .LE. DUALS) GO TO 160
      IF (NOAXLES .EQ. 6) GO TO 159
  150 VTYPE(PLANE) = TRCK131
      AXLUZD = 5
      GO TO 11
C
  160 IF (NOAXLES .GT. 6) GO TO 180
  170 VTYPE(PLANE) = TRCK132
      AXLUZD = 6
      GO TO 11
C
  180 IF (TAXBS(6) .GT. DUALS) GO TO 170
      VTYPE(PLANE) = TRCK133
      AXLUZD = 7
      GO TO 11
C
  111 IF (TAXBS(2) .GT. 18.0) GO TO 112
      IF (NOAXLES .GT. 3) GO TO 151
  141 VTYPE(PLANE) = CAR111
      AXLUZD = 3
      GO TO 11
C
  151 IF (TAXBS(3) .LE. DUALS) GO TO 192
      IF (NOAXLES .EQ. 4) GO TO 157
      IF (TAXBS(3) .LE. 18.0) GO TO 157
      GO TO 141
C
  112 IF (NOAXLES .GT. 3) GO TO 162
  152 VTYPE(PLANE) = TTRL111
      AXLUZD = 3
      GO TO 11
C
```

```
  162 IF (TAXBS(3) .LE. DUALS) GO TO 172
      IF (NOAXLES .EQ. 4) GO TO 157
      GO TO 152
C
  172 IF (NOAXLES .EQ. 4) GO TO 182
      IF (TAXBS(4) .LE. DUALS) GO TO 163
  182 VTYPE(PLANE) = TTRL112
      AXLUZD = 4
      GO TO 11
C
  115 IF (NOAXLES .GT. 4) GO TO 185
  175 VTYPE(PLANE) = TTRL121
      AXLUZD = 4
      GO TO 11
C
  185 IF (TAXBS(4) .GT. DUALS) GO TO 195
      IF (NOAXLES .GT. 5) GO TO 205
  215 VTYPE(PLANE) = TTRL122
      AXLUZD = 5
      GO TO 11
C
  195 IF (NOAXLES .GE. 6) GO TO 175
      GO TO 158
C
  205 IF (TAXBS(5) .GT. DUALS) GO TO 215
  206 VTYPE(PLANE) = TTRL123
      AXLUZD = 6
      GO TO 11
C
  117 IF (TAXBS(1) .LT. 12.1) GO TO 124
      IF (TAXBS(1) .GT. 14.5) GO TO 123
      IF (NOAXLES .GT. 2) GO TO 118
  177 VTYPE(PLANE) = TRUCK11
      AXLUZD = 2
      GO TO 11
C
  118 IF (TAXBS(2) .GT. DUALS) GO TO 122
  128 IF (NOAXLES .GT. 3) GO TO 119
      GO TO 158
C
  119 IF (TAXBS(3) .LE. DUALS) GO TO 149
C
  139 IF (TAXBS(3) .GT. 18.0) GO TO 115
  121 IF (NOAXLES .GT. 4) GO TO 113
C
  131 VTYPE(PLANE) = TRCK121
      AXLUZD = 4
      GO TO 11
C
  113 IF (TAXBS(4) .GT. DUALS) GO TO 131
      IF (NOAXLES .EQ. 5) GO TO 114
      IF (TAXBS(5) .GT. DUALS) GO TO 114
```

```
      VTYPE(PLANE) = TRCK123
      AXLUZD = 6
      GO TO 11
C
  114 VTYPE(PLANE) = TRCK122
      AXLUZD = 5
      GO TO 11
C
  122 IF (TAXBS(2) .GT. 18.0) GO TO 112
      IF (NOAXLES .GT. 3) GO TO 142
  132 VTYPE(PLANE) = TRCK111
      AXLUZD = 3
      GO TO 11
C
  142 IF (TAXBS(3) .GT. DUALS) GO TO 177
      IF (NOAXLES .EQ. 4) GO TO 222
      IF (TAXBS(4) .LE. DUALS) GO TO 213
C
  222 VTYPE(PLANE) = TRCK112
      AXLUZD = 4
      GO TO 11
C
  163 IF (NOAXLES .EQ. 5) GO TO 173
      IF (TAXBS(5) .LE. DUALS) GO TO 183
  173 VTYPE(PLANE) = HSTL113
      AXLUZD = 5
      GO TO 11
C
  183 VTYPE(PLANE) = HSTL114
      AXLUZD = 6
      GO TO 11
C
  192 IF (NOAXLES .EQ. 4) GO TO 202
      IF (TAXBS(4) .LE. DUALS) GO TO 212
  202 VTYPE(PLANE) = CAR112
      AXLUZD = 4
      GO TO 11
C
  212 IF (NOAXLES .EQ. 5) GO TO 213
      IF (TAXBS(5) .GT. DUALS) GO TO 213
      VTYPE(PLANE) = TRCK114
      AXLUZD = 6
      GO TO 11
C
  213 VTYPE(PLANE) = TRCK113
      AXLUZD = 5
      GO TO 11
C
C
  123 IF (TAXBS(1) .GT. 23.0) GO TO 120
      IF (NOAXLES .GT. 2) GO TO 143
  233 VTYPE(PLANE) = STRCK11
```

```
      AXLUZD = 2
      GO TO 11
C
  143 IF (TAXBS(2) .GT. DUALS) GO TO 193
      IF (NOAXLES .GT. 3) GO TO 144
  153 VTYPE(PLANE) = STRCK12
      AXLUZD = 3
      GO TO 11
C
  144 IF (TAXBS(3) .LE. DUALS) GO TO 888
      IF (NOAXLES .GT. 4) GO TO 146
      VTYPE(PLANE) = STRK121
      AXLUZD = 4
      GO TO 11
C
  146 IF (TAXBS(4) .GT. DUALS) GO TO 153
      GO TO 888
C
  193 IF (NOAXLES .GT. 3) GO TO 194
      VTYPE(PLANE) = STRK111
      AXLUZD = 3
      GO TO 11
C
  194 IF (TAXBS(3) .GT. DUALS) GO TO 233
      AXLUZD = 4
      GO TO 888
C
C     STRAIGHT TRUCKS WITH TWO AND THREE AXLE TRAILERS ARE CONSIDERED
C     RARE ENOUGH TO BE IGNORED.
C
C 203 IF (NOAXLES .GT. 4) GO TO 214
C     VTYPE(PLANE) = STRK112
C     AXLUZD = 4
C     GO TO 11
C
C 214 IF (TAXBS(4) .GT. DUALS) GO TO 204
C     VTYPE(PLANE) = STRCK113
C     AXLUZD = 5
C     GO TO 11
C
C
  120 IF (NOAXLES .GT. 2) GO TO 140
  130 VTYPE(PLANE) = BUS11
      AXLUZD = 2
      GO TO 11
C
  140 IF (TAXBS(2) .GT. DUALS) GO TO 130
      VTYPE(PLANE) = BUS12
      AXLUZD = 3
      GO TO 11
C
  124 IF (NOAXLES .GT. 2) GO TO 125
```

```
  154 VTYPE(PLANE) = PUVAN
      AXLUZD = 2
      GO TO 11
C
  125 IF (TAXBS(2) .GT. 18.0) GO TO 112
      IF (TAXBS(2) .LE. DUALS) GO TO 128
      IF (NOAXLES .GT. 3) GO TO 145
  165 VTYPE(PLANE) = PUVN111
      AXLUZD = 3
      GO TO 11
C
  145 IF (TAXBS(3) .GT. DUALS) GO TO 165
      IF (NOAXLES .GT. 4) GO TO 176
  155 VTYPE(PLANE) = PUVN112
      AXLUZD = 4
      GO TO 11
C
C
  147 IF (TAXBS(5) .GT. DUALS) GO TO 146
      GO TO 206
C
  176 IF (TAXBS(4) .LE. DUALS) GO TO 163
      GO TO 155
C
  186 IF (TAXBS(4) .LT. 20.0) GO TO 158
      GO TO 121
C
C        VEHICLE TYPE CANNOT BE DEDUCED - ASSIGN VEHICLE TO UNKNOWN GROUP.
C
  888 AXLUZD = NOAXLES
      IF (NOAXLES .GT. 6) NOAXLES = 6
      VTYPE(PLANE) = 35 + NOAXLES
C
   11 FRSWTM = TRIPTM(PLANE,FRSTSW,AX1PTR(PLANE))
      SNSWTM = TRIPTM(PLANE,SCNDSW,AX1PTR(PLANE))
C
C        DETERMINE VEHICLE SPEED BY TAKING THE AVERAGE OF THE AXLE SPEEDS.
C_______________________________________________________________________
C
      VSPEED(PLANE) = 0.0
      VNOAXLE(PLANE) = AXLUZD
      AXLMIT = AXFRNT + AXLUZD - 1
      DO 530 AXLE=AXFRNT,AXLMIT
      TAXBS(AXLE) = 0.0
  530 VSPEED(PLANE) = VSPEED(PLANE) + AXSPEED(PLANE,AXLE)
      VSPEED(PLANE) = VSPEED(PLANE) / FLOAT(AXLUZD)
      VPASS = .F.
      IF (VSPEED(PLANE) .LT. 0.0) VPASS = .T.
      AXLMIT = AXLMIT - 1
C
C        DETERMINE VEHICLE ENTRY TIME AS THE CLOCK TIME WHEN THE FIRST
C    AXLE OF THE VEHICLE ACTIVATED THE FIRST SWITCH.
```

```
C___________________________________________________________________________
C
      REGTIME = FRSWTM * POLTIME
      HOUR = REGTIME/3600
      MINUTE = (REGTIME - HOUR*3600)/60
      SEC = REGTIME - HOUR*3600 - MINUTE*60
C
C         ADD STARTING TIME TO REGISTER TIME FOR CLOCK ENTRY TIME
C
      ENTSEC(PLANE) = ISTRTM(3) + SEC
      IF (ENTSEC(PLANE) .GE. 60) MINUTE = MINUTE + 1
      IF (ENTSEC(PLANE) .GE. 60) ENTSEC(PLANE) = ENTSEC(PLANE) - 60.0
      ENTMINT(PLANE) = ISTRTM(2) + MINUTE
      IF (ENTMINT(PLANE) .GE. 60) HOUR = HOUR + 1
      IF (ENTMINT(PLANE) .GE. 60) ENTMINT(PLANE) = ENTMINT(PLANE) - 60
      ENTHR(PLANE) = ISTRTM(1) + HOUR
      IF (ENTHR(PLANE) .GE. 24) ENTHR(PLANE) = ENTHR(PLANE) - 24
C
      ENTRYT(PLANE) = ENTHR(PLANE)*10000.0 + ENTMINT(PLANE)*100.0
     -                                     + ENTSEC(PLANE)
      ENTSEC(PLANE) = ENTSEC(PLANE) + 0.5
      IF (DEBUG)
     -     WRITE (PRINTER,9015) SEQ, VTPNAME(VTYPE(PLANE)), AXLUZD,
     -     ENTHR(PLANE), ENTMINT(PLANE), INT(ENTSEC(PLANE))
 9015 FORMAT (/30X,"VEHICLE NO. =",I4,5X,"VTPNAME = ",A10,5X,"AXLUZD = "
     -        ,I2,5X,"TIME = ",I2.1,1H:,I2.2,1H:,I2.2)
C
C         DETERMINE LATERAL PLACEMENT BY TRIANGULATION.
C___________________________________________________________________________
C
      GO TO 6
    5 CALL DELETEQ(ZONESW,PLANE)
      IF (.NOT. DELFLAG(PLANE,ZONESW)) GO TO 6
   55 ZONE(PLANE) = 0.0
      GO TO 20
    6 ZONEPTR = POINTER(PLANE,ZONESW,FRONT)
      ZNSWTM = TRIPTM(PLANE,ZONESW,ZONEPTR)
      IF (FRSWTM .GT. ZNSWTM) GO TO 5
      IF (ZNSWTM .GT. SNSWTM) GO TO 55
      SPEED = ABS(VSPEED(PLANE))
C
C     IF VEHICLE  PASSING COMPUTE LATERAL PLACEMENT USING SECOND CLOSURE
C
      IF (.NOT. VPASS) GO TO 8
      IF (VTYPE(PLANE) .EQ. MCYCLE) GO TO 7
      CALL DELETEQ(ZONESW,PLANE)
      IF (DELFLAG(PLANE,ZONESW)) GO TO 55
      ZONEPTR = POINTER(PLANE,ZONESW,FRONT)
      ZNSWTM2 = TRIPTM(PLANE,ZONESW,ZONEPTR)
      IF (ZNSWTM2 .LT. SNSWTM) ZNSWTM = ZNSWTM2
    7 SWTIME = ZNSWTM - FRSWTM
      GO TO 10
```

1828

```
C
    8 SWTIME = SNSWTM - ZNSWTM
C
   10 ZONE(PLANE) = SWDIST - (SWTIME * SPEED/MPHFCT)
      IF (ZONE(PLANE) .GT. ZNWDTH+0.5  .OR. ZONE(PLANE) .LT. 0.0)
     -                                                  ZONE(PLANE) = 0.0
   20 IF (DEBUG)
     -     WRITE (PRINTER,9020) ZONE(PLANE), SWDIST, SWTIME
     -                                 ,VSPEED(PLANE), MPHFCT, ZNSWTM
 9020 FORMAT (/30X,"ZONE : ",F5.2," = ",F4.1, " - ( ",I8," * "
     -          ,F6.2,"(VSPEED)"," / ",F5.1,"   ZNSWTM = ",I8)
C
C
C        IF METRIC UNITS REQUESTED CONVERT SPEED AND LENGTH DATA.
C
      IF (.NOT. METRIC) GO TO 15
      ZONE(PLANE) = .3048 * ZONE(PLANE)
      VSPEED(PLANE) = 1.609344 * VSPEED(PLANE)
      DO 200 I=AXFRNT,AXLMIT
      AXBS(PLANE,I) = .3048 * AXBS(PLANE,I)
  200 CONTINUE
C
C        IF LIST OPTION REQUESTED PRINT VEHICLE INFORMATION AND CREATE
C    OUTPUT DISK FILE; ELSE JUST CREATE THE DISK FILE.
C_______________________________________________________________________
C
C
   15 IF (ENTRYT(PLANE) .LT. BEGLIST .OR. ENTRYT(PLANE) .GT. ENDLIST
     -                                      .OR. .NOT. LIST) GO TO 133
      IF (PGCNTRL) GO TO 12
      FRSTVH = SEQ
      PGCNTRL = .TRUE.
   12 LINEMOD = MOD((SEQ-FRSTVH),50)
      IF (LIST .AND. LINEMOD .EQ. 0)
     -          WRITE (PRINTER,9060) RUNNAME,SITENM,CONDITN,DATE,ISTRTM
C
      IF (LIST .AND. LINEMOD .EQ. 0) WRITE (PRINTER,6050)
 6050 FORMAT (T3,"VEHICLE",
     -       T18,"LOCATION",T35,"ENTRY TIME",T49,"SPEED",
     -       T58,"HEADWAY(SEC)",T74,"TYPE OF",T87,"NO. OF",
     -       T101,"DISTANCE BETWEEN AXLES",/
     -    1H+,T13,"_________________",T35,"__________",T49,"_____",
     -       T58,"____________",
     -       T96,"________________________________")
      IF (METRIC) GO TO 17
      IF (LIST .AND. LINEMOD .EQ. 0) WRITE (PRINTER,6051)
 6051 FORMAT (T3,"NUMBER",T13,"SITE LANE ZONE(FT)",T35,"HR MIN SEC",
     -       T50,"MPH",T58,"BACK   FRONT",T74,"VEHICLE",T87,"AXLES",
     -       T96,"1--2   2--3   3--4   4--5   5--6",/,1H+,
     -       T3,"______",T13,"____ ____ _________",T35,"__ ___ ___",
     -       T50,"___",T58,"____   _____",T74,"_______",T87,"_____",
     -       T96,"____   ____   ____   ____   ____",/)
```

```
      GO TO 18
C
C     PRINT METRIC HEADINGS.
C
   17 IF (LIST .AND. LINEMOD .EQ. 0) WRITE (PRINTER,6052)
 6052 FORMAT (T3,"NUMBER",T13,"SITE LANE ZONE(M)",T35,"HR MIN SEC",
     -         T50,"KM/H",T58,"BACK   FRONT",T74,"VEHICLE",T87,"AXLES",
     -         T96,"1--2   2--3   3--4   4--5   5--6",/,1H+,
     -         T3,"______",T13,"____ ____ ________",T35,"__ ___ ___",
     -         T50,"____",T58,"____   _____",T74,"_______",T87,"_____",
     -         T96,"____   ____   ____   ____   ____",/)
C
   18 IF (LIST .AND. PRNTLN(PLANE))
     -           WRITE (PRINTER,6100) SEQ,SITENO,PLANE,ZONE(PLANE)
     -                     ,ENTHR(PLANE),ENTMINT(PLANE),INT(ENTSEC(PLANE))
     -                     ,VSPEED(PLANE),BHEDWAY(PLANE),FHEDWAY(PLANE)
     -                     ,VTPNAME(VTYPE(PLANE)), VNOAXLE(PLANE)
     -                     ,(AXBS(PLANE,I),I=AXFRNT,AXLMIT)
 6100 FORMAT(1X,I6,7X,I1,4X,I1,4X,F4.2,6X,I2.2,":",1X,I2.2,":",1X,I2.2,
     -        4X,F5.1,2X,F6.2,2X,F6.2,4X,A10,5X,I1,6X,5(F4.1,3X))
C
C        WRITE RESULTS TO DISK FILE FOR REPORT PROGRAM PROCESSING UNLESS
C        THIS IS THE FIRST VEHICLE IN THIS LANE (ZERO HEADWAYS) OR THIS
C        VEHICLE IS GOING IN THE WRONG DIRECTION (PASSING).
C
  133 IF (BHEDWAY(PLANE) .EQ. 0.0 .OR. VSPEED(PLANE) .LE. 0.0) GO TO 13
C
      WRITE (OUTTAPE,6600) SITENO,PLANE,ZONE(PLANE),VTYPE(PLANE)
     -                     ,ENTRYT(PLANE)
     -                     ,VSPEED(PLANE),FHEDWAY(PLANE),BHEDWAY(PLANE)
 6600 FORMAT (2I1,F4.1,I2,F8.1,F4.1,2F5.1)
C
      SUMMARY(PLANE,VTYPE(PLANE)) = SUMMARY(PLANE,VTYPE(PLANE)) + 1
C
C        DETERMINE FRONT AND BACK HEADWAYS FOR NEXT VEHICLE.
C______________________________________________________________________
C
C
C        ADJUST VEHICLE POINTERS BEFORE ESTABLISHING HEADWAYS
C
   13 AXFRNT = AXFRNT + AXLUZD
      AXLCNT(PLANE) = AXLCNT(PLANE) - AXLUZD
      SEQ = SEQ + 1
  135 OLDAX1 = AX1PTR(PLANE)
      AX1PTR(PLANE) = AX1PTR(PLANE) + AXLUZD
      IF (AX1PTR(PLANE) .GT. AXLLMT)
     -                   AX1PTR(PLANE) = AX1PTR(PLANE) - AXLLMT
      PAXPTR = AX1PTR(PLANE)
      LAXPTR = PAXPTR - 1
      IF (PAXPTR .EQ. 1) LAXPTR = AXLLMT
C
      BHEDWAY(PLANE) = (TRIPTM(PLANE,FRSTSW,PAXPTR) -
```

```
     -                    TRIPTM(PLANE,FRSTSW,LAXPTR)) * POLTIME
C
      FHEDWAY(PLANE) = (TRIPTM(PLANE,FRSTSW,PAXPTR) -
     -                    TRIPTM(PLANE,FRSTSW,OLDAX1)) * POLTIME
C
      IF (DEBUG)
     -     WRITE(PRINTER,9030)BHEDWAY(PLANE),TRIPTM(PLANE,FRSTSW,PAXPTR)
     -                ,TRIPTM(PLANE,FRSTSW,LAXPTR),POLTIME
 9030 FORMAT (30X,"BHEDWAY : ",F6.1," = ( ",I8," - ",I8," ) * "
     -       ,F5.3,"(POLTIME)")
C
      IF (DEBUG)
     -     WRITE(PRINTER,9040)FHEDWAY(PLANE),TRIPTM(PLANE,FRSTSW,PAXPTR)
     -                ,TRIPTM(PLANE,FRSTSW,OLDAX1),POLTIME
 9040 FORMAT (30X,"FHEDWAY : ",F6.1," = ( ",I8," - ",I8," ) * "
     -       ,F5.3,"(POLTIME)")
C
      IF (DEBUG) WRITE(PRINTER,9050)
     -           POINTER(PLANE,FRSTSW,FRONT), POINTER(PLANE,SCNDSW,FRONT),
     -           POINTER(PLANE,ZONESW,FRONT), POINTER(PLANE,FRSTSW,REAR),
     -           POINTER(PLANE,SCNDSW,REAR),  POINTER(PLANE,ZONESW,REAR)
 9050 FORMAT (30X,"FRONT POINTERS",5X,"SWITCH TYPE1 = ",I2,5X,"SWITCH ",
     -       "TYPE2 = ",I2,5X,"ZONE SWITCH = ",I2,/,30X,"REAR  POINTERS"
     -       ,20X,I2,20X,I2,19X,I2)
C
C***********************************************************************
C
C         IF THE PROGRAM WERE TO BE MODIFIED TO ALLOW SIMULTANEOUS ANALYSIS
C    OF ADJACENT LANES THE NECESSARY CODE WOULD BE INSERTED HERE.
C
C***********************************************************************
C
C
C         CLEAR LANE STORAGE FOR THE VEHICLE JUST PROCESSED UNLESS AXLES
C         REMAIN TO BE PROCESSED
C
      IF (AXLCNT(PLANE) .GT. 0) GO TO 9
C
      TAXSPD = AXSPEED(PLANE,AXFRNT)
C
      DO 560 I=1,AXFRNT
      AXSPEED(PLANE,I) = 0.0
      AXBS(PLANE,I) = 0.0
  560 CONTINUE
C
      AXSPEED(PLANE,AXLE1) = TAXSPD
C
      AXLCNT(PLANE) = 1
      ZONE(PLANE) = 0.0
      ENTRYT(PLANE) = 0.0
      VSPEED(PLANE) = 0.0
      VTYPE(PLANE) = 0
```

```
      GO TO 1
C
C*********************************************************************
C
C         END OF DATA HAS BEEN FOUND - PRINT THE SUMMARY REPORT
C
C*********************************************************************
C
   90 WRITE (PRINTER,9060) RUNNAME,SITENM,CONDITN,DATE,ISTRTM
 9060 FORMAT (1H1,34X,"RUN NAME        - ",8A10/
     -              35X,"SITE NAME       - ",8A10/
     -              35X,"CONDITION       - ",8A10/
     -              35X,"DATE            - ",8A10/
     -              35X,"START TIME      - ",I2.2,":",I2.2,":",I2.2//)
C
      WRITE (PRINTER,9100)
 9100 FORMAT (51X,"SUMMARY(VEHICLE TYPE BY LANE)"/
     -             51X,"*****************************"//
     -             2(5X,"VEHICLE TYPE",4X,"LANE1",3X,"LANE2",3X,"LANE3",3X,
     -             "LANE4",6X,"TOTAL",5X),/,1H+,
     -             2(4X,"____________",4X,"_____",3X,"_____",3X,"_____",3X
     -             ,"_____",6X,"_____",6X)/)
C
C         CALCULATE LANE TOTALS
C
      LANESUM = 0
      LMT = NOVTYPE/2
      DO 570 J=1,LMT
      SUM1 = 0
      SUM2 = 0
      JJ = J + LMT
C
      DO 580 I=1,NOLANE
      SUM1 = SUM1 + SUMMARY(I,J)
      SUM2 = SUM2 + SUMMARY(I,JJ)
      LNSUM(I) = LNSUM(I) + SUMMARY(I,J) + SUMMARY(I,JJ)
  580 CONTINUE
      WRITE (PRINTER,9110) VTPNAME(J),(SUMMARY(I,J),I=1,NOLANE),SUM1,
     -                     VTPNAME(JJ),(SUMMARY(I,JJ),I=1,NOLANE),SUM2
 9110 FORMAT (6X,A10,6X,4(I4,4X),1X,I6,11X,A10,6X,4(I4,4X),1X,I6,/)
  570 CONTINUE
C
      LANESUM = LNSUM(1) + LNSUM(2) + LNSUM(3) + LNSUM(4)
C
      WRITE (PRINTER,9120) (LNSUM(I),I=1,NOLANE), LANESUM
 9120 FORMAT (72X,55(1H-)/
     -        72X,"TOTAL     ",6X,I4,4X,I4,4X,I4,4X,I4,5X,I6)
      STOP
C
C*********************************************************************
C
C         ERROR MESSAGES
```

```
C
C*********************************************************************
C
  860 WRITE (PRINTER,8600) SEQ
 8600 FORMAT (1H1,"*** FATAL ERROR *** -- SINGLE AXLE VEHICLE "
     -        "ENCOUNTERED AT VEHICLE NUMBER",I5)
      GO TO 899
C
  865 WRITE(PRINTER,8650) SEQ
 8650 FORMAT (1H1,"*** FATAL ERROR *** -- QUEUE INSERTION OVERFLOW"
     -              " ENCOUNTERED AFTER VEHICLE NUMBER",I5)
      GO TO 899
C
  870 WRITE (PRINTER,8700) SEQ
 8700 FORMAT (1H1,"*** FATAL ERROR *** -- AXLE COUNT OVERFLOW FOR "
     -        "VEHICLE NUMBER",I5)
      GO TO 899
C
  875 WRITE (PRINTER,8750) SVAXSPD, SEQ
 8750 FORMAT (1H1,"*** FATAL ERROR *** -- AXLE SPEED BOUNDS",F6.1,
     -          "  FOR VEHICLE NUMBER",I5)
      GO TO 899
C
  880 WRITE (PRINTER,8800) SWID
 8800 FORMAT (1H1,"*** FATAL ERROR *** -- LANE DESIGNATION FOR SWITCH #"
     -          ,I2," IS INVALID -- CHECK YOUR $PARAM CARDS")
      STOP
C
  890 WRITE (PRINTER,8900) PSWTYPE
 8900 FORMAT (1H1,"*** FATAL ERROR *** -- TYPE DESIGNATION FOR SWITCH #"
     -          ,I2," IS INVALID -- CHECK YOUR $PARAM CARDS")
      STOP
C
  899 FRNTPTR = POINTER(PLANE,1,FRONT)
      REARPTR = POINTER(PLANE,1,REAR)
      WRITE (PRINTER,8990) ENTHR(PLANE), ENTMINT(PLANE),
     -   INT(ENTSEC(PLANE)), PLANE,PASS(PLANE), REARPTR, SVLAXTM
 8990 FORMAT (//,20X,"ENTRY TIME FOR LAST IDENTIFIABLE VEHICLE IS ",
     -        I2.1,1H:,I2.2,1H:,I2.2,
     -        //,28X,"QUEUE DUMP OF SWITCH TIMES FOR LANE ",I1,//
     -        7X,"PASS = ",L1,5X,"QUEUE POINTER = "I2,5X,"LAST USEABLE "
     -       "SWITCH TIME =",I10,//,7X,"QUEUE POSITION",
     -       9X,"SWITCH TYPE 1",15X,"SWITCH TYPE 2",/,1H+,6X,
     -       "______________",5X,"____________________",9X,
     -       "____________________",/)
C
      DO 901 J=FRNTPTR,AXLLMT
      WRITE (PRINTER,8991) J,
     -                   ((DIAGNZ(PLANE,I,J),DIAGNZ(PLANE,I+2,J)),I=1,2)
 8991 FORMAT (13X,I2,10X,I2,I18,8X,I2,I18)
  901 CONTINUE
      IF (FRNTPTR .EQ. 1) STOP
```

```
      FRNTPTR = FRNTPTR - 1
      DO 902 J=1,FRNTPTR
      WRITE (PRINTER,8991) J,
     -                     ((DIAGNZ(PLANE,I,J),DIAGNZ(PLANE,I+2,J)),I=1,2)
  902 CONTINUE
      STOP
      END
```

```
      SUBROUTINE DELETEQ(PSWTYPE,PLANE)
C
C***********************************************************************
C
C        THIS SUBROUTINE IS CALLED TO REMOVE THE ENTRY/EXIT SWITCH
C      ACTUATION TIME FROM EACH ENTRY/EXIT SWITCH QUEUE WHEN AN EXIT
C      SWITCH ACTIVATION IS FOUND.
C
C        THIS SUBROUTINE IS ALSO CALLED TO REMOVE THE ZONE SWITCH
C      ACTUATION TIMES FROM THE ZONE SWITCH QUEUE WHEN A NEW VEHICLE
C      IS DETECTED.
C
C        NOTE: THE QUEUE INSERTION FUNCTION WAS IMPLEMENTED AS AN INLINE
C      FUNCTION.
C
C***********************************************************************
C
      IMPLICIT INTEGER(A-Z)
      LOGICAL    DELFLAG(4,3)
      COMMON /JOINT/ REAR,AXLLMT,TRIPTM(4,3,40),POINTER(4,3,2)
     -       /DELETEQ/ FRONT,DELFLAG,PRINTER
C
      REARPTR = POINTER(PLANE,PSWTYPE,REAR)
      FRONPTR = POINTER(PLANE,PSWTYPE,FRONT)
C
C     CHECK FOR DELETION UNDERFLOW - IF SO, SET ERROR FLAG
C
      IF (REARPTR .NE. FRONPTR) GO TO 1
      DELFLAG(PLANE,PSWTYPE) = .TRUE.
      RETURN
C
C     CHECK FOR WRAP AROUND OF CIRCULAR QUEUE. IF NOT,INCREMENT POINTER.
C
    1 IF (FRONPTR .EQ. AXLLMT) GO TO 2
      POINTER(PLANE,PSWTYPE,FRONT) = POINTER(PLANE,PSWTYPE,FRONT) + 1
      RETURN
C
C     WRAP AROUND  - SET POINTER TO PHYSICAL BEGINNING OF STORAGE AREA.
C
    2 POINTER(PLANE,PSWTYPE,FRONT) = 1
      RETURN
      END
```

```
      SUBROUTINE PASSING (PSWTYPE)
C
C        SUBROUTINE TO EXCHANGE THE TYPE ONE AND TWO SWITCH DESIGNATIONS
C        TO HANDLE PASSING VEHICLES PROPERLY.
C
      INTEGER PSWTYPE
      IF (PSWTYPE .EQ. 3) RETURN
      IF (PSWTYPE .EQ. 1) GO TO 1
      PSWTYPE = 1
      RETURN
    1 PSWTYPE = 2
      RETURN
      END
```

1836

```
      PROGRAM CORECT(INPUT,OUTPUT,TAPE1,TAPE2,TAPE5=INPUT,TAPE6=OUTPUT)
      INTEGER SWTST, TMTST, SWTCH, TIME, SVTIM(10), NSWT
C
C
C*************************************************************************
C
C               TRAFFIC DATA PROGRAM TO CORRECT DATA FILE
C
C                               WRITTEN FOR
C
C          THE VIRGINIA HIGHWAY AND TRANSPORTATION RESEARCH COUNCIL
C
C                                   BY
3                              JERRY L. KORF
C                            RESEARCH SCIENTIST
C                                 6/1/79
C
C
C*************************************************************************
C
C
      LOGICAL DONE
      DATA DONE/.FALSE./, SVTIM/10*0/
C
C         THIS PROGRAM IS DESIGNED TO PROVIDE THE TDAS SYSTEM USER THE
C     MEANS TO MODIFY THE DATA FILE PRODUCED BY PROGRAM "RDTAPE". THE
C     CORRECTIVE POWER NECESSARY TO MAKE A DATA FILE ACCEPTABLE TO
C     PROGRAM PROCESS.
C
C         FOR DELETIONS ENTER THE SWITCH NUMBER RIGHT JUSTIFIED IN
C     COLUMNS 1-2 AND THE TIME RIGHT JUSTIFIED IN COLUMNS 3-20.
C
C         FOR INSERTIONS ENTER 99 IN COLUMNS 1-2, THE TIME RIGHT JUSTIFIED
C     INCOLUMNS 3-20, AND THE SWITCH NUMBER RIGHT JUSTIFIED IN COLUMNS
C     31-32.
C
      WRITE (6,2010)
 2010 FORMAT (1H1)
C
C         READ A COMMAND CARD.
C
    1 READ (5,1000) SWTST, TMTST, NSWT
 1000 FORMAT (I2,I18,10X,I2)
      IF (EOF(5)) 88, 2
C
C         READ A DISC FILE DATA RECORD.
C
    2 READ (1,1000) SWTCH, TIME
      IF (EOF(1)) 99,3
C
C         WHEN COMMAND CARDS ARE EXHAUSTED DONE IS TRUE.
```

```
C
    3 IF (DONE) GO TO 4
      IF (TIME .LT. TMTST) GO TO 4
C
C        CHECK FOR INSERT COMMAND.
C
      IF (SWTST .EQ. 99) GO TO 5
      IF (TIME .GT. TMTST) GO TO 6
      IF (SWTCH .NE. SWTST) GO TO 4
C
C        DELETION FOUND.
C
      WRITE (6,2000) SWTCH, TIME
 2000 FORMAT (10X,I2,I18,5X,"RECORD FOUND AND DELETED")
      GO TO 1
C
C        CONTINUE SEARCHING DATA FILE.
C
    4 WRITE (2,1000) SWTCH, TIME
      GO TO 2
C
C        INSERTION LOCATION FOUND  -  INSERT RECORD.
C
    5 WRITE (2,1000) NSWT, TMTST
      WRITE (6,2004) NSWT, TMTST
 2004 FORMAT (10X,I2,I18,5X,"RECORD INSERTED")
C
    7 READ (5,1000) SWTST, TMTST, NSWT
      IF (EOF(5)) 77, 3
C
   77 DONE = .TRUE.
      GO TO 4
C
C        INCORRECT TIME OR SWITCH NUMBER ON DETETION RECORD.
C
    6 WRITE (6,2006) SWTST, TMTST
 2006 FORMAT (" WARNING  ",I2,I18,5X,"NO MATCH FOUND FOR THIS RECORD")
      GO TO 7
C
   88 DONE = .TRUE.
      GO TO 2
C
   99 STOP
      END
```

1838

```
      PROGRAM REPORT(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
C
C                                    INPUT DISK FILE
C
     -                                    TAPE1, TAPE2)
C
C***********************************************************************
C
C             TRAFFIC DATA PROGRAM TO PRODUCE TRAFFIC FLOW REPORTS
C
C                                WRITTEN FOR
C
C           THE VIRGINIA HIGHWAY AND TRANSPORTATION RESEARCH COUNCIL
C
C                                    BY
C                                JERRY L. KORF
C                             RESEARCH SCIENTIST
C
C                                    AND
C
C                                WOON-HO SONG
C                             GRADUATE ASSISTANT
C                                  4/1/77
C
C                       REVISED 11/20/77 BY J. L. KORF
C                       REVISED  5/18/79 BY J. L. KORF
C
C***********************************************************************
C
C
C        'REPORT' CONSTRUCTS THE FOLLOWING TRAFFIC FLOW DATA TABLES
C     FROM THE DISK FILES GENERATED BY 'PROCESS'.
C
C          TRAFFIC VOLUME TABLE: THIS TABLE INDICATES TRAFFIC VOLUMES FOR
C               EACH LATERAL PLACEMENT ZONE AND ITS PERCENTAGE OF THE LANE
C               VOLUME FOR EACH VEHICLE TYPE. THE ORIGINAL 45 VEHICLE
C               TYPES ARE MAPPED INTO 5 CATAGORIES; CARS, CARS PULLING
C               TRAILERS, TRUCKS, TRACTOR-TRAILERS, AND OTHERS VIA THE
C               ARRAY 'VMASK' AND ITS ASSOCIATED DATA STATEMENT.
C
C          VEHICLE SPEED TABLE: THIS TABLE INDICATES THE AVERAGE,
C               STANDARD DEVIATION, AND 85 PERCENTILE FOR THE SPEEDS
C               OF EACH VEHICLE TYPE BY LANE AND ZONE.
C
C          HEADWAY INFORMATION TABLES: ONE OF THESE TABLES SHOWS THE
C               AVERAGE, STANDARD DEVIATION, AND THE MEDIAN OF THE
C               HEADWAY DISTRIBUTION OF EACH LANE. THE OTHER TABLE SHOWS
C               THE HEADWAY DISTRIBUTION BY TIME INTERVALS IN SECONDS.
C               NOTE THAT THE LAST CATEGORY INCLUDES ALL HEADWAYS GREATER
C               THAN 59 SECONDS.
C
```

```
C          QUEUE INFORMATION TABLES: ONE OF THESE TABLES SHOWS THE NUMBER
C                OF QUEUES ENCOUNTERED, THE AVERAGE AND STANDARD DEVIATION
C                OF THE NUMBER OF VEHICLES IN THE QUEUES, THE AVERAGE AND
C                STANDARD DEVIATION OF THE QUEUE SPEED DISTRIBUTION.
C                THE OTHER TABLE SHOWS THE FREQUENCY OF VEHICLE TYPE IN
C                EACH OF THE FIRST FIVE QUEUE POSITIONS.
C
C          THIS PROGRAM IS DESIGNED TO DEAL WITH UP TO FIVE DATA
C     COLLECTION SITES WHERE THERE ARE A MAXIMUM OF FOUR LANES PER SITE.
C     IT SHOULD BE FURTHER NOTED THAT PRESENTLY THE HARDWARE IS LIMITED
C     THIRTY TAPE SWITCHES.
C
C       'REPORT' ASSUMES THE INPUT IS PRESENTED IN VEHICLE ENTRY-TIME
C     ORDER.
C
C*************************************************************************
C
C          DECLARE VARIABLES AND INITIALIZE THEM VIA DATA STATEMENTS.
C
C*************************************************************************
C
      IMPLICIT INTEGER(A-Z)
C
      REAL      QCUTOFF(5),ZNSIZE,ZNWIDTH
     -         ,LNSTAT(44,6,20),QSTAT(2,10,20)
     -         ,PLACE,ENTTIME,VSPEED,FHDWY,BHDWY
     -         ,HDWY
     -         ,HCUTOFF
     -         ,VOLUMMD,VOLUM85
     -         ,MLFCTR, FTFCTR
     -         ,DIVISR
C
      INTEGER   STARTTM,ENDTM,FRSTLN,LASTLN
C
C          INPUT PARAMETER VARIABLES FOR HEADING INFORMATION
C
      INTEGER   RUNNAME(8),SITENM(8),CONDITN(8),DATE(8)
C
      INTEGER   SPDFQ(44,6,80),HDWYFQ(2,6,60),QPOSITN(2,25,5)
     -         ,POSTIDX(4,5),VMASK(40),LNTOTS(4),INFILE,SVTYPE(4,5)
C
      LOGICAL   BKHDWY, FIRSTQ(4,5), METRIC, METRC
C
C          PROGRAM CONTROL PARAMETER LIST
C
C
      NAMELIST /PARAM/ STARTTM,ENDTM,QCUTOFF,ZNSIZE,ZNWIDTH,BKHDWY
     -                ,FRSTLN,LASTLN,METRIC
C
C          DEFAULT INPUT PARAMETER VALUES FOR PROCESSING CONTROL
C
      DATA      STARTTM/000000/
```

```
     -          ,ENDTM/235959/
     -          ,QCUTOFF/6.0,0.0,0.0,0.0,0.0/
     -          ,ZNSIZE/0.0/
     -          ,ZNWIDTH/0.0/
     -          ,BKHDWY/.FALSE./
     -          ,METRIC/.FALSE./
     -          ,FRSTLN/1/
     -          ,LASTLN/1/
C
      DATA      LANE1/1/, LNTOTS/1,12,23,34/
     -          ,VEHTOT,CAR,CARTRL,TRUCK,TT,OTHERS/ 6,1,2,3,4,5/
     -          ,VOLUME,VOLPRCT,VOLSUM,VOLSQ,VOLAVG,VOLSD
     -          ,SPDSUM,SPDSQ,SPDAVG,SPDMD,SPD85,SPDSD
     -          ,HDWYSUM,HDWYSQ,HDWYAVG,HDWYMD,HDWY85,HDWYSD
     -          ,NOQ
     -               /1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,20/
C
      DATA      LN2DISP/10/
     -          ,SPDFQSZ,HDFQSZ/80,60/
     -          ,BSPSTQ,BSQTOT/0,5/
     -          ,HCUTOFF/60.0/
     -          ,FIRSTQ/20*.TRUE./
     -          ,MLFCTR,FTFCTR/2*1.0/
     -          ,LUNITS,SUNITS/"INCHES","(MPH)"/
     -          ,DIVISR/12.0/
C
      DATA      READER,PRINTER/5,6/
     -          ,INTAPE /1/
     -          ,NOVTYPE/5/
     -          ,MAXQSEQ/10/
     -          ,NOQCOFF/5/
     -          ,MAXPOTN/25/
     -          ,NOPOSTN/5/
     -          ,VMASK/5,5,1,2,2,1,2,2,17*3,7*4,2*3,6*5/
C
C
C         READ INPUT PARAMETERS FROM CARD READER
C
      READ (READER,PARAM)
      IF (EOF(READER)) 1,2
C
    1 WRITE (PRINTER,*) "PARAMETERS MISSING - ZNSIZE AND/OR ZNWIDTH"
      STOP
C
    2 IF (ZNSIZE .LE. 0.0 .OR. ZNWIDTH .LE. 0.0) GO TO 1
C
      IF (LASTLN .GT. 4) LASTLN = 4
      IF (FRSTLN .GT. 4) FRSTLN = 4
      ZNLNLMT = LNTOTS(LASTLN) + 10
      LIMIT = HDWYSD
C
C         CLEAR STORAGE AREAS
```

```
C
      DO 120 ZNTYPE=1,ZNLNLMT
      DO 120 VEHTYPE=1,VEHTOT
      DO 120 STATICS=1,20
             LNSTAT(ZNTYPE,VEHTYPE,STATICS) = 0.0
  120 CONTINUE
C
      DO 123 ZNTYPE=1,ZNLNLMT
      DO 123 VEHTYPE=1,VEHTOT
      DO 123 SPDIDX=1,SPDFQSZ
             SPDFQ(ZNTYPE,VEHTYPE,SPDIDX)=0
  123 CONTINUE
C
      DO 127 ILANE=FRSTLN,LASTLN
      DO 127 VEHTYPE=1,VEHTOT
      DO 127 HDWYIDX=1,HDFQSZ
             HDWYFQ(ILANE,VEHTYPE,HDWYIDX)=0
  127 CONTINUE
C
      DO 130 ILANE=FRSTLN,LASTLN
      DO 130 QTYPE=1,MAXQSEQ
      DO 130 STATICS=1,NOQ
             QSTAT(ILANE,QTYPE,STATICS) = 0.0
  130 CONTINUE
C
      DO 140 ILANE=FRSTLN,LASTLN
         DO 141 QSEQ=1,NOQCOFF
  141    POSTIDX(ILANE,QSEQ)=0
      DO 140 POSITN=1,MAXPOTN
      DO 140 VEHTYPE=1,NOVTYPE
             QPOSITN(ILANE,POSITN,VEHTYPE) = 0
  140 CONTINUE
C
C        PREPARE FOR OUTPUT BY ADVANCING TO NEW PAGE
C
      WRITE (PRINTER,5050)
 5050 FORMAT (1H1)
C
C        READ HEADING INFORMATION FROM DISK FILE THEN PRINT HEADING
C     INFORMATION FOLLOWED BY CONTROL PARAMETERS THAT ARE IN EFFECT.
C
    3 READ (INTAPE,2020) RUNNAME,SITENM,CONDITN,DATE,METRC
 2020 FORMAT (8A10/8A10/8A10/8A10,L1)
C
      IF (EOF(INTAPE)) 20,4
    4 IF (INTAPE .GT. 1) GO TO 9
      WRITE (PRINTER,5060) RUNNAME,SITENM,CONDITN,DATE
 5060 FORMAT (30X,"HEADING INFORMATION"/
     -        1H+,29X,"___________________"///
     -        30X,"RUN NAME  - ",8A10//
     -        30X,"SITE NAME - ",8A10//
     -        30X,"CONDITION - ",8A10//
```

```
     -          30X,"DATE       - ",8A10//////)
C
      WRITE (PRINTER,5070) STARTTM,ENDTM,QCUTOFF,METRIC,ZNSIZE,ZNWIDTH,
     -                     BKHDWY,FRSTLN,LASTLN
 5070 FORMAT (30X,"CONTROL PARAMETERS"/
     -        1H+,29X,"__________________"///
     -        30X,"$PARAM"//
     -        30X,"STARTTM   = ",I6//
     -        30X,"ENDTM     = ",I6//
     -        30X,"QCUTOFF   = ",5(F5.1,",")//
     -        30X,"METRIC    = ",L1//
     -        30X,"ZNSIZE    = ",F5.1//
     -        30X,"ZNWIDTH   = ",F5.1//
     -        30X,"BKHDWY    = ",L1//
     -        30X,"FRSTLN    = ",I2,5X,"LASTLN    = ",I2,//
     -        30X,"$END"//////)
C
C        DECOMPOSE STARTING TIME AND ENDING TIME INTO HOUR, MINUTE, AND
C     SECOND.
C
      BEGHR = STARTTM/10000
      BEGMINT= (STARTTM-BEGHR*10000)/100
      BEGSEC = STARTTM-BEGHR*10000-BEGMINT*100
C
      ENDHR = ENDTM/10000
      ENDMINT= (ENDTM-ENDHR*10000)/100
      ENDSEC = ENDTM-ENDHR*10000-ENDMINT*100
C
C        DETERMINE PROPER UNITS AND SET APPROPRIATE CONVERSION FACTORS.
C
      IF (METRIC) SUNITS = "(KM/H)"
      IF (METRIC) LUNITS = "CENTIMETER"
      IF (METRIC) DIVISR = 1.0
      IF (METRIC .AND. METRC) GO TO 8
      IF (.NOT. METRIC .AND. .NOT. METRC) GO TO 8
      IF (METRIC .AND. .NOT. METRC) GO TO 7
      MLFCTR = .621371192
      FTFCTR = 0.03280839895
      GO TO 8
C
    7 MLFCTR = 1.609344
      FTFCTR = 30.4800
C
C        CALCULATE NUMBER OF ZONES PER LANE BY USING ZONE LANE WIDTH AND
C     ZONE SIZE.
C
    8 NOLNZN = IFIX(ZNWIDTH/ZNSIZE + 1.0)
      IF (NOLNZN .GT. 10) NOLNZN = 10
C
C        READ VEHICLE DATA FROM INPUT DISK FILE.
C
    9 READ (INTAPE,3100) SITE,LANE,PLACE,VEHTYPE,ENTTIME,VSPEED
```

```
-                                   ,FHDWY,BHDWY
 3100 FORMAT (2I1,F4.1,I2,F8.1,F4.1,2F5.1)
      IF (EOF(INTAPE)) 20,10
C
   10 IF (LANE .LT. FRSTLN .OR. LANE .GT. LASTLN) GO TO 9
C
C          SELECT ONLY THOSE VEHICLES THAT LIE WITHIN THE PERIOD TO BE
C     ANALYZED.
C
      IF (ENTTIME .LT. STARTTM) GO TO 9
      IF (ENTTIME .GT. ENDTM) GO TO 20
C
C*********************************************************************
C
C     THIS RECORD IS TO BE USED IN THE ANAYSIS STORE ITS CHARACTERISTICS.
C
C*********************************************************************
C
C          FIRST CONVERT SPEED AND LATERAL PLACEMENT TO PROPER UNITS.
C
      VSPEED = VSPEED * MLFCTR
      PLACE = PLACE * FTFCTR
C
      VTYPE=VMASK(VEHTYPE)
C
C      LANE LATERAL PLACEMENT INFORMATION IS STORED BY OFFSETTING TO THE
C     APPROPRIATE AREA OF THE ARRAY.
C
      ZONE = IFIX(PLACE/(ZNSIZE/DIVISR) + 1.0)
      IF (ZONE .GT. NOLNZN) ZONE = NOLNZN
      ZONE = ZONE + LNTOTS(LANE)
C
      LNSTAT(ZONE,VTYPE,VOLUME) = LNSTAT(ZONE,VTYPE,VOLUME) + 1.0
      LNSTAT(ZONE,VTYPE,SPDSUM) = LNSTAT(ZONE,VTYPE,SPDSUM) + VSPEED
      LNSTAT(ZONE,VTYPE,SPDSQ) = LNSTAT(ZONE,VTYPE,SPDSQ) + VSPEED**2
      SPDIDX = VSPEED + 1.0
      SPDFQ(ZONE,VTYPE,SPDIDX) = SPDFQ(ZONE,VTYPE,SPDIDX) + 1
C
      HDWY = FHDWY
      IF (BKHDWY) HDWY = BHDWY
C
      LNSTAT(ZONE,VTYPE,HDWYSUM) = LNSTAT(ZONE,VTYPE,HDWYSUM) + HDWY
      LNSTAT(ZONE,VTYPE,HDWYSQ) = LNSTAT(ZONE,VTYPE,HDWYSQ) + HDWY**2
      HDWYIDX = IFIX(HDWY + 1.0)
      IF (HDWY .GT. HDFQSZ) HDWYIDX = HDFQSZ
      HDWYFQ(LANE,VTYPE,HDWYIDX) = HDWYFQ(LANE,VTYPE,HDWYIDX) + 1
C
C
C         CONSTRUCT EACH QUEUE BY SCANNING EACH QUEUE CUTOFF TIME
C
      I = 1
   11 BSIDX = (I-1) * NOPOSTN
```

```
      QTOT = BSQTOT + I
      IF (QCUTOFF(I) .LE. 0.0) GO TO 19
      IF (HDWY .GT. QCUTOFF(I)) GO TO 15
      QSTAT(LANE,I,VOLUME) = QSTAT(LANE,I,VOLUME) + 1
      QSTAT(LANE,I,SPDSUM) = QSTAT(LANE,I,SPDSUM) + VSPEED
      QSTAT(LANE,I,HDWYSUM) = QSTAT(LANE,I,HDWYSUM) + HDWY
      POSTIDX(LANE,I) = POSTIDX(LANE,I) + 1
      POSTSEQ = BSIDX + POSTIDX(LANE,I)
      IF (POSTIDX(LANE,I) .LE. NOPOSTN)
     -    QPOSITN(LANE,POSTSEQ,VTYPE) = QPOSITN(LANE,POSTSEQ,VTYPE) + 1
      GO TO 19
C
C         CHECK FOR FIRST QUEUE AND SINGLE VEHICLE QUEUE - IF SO, DISCARD
C     THIS VEHICLE AND RESET ELSE SUMMARIZE THE DATA FROM THE PREVIOUS
C     QUEUE AND THEN RESET.
C
   15 IF (FIRSTQ(LANE,I)) GO TO 17
      IF (QSTAT(LANE,I,VOLUME) .EQ. 1) GO TO 18
      SVDTYPE = SVTYPE(LANE,I)
      POSTSEQ = BSIDX + 1
      QPOSITN(LANE,POSTSEQ,SVDTYPE) = QPOSITN(LANE,POSTSEQ,SVDTYPE) + 1
      QSTAT(LANE,I,SPDAVG) = QSTAT(LANE,I,SPDSUM)/QSTAT(LANE,I,VOLUME)
      QSTAT(LANE,I,HDWYAVG) = QSTAT(LANE,I,HDWYSUM)/QSTAT(LANE,I,VOLUME)
      QSTAT(LANE,QTOT,VOLSUM) =
     -           QSTAT(LANE,QTOT,VOLSUM) + QSTAT(LANE,I,VOLUME)
      QSTAT(LANE,QTOT,VOLSQ)=
     -            QSTAT(LANE,QTOT,VOLSQ) + QSTAT(LANE,I,VOLUME)**2
      QSTAT(LANE,QTOT,NOQ) = QSTAT(LANE,QTOT,NOQ) + 1
      QSTAT(LANE,QTOT,SPDSUM) =
     -           QSTAT(LANE,QTOT,SPDSUM) + QSTAT(LANE,I,SPDAVG)
      QSTAT(LANE,QTOT,SPDSQ) =
     -            QSTAT(LANE,QTOT,SPDSQ) + QSTAT(LANE,I,SPDAVG)**2
      GO TO 18
C
   17 FIRSTQ(LANE,I) = .FALSE.
C
C        SAVE PRESENT VEHICLE INFO IN PREPARATION FOR NEW QUEUE
C
   18 QSTAT(LANE,I,VOLUME) = 1
      QSTAT(LANE,I,SPDSUM) = VSPEED
      QSTAT(LANE,I,HDWYSUM) = HDWY
      POSTIDX(LANE,I) = 1
      SVTYPE(LANE,I) = VTYPE
C
   19 I = I + 1
      IF (I .GT. 5) GO TO 9
      GO TO 11
C
C        PREPARE TOTALS BY LATERAL PLACEMENT ZONE
C
   20 INTAPE = INTAPE + 1
      IF (INTAPE .LE. NOLANS) GO TO 3
```

```
      LIMIT = HDWYSD
      DO 210 ZNTYPE=1,ZNLNLMT
      DO 210 STATICS=VOLUME,LIMIT
      DO 210 VEHTYPE=1,NOVTYPE
      LNSTAT(ZNTYPE,VEHTOT,STATICS) = LNSTAT(ZNTYPE,VEHTOT,STATICS)
     -                               +LNSTAT(ZNTYPE,VEHTYPE,STATICS)
  210 CONTINUE
C
      DO 213 ZNTYPE=1,ZNLNLMT
      DO 213 SPDIDX=1,SPDFQSZ
      DO 213 VEHTYPE=1,NOVTYPE
         SPDFQ(ZNTYPE,VEHTOT,SPDIDX) =
     -      SPDFQ(ZNTYPE,VEHTOT,SPDIDX) + SPDFQ(ZNTYPE,VEHTYPE,SPDIDX)
  213 CONTINUE
C
      DO 215 ILANE=FRSTLN,LASTLN
      DO 215 HDWYIDX=1,HDFQSZ
      DO 215 VEHTYPE=1,NOVTYPE
         HDWYFQ(ILANE,VEHTOT,HDWYIDX) =
     -      HDWYFQ(ILANE,VEHTOT,HDWYIDX) + HDWYFQ(ILANE,VEHTYPE,HDWYIDX)
  215 CONTINUE
C
C        TOTALS FOR ALL LANES
C
      DO 230 II=FRSTLN,LASTLN
      LANETOT = LNTOTS(II)
      ZNLLMT = LANETOT + 1
      ZNUPLMT = LANETOT + 10
C
      DO 220 STATICS=VOLUME,LIMIT
      DO 220 VEHTYPE=1,VEHTOT
      DO 220 IZONE=ZNLLMT,ZNUPLMT
      LNSTAT(LANETOT,VEHTYPE,STATICS) = LNSTAT(LANETOT,VEHTYPE,STATICS)
     -                                + LNSTAT(IZONE,VEHTYPE,STATICS)
  220 CONTINUE
C
      DO 225 SPDIDX=1,SPDFQSZ
      DO 225 VEHTYPE=1,VEHTOT
      DO 225 IZONE=ZNLLMT,ZNUPLMT
         SPDFQ(LANETOT,VEHTYPE,SPDIDX) = SPDFQ(LANETOT,VEHTYPE,SPDIDX)
     -                     +  SPDFQ(IZONE,VEHTYPE,SPDIDX)
  225 CONTINUE
  230 CONTINUE
C
C*********************************************************************
C
C        CALCULATE FLOW CHARACTERISTICS BY LANE, LATERAL PLACEMENT ZONE,
C    AND VEHICLE TYPE.
C
C*********************************************************************
C
      DO 250 II=FRSTLN,LASTLN
```

```
      LANETOT = LNTOTS(II)
      ZNLLMT = LANETOT + 1
      ZNUPLMT = LANETOT + 10
      IF (LNSTAT(LANETOT,VEHTOT,VOLUME) .EQ. 0.0) GO TO 250
C
C     CALCULATE LATERAL PLACEMENT ZONE PERCENTAGES FOR EACH VEHICLE TYPE
C
      DO 245 J=1,VEHTOT
      LNSTAT(LANETOT,J,VOLPRCT) =
     -   (LNSTAT(LANETOT,J,VOLUME)/LNSTAT(LANETOT,VEHTOT,VOLUME))*100.0
      DO 240 I=ZNLLMT,ZNUPLMT
      IF (LNSTAT(LANETOT,J,VOLUME) .EQ. 0.0) GO TO 240
      LNSTAT(I,J,VOLPRCT) = LNSTAT(I,J,VOLUME)/LNSTAT(LANETOT,J,VOLUME)
     -                                                        * 100.0
  240 CONTINUE
  245 CONTINUE
  250 CONTINUE
C
C        DETERMINE AVERAGES FOR VEHICLE SPEEDS AND HEADWAYS
C
      DO 405 I=1,ZNLNLMT
      DO 405 J=1,VEHTOT
      IF (LNSTAT(I,J,VOLUME) .EQ. 0) GO TO 405
      LNSTAT(I,J,SPDAVG) = LNSTAT(I,J,SPDSUM)/LNSTAT(I,J,VOLUME)
      LNSTAT(I,J,HDWYAVG) = LNSTAT(I,J,HDWYSUM)/LNSTAT(I,J,VOLUME)
C
C        DETERMINE STANDARD DEVIATION OF VEHICLE SPEEDS AND HEADWAYS
C
      IF (LNSTAT(I,J,VOLUME) .LE. 1) GO TO 405
      LNSTAT(I,J,SPDSD) =
     -   (LNSTAT(I,J,SPDSQ) - LNSTAT(I,J,SPDSUM)**2/LNSTAT(I,J,VOLUME))
     -                           /(LNSTAT(I,J,VOLUME) - 1.0)
      LNSTAT(I,J,HDWYSD) =
     -  (LNSTAT(I,J,HDWYSQ) - LNSTAT(I,J,HDWYSUM)**2/LNSTAT(I,J,VOLUME))
     -                          /(LNSTAT(I,J,VOLUME) - 1.0)
      LNSTAT(I,J,SPDSD) = SQRT(LNSTAT(I,J,SPDSD))
      LNSTAT(I,J,HDWYSD) = SQRT(LNSTAT(I,J,HDWYSD))
C
C     DETERMINE MEDIAN AND 85TH PERCENTILE OF VEHICLE SPEEDS AND HEADWAYS
      VOLUMMD = LNSTAT(I,J,VOLUME)*0.50
      VOLUM85 = LNSTAT(I,J,VOLUME)*0.85
      CUMULAT = 0
C
      DO 400 SPDIDX=1,SPDFQSZ
      CUMULAT = CUMULAT + SPDFQ(I,J,SPDIDX)
      IF (CUMULAT .GE. VOLUMMD .AND. LNSTAT(I,J,SPDMD) .EQ. 0.0)
     -                           LNSTAT(I,J,SPDMD) = SPDIDX
      IF (CUMULAT .GE. VOLUM85 .AND. LNSTAT(I,J,SPD85) .EQ. 0.0)
     -                           LNSTAT(I,J,SPD85) = SPDIDX
  400 CONTINUE
  405 CONTINUE
C
```

```
      DO 410 I=FRSTLN,LASTLN
      ZNTOT = LNTOTS(I)
      DO 410 J=1,VEHTOT
      VOLUMMD = LNSTAT(ZNTOT,J,VOLUME)*0.50
      VOLUM85 = LNSTAT(ZNTOT,J,VOLUME)*0.85
      CUMULAT = 0
      DO 410 HDWYIDX=1,HDFQSZ
      CUMULAT = CUMULAT + HDWYFQ(I,J,HDWYIDX)
      IF (CUMULAT .GE. VOLUMMD .AND. LNSTAT(ZNTOT,J,HDWYMD) .EQ. 0.0)
     -                         LNSTAT(ZNTOT,J,HDWYMD) = HDWYIDX
      IF (CUMULAT .GE. VOLUM85 .AND. LNSTAT(ZNTOT,J,HDWY85) .EQ. 0.0)
     -                         LNSTAT(ZNTOT,J,HDWY85) = HDWYIDX
  410 CONTINUE
C
C*********************************************************************
C
C         CALCULATE QUEUE STATISTICS FOR EACH LANE AND CUTOFF VALUE
C
C*********************************************************************
C
      DO 430 I=FRSTLN,LASTLN
      DO 420 II=1,NOQCOFF
      J = BSQTOT + II
      IF (QCUTOFF(II) .EQ. 0.0) GO TO 420
      IF (QSTAT(I,J,NOQ) .EQ. 0) GO TO 420
C
C         DETERMINE AVERAGES FOR QUEUE SIZE AND SPEED
C
      QSTAT(I,J,VOLAVG) = QSTAT(I,J,VOLSUM) / QSTAT(I,J,NOQ)
      QSTAT(I,J,SPDAVG) = QSTAT(I,J,SPDSUM) / QSTAT(I,J,NOQ)
C
C         DETERMINE STANDARD DEVIATION FOR QUEUE SIZES AND SPEEDS
C
      IF (QSTAT(I,J,NOQ) .LE. 1) GO TO 420
      QSTAT(I,J,VOLSD) =
     -           (QSTAT(I,J,VOLSQ) - QSTAT(I,J,VOLSUM)**2/QSTAT(I,J,NOQ))
     -                      /(QSTAT(I,J,NOQ) - 1.0)
C
      QSTAT(I,J,SPDSD) =
     -           (QSTAT(I,J,SPDSQ) - QSTAT(I,J,SPDSUM)**2/QSTAT(I,J,NOQ))
     -                       /(QSTAT(I,J,NOQ) - 1.0)
C
      QSTAT(I,J,VOLSD) = SQRT(QSTAT(I,J,VOLSD))
      QSTAT(I,J,SPDSD) = SQRT(QSTAT(I,J,SPDSD))
  420 CONTINUE
  430 CONTINUE
C
C*********************************************************************
C
C         PRINT TABLES OF RESULTS
C
C*********************************************************************
```

```
C
C
C         PRINT GENERAL HEADING INFORMATION
C
      WRITE (PRINTER,6010) RUNNAME,SITENM,CONDITN,DATE
     -                    ,BEGHR,BEGMINT,BEGSEC,ENDHR,ENDMINT,ENDSEC
     -                    ,ZNSIZE,LUNITS,ZNWIDTH,LUNITS
 6010 FORMAT (1H1,34X,"RUN NAME          - ",8A10/
     -            35X,"SITE NAME         - ",8A10/
     -            35X,"CONDITION         - ",8A10/
     -            35X,"DATE              - ",8A10/
     -            35X,"PERIOD ANALYZED - ",I2.1,":",I2.2,":",I2.2,
     -                      "     TO     ",I2.1,":",I2.2,":",I2.2/
     -            35X,"ZONE SIZE         - ",F5.1,1X,A10,/
     -            35X,"ZONE LANE WIDTH - ",F5.1,1X,A10,///)
C
C         PRINT TRAFFIC VOLUME REPORT
C
      WRITE (PRINTER,6020)
 6020 FORMAT (51X,*T R A F F I C   V O L U M E*,//)
      WRITE (PRINTER,6030)
 6030 FORMAT (23X,"CARS",11X,"CAR-TRAILERS",10X,"TRUCKS"
     -        8X,"TRACTOR-TRAILERS",8X,"OTHERS",10X,"ALL VEHICLES"/,1H+,
     -       22X,"____",11X,"____________",10X,"______",
     -        8X,"________________",8X,"______",10X,"____________")
      WRITE (PRINTER,6040)
 6040 FORMAT (4X,"LANE",2X,"ZONE",3X,6("VOLUME   PERCENT   ")/,1H+,
     -        3X,"____",2X,"____",3X,6("______   _______   "))
C
      DO 715 ILANE=FRSTLN,LASTLN
      LANETOT = LNTOTS(ILANE)
      ZNLLMT = LANETOT + 1
      ZNUPLMT = LANETOT + 10
      WRITE (PRINTER,6050) ILANE,
     -        (INT(LNSTAT(LANETOT,J,VOLUME)),LNSTAT(LANETOT,J,VOLPRCT),
     -                                   J=1,VEHTOT)
 6050 FORMAT (/,6X,I1,10X,6(I5,5X,F5.1,4X)/)
      DO 710 I=ZNLLMT,ZNUPLMT
      IZONE = I - ZNLLMT + 1
  710 WRITE (PRINTER,6055) IZONE,
     -         (INT(LNSTAT(I,J,VOLUME)),LNSTAT(I,J,VOLPRCT),J=1,VEHTOT)
 6055 FORMAT (11X,I2, 4X,6(I5,5X,F5.1,4X)/)
  715 CONTINUE
C
C         PRINT VEHICLE SPEED TABLES
C
      WRITE (PRINTER,6010) RUNNAME,SITENM,CONDITN,DATE
     -                    ,BEGHR,BEGMINT,BEGSEC,ENDHR,ENDMINT,ENDSEC
     -                    ,ZNSIZE,LUNITS,ZNWIDTH,LUNITS
      WRITE (PRINTER,6110) SUNITS
 6110 FORMAT (51X,*V E H I C L E   S P E E D*,/,63X,A6,/)
      WRITE (PRINTER,6115)
```

```
 6115 FORMAT (21X,"CARS",12X,"CAR-TRAILERS",11X,"TRUCKS",
     -         9X,"TRACTOR-TRAILERS",9X,"OTHERS",11X,"ALL VEHICLES"/1H+,
     -         20X,"____",12X,"____________",11X,"______",
     -         9X,"________________",9X,"______",11X,"____________")
      WRITE (PRINTER,6120)
 6120 FORMAT (1X,"LANE",2X,"ZONE",4X,6("AVG.  S.D.  85%     ")/,1H+,
     -            "____",2X,"____",4X,6("____  ____  ___     "))
C
      DO 725 ILANE=FRSTLN,LASTLN
      LANETOT = LNTOTS(ILANE)
      ZNLLMT = LANETOT + 1
      ZNUPLMT = LANETOT + 10
      WRITE (PRINTER,6130) ILANE,
     -               (LNSTAT(LANETOT,J,SPDAVG),LNSTAT(LANETOT,J,SPDSD),
     -                   INT(LNSTAT(LANETOT,J,SPD85)),J=1,VEHTOT)
 6130 FORMAT (/,3X,I1,11X,6(F4.1,2X,F4.1,3X,I2,5X)/)
C
      DO 720 I=ZNLLMT,ZNUPLMT
      IZONE = I - ZNLLMT + 1
      WRITE (PRINTER,6135) IZONE,
     -                (LNSTAT(I,J,SPDAVG),LNSTAT(I,J,SPDSD),
     -                INT(LNSTAT(I,J,SPD85)),J=1,VEHTOT)
 6135 FORMAT (8X,I2, 5X,6(F4.1,2X,F4.1,3X,I2,5X)/)
  720 CONTINUE
  725 CONTINUE
C
      DO 735 ILANE=FRSTLN,LASTLN
      ZNTOT = LNTOTS(ILANE)
      IF (LNSTAT(ZNTOT,VEHTOT,VOLUME) .LE. 0) GO TO 735
C
C        PRINT HEADWAY SUMMARY INFORMATION FOR THIS LANE
C
      WRITE (PRINTER,6010) RUNNAME,SITENM,CONDITN,DATE
     -                  ,BEGHR,BEGMINT,BEGSEC,ENDHR,ENDMINT,ENDSEC
     -                  ,ZNSIZE,LUNITS,ZNWIDTH,LUNITS
C
      WRITE(PRINTER,6210)
 6210 FORMAT( 51X,"H E A D W A Y    I N F O R M A T I O N"//)
      HDWYID = "FRONT"
      IF (BKHDWY) HDWYID = "BACK"
      WRITE (PRINTER,6220) HCUTOFF,HDWYID
 6220 FORMAT (/,39X,*HEADWAY CUTOFF TIME* ,F5.1,* SECONDS*
     -             ,*  USING *,A5,* HEADWAY DATA *//)
      WRITE (PRINTER,6230)
 6230 FORMAT (76X,"H E A D W A Y S"/,1H+,
     -        75X,"_______________"//
     -       40X,"LANE",6X,"VOLUME",14X,"AVG.",6X,"S.D.",6X,"MEDIAN"/1H+
     -        ,39X,"____",6X,"______",14X,"____",6X,"____",6X,"______"/)
      WRITE (PRINTER,6240) ILANE
     -                            ,INT(LNSTAT(ZNTOT,VEHTOT,VOLUME))
     -                         ,LNSTAT(ZNTOT,VEHTOT,HDWYAVG)
     -                         ,LNSTAT(ZNTOT,VEHTOT,HDWYSD)
```

```
     -                            ,LNSTAT(ZNTOT,VEHTOT,HDWYMD)
 6240 FORMAT (42X,I1,7X,I5,14X,F5.1,5X,F5.1,6X,F5.1)
C
C        PRINT HEADWAY DISTRIBUTIONS IN ONE SECOND INTERVALS
C
      WRITE (PRINTER,6250)
 6250 FORMAT (/////39X,"DISTRIBUTION OF HEADWAYS BY TIME INTERVALS ",
     -                 "IN SECONDS"//)
      WRITE (PRINTER,6255)
 6255 FORMAT (12X, 4("TIME",7X,"VEHICLE",12X)/,1H+,
     -          11X, 4("____",7X,"_______",12X)/
     -          10X,4("INTERVAL",5X,"NUMBERS",10X)/,1H+,
     -          9X, 4("________",5X,"_______",10X)/)
C
      L = HDFQSZ/4
      DO 730 I=1,L
      WRITE (PRINTER,6260)
     -    (I-1),I,HDWYFQ(ILANE,VEHTOT,I),
     -    ((I-1 + K*L),(I + K*L),HDWYFQ(ILANE,VEHTOT,(I + K*L)),K=1,3)
 6260 FORMAT (11X,4(I2," - ",I2,5X,I5,13X))
  730 CONTINUE
  735 CONTINUE
C
C        PRINT QUEUE STATISTICS TABLE FOR THIS LANE
C
      DO 760 QSEQ=1,NOQCOFF
      QTOT = BSQTOT + QSEQ
      IF (QCUTOFF(QSEQ) .EQ. 0.0) GO TO 760
C
      WRITE (PRINTER,6010) RUNNAME,SITENM,CONDITN,DATE
     -                    ,BEGHR,BEGMINT,BEGSEC,ENDHR,ENDMINT,ENDSEC
     -                    ,ZNSIZE,LUNITS,ZNWIDTH,LUNITS
C
      WRITE (PRINTER,6310)
 6310 FORMAT (51X,"Q U E U E   I N F O R M A T I O N"//)
      WRITE (PRINTER,6315) QCUTOFF(QSEQ)
 6315 FORMAT (/,39X,*QUEUE CUTOFF TIME*,F5.1,* SECONDS*//)
      WRITE (PRINTER,6320)
 6320 FORMAT (40X,"QUEUE LENGTH",8X,"QUEUE  SPEED"/,1H+,
     -          39X,"____________",8X,"_____________"/
     -          10X,"LANE",6X,"NO. OF QUEUES",
     -          7X,"AVG.",4X,"S.D.",8X,"AVG.",4X,"S.D."/,1H+,
     -          9X,"____",6X,"_____________",
     -        7X,"____",4X,"____",8X,"____",4X,"____"/)
C
      DO 740 LANE=FRSTLN,LASTLN
      WRITE (PRINTER,6330)
     -          LANE,INT(QSTAT(LANE,QTOT,NOQ))
     -          ,QSTAT(LANE,QTOT,VOLAVG),QSTAT(LANE,QTOT,VOLSD)
     -          ,QSTAT(LANE,QTOT,SPDAVG),QSTAT(LANE,QTOT,SPDSD)
 6330 FORMAT (12X,I1,10X,I4,13X,F4.1,4X,F4.1,8X,F4.1,4X,F4.1)
  740 CONTINUE
```

```
C
      WRITE (PRINTER,6340)
 6340 FORMAT (///10X,"LANE",2X,"POSITION",6X,"CAR ",6X,"CAR-TRL ",
     -          6X,"TRUCK",8X,"TRACTOR TRAILER",7X,"OTHERS"/,1H+,
     -          9X,"____",2X,"________",6X,"___ ",6X,"_______ ",
     -          6X,"_____",8X,"_______________",7X,"______"/)
C
      DO 750 LANE=FRSTLN,LASTLN
      WRITE(PRINTER,6345) LANE
 6345 FORMAT (12X,I1)
      DO 750 POSITN=1,NOPOSTN
      BSIDX = (QSEQ - 1)*NOPOSTN
      POSTSEQ = BSIDX + POSITN
      WRITE (PRINTER,6350) POSITN
     -                  ,(QPOSITN(LANE,POSTSEQ,K),K=CAR,OTHERS)
 6350 FORMAT (13X,6X,I1,11X,I2,2(10X,I2),2(16X,I2))
  750 CONTINUE
  760 CONTINUE
C
      STOP
      END
```

1950

APPENDIX C

FIELD DATA FORMS

1854

MOUNTAIN PAVEMENT MARKING PROJECT

Field Data Collection Form

Date ______________________ PROJECT STATUS ______________________

Route ______________ County ______________ District ______________

Station Location ______________________ Mile Post ______________

Pavement width ______________ Shoulder width ______________

Horizontal alignment ______________ Vertical alignment ______________

Posted speed limit ________ Type of area ______________________

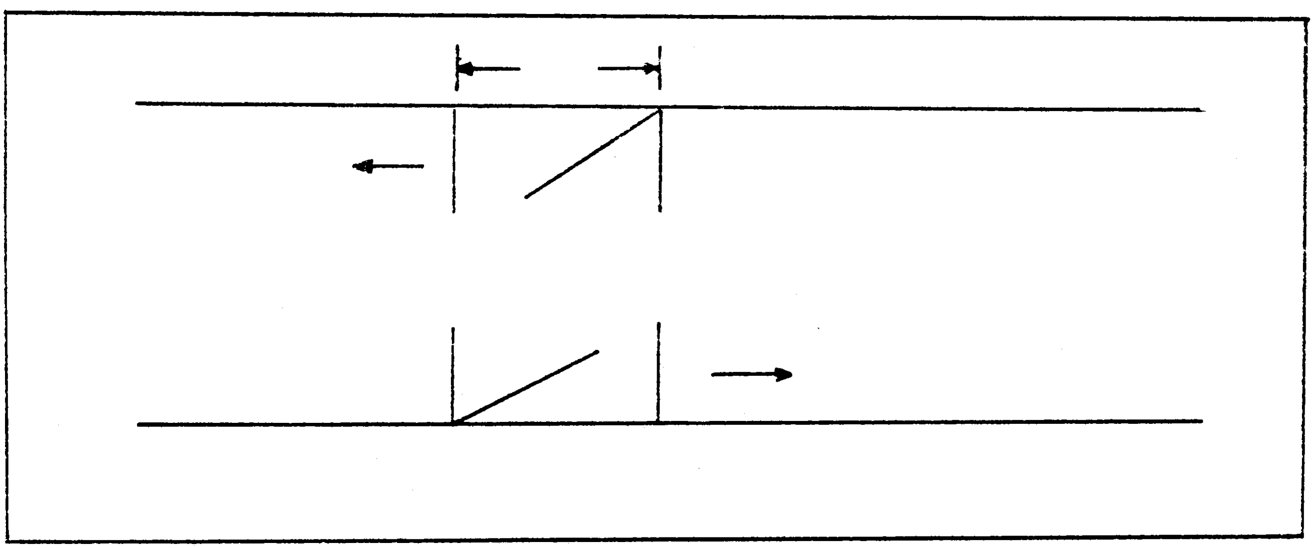
Signs related to passing ______________________________

Pavement markings: Centerline ______________________________

(Indicate on sketch) ______________________________

Edgeline ______________________________

Weather ______________________ Temperature ______________



Tape No.	Date	Recording Time Start	Recording Time Stop	Number of Descriptive Pages
______	______	______	______	______
______	______	______	______	______
______	______	______	______	______
______	______	______	______	______
______	______	______	______	______

1855

TDAS FIELD FORM

PROJECT TITLE ____________________

SITE NAME ____________________

DATE ____________________ SITE NUMBER ____________________

WEATHER CONDITIONS ____________________ TEMPERATURE ____________________

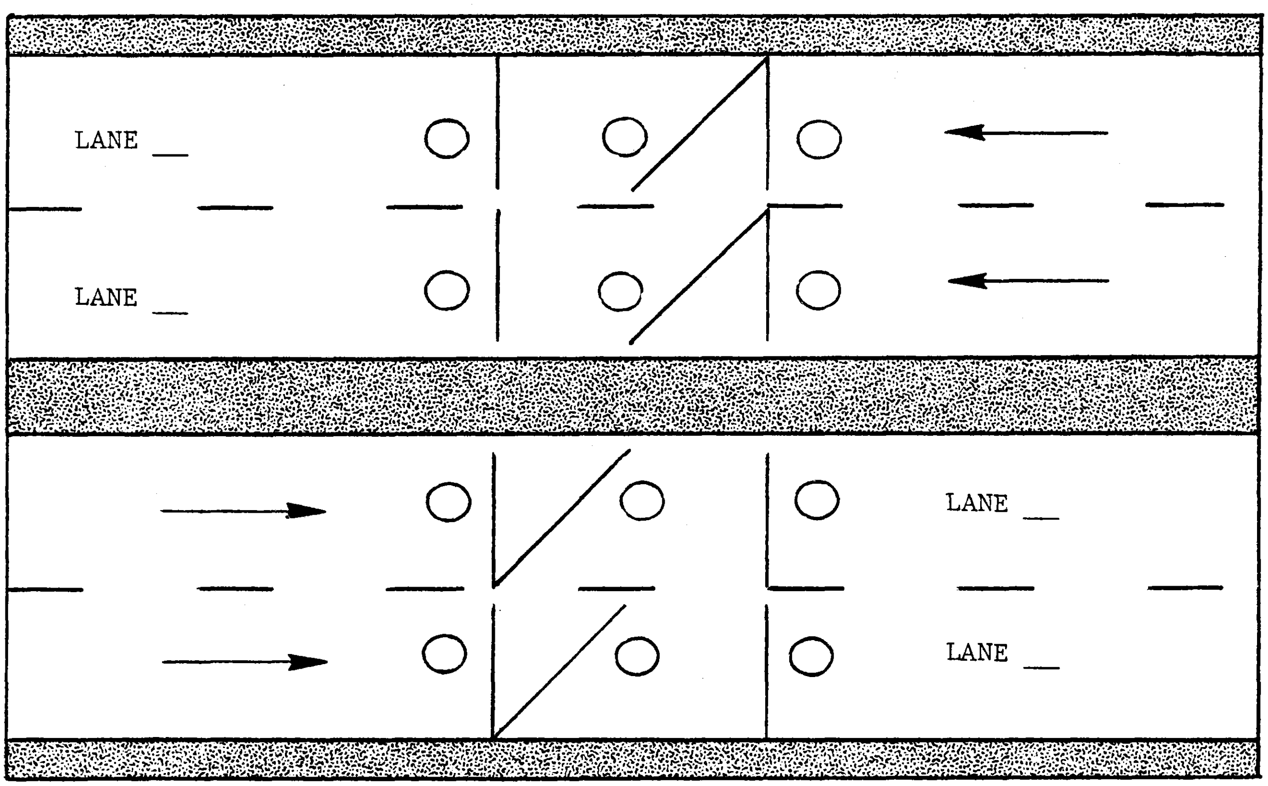
REMARKS ____________________

____

____

____

____



Note! Switches should be assigned input numbers shown on remote box.

Tape No.	Date	Start Time	Stop Time	Descriptive Pages

1856

MOUNTAIN PAVEMENT MARKING PROJECT

Descriptive Events

Date ______________________ Route ______________________

Tape No. __________

Time	Event
	Begin Data Collection

APPENDIX D

VEHICLE CHARACTERIZATIONS

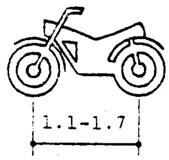
(Dimensions shown are in meters. For conversion use 3.28 feet per meter.)

1858

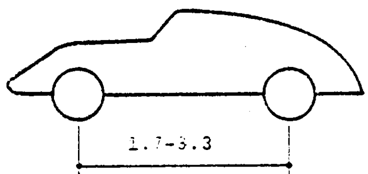
VEHICLE TYPE

A. 2-Axle

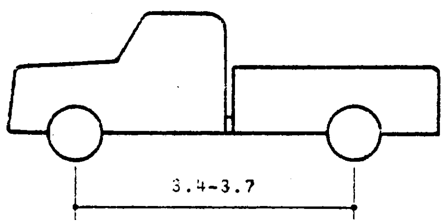
1. Motorcycle



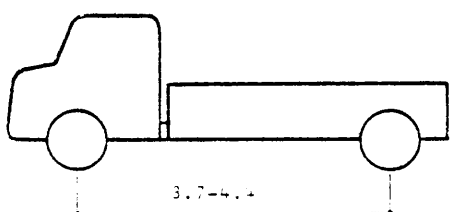
2. Car



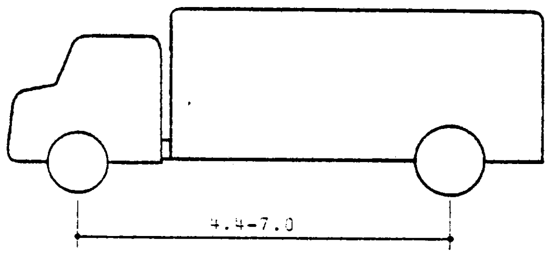
3. Pickup or Van



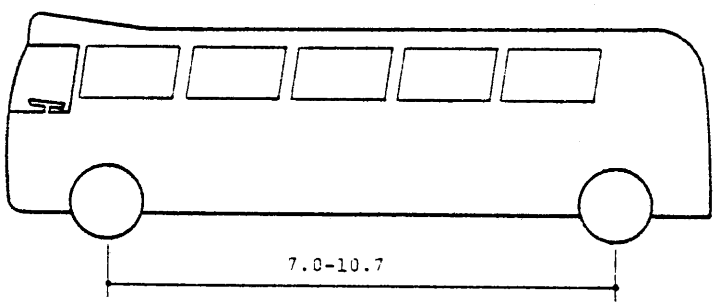
4. Truck



5. Straight Truck

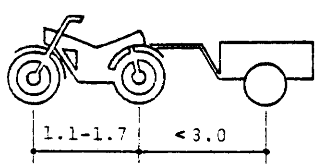


6. Bus

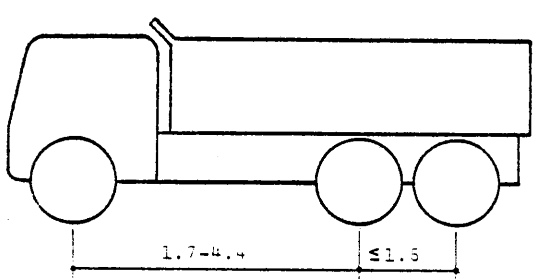


B. 3-Axle

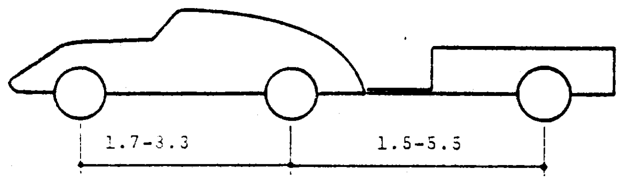
7. Motorcycle and Trailer



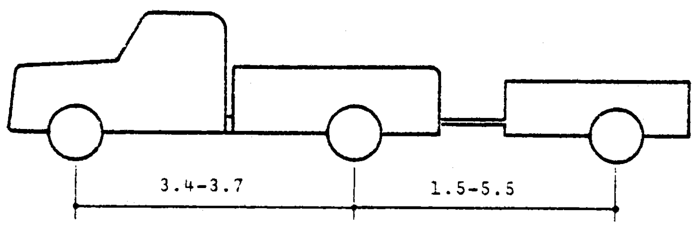
8. Dual Truck



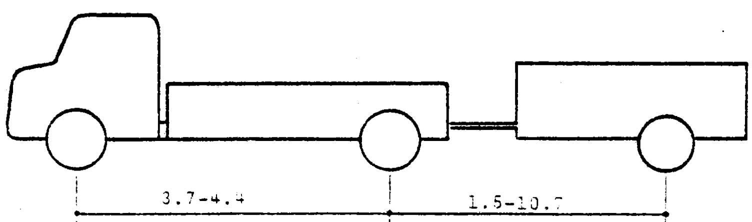
9. Car-Trailer (1-Axle)



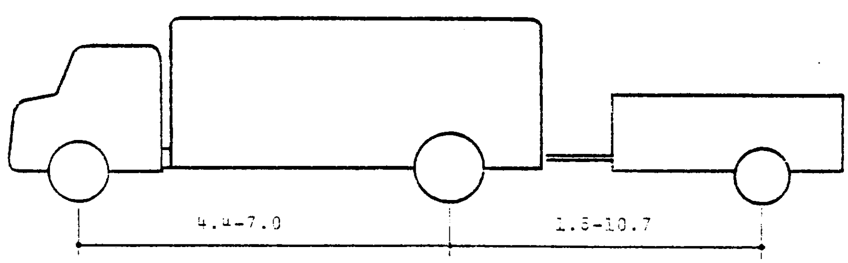
10. Pickup or Van — Trailer (1-Axle)



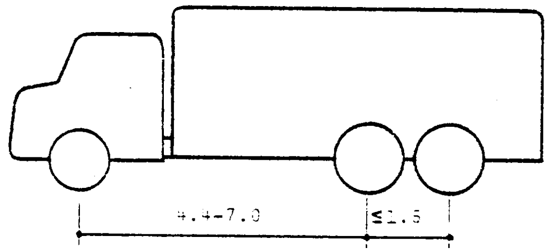
11. Truck-Trailer (1-Axle)



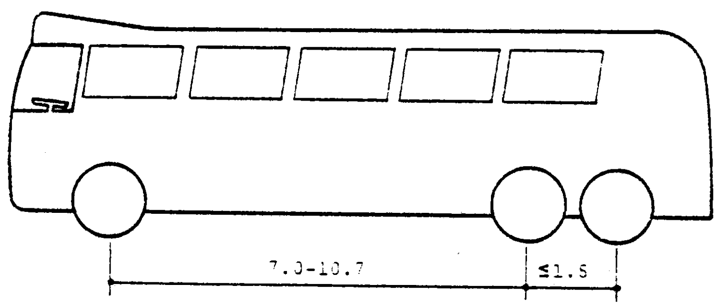
12. Straight Truck and Trailer



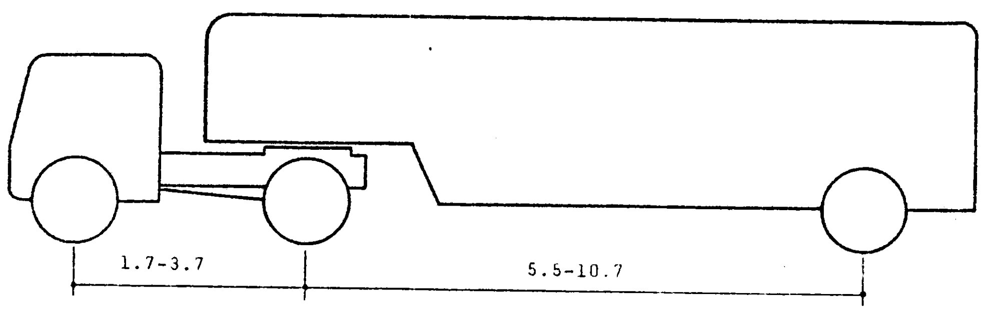
13. Dual-Axle Straight Truck



14. Dual-Axle Bus

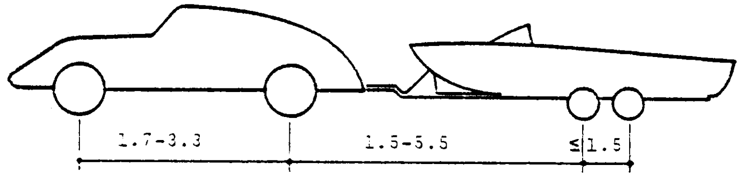


15. Tractor-Trailer



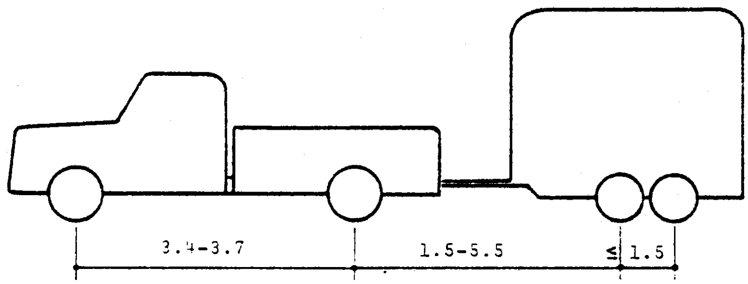
C. 4-Axle

16. Car and 2-Axle Trailer

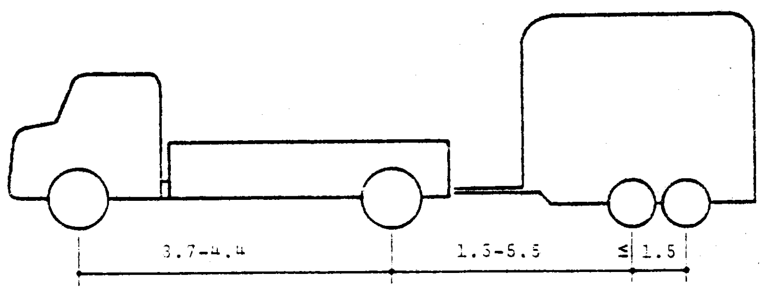


1862

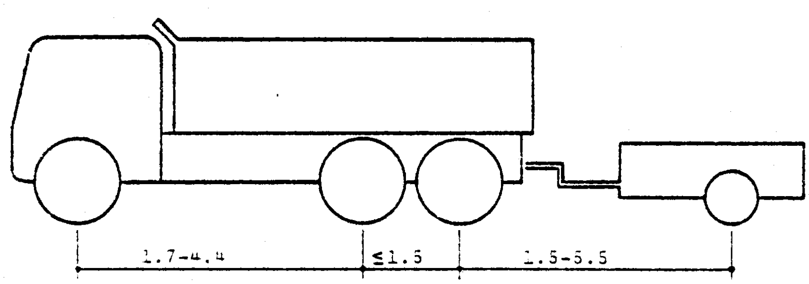
17. Pickup or Van and 2-Axle Trailer



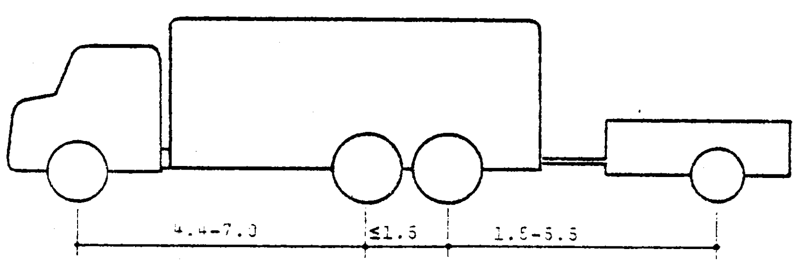
18. Truck and 2-Axle Trailer



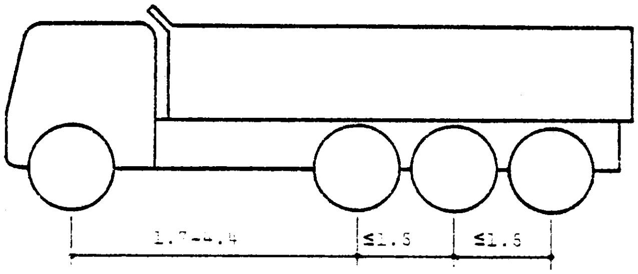
19. Dual-Axle Truck and Trailer



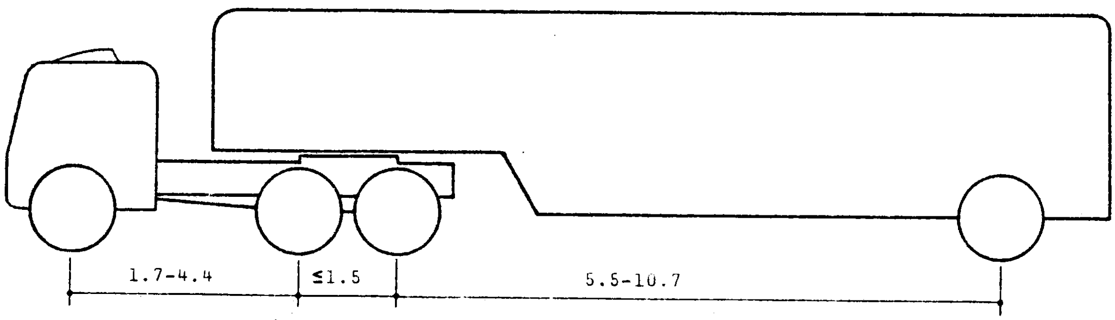
20. Dual-Axle Straight Truck and Trailer



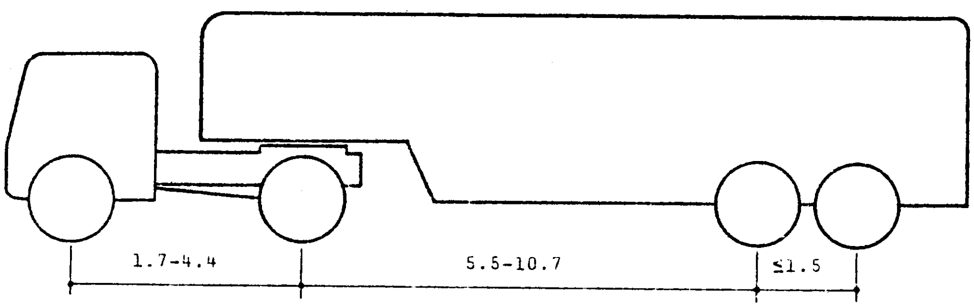
21. Triple-Axle Truck



22. Tractor Trailer: Dual-Axle Tractor

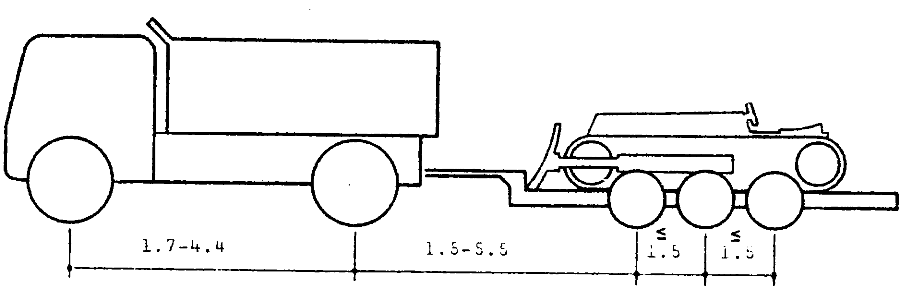


23. Tractor Trailer: Dual-Axle Trailer

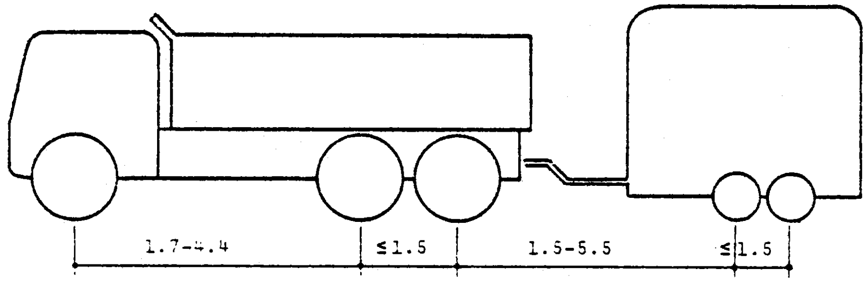


D. 5-Axle

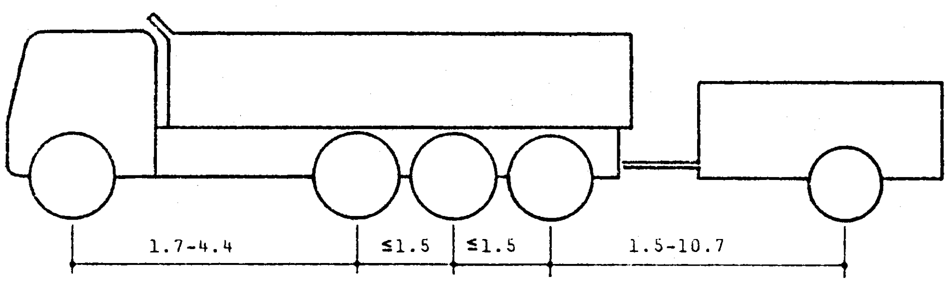
24. Truck and 3-Axle Trailer



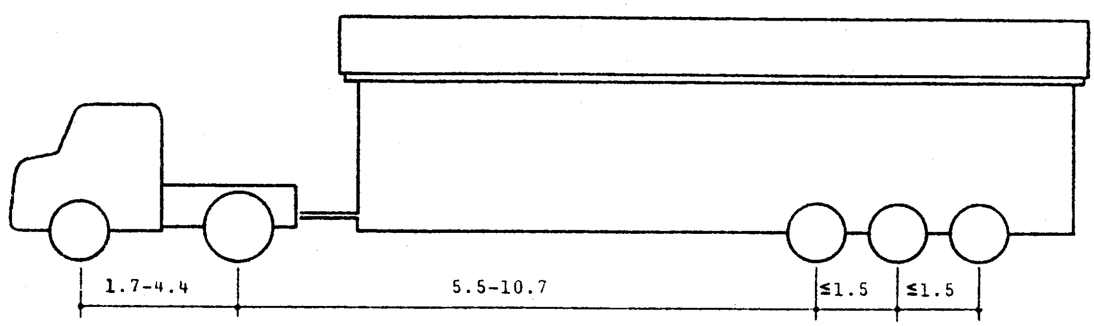
25. Dual-Truck and 2-Axle Trailer



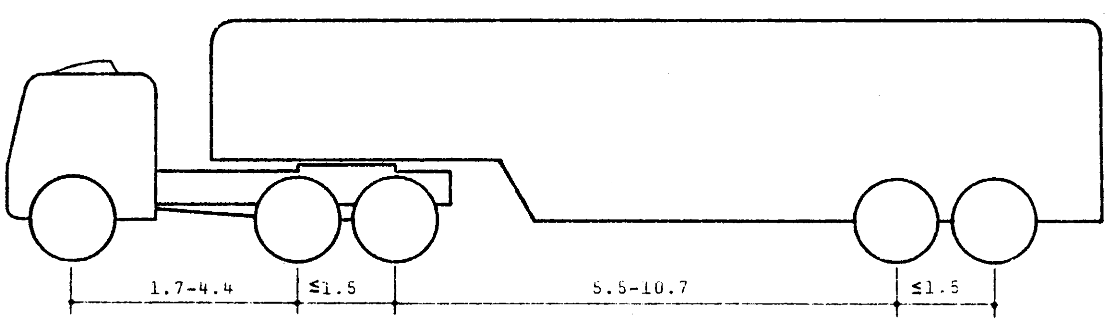
26. Triple-Axle Truck and Trailer



27. Truck and 3-Axle House Trailer



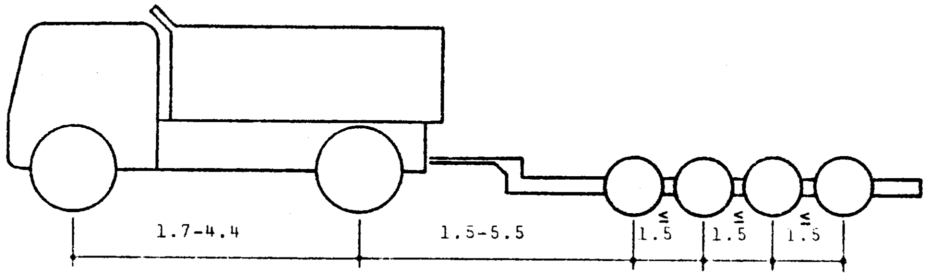
28. Tractor-Trailer: Dual-Axle on Both



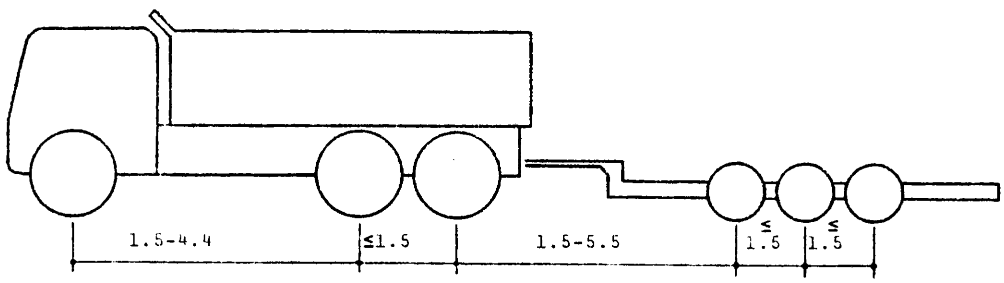
E. 6-Axle

29. Truck and 4-Axle Trailer

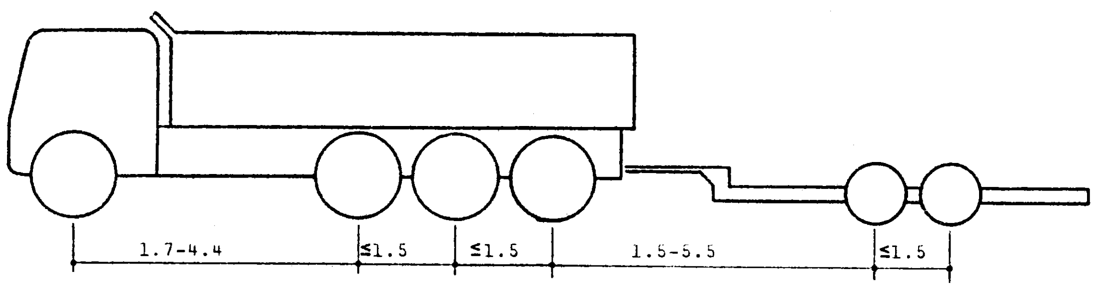
1865



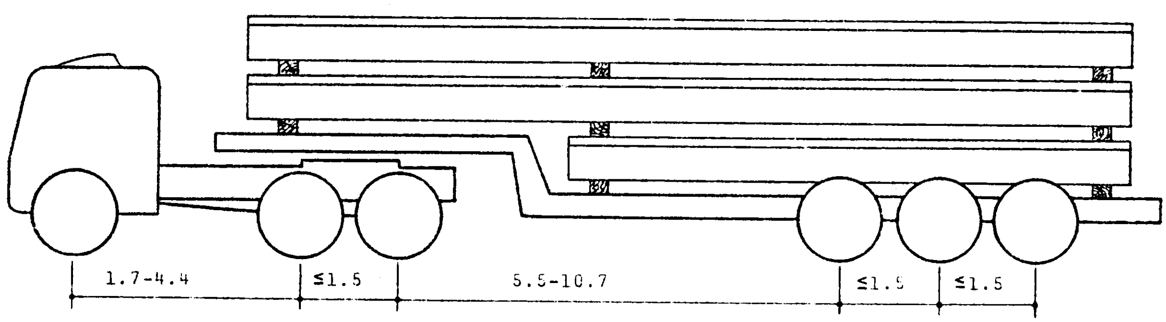
30. Dual-Axle Truck and 3-Axle Trailer



31. Triple-Axle Truck and 2-Axle Trailer

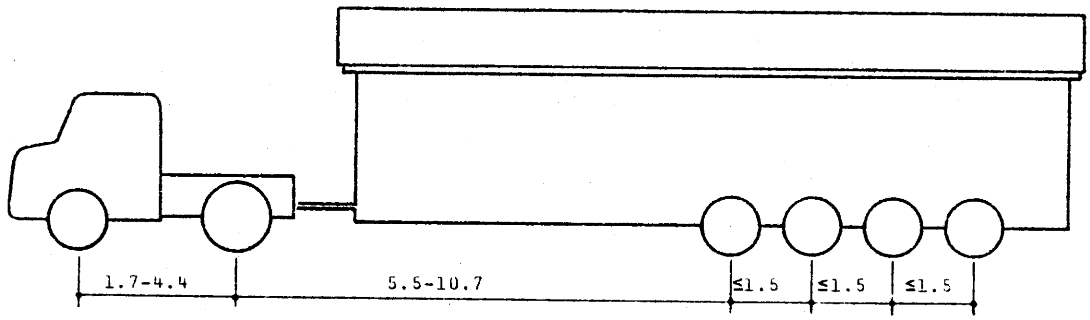


32. Dual-Tractor and 3-Axle Trailer



1866

33. Truck and 4-Axle House Trailer



F. 7-Axle

34. Triple-Axle Truck and 3-Axle Trailer