TECHNICAL REPORT STANDARD TITLE PAGE

| 1. Report No. | 2. Government Accession No. | 3. Recipient's Catalog No. |
|---|---|---|
| FHWA/TX-84/59+290-2 | | |

| 4. Title and Subtitle | 5. Report Date |
|---|---|
| Data Reduction System Utilizing Golden River Counting and Recording Equipment | November 1983 |
| | 6. Performing Organization Code |
| 7. Author(s) | 8. Performing Organization Report No. |
| Jae Y. Lee and Gene P. Ritch | Research Report 290-2 |
| 9. Performing Organization Name and Address | 10. Work Unit No. |
| Texas Transportation Institute<br>The Texas A&M University System<br>College Station, Texas 77843 | 11. Contract or Grant No.<br>Study No. 2-18-81-290 |
| | 13. Type of Report and Period Covered |
| 12. Sponsoring Agency Name and Address | Interim - September 1980<br>November 1983 |
| Texas State Department of Highways and Public Transportation; State Transportation Planning Division,<br>P. O. Box 5051<br>Austin, Texas 78763 | 14. Sponsoring Agency Code |

15. Supplementary Notes

Research performed in cooperation with DOT, FHWA.
Research Study Title: Developing a Freeway Data Collection System.

16. Abstract

There is a need for large quantities of traffic data to plan, design and operate urban freeways. Portable traffic counters and records, utilizing microprocessors and memory storage, have increased the efficiency in the collection of volume data, and to a limited extent, the processing and reporting of the data. The increased need for timely data requires systems that automatically process and store data in a computer data management System

For the current microprocessor technology in portable data collection systems of traffic counters, this project develops software programs that transfer the raw data to a large computer, process and store the data in forms from which the user can request a large number of analyses and report formats with little manual intervention.

| 17. Key Words | 18. Distribution Statement |
|---|---|
| Traffic Counting Systems, Data Analysis, Data Storage, Data Retrieval, Traffic Data Base, Microprocessor Applications. | No restriction. This document is available to the public through the National Technical Information Service, 5285 Port Royal Road, Springfield, Virginia 22161. |

| 19. Security Classif. (of this report) | 20. Security Classif. (of this page) | 21. No. of Pages | 22. Price |
|---|---|---|---|
| Unclassified | Unclassified | 69 | |

Form DOT F 1700.7 (8-69)

# DATA REDUCTION SYSTEM UTILIZING GOLDEN RIVER COUNTING AND RECORDING EQUIPMENT

by

Jae Y. Lee

and

Gene P. Ritch

Research Report 290-2
Research Study Number 2-18-81-290
Developing a Freeway Data Collection System

Sponsored by

Texas State Department of Highways and Public Transportation
in cooperation with the
U. S. Department of Transportation, Federal Highway Administration

TEXAS TRANSPORTATION INSTITUTE
The Texas A&M University System
College Station, Texas

November 1983

## ABSTRACT

There is a need for large quantities of traffic data to plan, design and operate urban freeways. Portable traffic counters and recorders, utilizing microprocessors and memory storage, have increased the efficiency in the collection of volume data, and to a limited extent, the processing and reporting of the data. The increased need for timely data requires systems that automatically process and store data in a computer data management system.

For the current microprocessor technology in portable data collection systems of traffic counters, this project develops software programs that transfer the raw data to a large computer, process and store the data in forms from which the user can request a large number of analyses and report formats with little manual intervention.

## DISCLAIMER

The contents of the report reflect the views of the authors who are responsible for the facts and accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the Federal Highway Administration. This report does not constitute a standard, specification, or regulation.

Key Words: Traffic Counting Systems, Data Analysis, Data Storage, Data Retrieval, Traffic Data Base, Microprocessor Applications

## SUMMARY

This report is concerned with expanding the analysis of traffic volume data as counted and recorded by the Golden River traffic counter system. This system stores data in the counter in a memory controlled by a microprocessor. These data are transferred to another larger microprocessor called a Retriever for further processing. Two reports can be printed on a terminal directly from the Retriever. The formats of these reports are provided by the suppliers of the equipment. Different formats can be ordered from the suppliers, but the flexibility of the type of analyses and reports provided by the Retriever is limited.

This study developed programs that transfer the raw data from the Retriever processes and stores the data in a large computer in a manner that permits further analysis and reports specified by the user. Programs for daily and weekly summary reports are provided.

## IMPLEMENTATION STATEMENT

The traffic data analysis programs are written in the WATFIV FORTRAN language and, with modifications, can be utilized on the Department's computer system. The software modifications are mainly associated with the input/output Fortran functions and can be implemented with minimal time and effort. The more difficult task is transmitting the traffic data from the Golden River retriever to the Department's computer system. The retriever transmits data as an asynchronous communications device. The Department's communications network is built around IBM's System Network Architecture (SNA) concept and its ability to communicate with synchronous data link controllers (SDLC). As such, the retriever must appear to be one of several SDLC devices to the Department's computer system. At present, there are only two methods to employ that will enable the retriever to communicate with the Department's computer system in this manner and both methods include additional equipment. The present FACTS* system can communicate to both the retriever and the main computer system, but software must be developed for the FACTS that will enable the traffic data to be read. The FACTS system will in turn communicate to the main computer system via a SDLC linkup. The other method is to utilize a microprocessor to read the traffic data and through software emulate a SDLC device when communicating to the Department's computer system. This approach is being studied in the current Research Project 2-18-84-421, "Developing A Freeway Data Base." The microprocessor will emulate the current data terminal through a software program and a hardware 3270 SNA/SDLC communication package.

*Flexible Advanced Computer Traffic Signal

TABLE OF CONTENTS

LIST OF FIGURES

# LIST OF APPENDICES

## INTRODUCTION

The Golden River traffic counter system is one of the most advanced traffic counter systems available. Until recently, a sizeable and time consuming effort was required to transcribe the data from the output format of the Retriever to one which was more easily understood. This report explains the development of a computer assisted method to reduce this data reduction effort. The programs which were developed aid the user in removing the data from the Retriever, storing them in a computer file, and producing printed outputs similar to those which were obtained manually.

The procedures and programs discussed in this report were developed under Wylbur/370 system at Texas A&M University. Wylbur/370 is an on-line interactive text editing system that helps the user to create files, submit and retrieve jobs from remote terminals. Wylbur also has executing capabilities that can be used as a preprocessor to better aid the user in executing the programs. The execute files discussed in this report uses this preprocessor capability to enter different header information to the program.

The use of these programs results in a simplified and efficient reduction of traffic volume data for the analysis process.

## SITUATION

The Golden River Retriever is being used in connection with a Silent-700 terminal to produce a raw data output of traffic volume counts. The Silent-700 terminal is used as an auxiliary printer connected to the

Golden River Retriever. The reduction of these raw data into useful information is a tedious manual process that occupies much of the user's time. For more efficiency, a computer can be utilized to remove the data from the retriever and store it in a computer file. Once the data are stored, a program can manipulate the data and produce reports similar to those obtained manually. This process could aid the user in a faster and more efficient analysis of traffic volume data. The data can be stored permanently in a computer memory and be used at a later date to produce reports. This is advantageous when the Retriever memory space is filled. The data can be transferred to the computer, making the Retriever available for continued data collection.

## PROCEDURES

The first step in computer utilization is to remove the data from the Retriever and store it in a computer file. It is desirable that the data be transferred directly from the Retriever into a computer file with the least amount of human intervention. The transmission of data by individual files is more time consuming. It requires an interaction with Wylbur after each file is transmitted to the computer. Direct data transfer allows for a faster and more accurate data transfer and frees the user to concentrate on other activities during the data transmission.

A special cable was developed to serve as an interface between the Retriever and a Silent-700 terminal. This cable enables communication between the Retriever and the Wylbur system at the Texas A&M host computer. The data are transmitted via the RS-232 serial port connector of the

GR-0308 Interface/Charger unit to the Silent-700 terminal. The Silent/700 terminal is used as a modem connection that transmits the serial data to the host computer (Amdahl 470). The transmitted data are stored as a sequential data file on a computer disk memory.

Once the data is removed from the Retriever and stored in a computer memory, the Retriever may be cleared and used for more data retrieval. The procedure to remove the data from the Retriever and store them in a computer file is described in the following "DATA FILE SET-UP" procedure.

## DATA FILE SET-UP PROCEDURE

The following procedure is to be utilized for a direct transfer of traffic volume data from a Golden River Retriever to a data file on the Amdahl 470 V6/V8 system at Texas A&M University.

- Turn on the Interface/charge unit.
- Turn on the Silent-700 terminal.
- Make the following cable hook-ups on the Interface unit:
  1) Connect P01 to P01-A.
  2) Connect P02 to On-Line Computer.
  3) Connect P03 to On-Line Terminal.
  4) Connect P04 to Silent-700 rear plug.
  5) Set the toggle switch on the switch box on "ON" position. This switch box disables the data transmission to the printer but allows the communication between the Retriever and the Wylbur system. The user is advised not to disturb this switch position during the data transmission.

See Figure 1 for description of the Interface unit. See Figure 2 for the block diagram of the entire system hook up.

- Set the toggle switch on the Interface unit to the "On-Line Terminal" side (Right).
- Make the following settings on the Silent-700 terminal.
  1) Speed setting to High-Speed.
  2) Duplex setting to FULL Duplex.
  3) On-Line switch to "On-Line" position.

The interface signals transmitted during the data transfer from the Retriever are an important factor of data transfer. These signals control "line feed" and "carriage return" as well as other signals that can be programmed to be transmitted at the end of each line and end of each file.

- To make the proper signals, the following steps should be used on the Retriever:
  1) Place Retriever into the Interface mode by sliding the indicator bar next to "INTERFACE".
  2) The display will show three numbers.
  3) Push "2" to step to Interface mode 2. Interface 2 should be set at 300.
  4) Push "3" and the Retriever will display three numbers.
- Press "#" and "*" simultaneously and the Retriever will display "_ . _ _".

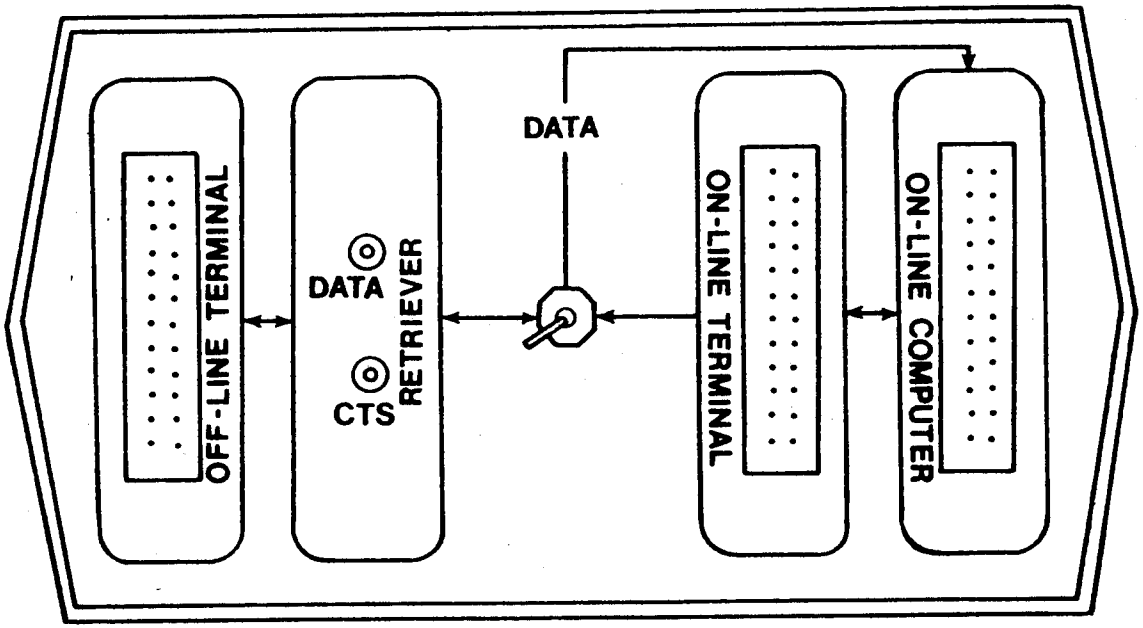



Figure 1

GR-0308 Interface/Charger Unit

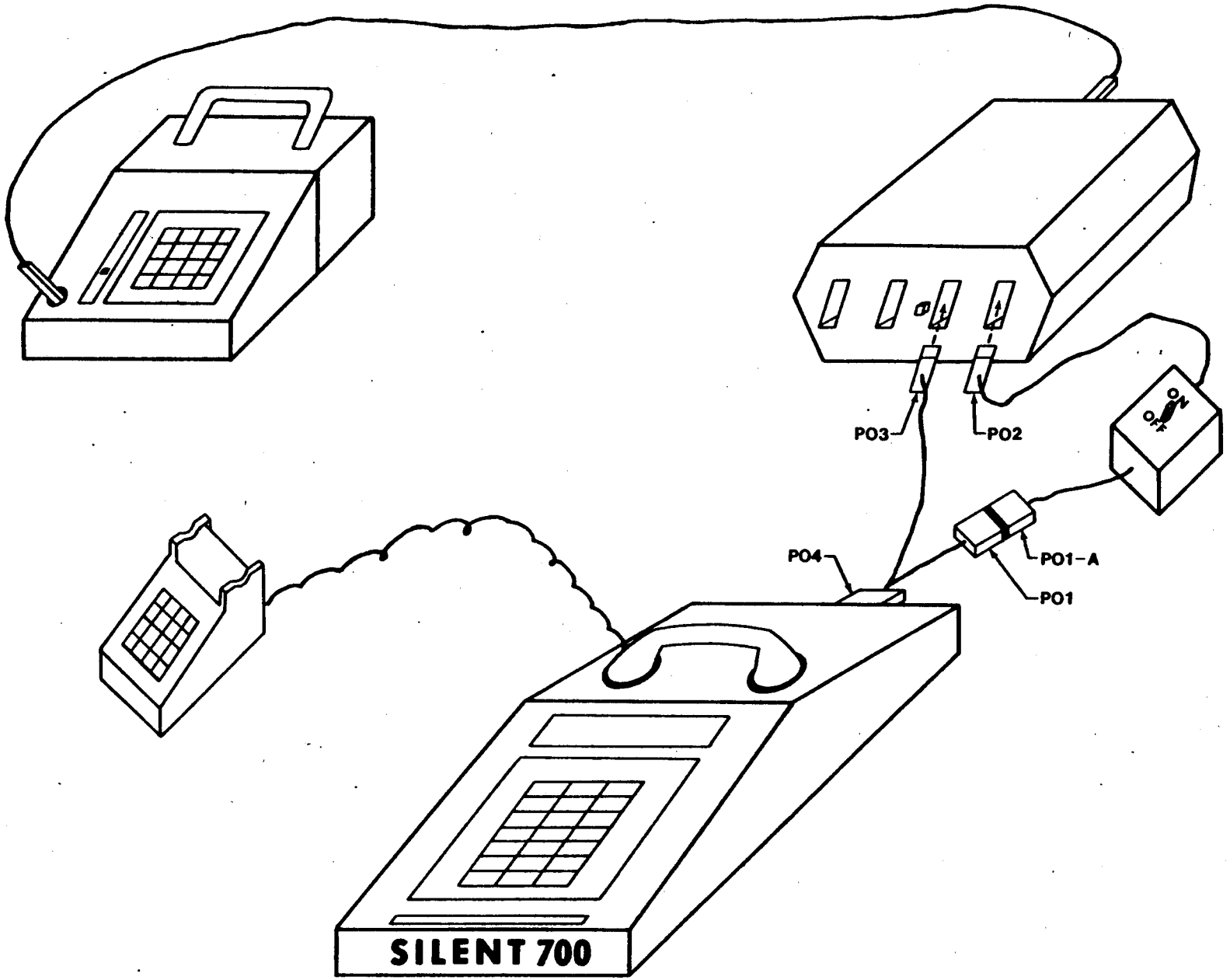



Figure 2

Block Diagram of System Hook-Up

- Type in the following sequence of numbers:

  1) 013 + Enter key

  2) 999 + Enter key

  3) 999 + Enter key

  These are minimum required signals and more can be found in the user's manual for the Golden River Retriever. See Figure 3 for the description of Retriever keyboard.

- Push "4" and make sure that the Interface 4 is set to "1" which automatically stops data transmission after each line, and wait for DC4 signal from Wylbur before continuing to the next line of data.
- Push 5; Interface 5 should be set on "0". See Appendix D for more information on interface signal setting procedure.
- After all the hook-ups and settings are made, on the telephone dial the dial-up number of the host computer, and wait for the high pitch. Place the phone receiver onto the cups on the Silent-700 terminal.
- The "On-Line" light should appear on the right corner of the terminal.
- Press "Return" twice and the computer will respond with a beep and a "GO".
- Press "Return" twice more and the computer will respond with "Invalid Switch Chars".
- Log-on to computer using the appropriate Texas A&M Data Processing Service Request number.
- Once logged on, get into collect mode by pressing "BREAK" key on the right-upper corner of the terminal. The computer will respond with a line number and a question mark, i.e., "1?".

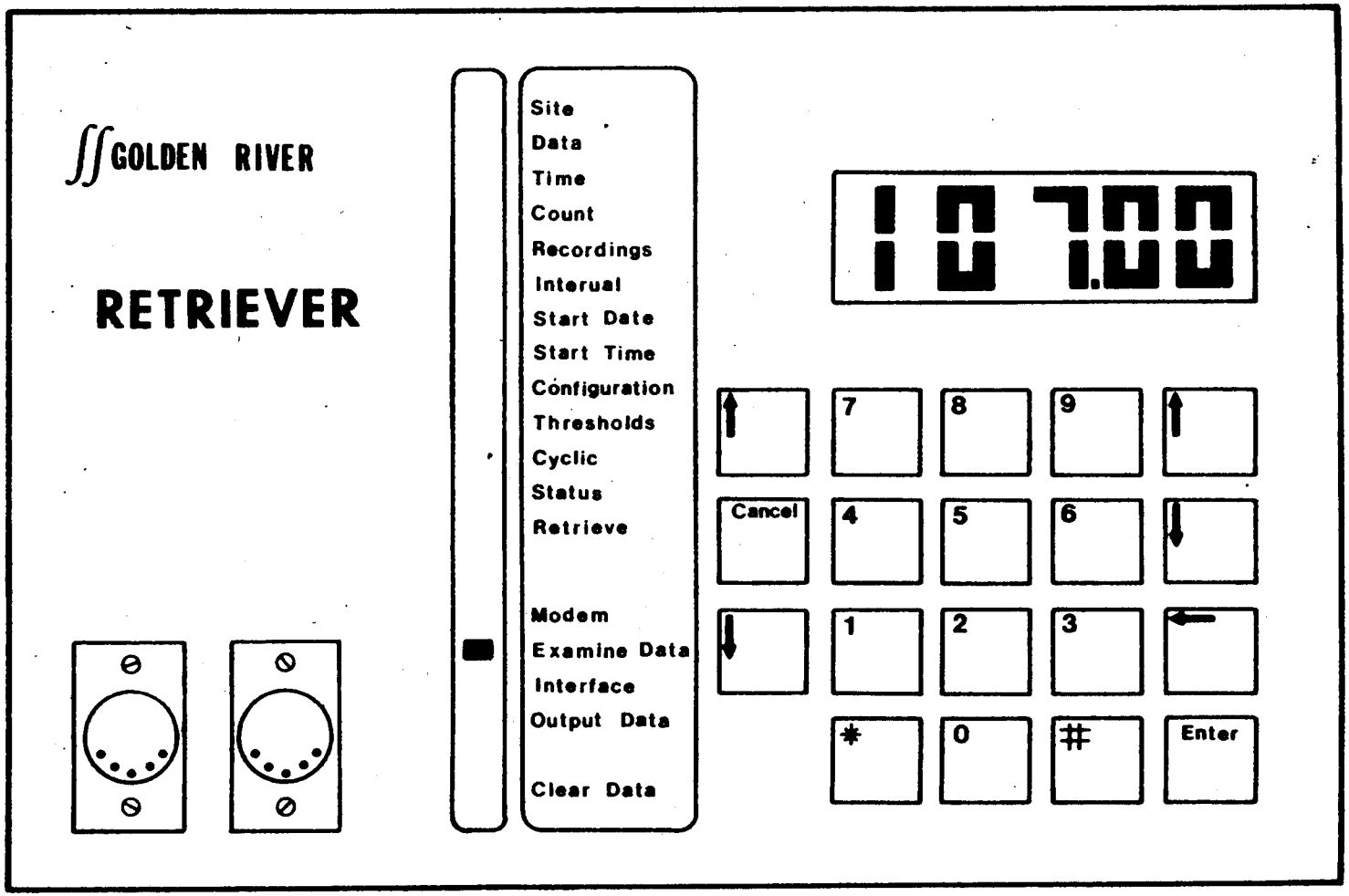



Figure 3

Golden River Retriever Face

- Place the toggle switch on the Interface unit to the "RETRIEVER" side (Left).
- Get into the Output mode in the Retriever by lining up the indicator bar next to the "OUTPUT DATA" column.
- Press "*" and "#" simultaneously. Retriever will display "_ _".
- Enter a file number to transfer a single file. The user could enter "99" which will transfer all the data files on the Retriever. The user may use a special command, Collect Unnumbered, which will squeeze each line of data into a single line on the printer. To use this command, type in "COL UNN" under "COMMAND" mode and the computer will no longer print out the line numbers.
- Examine the data as they are being transferred to the Amdahl computer.

Sometimes the user may want to disable the printer and proceed with the data transfer silently.

- To disable the printer, turn the "On-Line" switch on the Silent-700 to "OFF" position. The "On-Line" light should remain on.
  1) Data Transfer is shown by the blinking light on "DATA" indicator bulb on the Interface unit.
  2) The Retriever should display four dashes during the data transfer. Wait until both the blinking light and the dashes disappear which indicates the end of data transfer.
- Now the user must save the data transferred under a unique name for later usage.
- To get back to Wylbur, set the toggle switch on the Interface unit back to "On-Line Terminal" side and the "On-Line" switch on the

Silent-700 back to "On-Line" position.

- Press "BREAK" and the computer will respond with "COMMAND?"
- Save the data file under a unique alphabetic name of six characters or less.
- Examine the data type "LIST" and inspect the data as they are printed. across the printer. "LIST UNN" will squeeze each line of data onto single line on the printer.
- Log-off by typing LOGOFF CLE.

DATA OUTPUT PROCEDURE

Sometimes the user may wish to obtain raw data printout without creating a computer file. The following procedure can be used to obtain print out of the data from the Retriever.

- Turn on the GR-0308 Interface/Charger unit.
- Turn on the Silent-700 terminal.
- Make the following cable hook-ups on the GR-0308 Interface unit:
    1) Connect P01 to P01-A.
    2) Connect P02 to On-Line Computer.
    3) Connect P03 to Off-Line Terminal.
    4) Connect P04 to Silent-700 rear plug.
    5) Set the toggle switch on the switch box on "ON" position.
- Set the toggle switch on the GR-0308 Interface unit to the "Off-Line Terminal" side (Left).
- Following the same procedure described on page 4, make the following

interface signals:

1) 013 + Enter key

2) 010 + Enter key

3) 099 + Enter key

4) 099 + Enter key

- The Interface 4 should be set to "Ø" which allows continuous data transmission.
- Get the Retriever into output mode by lining up the bar indicator next to the "OUTPUT DATA" column.
- Press "*" and "#" simultaneously. Retriever will display "_ _".
- Enter a file number to transfer a specific file. The user could enter "99" which will transfer all the data files on the Retriever.
- Examine the data as they are printed.

This procedure produces simple listing of the collected data and does not create a computer file.

FILE SEPARATION

Once the data are stored as a sequential file, it must be subdivided into several files. Each segment of the original file is separated by marker statements "*BEGIN" and "*END" as shown in Figure 4. The segments delimited by "*BEGIN" and "*END" of the original data file must be subdivided into individual files. Then the program can reduce the data within the individual files and produce a report. Partitioned data sets are used to group the individual files together under a master file. A partitioned data set will save the entire file under a master file name and each individual file will be given a distinct member name. This allows the user to

```
 1.  *BEGIN 00 01 03310482 0015 00
 2.  830613 1415  0080 0097 0101 0115 0092 0099 0098 0096 0080 0073 0100 0099
 3.  830613 1715  0122 0125 0092 0098 0074 0081 0101 0065 0068 0065 0051 0049
 4.  830613 2015  0050 0048 0051 0043 0052 0039 0024 0036 0036 0027 0023 0019
 5.  830613 2315  0021 0027 0021 0013 0019 0004 0007 0011 0002 0006 0007 0002
 6.  830614 0215  0007 0011 0014 0002 0005 0004 0003 0003 0003 0006 0011 0010
 7.  830614 0515  0012 0024 0043 0050 0064 0122 0126 0136 0132 0171 0197 0223
 8.  830614 0815  0196 0172 0176 0139 0131 0134 0145 0142 0098 0087 0093 0096
 9.  *END   00 01 03310482 0015 00
10.  *BEGIN 00 02 03310780 0015 00
11.  830613 1100  0114 0142 0141 0162 0136 0151 0165 0157 0157 0162 0179 0169
12.  830613 1400  0198 0158 0142 0181 0161 0155 0151 0261 0152 0218 0168 0207
13.  830613 1700  0147 0233 0128 0145 0111 0094 0084 0090 0065 0072 0064 0053
14.  830613 2000  0067 0051 0044 0046 0047 0031 0023 0026 0033 0027 0021 0031
15.  830613 2300  0014 0018 0020 0009 0013 0008 0015 0006 0002 0012 0004 0011
16.  830614 0200  0012 0009 0010 0004 0003 0003 0003 0007 0005 0006 0006 0012
17.  830614 0500  0009 0019 0015 0018 0018 0053 0050 0101 0118 0143 0185 0225
18.  830614 0800  0208 0185 0168 0149 0141 0129 0134 0127 0120 0133 0117 0128
19.  830614 1100  0129 0130
20.  *END   00 02 03310780 0015 00
21.  *BEGIN 00 03 03310777 0015 00
22.  830613 1115  0056 0081 0070 0048 0062 0068 0063 0041 0055 0059 0054 0065
23.  830613 1415  0060 0052 0069 0058 0053 0060 0082 0075 0066 0084 0071 0074
24.  830613 1715  0084 0096 0074 0077 0057 0043 0049 0042 0046 0036 0041 0032
25.  830613 2015  0034 0035 0026 0036 0034 0033 0029 0027 0021 0025 0022 0028
26.  830613 2315  0013 0018 0005 0013 0009 0010 0005 0007 0003 0003 0007 0002
27.  830614 0215  0003 0005 0002 0004 0007 0001 0003 0003 0001 0004 0002 0003
28.  830614 0515  0005 0003 0016 0028 0038 0024 0048 0047 0049 0064 0063 0054
29.  830614 0815  0067 0067 0062 0061 0061 0065 0062 0068 0061 0036 0049 0050
30.  830614 1115  0050
31.  *END   00 03 03310777 0015 00
32.  *BEGIN 00 04 03310483 0015 00
33.  830613 1130  0163 0176 0145 0141 0118 0116 0130 0106 0120 0134 0171 0129
34.  830613 1430  0152 0177 0169 0149 0154 0150 0100 0096 0135 0146 0151 0192
35.  830613 1730  0204 0176 0149 0085 0093 0102 0113 0086 0081 0058 0064 0059
36.  830613 2030  0041 0058 0054 0040 0043 0042 0037 0035 0035 0040 0032 0026
37.  830613 2330  0026 0024 0021 0014 0011 0014 0007 0010 0011 0005 0009 0005
38.  830614 0230  0006 0001 0002 0004 0002 0002 0003 0003 0002 0007 0003 0006
39.  830614 0530  0012 0025 0029 0036 0076 0096 0124 0105 0120 0143 0156 0142
40.  830614 0830  0153 0130 0159 0136 0149 0141 0198 0139 0100 0121 0110 0149
41.  *END   00 04 03310483 0015 00
```

Figure 4

Data File Listing

address the individual files and also store all the individual files together in one master file.

The execute file named PARTSAVE was developed to aid the user in dividing the original data set into a partitioned data set. The following procedure is used to run the execute file PARTSAVE.

- Log-On to Wylbur
- Type "EXEC FROM PARTSAVE CLE"
- The execute file will interact with the user in conversational style to aid the user in dividing the data set.

The following algorithm is used in the execute file PARTSAVE. The execute file prompts the user to create a master file and provides instructions if the user needs assistance. Once the master file is set-up, the execute file will ask the user for the original file, working from end of the file and prints each individual section for verification. The user has the choice to either save or delete that portion of the data file. If the user chooses to save that segment, the execute file will ask for a member name to be assigned to that segment. The execute file will continue this process until it reaches the end of data set or the user "BREAK" operation. After the completion of the execute file "PARTSAVE", the user will have the original data file plus a master partitioned data set with its individual members. The system logic of PARTSAVE described in the flowchart in Figure 5. See Appendix A for the complete listing of the execute file PARTSAVE.

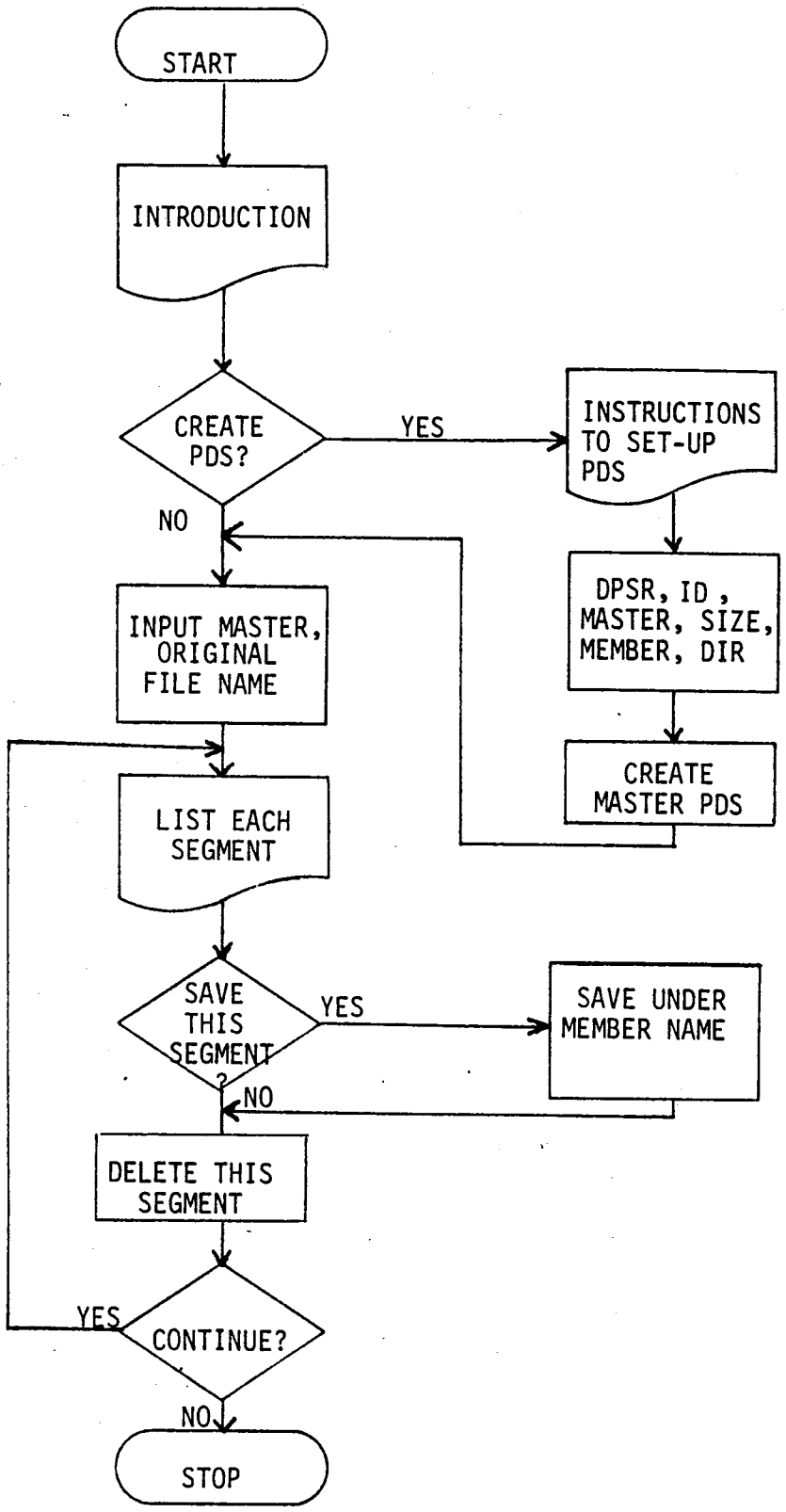



Figure 5

System Logic of PARTSAVE

## DATA REDUCTION

After the data set is divided into a partitioned data set, the user is ready to use the program named REPORTER, and reduce the raw data. The execute file "RUNNER" has been developed to aid the user in running the program REPORTER, which reduces data and produces the reports. The program RUNNER is used in the following manner. It interacts with the user to obtain the specific data set to be reduced, the specific count location, time interval, weather condition, counter, etc. Once this information is obtained, it passes on these parameters to the program REPORTER. They are to be used in producing reports. Program RUNNER will initiate the execution of the program REPORTER. REPORTER produces the report and returns to the execute file to check for continuing execution. If there are multiple files to be reduced, the user simply replies with the name of other files to be reduced. The system logic of RUNNER is described by the flowchart in Figure 6. See Appendix B for the complete listing of the execute file RUNNER.

The program REPORTER reduces the data according to the time interval and the appropriate configuration. REPORTER produces two reports: Daily time interval sums and weekly hourly totals. It also prints running subtotals and hourly totals for better data analysis. Two reports produced by REPORTER are shown in Figures 7 and 8.

The program REPORTER is designed with several modules, each with distinct functions which aid in producing the reports. Subroutine READER reads the specified data file and passes this information to module PREP. PREP is the core segment of REPORTER. It reduces the data

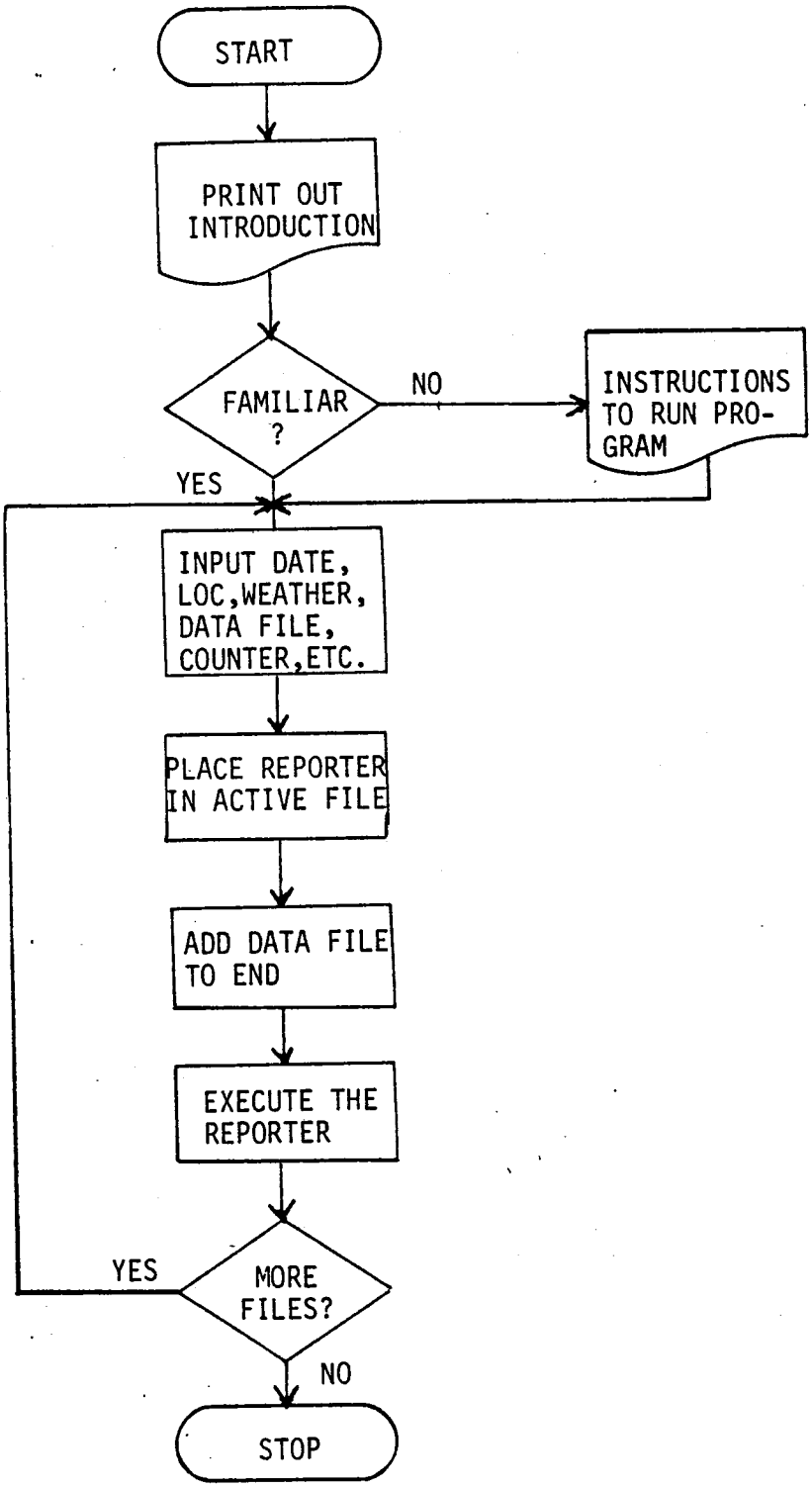



Figure 6

System Logic of RUNNER

FREEWAY VOLUME COUNT - TEXAS TRANSPORTATION INSTITUTE

DATE: 6-14-1983 SITE NO: 3310775 COUNTER: G.P. RITCH
WEATHER: GOOD DIRECTION OF FLOW: NORTHBOUND ON I45
LOCATION: I45 @ GREENS ROAD

SOURCE : THIS IS 1 OF 2

| TIME | VOLUME | HOURLY VOLUME | TIME | VOLUME | HOURLY VOLUME | TIME | VOLUME | HOURLY VOLUME | TIME | VOLUME | HOURLY VOLUME |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.00- 0.15 | 25 | | 6.00- 6.15 | 87 | | 12.00-12.15 | 0 | | 18.00-18.15 | 0 | |
| 0.15- 0.30 | 14 | | 6.15- 6.30 | 113 | | 12.15-12.30 | 0 | | 18.15-18.30 | 0 | |
| 0.30- 0.45 | 19 | 69 | 6.30- 6.45 | 177 | 562 | 12.30-12.45 | 0 | 0 | 18.30-18.45 | 0 | 0 |
| 0.45- 1.00 | 11 | | 6.45- 7.00 | 185 | | 12.45-13.00 | 0 | | 18.45-19.00 | 0 | |
| | | ( 69) | | | ( 949) | | | ( 4355) | | | ( 4355) |
| 1.00- 1.15 | 15 | | 7.00- 7.15 | 232 | | 13.00-13.15 | 0 | | 19.00-19.15 | 0 | |
| 1.15- 1.30 | 11 | | 7.15- 7.30 | 267 | | 13.15-13.30 | 0 | | 19.15-19.30 | 0 | |
| 1.30- 1.45 | 5 | 41 | 7.30- 7.45 | 299 | 1119 | 13.30-13.45 | 0 | 0 | 19.30-19.45 | 0 | 0 |
| 1.45- 2.00 | 10 | | 7.45- 8.00 | 321 | | 13.45-14.00 | 0 | | 19.45-20.00 | 0 | |
| | | ( 110) | | | ( 2068) | | | ( 4355) | | | ( 4355) |
| 2.00- 2.15 | 9 | | 8.00- 8.15 | 252 | | 14.00-14.15 | 0 | | 20.00-20.15 | 0 | |
| 2.15- 2.30 | 11 | | 8.15- 8.30 | 262 | | 14.15-14.30 | 0 | | 20.15-20.30 | 0 | |
| 2.30- 2.45 | 6 | 31 | 8.30- 8.45 | 206 | 907 | 14.30-14.45 | 0 | 0 | 20.30-20.45 | 0 | 0 |
| 2.45- 3.00 | 5 | | 8.45- 9.00 | 187 | | 14.45-15.00 | 0 | | 20.45-21.00 | 0 | |
| | | ( 141) | | | ( 2975) | | | ( 4355) | | | ( 4355) |
| 3.00- 3.15 | 11 | | 9.00- 9.15 | 161 | | 15.00-15.15 | 0 | | 21.00-21.15 | 0 | |
| 3.15- 3.30 | 17 | | 9.15- 9.30 | 140 | | 15.15-15.30 | 0 | | 21.15-21.30 | 0 | |
| 3.30- 3.45 | 7 | 47 | 9.30- 9.45 | 153 | 628 | 15.30-15.45 | 0 | 0 | 21.30-21.45 | 0 | 0 |
| 3.45- 4.00 | 12 | | 9.45-10.00 | 174 | | 15.45-16.00 | 0 | | 21.45-22.00 | 0 | |
| | | ( 188) | | | ( 3603) | | | ( 4355) | | | ( 4355) |
| 4.00- 4.15 | 4 | | 10.00-10.15 | 133 | | 16.00-16.15 | 0 | | 22.00-22.15 | 0 | |
| 4.15- 4.30 | 7 | | 10.15-10.30 | 120 | | 16.15-16.30 | 0 | | 22.15-22.30 | 0 | |
| 4.30- 4.45 | 12 | 46 | 10.30-10.45 | 126 | 482 | 16.30-16.45 | 0 | 0 | 22.30-22.45 | 0 | 0 |
| 4.45- 5.00 | 23 | | 10.45-11.00 | 103 | | 16.45-17.00 | 0 | | 22.45-23.00 | 0 | |
| | | ( 234) | | | ( 4085) | | | ( 4355) | | | ( 4355) |
| 5.00- 5.15 | 17 | | 11.00-11.15 | 125 | | 17.00-17.15 | 0 | | 23.00-23.15 | 0 | |
| 5.15- 5.30 | 30 | | 11.15-11.30 | 145 | | 17.15-17.30 | 0 | | 23.15-23.30 | 0 | |
| 5.30- 5.45 | 33 | 153 | 11.30-11.45 | 0 | 270 | 17.30-17.45 | 0 | 0 | 23.30-23.45 | 0 | 0 |
| 5.45- 6.00 | 73 | | 11.45-12.00 | 0 | | 17.45-18.00 | 0 | | 23.45-24.00 | 0 | |
| | | ( 387) | | | ( 4355) | | | ( 4355) | | | ( 4355) |

AM PEAK (6:30 - 9:30) VOLUME: 2689
AM PEAK HIGHEST HOUR: 1139
(NEAREST 15 MINUTE COUNT)

STARTING TIME : 7:15

PM PEAK (3:30 - 6:30) VOLUME: 0
PM PEAK HIGHEST HOUR: 0
(NEAREST 15 MINUTE COUNT)

STARTING TIME : 0:00

DT: 4355

Figure 7

Daily Time Interval Report

FREEWAY VOLUME COUNT SUMMARY SHEET

LOCATION : I45 @ GREENS ROAD

DATE : 6-13-1983 DIRECTION OF FLOW : NORTHBOUND ON I45

SOURCE : THIS IS 1 OF 2 SITE NO : 3310775

| DAY OF WEEK: | MONDAY | TUESDAY | WEDNESDAY | THURSDAY | FRIDAY | SATURDAY | SUNDAY |
|---|---|---|---|---|---|---|---|
| | 6 / 13 | 6 / 14 | 6 / 15 | 6 / 16 | 6 / 17 | 6 / 18 | 6 / 19 |
| TIME | HOURLY VOLUME | HOURLY VOLUME | HOURLY VOLUME | HOURLY VOLUME | HOURLY VOLUME | HOURLY VOLUME | HOURLY VOLUME |
| 0- 1 | 0 | 69 | 0 | 0 | 0 | 0 | 0 |
| 1- 2 | 0 | 41 | 0 | 0 | 0 | 0 | 0 |
| 2- 3 | 0 | 31 | 0 | 0 | 0 | 0 | 0 |
| 3- 4 | 0 | 47 | 0 | 0 | 0 | 0 | 0 |
| 4- 5 | 0 | 46 | 0 | 0 | 0 | 0 | 0 |
| 5- 6 | 0 | 153 | 0 | 0 | 0 | 0 | 0 |
| 6- 7 | 0 | 562 | 0 | 0 | 0 | 0 | 0 |
| 7- 8 | 0 | 1119 | 0 | 0 | 0 | 0 | 0 |
| 8- 9 | 0 | 907 | 0 | 0 | 0 | 0 | 0 |
| 9-10 | 0 | 628 | 0 | 0 | 0 | 0 | 0 |
| 10-11 | 0 | 482 | 0 | 0 | 0 | 0 | 0 |
| 11-12 | 0 | 270 | 0 | 0 | 0 | 0 | 0 |
| 12-13 | 532 | 0 | 0 | 0 | 0 | 0 | 0 |
| 13-14 | 555 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14-15 | 561 | 0 | 0 | 0 | 0 | 0 | 0 |
| 15-16 | 525 | 0 | 0 | 0 | 0 | 0 | 0 |
| 16-17 | 543 | 0 | 0 | 0 | 0 | 0 | 0 |
| 17-18 | 700 | 0 | 0 | 0 | 0 | 0 | 0 |
| 18-19 | 461 | 0 | 0 | 0 | 0 | 0 | 0 |
| 19-20 | 363 | 0 | 0 | 0 | 0 | 0 | 0 |
| 20-21 | 300 | 0 | 0 | 0 | 0 | 0 | 0 |
| 21-22 | 315 | 0 | 0 | 0 | 0 | 0 | 0 |
| 22-23 | 258 | 0 | 0 | 0 | 0 | 0 | 0 |
| 23-24 | 172 | 0 | 0 | 0 | 0 | 0 | 0 |
| AM PEAK (6-9) VOLUME: | 0 | 2588 | 0 | 0 | 0 | 0 | 0 |
| AM PEAK HIGHEST HOUR: | 0 | 1119 | 0 | 0 | 0 | 0 | 0 |
| PM PEAK (3-6) VOLUME: | 1768 | 0 | 0 | 0 | 0 | 0 | 0 |
| PM PEAK HIGHEST HOUR: | 700 | 0 | 0 | 0 | 0 | 0 | 0 |
| DT: | 5285 | 4355 | 0 | 0 | 0 | 0 | 0 |



Figure 8

according to time interval and configuration. PREP calculates the hourly totals, AM and PM peak period volumes, running subtotals, and other useful information. PREP then passes this information to subroutines DAILY and WEEKLY.

There are three possible ways to obtain the print out. DAILY produces the 24-hour summary and weekly produces the weekly hourly volume report. Both DAILY and WEEKLY produces the reports on each individual channel. They can produce up to seven reports for six channel configuration where the seventh report will contain the sum of all the channels. The user may choose to print out DAILY and/or WEEKLY report. On a multichannel configuration, the channels with no data, all zero recordings, will not be printed. The user makes the choice during the interaction with the execute file RUNNER which will make the proper changes in the REPORTER to print out the specified reports. The system logic of the program RUNNER is described in a flow chart in Figure 9. See Appendix C for the complete listing of the program REPORTER.

The entire system is designed in modular style to ease the modification of the programs to obtain other information needed by the user. It can be also modified to be used under different computer systems.

## CONCLUSIONS

The proper use of the computer programs described within this report results in a more efficient use of the Golden River traffic counter system. The developed programs aid the users in producing printed reports with less overall effort than was previously required using manual methods.

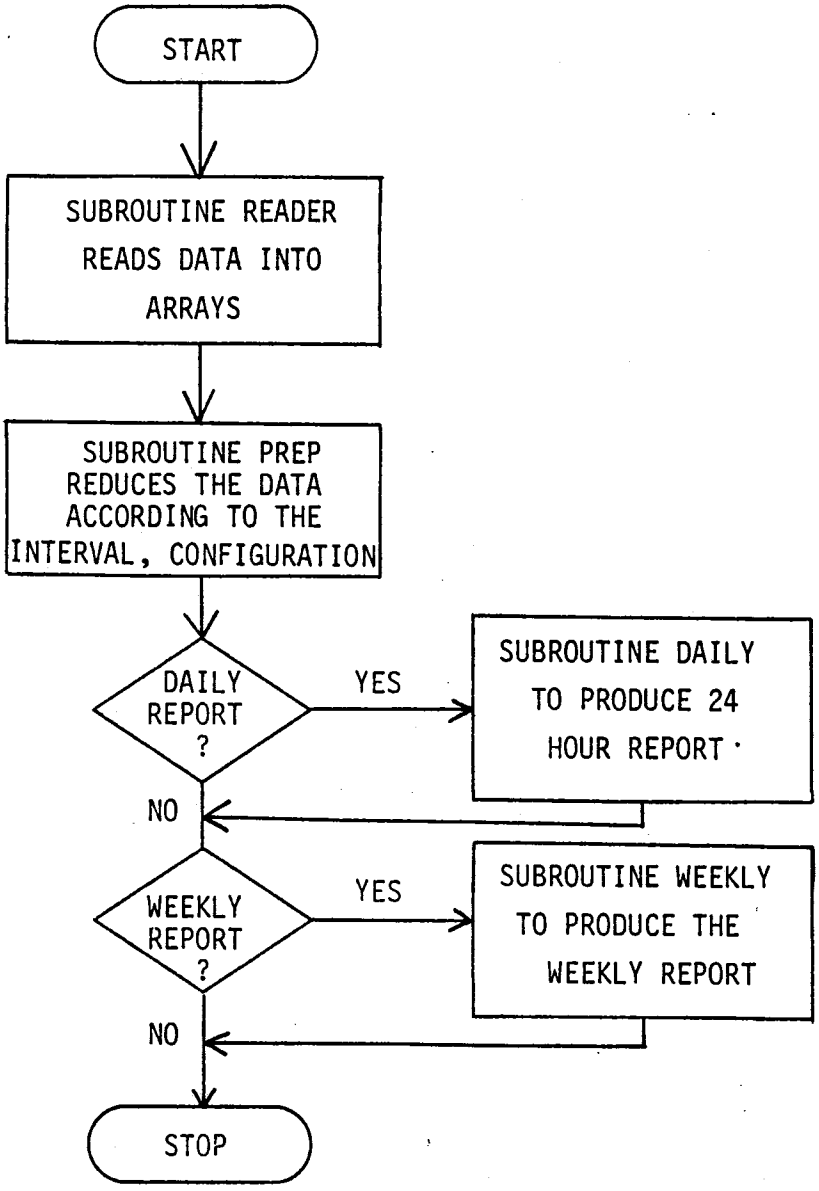



Figure 9
System Logic of REPORTER

Computer disk storage of the data file, as developed here, will aid in the long term storage and accessibility of the field data. The future developments will be in the area of utilizing a microcomputer to aid in data reduction system. The data will be transmitted to a microcomputer and stored on a floppy diskette for permanent storage.

# APPENDIX A

## Listing of Execute File PARTSAVE

```
 1.      SET ESCAPE %
 2.     ; THIS IS AN EXECUTE FILE TO DIVIDE A DATA FILE
 3.     ; INTO EACH INDIVIDUAL SEGMENT.  EACH SEGMENT IS
 4.     ; DELIMITERED BY BOTH '*BEGIN' AND '*END' STATEMENT
 5.     ;
 6.     ;
 7.     COMM  THIS PROGRAM WILL AID YOU IN DIVIDING YOUR DATA FILE INTO
 8.     COMM  SEVERAL INDIVIDUAL FILES.  YOU MUST ALREADY HAVE CREATED A
 9.     COMM  DUMMY PARTITIONED DATA SET WITH ENOUGH DIRECTORY AND SPACE
10.     COMM  ALLOCATED.  IF YOU HAVE NOT DONE SO, THIS PROGRAM WILL NOT
11.     COMM  WORK FOR YOU.  IF YOU NEED INSTRUCTIONS TO SET UP THIS DUMMY
12.     COMM  FILE, TYPE "YES" AFTER THE PROMPT "DO YOU NEED INSTRUCTIONS ?"
13.     COMM  OTHERWISE THIS PROGRAM WILL ASSUME THAT YOU HAVE ALREADY DONE
14.     COMM  THE PREREQUISITE STEPS.
15.     COMM
16.      READ STRING S7 PROMPT 'DO YOU NEED INSTRUCTIONS ? (Y/N)'
17.      IF (SUBSTR(S7,1) EQ 'Y') POINT ';CREATE'1 EXEC NOL
18.     COMM
19.     COMM  THIS PROGRAM NEEDS THE FOLLOWING INFORMATION.
20.     ;
21.     ;
22.     ;   CREATE PDS ?
23.     ;
24.     ;
25.     COMM
26.     COMM    NAME OF FILE TO BE DIVIDED -- YOUR ORIGINAL DATAFILE
27.     COMM    MASTER FILE NAME? -- MASTER NAME IN WHICH THE PARTITIONED
28.     COMM           DATA SETS WILL BE SAVED.
29.     COMM    MEMBER NAME? --  NAME OF THE INDIVIDUAL DATA FILE TO BE
30.     COMM         SAVED UNDER.  MUST BE LESS THAN 8 CHARACTERS.
31.     COMM
32.     COMM  REMEMBER THAT THE MASTER NAME CANNOT BE SAME AS THE
33.     COMM  ORIGINAL DATAFILE NAME!!!
34.     COMM
35.     COMM
36.     ;RETURN POINT
37.     ;
38.     ;
39.     ;  INPUT MASTER, ORIGINAL FILE NAME
40.     ;
41.     ;
42.      READ STRING S4 PROMPT 'NAME OF THE FILE TO BE DIVIDED: '
43.      READ STRING S2 PROMPT 'MASTER FILE NAME: '
44.      USE %S4 CLE
45.     ;START
46.      POINT '*BEGIN'1 NOL
47.      SET VAL W1=CURRENT
48.      POINT '*END'1 IN %W1/L NOL
49.      SET VAL W2=CURRENT
50.     ;
51.     ;
52.     ;  LIST EACH SEGMENT
53.     ;
54.     ;
55.      LIST %W1/%W2
56.     ; LIST DATA TO VERIFY THE FILE
57.      READ STRING S1 PROMPT 'SAVE THIS FILE? (Y/N) '
58.     IF (SUBSTR(S1,1) EQ 'Y') POINT ';SAVE'1 EXEC NOL
59.     DEL %W1/%W2
```

```
60.     COMM
61.     COMM  LET'S CONTINUE WITH THE NEXT SEGMENT....
62.     COMM
63.     POINT ';START'1 EXEC NOL
64.     ;
65.     ;
66.     ;  SAVE UNDER MEMBER NAME
67.     ;
68.     ;
69.     ;SAVE
70.     READ STRING S3 PROMPT 'MEMBER NAME: '
71.     SAVE %S2(%S3) LINES %W1/%W2
72.     DELETE %W1/%W2
73.     ;
74.     ;
75.     ;   CONTINUE ?
76.     ;
77.     ;
78.     ;  CHECK FOR END OF DATA FILE
79.      IF (W1 EQ 1) POINT ';QUIT'1 EXEC NOL
80.     ;
81.     ;
82.     COMM  LET'S CONTINUE WITH THE NEXT SEGMENT...
83.     POINT ';START'1 EXEC NOL
84.     ;
85.     ;
86.     ;CREATE DUMMY PARTITIONED DATASET
87.     ;
88.     ; INSTRUCTIONS TO SET UN DUMMY PARTITIONED DATASET
89.     COMM
90.     COMM  FOLLOWING INFORMATION IS NEEDED TO SET UP A PARTITIONED DATASET
91.     COMM
92.     COMM     DIRECTORY SIZE --  SINGLE DIRECTORY WILL HOLD ABOUT 10
93.     COMM          MEMBER FILES.  DIVIDE THE TOTAL NUMBER OF FILES BY 10
94.     COMM          AND ADD 20% TO THIS SIZE FOR LATER ADDIDTION.
95.     COMM     SPACE  --  NUMBER OF TRACKS TO BE ALLOCATED FOR THE DUMMY
96.     COMM          DATASET.  SPACE SHOULD BE ESTIMATED 20% HIGHER THAN THE
97.     COMM          CURRENT NUMBER OF TRACKS OCCUPIED BY THE ORIGINAL FILE
98.     COMM
99.     ;
100.      READ STRING S9 PROMPT 'DO YOU NEED THIS INFORMATION? (Y/N)'
101.      IF (SUBSTR(S9,1) EQ 'N') POINT ';DUMMY'1 EXEC NOL
102.      READ STRING S8 PROMPT 'YOUR DPSR ? '
103.      READ STRING S7 PROMPT 'YOUR ID ? '
104.      READ STRING S6 PROMPT 'ORIGINAL DATAFILE NAME ? '
105.      ;
106.    ;
107.      SHO DSN LIKE $USR.%S8.%S7.%S6 ALL
108.    ;
109.    ;
110.    ;  DPSR, ID, MASTER, SIZE, ETC.
111.    ;
112.    ;
113.    COMM
114.    ;DUMMY
115.      READ STRING S5 PROMPT 'DO YOU WANT TO SET UP DUMMY FILE ? (Y/N)'
116.      IF (SUBSTR(S5,1) EQ 'N') POINT ';QUIT'1 EXEC NOL
117.      READ STRING S9 PROMPT 'MASTER FILE NAME ? '
118.     READ STRING S8 PROMPT 'FIRST MEMBER FILE NAME? '
119.      READ VALUE N1 PROMPT 'DIRECTORY SIZE ? '
```

```
120.       READ VALUE N2 PROMPT 'SPACE SIZE ? '
121.     COMM
122.     COMM
123.     ;
124.     ;
125.     ;  SET UP MASTER PDS
126.     ;
127.     ;
128.      SAVE %S9(%S8) DIR %N1 CARD SPACE %N2
129.     COMM
130.     COMM
131.     COMM  YOU ARE NOW READY TO DIVIDE THE FILES.
132.     POINT ';RETURN'1 EXEC NOL
133.     ;QUIT
134.     COMM  HAVE A NICE DAY !!!!
135.     COMM
```

# APPENDIX B

```
1.    SET ESC &
2.    ;  THIS EXECUTE FILE AIDS THE USED IN PRODUCING THE SUMMARY REPORTS.
3.    ;  THIS EXECUTE FILE INTERACTS WITH THE USER TO OBTAIN INFORMATION
4.    ;  NEEDED TO PRODUCE THE SUMMARY REPORTS SUCH AS DATE, LOCATION,
5.    ;  TIME INTERVAL, WEATHER CONDITION, SPECIFIC DATA FILE, ETC.
6.    ;  IT THEN SELECTS THE SPECIFIED DATA FILE AND INITIATE THE ACTUAL
7.    ;  PROGRAM WHICH PRODUCES THE SUMMARY REPORTS.
8.    ;
9.    COMM  WELCOME TO CONVERSATIONAL COMPUTER SYSTEM.  IF YOU ARE ALREADY
10.   COMM  FAMILIAR WITH THIS PROGRAM PROCEDURES, TYPE "YES" TO SKIP THE
11.   COMM  INTRODUCTION.  OTHERWISE YOU WILL BE GIVEN THE INSTRUCTIONS ON
12.   COMM  HOW TO USE THIS PROGRAM.
13.   COMM
14.   ;
15.   ; FAMILIAR USER?
16.   ;
17.     READ STRING S9 PROMPT 'ARE YOU ALREADY FAMILIAR ? '
18.     IF (S9 EQ 'YES') POINT ';START'1 EXEC NOL
19.   COMM
20.   ;
21.   ;  INSTRUCTIONS TO RUN THE PROGRAM
22.   ;
23.   ;
24.   COMM  YOU ARE NOW IN A POSITION TO CONTROL THE COMPUTER PROCESSING.
25.   COMM  COMPUTER WILL PROMPT YOU FOR SOME SPECIFIC INFORMATION WHICH
26.   COMM  YOU MUST SUPPLY BEFORE PROPER PROCESSING CAN BEGIN.  YOUR
27.   COMM  COOPERATION IS A VITAL PART OF THIS PROGRAM.
28.   COMM
29.   COMM
30.   COMM
31.   COMM  IF YOU MAKE A TYPING ERROR DURING YOUR CONVERSATION WITH
32.   COMM THE COMPUTER, YOU CAN DO EITHER OF TWO THINGS.
33.   COMM
34.   COMM     1.  IF YOU HAVE ALREADY PRESSED "RETURN", YOU WILL SIMPLY
35.   COMM         HAVE TO START ALL OVER.
36.   COMM     2.  IF YOU HAVE NOT PRESSED "RETURN", SIMPLY BACK SPACE
37.   COMM         UNTIL YOU REACH YOUR ERROR, THEN RETYPE THE REMAINING
38.   COMM         LINE OVER AGAIN.  ONCE YOU BACK TRACK, YOU LOOSE THAT
39.   COMM         LETTER EVEN IF IT REMAINS ON YOUR TERMINAL.
40.   COMM
41.   COMM
42.   COMM  THANK YOU FOR YOUR COOPERATION.
43.   COMM
44.   COMM  LET'S  BEGIN....
45.   COMM
46.   COMM
47.   COMM
48.   ;
49.   ;
50.   ;  INPUT DATE, LOC, WEATHER, ETC.
51.   ;
52.   ;START PROCESSING
53.   COMM  TYPE IN DATE AS MM-DD-YY
54.    READ STRING S1 PROMPT 'DATE ? '
55.   ;
56.    READ STRING S2 PROMPT 'LOCATION ? '
57.    READ STRING S3 PROMPT 'DIRECTION OF FLOW ? '
58.    READ STRING S4 PROMPT 'WEATHER ? '
59.    READ STRING S5 PROMPT 'COUNTER ? '
```

```
 60.      READ STRING S6 PROMPT 'WHICH DATA FILE ? '
 61.      READ VALUE N1 PROMPT 'TIME INTERVAL IN MINUTES ? '
 62.     ;
 63.     ;
 64.     COMM
 65.     COMM  THANK YOU FOR YOUR COOPERATION
 66.     COMM
 67.     ;
 68.     ;
 69.     ;  PLACE REPORTER IN ACTIVE FILE SPACE
 70.     ;
 71.     ;
 72.      USE REPORTER CLE
 73.      CH '&DATE' TO '&S1' N
 74.      CH '&LOC' TO '&S2' N
 75.      CH '&DIR' TO '&S3' N
 76.      CH '&WEATHER' TO '&S4' N
 77.      CH '&COUNTER' TO '&S5' N
 78.      CH '&DELTA' TO &N1 N
 79.      CH '&TITLE' TO '&S6' N
 80.     ;
 81.     ;  SELECT THE DATA FILE TO BE READ IN
 82.     ;  ADD DATA FILE TO END OF THE PROGRAM
 83.     ;
 84.     ;
 85.     ;
 86.      COPY FROM  &S6  TO END
 87.      SET VALUE W0=LAST-1
 88.      &W0+.1 999999
 89.     ;
 90.     ;
 91.     ;  MAKE CHANGES FOR PRINTING OUT THE TOTAL VOLUMES
 92.     ;
 93.     ;
 94.      COMM   WHICH SUMMARY REPORT DO YOU WANT?
 95.      COMM      D - DAILY REPORT
 96.      COMM      W - WEEKLY REPORT
 97.      COMM      B - BOTH DAILY AND WEEKLY REPORTS
 98.      COMM
 99.      READ STRING S1 PROMPT 'TYPE IN YOUR CHOICE (D,W,B) '
100.      ;
101.      IF (S1 EQ 'D') CH '777' TO 'C77' NOL
102.      IF (S1 EQ 'W') CH '888' TO 'C88' NOL
103.     ;
104.     ;
105.      READ STRING S2 PROMPT 'DO YOU WANT THE CHANNEL TOTALS? (Y/N) '
106.      IF (S2 EQ 'Y') CH 'LT. 7 ' TO 'LT. 8' NOL
107.     ;
108.     ;
109.     ;  EXECUTE THE PROGRAM REPORTER
110.     ;
111.     ;
112.     ;
113.      RUN H
114.     COMM  IF YOU WISH TO CONTINUE PROCESSING, TYPE 'YES'
115.     COMM  AFTER THE PROMPT 'CONTINUE ?'.
116.     COMM  OTHERWISE, SIMPLY RETURN AND THIS PROGRAM WILL
117.     COMM  TERMINATE AUTOMATICALLY.
118.     ;
119.     ;  MORE FILES ?
```

```
120.      ;
121.      ;
122.      READ STRING S7 PROMPT 'CONTINUE ? '
123.       IF (S7 EQ 'YES') POINT ';START'1 EXEC NOL
124.      COMM  HAVE A NICE DAY!!!!!
125.      COMM
```

# APPENDIX C

```
1.      //&TITLE JOB (W262,501D,S02,003,JL),'REPORT'
2.      //*XBM WATFIV
3.      C
4.      C
5.      C
6.      C
7.      C*****************************************************************
8.      C*                                                               *
9.      C*   THIS PROGRAM WAS DEVELOPED TO AID THE USER IN REDUCING THE  *
10.     C*   TRAFFIC VOLUME DATA.  DATA IS TRANSFERRED FROM THE GOLDEN   *
11.     C*   RIVER RETRIEVER TO THE COMPUTER MEMORY BY DIRECT DATA COM-  *
12.     C*   MUNICATION PROCEDURE.  THE EXECUTE FILE RUNNER PASSES THE   *
13.     C*   HEADER INFORMATION TO THIS PROGRAM AND CONCATENATE THE DATA *
14.     C*   FILE TO THE END OF THIS PROGRAM.  THIS PROGRAM IS DESIGNED  *
15.     C*   WITH MANY SUBROUTINES TO EASE THE MODIFICATION OF THE THIS  *
16.     C*   PROGRAM AT A LATER DATE.  THIS PROGRAM CAN BE MODIFIED TO   *
17.     C*   WORK UNDER A DIFFERENT SYSTEM AS WELL AS FOR ADDITIONAL IN- *
18.     C*   FORMATION NEEDED BY THE USER.                               *
19.     C*                                                               *
20.     C*****************************************************************
21.     C
22.     C
23.           COMMON HEADER,TRAIL,TEST,CHOTOT(7,24)
24.           COMMON BIN1(200),BIN2(200),BIN3(200),BIN4(200)
25.           COMMON BIN5(200),BIN6(200),BIN7(200),BIN8(200)
26.           COMMON BIN9(200),BIN10(200),BIN11(200),BIN12(200)
27.           COMMON DATE(200),TIME(200)
28.           COMMON CONFIG,TCONFG,FILENO,TSDATE,MULTPL,MDAYZ(7,288)
29.           COMMON TFILEN,SITENO,TSITEN,INTVAL,TINTVL,STATUS,INDEX
30.           COMMON TSTATS,AMPEAK(7),PMPEAK(7),AMHIGH(7),PMHIGH(7),RTIME
31.           INTEGER P,NWTOT(7,7,24),CHECK,APEAK(7),PPEAK(7),AHI(7),PHI(7)
32.           CHARACTER*8 HEADER,TRAIL,TEST
33.           INTEGER  TSDATE,DATE,CHOTOT,TESDAY
34.           INTEGER COUNT,INDEX,TSTIME
35.           INTEGER  CONFIG,TCONFG,FILENO,TFILEN,SITENO,TSITEN
36.           INTEGER  ARMON(7),ARDAY(7)
37.           INTEGER  INTVAL,TINTVL,STATUS,TSTATS,TIME
38.           INTEGER  BIN1,BIN2,BIN3,BIN4
39.           INTEGER  BIN5,BIN6,BIN7,BIN8
40.           INTEGER  BIN9,BIN10,BIN11,BIN12,MDAYZ
41.           INTEGER  AMPEAK, PMPEAK, AMHIGH, PMHIGH,RTIME,IMAX
42.           INTEGER  ENDEX,D3,DAY,MONTH,YEAR
43.     C
44.     C
45.     C
46.           IFLAG=0
47.     C
48.     C   IFLAG IS USED TO DETECT THE MISMATCH OF THE TIME INTERVAL
49.     C
50.     C
51.           DO 10 I = 1,7
52.                APEAK(I) = 0
53.                PPEAK(I) = 0
54.                AHI(I)   = 0
55.                PHI(I)   = 0
56.     C
57.              DO 5 J = 1,24
58.     C
59.     C*****************************************************************
```

```
C*                                                                    *
C*     THIS SEGMENT OF THE PROGRAM IS DESIGNED TO SET UP A THREE      *
C*     DIMENSIONAL ARRAY IN ORDER TO ASSIST IN PRINTING OUT THE       *
C*     TABLE. THE FOLLOWING IS A DESCRIPTION OF THE ARRAY 'NWTOT'     *
C*          K - THE FIRST SUBSCRIPT IS FOR THE CHANNEL NUMBER 1 - 6*
C*          I - THE SECOND SUBSCRIPT IS FOR THE DAY OF WEEK (1 - 7)*
C*          J - THE LAST SUBSCRIPT IS FOR THE NUMBER OF HOURS      *
C*               IN THE DAY (1 - 24)                               *
C*                                                                    *
C**********************************************************************
          DO 4 K = 1,7
          NWTOT(K,I,J) = 0
4         CONTINUE
C**********************************************************************
5         CONTINUE
10        CONTINUE
C
C
C
C
C
C
C
C
C
C  CALL SUBROUTINE READER TO READ THE DATA FROM DISK FILE
C
          CALL READER(IMAX)
C
C
C   CALL SUBROUTINE DAWEEK TO FIND THE DAY-OF-WEEK GIVEN THE
C   DATE OF DATA COLLECTION.
C
          TSDATE = DATE(1)
          TESDAY = DATE(1)
          CALL DAWEEK(TSDATE,ENDEX,ARMON,ARDAY,CHECK,DAY,MONTH,YEAR)
C
          P = CHECK
C
          IND=ENDEX*24+1
          IF (IND .EQ. 0) IND = 1
C
C
C
          INDEX=1
          ITIME = TIME(INDEX)/100
          IT = 1
          IJ = 1
          WHILE(INDEX .LE. IMAX) DO
C
C   INDEX IS THE VARIABLE THAT CHECKS AGAINST THE END OF FILE
C   CONTINUE LOOP UNTIL END OF FILE HAS BEEN REACHED
C
C
C
C  CALL SUBROUTINE INITAL TO INITIALIZE ARRAYS
C
           CALL  INITAL(MDAYZ,CHOTOT)
C
C
```

```
120.     C    CALL SUBROUTINE PREP TO PREPARE THE OUTPUT REPORT
121.     C
122.     C
123.               CALL PREP (P,NWTOT,D3,IJ,ITIME,IT,IFLAG)
124.     C
125.     C  CHECK FOR ABORTED ENDING DUE TO DIFFERENT TIME INTERVALS
126.     C
127.     C
128.              IF (IFLAG .EQ. 1) GOTO 99
129.     C
130.     C
131.     C
132.                P = P + 1
133.     C
134.     C
135.     C
136.               IF (TIME(INDEX) .GT.0) ITIME = TIME(INDEX)/100
137.     C
138.     C
139.     C    CALL SUBROTINE DAILY TO PRINT OUT DAILY REPORT
140.     C
141.     888       CALL DAILY (APEAK,PPEAK,AHI,PHI,P,D3,DAY,MONTH,YEAR)
142.     C
143.                    DAY = DAY + 1
144.     C
145.     C
146.     C  THIS IS THE END OF THE PROGRAM.  LOOPING WITHIN THE SUBROUTINES
147.     C  ARE DONE THROUGH SUBROUTINE CALLS INSIDE SUBROUTINES.
148.     C
149.     C
150.            END WHILE
151.     C
152.     C
153.     777       CALL WEEKLY(ARMON,ARDAY,NWTOT,APEAK,PPEAK,AHI,PHI,D3,
154.     7777 *                  SITENO,TESDAY)
155.     C
156.     99       STOP
157.              END
158.     C
159.     C
160.     C*********************************************************************
161.     C********                                                      ******
162.     C********       FOLLOWING ARE SUBROUTINES                      ******
163.     C********                                                      ******
164.     C*********************************************************************
165.     C
166.     C
167.     C
168.     C*********************************************************************
169.     C*                                                                   *
170.     C*     SUBROUTINE READER -- READS THE ENTIRE DATA FILE AND STORE     *
171.     C*          THE DATA INTO BIN1 - BIN12 ARRAYS.  PRINTS OUT THE       *
172.     C*          DATA AT THE END OF THE SUBROUTINE TO VERIFY THE DATA     *
173.     C*                                                                   *
174.     C*********************************************************************
175.     C
176.              SUBROUTINE READER(IMAX)
177.     C
178.     C
179.     C
```

```
180.            COMMON HEADER,TRAIL,TEST,CHOTOT(7,24)
181.            COMMON BIN1(200),BIN2(200),BIN3(200),BIN4(200)
182.            COMMON BIN5(200),BIN6(200),BIN7(200),BIN8(200)
183.            COMMON BIN9(200),BIN10(200),BIN11(200),BIN12(200)
184.            COMMON DATE(200),TIME(200)
185.            COMMON CONFIG,TCONFG,FILENO,TSDATE,MULTPL,MDAYZ(7,288)
186.            COMMON TFILEN,SITENO,TSITEN,INTVAL,TINTVL,STATUS,INDEX
187.            COMMON TSTATS,AMPEAK(7),PMPEAK(7),AMHIGH(7),PMHIGH(7),RTIME
188.            CHARACTER*8 HEADER,TRAIL,TEST
189.            INTEGER P,NWTOT(7,7,24),CHECK,APEAK(7),PPEAK(7),AHI(7),PHI(7)
190.            INTEGER CHECKR
191.            INTEGER  TSDATE,DATE,CHOTOT
192.            INTEGER COUNT,INDEX,TSTIME
193.            INTEGER  CONFIG,TCONFG,FILENO,TFILEN,SITENO,TSITEN
194.            INTEGER  INTVAL,TINTVL,STATUS,TSTATS,TIME
195.            INTEGER  BIN1,BIN2,BIN3,BIN4
196.            INTEGER  BIN5,BIN6,BIN7,BIN8
197.            INTEGER  BIN9,BIN10,BIN11,BIN12,MDAYZ
198.            INTEGER  AMPEAK, PMPEAK, AMHIGH, PMHIGH,RTIME,IMAX
199.     C
200.     C
201.     C  READ THE HEADER LINE
202.     C
203.            READ(5,201) HEADER,CONFIG,FILENO,SITENO,INTVAL,STATUS
204.     201    FORMAT(A6,1X,I2,1X,I2,1X,I8,1X,I4,1X,I2)
205.     C
206.     C
207.     C  SET UP LOOP TO READ AND STORE ALL THE DATA
208.     C
209.     C
210.            I=1
211.            J=1
212.     C
213.     C      READ THE DATA AND STORE INTO ARRAYS
214.     C
215.     49     READ(5,202) DATE(I),TIME(I),BIN1(I),BIN2(I),BIN3(I),BIN4(I),
216.          *             BIN5(I),BIN6(I),BIN7(I),BIN8(I),BIN9(I),BIN10(I),
217.          *             BIN11(I),BIN12(I)
218.     C
219.     C
220.     C
221.     C
222.     C
223.     202    FORMAT(I6,1X,I4,2X,12(I4,1X))
224.           IF (DATE(I) .EQ. 999999 ) GOTO 1001
225.            I=I+1
226.           GOTO 49
227.     1001   READ(5,201)TRAIL,TCONFG,TFILEN,TSITEN,TINTVL,TSTATS
228.     C
229.     C   COMPLETED READING THE ENTIRE FILE
230.     C
231.              IMAX=I-1
232.            WRITE(6,403) HEADER,CONFIG,FILENO,SITENO,INTVAL,STATUS
233.           DO 13000 J=1,IMAX
234.           WRITE(6,402)DATE(J),TIME(J),BIN1(J),BIN2(J),BIN3(J),BIN4(J),
235.          *             BIN5(J),BIN6(J),BIN7(J),BIN8(J),BIN9(J),BIN10(J),
236.          *             BIN11(J),BIN12(J)
237.     13000   CONTINUE
238.            WRITE(6,404)TRAIL,TCONFG,TFILEN,TSITEN,TINTVL,TSTATS
239.     403    FORMAT('1',10X,A6,1X,I2,1X,I2,1X,I8,1X,I4,1X,I2,/)
```

```
404    FORMAT(/,10X,A6,1X,I2,1X,I2,1X,I8,1X,I4,1X,I2,/)
402    FORMAT(10X,I6,1X,I4,2X,12(I4,1X))
C
C
       RETURN
       END
C
C
C*****************************************************************
C*                                                               *
C*    SUBROUTINE PREP -- PREP REDUCES THE DATA ACCORDING TO THE  *
C*      TIME INTERVAL AND THE CONFIGURATION.  IT CALCULATES THE  *
C*      MINUTE INTERVAL TOTAL, HOURLY TOTAL, RUNNING SUBTOTAL,   *
C*      AND PEAK HOUR TOTALS.  THIS SUBROUTINE IS THE MAIN CORE  *
C*      OF THE ENTIRE PROGRAM.                                   *
C*                                                               *
C*****************************************************************
       SUBROUTINE PREP(P,NWTOT,D3,IJ,ITIME,IT,IFLAG)
C
C
       COMMON HEADER,TRAIL,TEST,CHOTOT(7,24)
       COMMON BIN1(200),BIN2(200),BIN3(200),BIN4(200)
       COMMON BIN5(200),BIN6(200),BIN7(200),BIN8(200)
       COMMON BIN9(200),BIN10(200),BIN11(200),BIN12(200)
       COMMON DATE(200),TIME(200)
       COMMON CONFIG,TCONFG,FILENO,TSDATE,MULTPL,MDAYZ(7,288)
       COMMON TFILEN,SITENO,TSITEN,INTVAL,TINTVL,STATUS,INDEX
       COMMON TSTATS,AMPEAK(7),PMPEAK(7),AMHIGH(7),PMHIGH(7),RTIME
       CHARACTER*8 HEADER,TRAIL,TEST
       INTEGER P,NWTOT(7,7,24),CHECK,APEAK(7),PPEAK(7),AHI(7),PHI(7)
       INTEGER  TSDATE,DATE,CHOTOT
       INTEGER COUNT,INDEX,TSTIME
       INTEGER  CONFIG,TCONFG,FILENO,TFILEN,SITENO,TSITEN
       INTEGER  INTVAL,TINTVL,STATUS,TSTATS,TIME
       INTEGER  BIN1,BIN2,BIN3,BIN4
       INTEGER  BIN5,BIN6,BIN7,BIN8
       INTEGER  BIN9,BIN10,BIN11,BIN12,MDAYZ
       INTEGER  AMPEAK, PMPEAK, AMHIGH, PMHIGH,RTIME,IMAX,D3
       INTEGER P2,P1
C
       FIRST=1
C
C  CALCULATE MINUTE TOTAL AND HOURLY TOTAL
C
C
C
       TSDATE=DATE(INDEX)
       TSTIME=TIME(INDEX)
       INDX = INDEX
       RTIME=&DELTA
C
C
C   CHECK FOR THE INTERVAL TIME BEFORE REDUCING THE DATA
C   IF THE TIME INTERVAL IS DIFFERENT, THEN PRINT THE ERROR
C   MESSAGE AND ABORT.
C
C
       IF (RTIME .NE. INTVAL) THEN DO
         WRITE(6,21)
21       FORMAT (///,110('*'),/,' ',T20,'***  ERROR ****',
```

```
300.          *            T20,'TIME INTERVAL SUPPLIED BY THE USER DOES NOT',
301.          *            'MATCH THE DATA FILE INFORMATION',//,' ',
302.          *            110('*'))
303.               IFLAG=1
304.               RETURN
305.             END IF
306.      C
307.      C
308.      C
309.             P1 = 0
310.             MULTPL=60./RTIME
311.             COUNT=(TSTIME/100)*MULTPL + (MOD(TSTIME,100)/RTIME)
312.             IF (COUNT .EQ. 0) COUNT=1
313.      222    WHILE (DATE(INDEX) .EQ. TSDATE) DO
314.      C
315.      C
316.      C   SET UP LOOP TO CALCULATE THE MINUTE TOTALS
317.      C
318.               IF (CONFIG .LE. 3) THEN DO
319.                  MDAYZ(7,COUNT)=BIN1(INDEX)
320.                  MDAYZ(7,COUNT+1)=BIN2(INDEX)
321.                  MDAYZ(7,COUNT+2)=BIN3(INDEX)
322.                  MDAYZ(7,COUNT+3)=BIN4(INDEX)
323.                  MDAYZ(7,COUNT+4)=BIN5(INDEX)
324.                  MDAYZ(7,COUNT+5)=BIN6(INDEX)
325.                  MDAYZ(7,COUNT+6)=BIN7(INDEX)
326.                  MDAYZ(7,COUNT+7)=BIN8(INDEX)
327.                  MDAYZ(7,COUNT+8)=BIN9(INDEX)
328.                  MDAYZ(7,COUNT+9)=BIN10(INDEX)
329.                  MDAYZ(7,COUNT+10)=BIN11(INDEX)
330.                  MDAYZ(7,COUNT+11)=BIN12(INDEX)
331.      C*****************************************************
332.      C*                                                   *
333.      C*   THIS SEGMENT PREPARES DATA FOR INDIVIDUAL LANE  *
334.      C*   DAILY TIME INTERVAL REPORT.  IT STORES THE      *
335.      C*  VALUES INTO MDAYZ(7,288) WHICH COVERS ALL SIX    *
336.      C*   LANES IN THE CASE OF SIX CHANNEL CONFIGURATION. *
337.      C*                                                   *
338.      C*****************************************************
339.      C
340.                  MDAYZ(FIRST,COUNT)=BIN1(INDEX)
341.                  MDAYZ(FIRST,COUNT+1)=BIN2(INDEX)
342.                  MDAYZ(FIRST,COUNT+2)=BIN3(INDEX)
343.                  MDAYZ(FIRST,COUNT+3)=BIN4(INDEX)
344.                  MDAYZ(FIRST,COUNT+4)=BIN5(INDEX)
345.                  MDAYZ(FIRST,COUNT+5)=BIN6(INDEX)
346.                  MDAYZ(FIRST,COUNT+6)=BIN7(INDEX)
347.                  MDAYZ(FIRST,COUNT+7)=BIN8(INDEX)
348.                  MDAYZ(FIRST,COUNT+8)=BIN9(INDEX)
349.                  MDAYZ(FIRST,COUNT+9)=BIN10(INDEX)
350.                  MDAYZ(FIRST,COUNT+10)=BIN11(INDEX)
351.                  MDAYZ(FIRST,COUNT+11)=BIN12(INDEX)
352.      C
353.      C
354.                  COUNT=COUNT+12
355.                  TSDATE=DATE(INDEX)
356.                  INDEX=INDEX+1
357.      C
358.      C*************************************
359.      C*        DIMENSION 1 FOR SINGLE CHANNEL*
```

```
C*****************************************
C*
                D3 = 1
C
C
              ELSE DO
                IF (CONFIG .LE. 15) THEN DO
                  MDAYZ(7,COUNT)=BIN1(INDEX)+BIN2(INDEX)
                  MDAYZ(7,COUNT+1)=BIN3(INDEX)+BIN4(INDEX)
                  MDAYZ(7,COUNT+2)=BIN5(INDEX)+BIN6(INDEX)
                  MDAYZ(7,COUNT+3)=BIN7(INDEX)+BIN8(INDEX)
                  MDAYZ(7,COUNT+4)=BIN9(INDEX)+BIN10(INDEX)
                  MDAYZ(7,COUNT+5)=BIN11(INDEX)+BIN12(INDEX)
C
C*********************************************************
C*    THIS SEGMENT PREPARES DATA FOR DUAL LANE REPORT    *
C*    TWO OF THE SIX CHANNELS ARE USED IN THIS ARRAY     *
C*********************************************************
C
C
                  MDAYZ(FIRST,COUNT)=BIN1(INDEX)
                  MDAYZ(FIRST,COUNT+1)=BIN3(INDEX)
                  MDAYZ(FIRST,COUNT+2)=BIN5(INDEX)
                  MDAYZ(FIRST,COUNT+3)=BIN7(INDEX)
                  MDAYZ(FIRST,COUNT+4)=BIN9(INDEX)
                  MDAYZ(FIRST,COUNT+5)=BIN11(INDEX)
                  MDAYZ(FIRST+1,COUNT)=BIN2(INDEX)
                  MDAYZ(FIRST+1,COUNT+1)=BIN4(INDEX)
                  MDAYZ(FIRST+1,COUNT+2)=BIN6(INDEX)
                  MDAYZ(FIRST+1,COUNT+3)=BIN8(INDEX)
                  MDAYZ(FIRST+1,COUNT+4)=BIN10(INDEX)
                  MDAYZ(FIRST+1,COUNT+5)=BIN12(INDEX)
C
C
                  COUNT=COUNT+6
                  TSDATE=DATE(INDEX)
                  INDEX=INDEX+1
C
C*****************************************
C*     DIMENSION 2 FOR DUAL CHANNEL      *
C*****************************************
C
                  D3 = 2
C
                ELSE DO
                  IF (CONFIG .EQ. 18) THEN DO
                MDAYZ(7,COUNT)=BIN1(INDEX)+BIN2(INDEX)+BIN3(INDEX)+
     *                    BIN4(INDEX)+BIN5(INDEX)+BIN6(INDEX)
                MDAYZ(7,COUNT+1)=BIN7(INDEX)+BIN8(INDEX)+BIN9(INDEX)+
     *                     BIN10(INDEX)+BIN11(INDEX)+BIN12(INDEX)
C
C
C*****************************************************
C*    THIS SEGMENT PREPARES DATA FOR SIX CHANNEL     *
C*    CONFIGURATION.  IT UTILIZES ALL SIX ROWS OF    *
C*    THE ARRAY MDAYZ.  IT WILL HOLD VALUES TO PRINT*
C*    ALL SIX REPORTS IF NECESSARY.                  *
C*****************************************************
C
C
```

```
420.                  MDAYZ(FIRST,COUNT)=BIN1(INDEX)
421.                  MDAYZ(FIRST+1,COUNT)=BIN2(INDEX)
422.                  MDAYZ(FIRST+2,COUNT)=BIN3(INDEX)
423.                  MDAYZ(FIRST+3,COUNT)=BIN4(INDEX)
424.                  MDAYZ(FIRST+4,COUNT)=BIN5(INDEX)
425.                  MDAYZ(FIRST+5,COUNT)=BIN6(INDEX)
426.                  MDAYZ(FIRST,COUNT+1)=BIN7(INDEX)
427.                  MDAYZ(FIRST+1,COUNT+1)=BIN8(INDEX)
428.                  MDAYZ(FIRST+2,COUNT+1)=BIN9(INDEX)
429.                  MDAYZ(FIRST+3,COUNT+1)=BIN10(INDEX)
430.                  MDAYZ(FIRST+4,COUNT+1)=BIN11(INDEX)
431.                  MDAYZ(FIRST+5,COUNT+1)=BIN12(INDEX)
432.     C
433.     C
434.                 COUNT=COUNT+2
435.                 TSDATE=DATE(INDEX)
436.                 INDEX=INDEX+1
437.     C
438.     C*************************************************
439.     C*     DIMENSION 6 FOR MULTI-CHANNEL              *
440.     C*************************************************
441.     C
442.                  D3 = 6
443.     C
444.                   ELSE DO
445.                     MDAYZ(7,COUNT)=BIN1(INDEX)+BIN2(INDEX)+BIN3(INDEX)+
446.          *              BIN4(INDEX)+BIN5(INDEX)+BIN6(INDEX)+BIN7(INDEX)+
447.          *              BIN8(INDEX)+BIN9(INDEX)+BIN10(INDEX)+BIN11(INDEX)+
448.          *              BIN12(INDEX)
449.                      COUNT=COUNT+1
450.                      TSDATE=DATE(INDEX)
451.                      INDEX=INDEX+1
452.                   ENDIF
453.                  ENDIF
454.                 ENDIF
455.     C
456.     C
457.     C
458.              END WHILE
459.     C
460.     C
461.     C  SET DO LOOP TO CALCULATE THE HOURLY TOTAL VOLUME
462.     C
463.             NTEVAL=60/INTVAL
464.             MAXI=24*NTEVAL
465.     C
466.     C
467.     C
468.     C   NTEVAL ---  NUMBER OF ELEMENTS TO ADD TO FIND HOURLY TOTAL
469.     C
470.     C*************************************************
471.     C**  LOOP THROUGH SIX TIMES TO FIND HOURLY TOTALS *
472.     C**  OF ALL THE CHANNELS FOR LATER USAGE          *
473.     C*************************************************
474.     C
475.     C
476.             DO 6000 M=1,7
477.             NPLUS=0
478.             L = 1
479.             K=0
```

```
480.     C
481.             DO 1000 I=1,MAXI,NTEVAL
482.               NPLUS=NPLUS+NTEVAL
483.               K=K+1
484.               DO 100  J=I,NPLUS
485.                  CHOTOT(M,K)=CHOTOT(M,K)+MDAYZ(M,J)
486.     100      CONTINUE
487.               NWTOT(M,P,K)=CHOTOT(M,K)
488.     1000     CONTINUE
489.     6000   CONTINUE
490.     C
491.     C*****************************************************************
492.     C*                                                               *
493.     C*      THE NEXT SEGMENT TO THIS ROUTINE IS TO DETERMINE IF THE  *
494.     C*      CHANNEL IS A DUAL CHANNEL, AND IF SO, IT WILL COMPUTE    *
495.     C*      THE NECESSARY HOURLY TOTALS FOR EACH OF THE CHANNELS     *
496.     C*                                                               *
497.     C*       EXAMPLE:                                                *
498.     C*             NWTOT(1,K,J) - WILL CONTAIN INFORMATION FOR LANE 1*
499.     C*             NWTOT(2,K,J) - WILL CONTAIN INFORMATION FOR LANE 2*
500.     C*                                                               *
501.     C*****************************************************************
502.     C*                                                               *
503.     C*      TAKE NOTE THAT THE THE FIRST DIMENSION OF THE ARRAY      *
504.     C*      'NWTOT' HAS BEEN SET TO A MAXIMUN LIMIT OF '6' FOR SIX   *
505.     C*      LANES.  THIS HAS BEEN DONE FOR FUTURE USE OF A SIX CHANNEL*
506.     C*      TABLE.                                                   *
507.     C*                                                               *
508.     C** ***************************************************************
509.     C
510.     1999      RETURN
511.               END
512.     C
513.     C*****************************************************************
514.     C*                                                               *
515.     C*      SUBROUTINE DAILY --  THIS SUBROUTINE PRODUCES THE DAILY  *
516.     C*         REPORT WHICH INCLUDES THE MINUTE INTERVAL SUMS, HOURLY *
517.     C*         TOTALS, AM AND PM PEAK VOLUME COUNTS WITH THE HEADINGS *
518.     C*                                                               *
519.     C*****************************************************************
520.     C
521.     C
522.             SUBROUTINE DAILY(APEAK,PPEAK,AHI,PHI,P,D3,DAY,MONTH,YEAR)
523.     C
524.             COMMON HEADER,TRAIL,TEST,CHOTOT(7,24)
525.             COMMON BIN1(200),BIN2(200),BIN3(200),BIN4(200)
526.             COMMON BIN5(200),BIN6(200),BIN7(200),BIN8(200)
527.             COMMON BIN9(200),BIN10(200),BIN11(200),BIN12(200)
528.             COMMON DATE(200),TIME(200)
529.             COMMON CONFIG,TCONFG,FILENO,TSDATE,MULTPL,MDAYZ(7,288)
530.             COMMON TFILEN,SITENO,TSITEN,INTVAL,TINTVL,STATUS,INDEX
531.             COMMON TSTATS,AMPEAK(7),PMPEAK(7),AMHIGH(7),PMHIGH(7),RTIME
532.             DIMENSION NUAMHI(7),NUPMHI(7)
533.             CHARACTER*8 HEADER,TRAIL,TEST
534.             CHARACTER*2 CAMMIN(7),CPMMIN(7)
535.             INTEGER NWTOT(7,7,24)
536.             INTEGER  TSDATE,DATE,CHOTOT
537.             INTEGER COUNT,INDEX,TSTIME
538.             INTEGER  CONFIG,TCONFG,FILENO,TFILEN,SITENO,TSITEN
539.             INTEGER  INTVAL,TINTVL,STATUS,TSTATS,TIME
```

```
540.            INTEGER  BIN1,BIN2,BIN3,BIN4
541.            INTEGER  BIN5,BIN6,BIN7,BIN8
542.            INTEGER  BIN9,BIN10,BIN11,BIN12,MDAYZ
543.            INTEGER  AMPEAK, PMPEAK, AMHIGH, PMHIGH,RTIME
544.            INTEGER APEAK(7),PPEAK(7),AHI(7),PHI(7),P
545.            INTEGER  HILIM,HIGH,AMINDX(7),PMINDX(7)
546.            INTEGER  RUNSUB(7,24),T,B,D3,DAY,MONTH,YEAR
547.            INTEGER AMHOUR(7),PMHOUR(7),AMMIN(7),PMMIN(7)
548.            DATA CAMMIN/7*'99'/, CPMMIN /7*'99'/
549.     C
550.           DO 6 K=1,7
551.            DO 5 T = 1,24
552.            RUNSUB(K,T) = 0
553.     5      CONTINUE
554.     6     CONTINUE
555.     C
556.            DELTA=RTIME/100.
557.     C
558.             B = 1
559.                DO 601 N1=1,7
560.     C
561.                   RUNSUB(N1,1)=CHOTOT(N1,1)
562.     C
563.                  DO 602 N2=2,24
564.                     RUNSUB(N1,N2)=RUNSUB(N1,N2-1) + CHOTOT(N1,N2)
565.     C
566.     602           CONTINUE
567.     C
568.              K3=1
569.              K4=1
570.     601        CONTINUE
571.     C
572.     C
573.                COLONE=K3
574.                COLTWO=6*MULTPL
575.                COLTRE=12*MULTPL
576.                COLFOR=18*MULTPL
577.     C
578.     C
579.     C   CALCULATE THE AM-PEAK AND PM-PEAK VOLUME
580.     C   CALCUALTE THE AM-HIGH ANDP PM-HIGH VOLUME
581.     C
582.     C
583.     C*********************************************************
584.     C**   ZERO OUT THE ARRAYS TO HOLD THE PEAK VOLUMES     **
585.     C*********************************************************
586.     C
587.             DO 700 J1=1,7
588.             AMHIGH(J1)=0
589.             PMHIGH(J1)=0
590.             AMPEAK(J1)=0
591.             PMPEAK(J1)=0
592.     700     CONTINUE
593.     C
594.     C   SET UP LOOPS TO CALCULATE AMPEAK, PMPEAK, AMHIGH
595.     C   AND PMHIGH VOLUMES
596.     C
597.     C
598.             LOW=4*MULTPL
599.             HIGH=11*MULTPL
```

```
600.              LOWX2=12*MULTPL
601.      C
602.      C   LOW -- LOWER BOUND FOR THE LOOP TO CALCULATE VALUES
603.      C   HIGH - HIGHER   ""       "            "          "
604.      C
605.      C
606.             DO 35 IT=1,7
607.              AMINDX(IT)=0
608.              PMINDX(IT)=0
609.              AMHOUR(IT)=0
610.              AMMIN(IT)=0
611.              PMHOUR(IT)=0
612.              PMMIN(IT)=0
613.      C
614.      C
615.      C
616.              DO  33 I1=LOW,HIGH
617.      C   INITIALIZE NUAMHI AND NUPMHI TO ZEROS
618.      C
619.                 NUAMHI(IT)=0
620.                 NUPMHI(IT)=0
621.      C
622.               DO 34 I2=1,MULTPL
623.                  NUAMHI(IT)=NUAMHI(IT)+MDAYZ(IT,I1+I2)
624.                  NUPMHI(IT)=NUPMHI(IT)+MDAYZ(IT,LOWX2+I2)
625.      34         CONTINUE
626.      C
627.      C   CHECK TO EXCHANGE THE HIGHER VALUES TO AMHIGH AND PMHIGH
628.      C
629.               IF (NUAMHI(IT) .GT. AMHIGH(IT)) THEN DO
630.                  AMHIGH(IT)=NUAMHI(IT)
631.                  AMINDX(IT)=I1
632.                ENDIF
633.      C
634.               IF (NUPMHI(IT) .GT. PMHIGH(IT)) THEN DO
635.                  PMHIGH(IT)=NUPMHI(IT)
636.                  PMINDX(IT)=LOWX2
637.                ENDIF
638.      C
639.      C
640.              LOWX2=LOWX2+1
641.      C    INCREMENT LOWX2 TO ADD THE NUMBERS TO CALCULATE PMHIGH
642.       33      CONTINUE
643.      C
644.      C    RESET  LOWX2 BACK TO ORIGINAL VALUE FOR FURTHER PRINT OUT
645.      C
646.               LOWX2=12*MULTPL
647.      C
648.      35      CONTINUE
649.      C
650.      C
651.      C    FIND THE AM AND PM PEAK VOLUME STARTING TIME
652.      C
653.      C
654.               DO 751 N1=1,7
655.                  AMHOUR(N1)=AMINDX(N1)/MULTPL
656.                  AMMIN(N1)=(MOD(AMINDX(N1),MULTPL)*INTVAL)
657.      C
658.      C
659.      C    CONVERT TO THE CHARACTER SET FOR MINUTES LESS THAN 10
```

```
660.      C
661.            IF (AMMIN(N1) .LT. 10) THEN DO
662.               IF (AMMIN(N1) .EQ. 0) CAMMIN(N1)='00'
663.               IF (AMMIN(N1) .EQ. 1) CAMMIN(N1)='01'
664.               IF (AMMIN(N1) .EQ. 2) CAMMIN(N1)='02'
665.               IF (AMMIN(N1) .EQ. 3) CAMMIN(N1)='03'
666.               IF (AMMIN(N1) .EQ. 4) CAMMIN(N1)='04'
667.               IF (AMMIN(N1) .EQ. 5) CAMMIN(N1)='05'
668.               IF (AMMIN(N1) .EQ. 6) CAMMIN(N1)='06'
669.               IF (AMMIN(N1) .EQ. 7) CAMMIN(N1)='07'
670.               IF (AMMIN(N1) .EQ. 8) CAMMIN(N1)='08'
671.               IF (AMMIN(N1) .EQ. 9) CAMMIN(N1)='09'
672.            ENDIF
673.      C
674.      C
675.            IF (PMMIN(N1) .LT. 10) THEN DO
676.               IF (PMMIN(N1) .EQ. 0) CPMMIN(N1)='00'
677.               IF (PMMIN(N1) .EQ. 1) CPMMIN(N1)='01'
678.               IF (PMMIN(N1) .EQ. 2) CPMMIN(N1)='02'
679.               IF (PMMIN(N1) .EQ. 3) CPMMIN(N1)='03'
680.               IF (PMMIN(N1) .EQ. 4) CPMMIN(N1)='04'
681.               IF (PMMIN(N1) .EQ. 5) CPMMIN(N1)='05'
682.               IF (PMMIN(N1) .EQ. 6) CPMMIN(N1)='06'
683.               IF (PMMIN(N1) .EQ. 7) CPMMIN(N1)='07'
684.               IF (PMMIN(N1) .EQ. 8) CPMMIN(N1)='08'
685.               IF (PMMIN(N1) .EQ. 9) CPMMIN(N1)='09'
686.            ENDIF
687.      C
688.      C
689.      C
690.                PMHOUR(N1)=PMINDX(N1)/MULTPL
691.                PMMIN(N1) =(MOD(PMINDX(N1),MULTPL)*INTVAL)
692.      C
693.      751      CONTINUE
694.      C
695.      C
696.      C  SET UP LOOP TO FIND AMPEAK AND PMPEAK VALUES
697.      C  BY SIMPLY ADDING UP THE VALUES OF 6:30-9:30 AND
698.      C  15:30-18:30 RESPECTIVELY.
699.      C
700.            LOWLIM = INT(6.5*MULTPL)
701.            MAXLIM=3*MULTPL
702.            HILIM= INT(15.5*MULTPL)
703.      C
704.      C
705.      C
706.            DO 46 I4=1,7
707.      C
708.            DO 43 I3=1,MAXLIM
709.               AMPEAK(I4)=AMPEAK(I4)+MDAYZ(I4,LOWLIM+I3)
710.               PMPEAK(I4)=PMPEAK(I4)+MDAYZ(I4,HILIM+I3)
711.      43       CONTINUE
712.      C
713.      46     CONTINUE
714.      C
715.      C    PRINT OUT THE SUMMARY PORTION OF THE REPORT
716.      C
717.      C      APEAK(P) = AMPEAK
718.      C      PPEAK(P) = PMPEAK
719.      C      AHI(P)   = AMHIGH
```

```
C      PHI(P)   = PMHIGH
C
C*****************************************************
C**   START LOOP TO PRINT OUT SIX DAILY REPORTS.   ***
C*****************************************************
C
        I6=1
        WHILE(I6 .LT. 7 ) DO
C
C   CHECK FOR AM AND PM PEAK VOLUMES TO PRINT OUT THE NEXT
C   CHANNEL REPORT.  IF THE PEAK VOLUMES ARE ZERO THEN DO
C   NOT PRINT OUT THE NEXT CHANNEL.
C
         IF (RUNSUB(I6,24) .NE. 0) THEN DO
C
C
C
CC
        WRITE(6,11)
11      FORMAT('1')
        WRITE(6,21)
21      FORMAT(' ')
        WRITE(6,31)
31      FORMAT(T40,'FREEWAY VOLUME COUNT - TEXAS TRANSPORTATION',
     *' INSTITUTE',///)
        WRITE(6,41)MONTH,DAY,YEAR,SITENO,I6,D3
41     FORMAT(T10,'DATE:',3X,I2,'-',I2,'-',I4,T50,'SITE NO:',
     *       3X,I8,T85,'COUNTER:',3X,'&COUNTER',/,
     *      T10,'WEATHER:',3X,'&WEATHER',
     *       T50,'DIRECTION OF FLOW:',3X,'&DIR',/,
     *      T10,'LOCATION:',3X,
     *      '&LOC',//,
     *      T10,'SOURCE :',10X,'THIS IS ',I1,' OF ',I1,//)
C
C
C
        WRITE(6,51)
51      FORMAT(T5,125('_'))
        WRITE(6,61)
61      FORMAT(T26,'HOURLY',T57,'HOURLY',T88,'HOURLY',T119,'HOURLY')
        WRITE(6,71)
71      FORMAT(4(9X,'TIME',4X,'VOLUME',2X,'VOLUME'),/)
        WRITE(6,51)
        WRITE(6,56)
56      FORMAT(' ',T5,125('_'))
        KOUNT = 1
        COL1 = 0.0
        COL2 = 6.0
        COL3 = 12.0
        COL4 = 18.0
        TDELTA = 0.0
C
C   START A LOOP TO COUNT THE TIMES AND PRINT OUT MINUTE TOTALS
C
C
C   K3 WILL BE USED TO WRITE OUT THE MINUTE & HOURLY TOTALS
C
         COL42 = 0.0
          WHILE (COL42 .LE. (24.0-DELTA)) DO
C
```

```
780.     C   IF COL42 24. THEN END OF THAT DAY HAS REACHED
781.     C
782.     C   KEEP COUNTER FOR PAGE EJECT IN CASE OF TIME INTERVAL 15 MINUTES
783.     C
784.                 KOUNT = KOUNT+1
785.     C
786.               TDELTA = DELTA + TDELTA
787.                   COL12 = COL1 + DELTA
788.                   COL22 = COL2 + DELTA
789.                   COL32 = COL3 + DELTA
790.                   COL42 = COL4 + DELTA
791.                IF (TDELTA .GE. (0.6-DELTA)) THEN DO
792.                       COL12 = COL12 + 0.4
793.                       COL22 = COL22 + 0.4
794.                       COL32 = COL32 + 0.4
795.                       COL42 = COL42 + 0.4
796.                ENDIF
797.     C
798.     C
799.     C
800.     C
801.     C   CHOOSE WHICH INFORMATION TO PRINT OUT AND CORRESPONDING FORMAT
802.     C
803.                IF (MOD((K3-(MULTPL-1)),MULTPL) .EQ. 0) THEN DO
804.     C  PRINT OUT MINUTE TOTAL AND HOURLY TOTAL
805.                    WRITE(6,93) COL1, COL12, MDAYZ(I6,K3),CHOTOT(I6,K4),
806.          *            COL2,COL22,MDAYZ(I6,K3+COLTWO),CHOTOT(I6,K4+6),COL3,
807.          *            COL32,MDAYZ(I6,K3+COLTRE),CHOTOT(I6,K4+12),COL4,COL42,
808.          *            MDAYZ(I6,K3+COLFOR),CHOTOT(I6,K4+18)
809.     C
810.     C
811.                 K4=K4+1
812.                IF (K4 .GT. 6) K4=1
813.     C
814.                 ELSE DO
815.     C*****************************************************************
816.     C**   PRINT OUT THE MINUTE INTERVALS WITHOUT THE HOURLY VOLUME  *
817.     C*****************************************************************
818.     C
819.                   WRITE(6,94) COL1,COL12,MDAYZ(I6,K3),COL2,COL22,
820.          *             MDAYZ(I6,K3+COLTWO),COL3,COL32,MDAYZ(I6,K3+COLTRE),
821.          *             COL4,COL42,MDAYZ(I6,K3+COLFOR)
822.                  ENDIF
823.     C
824.     C
825.     93         FORMAT(T5,4(F5.2,'-',F5.2,I6,I9,5X))
826.     94         FORMAT(T5,4(F5.2,'-',F5.2,I6,14X))
827.     C
828.     C
829.     C               WRITE(6,91) COL1, COL12, COL2, COL22, COL3, COL32,
830.     C     *                     COL4, COL42
831.     C
832.     C   PAGE CONTROL FOR LONG REPORTS.
833.     C
834.     C
835.     C
836.     C
837.     C91             FORMAT(T5,4(F5.2,'-',F5.2,20X))
838.                    COL1 = COL12
839.                    COL2 = COL22
```

```
840.                    COL3 = COL32
841.                    COL4 = COL42
842.                    IF (TDELTA .GT. (0.6-DELTA)) THEN DO
843.                        TDELTA=0.0
844.                        COL1 = IFIX(COL1+1.0)
845.                        COL2 = IFIX(COL2+1.0)
846.                        COL3 = IFIX(COL3+1.0)
847.                        COL4 = IFIX(COL4+1.0)
848.     C
849.                         WRITE(6,56)
850.     C
851.     C
852.                   WRITE(6,95) RUNSUB(I6,B),RUNSUB(I6,B+6),RUNSUB(I6,B+12),
853.          *            RUNSUB(I6,B+18)
854.     C
855.     95            FORMAT('+',T5,22X,'(',I5,')',24X,'(',I5,')',24X,
856.          *        '(',I5,')',24X,'(',I5,')')
857.                   B = B + 1
858.                   IF (B .GT. 6) B=1
859.     C
860.     C
861.     C    CHECK FOR PAGE EJECT AND LINE CONTROL FOR LONG REPORTS
862.     C
863.                     IF ((45-KOUNT) .LT. 15) THEN DO
864.                         KOUNT=1
865.     C
866.     C    IF END-OF-DAY THEN DO NOT SKIP TO NEW PAGE. PRINT TOTALS..
867.     C
868.                    IF (COL4 .LT. 24) THEN DO
869.     C
870.                       WRITE(6,10)
871.     C
872.                       WRITE(6,56)
873.                      ENDIF
874.                     ENDIF
875.                    ENDIF
876.                   K3=K3+1
877.     C
878.     C
879.     C             PRINT,'K4=',K4
880.                 END WHILE
881.     C
882.     C
883.     C
884.     C
885.              WRITE(6,101) AMPEAK(I6),PMPEAK(I6)
886.     101        FORMAT(//,T10,'AM PEAK (6:30 - 9:30) VOLUME:',I10,
887.          *              T60,'PM PEAK (3:30 - 6:30) VOLUME:',I10,
888.          *              T110,'DT:',10('_'))
889.                WRITE(6,102) RUNSUB(I6,24)
890.     102        FORMAT('+',T115,I5)
891.     C
892.     C
893.                WRITE(6,111)AMHIGH(I6),PMHIGH(I6)
894.     111        FORMAT(T10,'AM PEAK HIGHEST HOUR:',8X,I10,5X,T60,
895.          *           'PM PEAK HIGHEST HOUR:',8X,I10,5X)
896.     C
897.     C
898.                WRITE(6,112) INTVAL,INTVAL
899.     112        FORMAT(T10,'(NEAREST',I3,' MINUTE COUNT)',T60,
```

```
900.          *         '(NEAREST',I3,' MINUTE COUNT)')
901.     C
902.     C
903.     C   PRINT OUT THE STARTING TIME DEPENDING ON THE MINUTE COUNT
904.     C
905.     C
906.                IF ((AMMIN(I6) .LT. 10) .AND. (PMMIN(I6) .LT. 10)) THEN DO
907.                  WRITE(6,114) AMHOUR(I6),CAMMIN(I6),PMHOUR(I6),CPMMIN(I6)
908.     C
909.     C
910.     114        FORMAT(/,T10,2('STARTING TIME :',18X,I3,':',A2,11X))
911.               ELSE DO
912.     C
913.     C
914.               IF ((AMMIN(I6) .LT. 10) .OR. (PMMIN(I6) .LT. 10)) THEN DO
915.               IF (AMMIN(I6) .LT. 10) THEN DO
916.                 WRITE(6,115) AMHOUR(I6),CAMMIN(I6),PMHOUR(I6),PMMIN(I6)
917.     115         FORMAT(/,T10,'STARTING TIME :',18X,I3,':',A2,11X,
918.          *           'STARTING TIME :',18X,I3,':',I2)
919.                 ELSE DO
920.     C
921.     C
922.                 WRITE(6,116) AMHOUR(I6),AMMIN(I6),PMHOUR(I6),CPMMIN(I6)
923.     116         FORMAT(/,T10,'STARTING TIME :',18X,I3,':',I2,11X,
924.          *           'STARTING TIME :',18X,I3,':',A2)
925.                 END IF
926.     C
927.     C
928.                ELSE DO
929.     C
930.                 WRITE(6,113) AMHOUR(I6),AMMIN(I6),PMHOUR(I6),PMMIN(I6)
931.     113         FORMAT(/,T10,2('STARTING TIME :',18X,I3,':',I2,11X))
932.                 END IF
933.     C
934.     C
935.     C
936.                END IF
937.     C
938.     C
939.     C
940.     C     RESET K3 FOR NEXT CHANNEL PRINT OUT
941.     C
942.                     K3=1
943.     C
944.     C
945.                  I6=I6+1
946.     C
947.                  ELSE DO
948.                    I6=I6+1
949.                  ENDIF
950.     C
951.                  END WHILE
952.     C
953.     C
954.     C  CHECK TO SEE IF END HAS REACHED
955.     C
956.     C
957.     C         WRITE(6,10)
958.     10       FORMAT('1')
959.               RETURN
```

```
960.              END
961.      C
962.      C*************************************************************
963.      C*                                                           *
964.      C*     SUBROUTINE INITAL -- SUBROUTINE TO INITIALIZE THE ARRAY *
965.      C*        HTOTAL AND MINTOT THAT ACCUMULATES THE HOURLY AND    *
966.      C*        MINUTE INTERVAL TOTALS.                              *
967.      C*                                                           *
968.      C*************************************************************
969.      C
970.      C
971.              SUBROUTINE INITAL(MDAYZ,CHOTOT)
972.              DIMENSION  MDAYZ(7,288)
973.              INTEGER CHOTOT(7,24)
974.      C
975.      C
976.      C
977.      C
978.      C
979.      C
980.      C
981.              DO 80 K=1,7
982.                DO 20 J=1,24
983.              CHOTOT(K,J)=0
984.      20        CONTINUE
985.      80    CONTINUE
986.      C*****************************************************
987.      C*   ZERO OUT MDAYZ AFTER EACH DAY OF PREPARATION   **
988.      C*****************************************************
989.      C
990.              DO 40 I=1,7
991.                 DO 50 J=1,288
992.                    MDAYZ(I,J)=0
993.      50         CONTINUE
994.      40      CONTINUE
995.      C
996.      C
997.              RETURN
998.              END
999.      C
1000.     C
1001.     C
1002.     C*************************************************************
1003.     C*                                                           *
1004.     C*     SUBROTINE DAWEEK -- THIS SUBROUTINE FINDS THE DAY OF  *
1005.     C*     THE WEEK GIVEN THE INITIAL DATE IN THE FORM OF YY-MM  *
1006.     C*     DD.  IT DETERMINES THE PROPER ELEMENT TO STORE THE    *
1007.     C*                                                           *
1008.     C*************************************************************
1009.     C
1010.     C
1011.             SUBROUTINE DAWEEK(TSDATE,ENDEX,ARMON,ARDAY,CHECK,DAY,MONTH,YEAR)
1012.             CHARACTER*9 DATES(7),DAYOWK
1013.             INTEGER  DATE,TSDATE,DAY,MONTH,YEAR
1014.             INTEGER  ENDEX,D3,FACTOR,CHECK
1015.             INTEGER ARMON(7),ARDAY(7)
1016.     C
1017.     C
1018.     C  INITIALIZE THE DATES ARRAY
1019.     C
```

```
1020.     C
1021.     C
1022.     C
1023.           DO 21 I = 1,7
1024.               ARMON(I) = 0
1025.               ARDAY(I) = 0
1026.     21    CONTINUE
1027.     C
1028.     C
1029.           DATES(1) = 'SATURDAY'
1030.           DATES(2) = 'SUNDAY'
1031.           DATES(3) = 'MONDAY'
1032.           DATES(4) = 'TUESDAY'
1033.           DATES(5) = 'WEDNESDAY'
1034.           DATES(6) = 'THURSDAY'
1035.           DATES(7) = 'FRIDAY'
1036.     C
1037.     C
1038.     C   FIND THE DAY, MONTH AND YEAR FROM THE GIVEN TEST DATE
1039.     C
1040.     C
1041.     C
1042.           DAY = (MOD(TSDATE,100))
1043.           MONTH = (MOD(TSDATE,10000) - DAY)/100
1044.           YEAR = (TSDATE/10000) + 1900
1045.     C
1046.     C
1047.     C   PRINT OUT DAY, MONTH, AND YEAR
1048.     C
1049.     C        PRINT,'DAY=',DAY,'MONTH=',MONTH,'YEAR=',YEAR
1050.     C
1051.     C
1052.     C
1053.     C   CONVERT THE DATE INTO ABSOLUTE JULIAN DATES
1054.     C
1055.     C   IF MONTH IS EITHER JAN. OR FEB. USE THE FOLLOWING FORMULA
1056.     C
1057.           IF (MONTH .LE. 2) THEN DO
1058.              FACTOR = 365*YEAR + DAY + 31*(MONTH-1) +
1059.          *      IFIX((YEAR-1)/4.) - IFIX(3/4.*(IFIX(((YEAR-1)/100.)+1)))
1060.     C
1061.     C   FOR ALL OTHER MONTHS, USE THE FOLLOWINT FORMULA.
1062.     C
1063.             ELSE DO
1064.               FACTOR = 365*YEAR+DAY+31*(MONTH-1) - IFIX(0.4*MONTH+2.3)
1065.          *      +IFIX(YEAR/4.) - IFIX(3/4.*(IFIX(YEAR/100.)+1))
1066.     C
1067.             END IF
1068.     C
1069.     C
1070.     C   FIND THE INDEX FOR THE ARRAY DATES TO CONVERT TO DATES
1071.     C
1072.     C
1073.           ENDEX = FACTOR + (IFIX(-FACTOR/7.)*7)+1
1074.     C
1075.            DAYOWK = DATES(ENDEX)
1076.     C
1077.     C   PRINT OUT THE DAY OF WEEK
1078.     C
1079.     C        PRINT, 'DAY OF WEEK = ', DATES(ENDEX)
```

```
1080.      C
1081.      C       CALL A PROGRAM TO DETERMINE 7-DAY WEEK
1082.      C
1083.              CALL FINDAY(DAYOWK,DAY,MONTH,YEAR,ARMON,ARDAY,CHECK)
1084.      C
1085.      C
1086.      C
1087.      C
1088.      C
1089.      C
1090.      C
1091.              IF (ENDEX .GE. 3) THEN DO
1092.                  ENDEX = ENDEX-3
1093.                ELSE DO
1094.                  ENDEX = ENDEX+4
1095.                ENDIF
1096.      C
1097.      C
1098.              RETURN
1099.              END
1100.      C
1101.      C*****************************************************************
1102.      C*                                                               *
1103.      C*    SUBROUTINE WEEKLY -- SUBROUTINE TO PRODUCES THE WEEKLY     *
1104.      C*       SUMMARY REPORT.  THIS REPORT INCLUDES THE HOURLY TOTAL  *
1105.      C*       VOLUMES, SUMMARY OF WEEKLY DATA.  IT CAN PRODUCE UP TO  *
1106.      C*       SIX REPORTS FOR THE 6-CHANNEL CONFIGURATION.            *
1107.      C*                                                               *
1108.      C*****************************************************************
1109.      C
1110.              SUBROUTINE WEEKLY(ARMON,ARDAY,NWTOT,APEAK,PPEAK,AHI,PHI,D3,
1111.           *                    SITENO,TESDAY)
1112.              INTEGER ARMON(7),ARDAY(7),SITENO,TESDAY
1113.              INTEGER NWTOT(7,7,24),DAY,MONTH,YEAR
1114.              INTEGER APEAK(7),PPEAK(7),AHI(7),PHI(7)
1115.              INTEGER L,D3,IDTOT(7),ITOTAL(7)
1116.      C
1117.      C
1118.      C
1119.      C
1120.      C
1121.      C
1122.      C
1123.      C
1124.      C
1125.      C*******************************************************
1126.      C*   THIS WILL HELP PRINT THE NECESSARY AMOUNT OF TABLES*
1127.      C*******************************************************
1128.      C*
1129.      C
1130.      C*************************************************************
1131.      C*     FIND OUT THE MONTH, DAY, AND YEAR FROM THE GIVEN DATE
1132.      C*
1133.      C*
1134.                DAY=MOD(TESDAY,100)
1135.                MONTH=(MOD(TESDAY,10000)-DAY)/100
1136.                YEAR=(TESDAY/10000)+1900
1137.      C
1138.      C
1139.      C*************************************************************
```

```
1140.     C
1141.     C
1142.     C   CALCULATE THE TOTAL VALUES OF DAILY HOURLY VOLUMES
1143.     C
1144.               L=1
1145.     C
1146.     C
1147.               WHILE (L .LT. 7 ) DO
1148.     C
1149.     C
1150.     C   SET UP LOOP TO PRINT OUT THE WEEKLY SUMMARY REPORT
1151.     C
1152.     C
1153.                 ITOTAL(L)=0
1154.     C
1155.     C
1156.                 DO 10009 K=1,7
1157.                   IDTOT(K)=0
1158.                   DO 10010 IL=1,24
1159.                      IDTOT(K)=IDTOT(K)+NWTOT(L,K,IL)
1160.     10010         CONTINUE
1161.     C
1162.     C
1163.                    ITOTAL(L) = ITOTAL(L) + IDTOT(K)
1164.     C
1165.     C
1166.     10009       CONTINUE
1167.     C
1168.     C
1169.     C
1170.     C
1171.     C
1172.     C
1173.     C
1174.     C    IF IT IS A TWO CHANNEL , THEN CLEAR THE PREVIOUS PEAK PERIODS
1175.     C
1176.                   IF (D3 .EQ. 2) THEN DO
1177.                      DO 10001 IK = 1,7
1178.                         APEAK(IK) = 0
1179.                         PPEAK(IK) = 0
1180.                         AHI(IK)   = 0
1181.                         PHI(IK)   = 0
1182.     10001            CONTINUE
1183.                   END IF
1184.     C
1185.     C
1186.     C
1187.     C
1188.     C*************************************************************
1189.     C*                                                           *
1190.     C*     IF IT IS A TWO CHANNEL TABLE, THIS WILL HELP PRINT    *
1191.     C*     OUT THE NECESSARY AM & PM PEAK PERIODS.               *
1192.     C*                                                           *
1193.     C*************************************************************
1194.     C
1195.     C
1196.               IF (D3 .EQ. 2) THEN DO
1197.     C
1198.               DO 15000 J=1,7
1199.     C
```

```
1200.                DO 20000 I =1,24
1201.      C
1202.                ITEMP = NWTOT(L,J,I)
1203.      C
1204.                IF (I .LE. 12) THEN DO
1205.      C
1206.                IF ((I .GT. 6) .AND. (I .LE. 9)) APEAK(J) = APEAK(J) + ITEMP
1207.                    IF (ITEMP .GT. AHI(J)) AHI(J) =  ITEMP
1208.                ELSE DO
1209.      C
1210.                IF (ITEMP .GT. PHI(J)) PHI(J) =  ITEMP
1211.      C
1212.                IF ((I .GT. 15) .AND. (I .LE. 18)) PPEAK(J) = PPEAK(J) + ITEMP
1213.                END IF
1214.      20000     CONTINUE
1215.      15000     CONTINUE
1216.      C
1217.                END IF
1218.      C
1219.      C
1220.      C*****************************************************************
1221.      C*                                                               *
1222.      C*     START A LOOP TO PRINT OUT THE WEEKLY REPORTS.  IF ALL OF  *
1223.      C*     THE ENTRIES ARE ZEROS, DO NOT PRINT OUT THAT REPORT.      *
1224.      C*                                                               *
1225.      C*****************************************************************
1226.      C
1227.      C
1228.               IF (ITOTAL(L) .NE. 0) THEN DO
1229.      C
1230.      C
1231.              WRITE(6,10)
1232.      10      FORMAT('1')
1233.      C
1234.      C
1235.               WRITE(6,20) MONTH,DAY,YEAR
1236.      20       FORMAT(///,T48,'FREEWAY VOLUME COUNT SUMMARY SHEET',///,
1237.           *     T15,'LOCATION :',5X,'&LOC',
1238.           *     //,T15,'DATE :',9X,I2,'-',I2,'-',I4,T50,
1239.           *     'DIRECTION OF FLOW :',5X,'&DIR',
1240.           *      /)
1241.      C
1242.      C
1243.               WRITE(6,35)L,D3,SITENO
1244.      35       FORMAT(T15,'SOURCE :',T30,'THIS IS ',I2,' OF ',I2,
1245.           *     T50,'SITE NO :',5X,I8,//)
1246.      C
1247.      C
1248.              WRITE(6,40)
1249.      40      FORMAT(T15,'DAY OF WEEK:',T40,'MONDAY',T50,'TUESDAY',
1250.           *         T60,'WEDNESDAY',T70,'THURSDAY',T80,'FRIDAY',
1251.           *         T90,'SATURDAY',T100,'SUNDAY',/)
1252.      C
1253.              WRITE(6,41) ARMON(1),ARDAY(1),ARMON(2),ARDAY(2),
1254.           * ARMON(3),ARDAY(3),ARMON(4),ARDAY(4),ARMON(5),ARDAY(5),
1255.           * ARMON(6),ARDAY(6),ARMON(7),ARDAY(7)
1256.      C
1257.      41      FORMAT(T39,I2,' / ',I2,T50,I2,' / ',I2,T60,I2,' / ',I2,
1258.           *         T70,I2,' / ',I2,T79,I2,' / ',I2,T90,I2,' / ',I2,
1259.           *        T99,I2,' / ',I2,/)
```

```
1260.     C
1261.           WRITE(6,45)
1262.     45    FORMAT(T15,93('-'))
1263.           WRITE(6,50)
1264.     50    FORMAT(T20,'TIME',T40,'HOURLY',T50,'HOURLY',T60,'HOURLY',
1265.          *       T70,'HOURLY',T80,'HOURLY',T90,'HOURLY',T100,
1266.          *       'HOURLY')
1267.           WRITE(6,60)
1268.     60    FORMAT(T40,'VOLUME',T50,'VOLUME',T60,'VOLUME',T70,
1269.          *       'VOLUME',T80,'VOLUME',T90,'VOLUME',T100,'VOLUME'/)
1270.           N=0
1271.     C      PRINT,'NWTOT =',NWTOT(1,1),NWTOT(2,1),NWTOT(3,1),NWTOT(4,1)
1272.           WHILE(N .LT. 24) DO
1273.            NONE=N+1
1274.            K =  3
1275.            WRITE(6,70) N,NONE,NWTOT(L,1,NONE),NWTOT(L,2,NONE),
1276.          * NWTOT(L,3,NONE),NWTOT(L,4,NONE),NWTOT(L,5,NONE),
1277.          * NWTOT(L,6,NONE), NWTOT(L,7,NONE)
1278.     70    FORMAT(T20,I2,'-',I2,T42,7(I4,6X))
1279.                N=N+1
1280.           END WHILE
1281.            WRITE(6,46)
1282.     46    FORMAT(' ')
1283.            WRITE(6,45)
1284.            WRITE(6,80) APEAK(1),APEAK(2),APEAK(3),APEAK(4),APEAK(5),
1285.          * APEAK(6),APEAK(7)
1286.     80    FORMAT(/,T39,7(3X,I4,3X))
1287.            WRITE(6,85)
1288.     85    FORMAT('+',T15,'AM PEAK (6-9) VOLUME:',T38,7(3X,5('_'),2X))
1289.            WRITE(6,90) AHI(1),AHI(2),AHI(3),AHI(4),AHI(5),
1290.          * AHI(6),AHI(7)
1291.     90    FORMAT(/,T39,7(3X,I4,3X))
1292.            WRITE(6,95)
1293.     95    FORMAT('+',T15,'AM PEAK HIGHEST HOUR:',T38,7(3X,5('_'),2X))
1294.            WRITE(6,100) PPEAK(1),PPEAK(2),PPEAK(3),PPEAK(4),PPEAK(5),
1295.          * PPEAK(6),PPEAK(7)
1296.     100   FORMAT(/,T39,7(3X,I4,3X))
1297.            WRITE(6,105)
1298.     105   FORMAT('+',T15,'PM PEAK (3-6) VOLUME:',T38,7(3X,5('_'),2X))
1299.            WRITE(6,110) PHI(1),PHI(2),PHI(3),PHI(4),PHI(5),
1300.          * PHI(6),PHI(7)
1301.     110   FORMAT(/,T39,7(3X,I4,3X))
1302.            WRITE(6,115)
1303.     115   FORMAT('+',T15,'PM PEAK HIGHEST HOUR:',T38,7(3X,5('_'),2X))
1304.            WRITE(6,120)
1305.     120   FORMAT(/,T15,'DT:',T38,7(3X,5('_'),2X))
1306.     C
1307.            WRITE(6,125)IDTOT(1),IDTOT(2),IDTOT(3),IDTOT(4),IDTOT(5),
1308.          *             IDTOT(6),IDTOT(7)
1309.     125   FORMAT('+',T38,7(3X,I5,2X))
1310.     C
1311.     C
1312.     C
1313.     C
1314.            L=L+1
1315.            ELSE DO
1316.              L=L+1
1317.            ENDIF
1318.     C
1319.     C
```

```
1320.           END WHILE
1321.           WRITE(6,10)
1322.           RETURN
1323.           END
1324.      C
1325.      C
1326.      C*************************************************************
1327.      C*          SUBROUTINE 'FINDAY' TO FIND 7-DAY WEEK           *
1328.      C*************************************************************
1329.      C
1330.      C
1331.      C*************************************************************
1332.      C*                                                           *
1333.      C*          THIS SUBROUTINE INPUTS THE DAY,MONTH, & YEAR     *
1334.      C*     OF THE CURRENT WEEK OF THE DATA COLLECTION.  IT THEN*
1335.      C*     ATTEMPTS TO DETERMINE THE NUMBER OF THE DAY OF THAT *
1336.      C*     WEEK.  IN DOING SO THE SUBROTINE NEEDS TO KNOW WHAT *
1337.      C*     MONTH IT IS SO THAT IT CAN KNOW HOW MANY DAYS IN IT.*
1338.      C*************************************************************
1339.                SUBROUTINE FINDAY(DAYOWK,DAY,MONTH,YEAR,ARMON,ARDAY,CHECK)
1340.           INTEGER DAY,MONTH,YEAR
1341.           CHARACTER*9 DAYOWK
1342.           INTEGER NWTOT(7,7,24)
1343.           INTEGER DAYZ(12),CHECK,CHECK1,CHECK2,MON1,MON2,ARMON(7),ARDAY(7)
1344.      C
1345.      C*****************************************************************
1346.      C*                                                               *
1347.      C*          THE FOLLOWING STATEMENTS WILL DETERMINE HOW MANY     *
1348.      C*     DAYS ARE IN THE RESPECTIVE MONTHS OF THE YEAR AND PLACE*
1349.      C*     IN THEIR RESPECTIVE ARRAYS.                               *
1350.      C*                                                               *
1351.      C*****************************************************************
1352.      C
1353.           DAYZ(1)  = 31
1354.           DAYZ(2)  = 28
1355.           DAYZ(3)  = 31
1356.           DAYZ(4)  = 30
1357.           DAYZ(5)  = 31
1358.           DAYZ(6)  = 30
1359.           DAYZ(7)  = 31
1360.           DAYZ(8)  = 31
1361.           DAYZ(9)  = 30
1362.           DAYZ(10) = 31
1363.           DAYZ(11) = 30
1364.           DAYZ(12) = 31
1365.           IF (MOD(YEAR,4) .EQ. 0) DAYZ(2)=29
1366.      C
1367.      C
1368.      C*****************************************************************
1369.      C*                                                               *
1370.      C*          THE FOLLOWING USES A 'CHECK' TO FIND OUT WHAT DAY *
1371.      C*     HAS BEEN USED AS INPUT. (EX. MONDAY = 1, TUESDAY = 2.) *
1372.      C*                                                               *
1373.      C*****************************************************************
1374.      C
1375.           IF (DAYOWK .EQ. 'MONDAY') CHECK = 1
1376.           IF (DAYOWK .EQ. 'TUESDAY') CHECK = 2
1377.           IF (DAYOWK .EQ. 'WEDNESDAY') CHECK = 3
1378.           IF (DAYOWK .EQ. 'THURSDAY') CHECK = 4
1379.           IF (DAYOWK .EQ. 'FRIDAY') CHECK = 5
```

```
1380.              IF (DAYOWK .EQ. 'SATURDAY') CHECK = 6
1381.              IF (DAYOWK .EQ. 'SUNDAY') CHECK = 7
1382.      C
1383.              CHECK1 = CHECK
1384.              CHECK2 = CHECK - 1
1385.      C
1386.              MON1 = MONTH
1387.              MON2 = MONTH
1388.      C
1389.              IDAY1 = DAY
1390.              IDAY2 = DAY
1391.      C
1392.               WHILE(CHECK1 .GT. 0) DO
1393.      C
1394.               ARMON(CHECK1) = MON1
1395.               ARDAY(CHECK1) = IDAY1
1396.      C
1397.               CHECK1 = CHECK1 - 1
1398.               IDAY1 = IDAY1 - 1
1399.               IF(IDAY1 .EQ. 0) THEN DO
1400.                 IF (MON1 .EQ. 1) MON1 = 13
1401.      C
1402.                      MON1 = MON1 -1
1403.                      IDAY1 = DAYZ(MON1)
1404.               END IF
1405.                  END WHILE
1406.      C
1407.      C
1408.               WHILE(CHECK2 .LT. 7) DO
1409.      C
1410.                     CHECK2 = CHECK2 + 1
1411.                     ARMON(CHECK2) = MON2
1412.                     ARDAY(CHECK2) = IDAY2
1413.                     IDAY2 = IDAY2 + 1
1414.      C
1415.              IF(IDAY2 .GT. DAYZ(MON2)) THEN DO
1416.      C
1417.                     IF(MON2 .EQ. 12) MON2 = 0
1418.      C
1419.                     MON2 = MON2 + 1
1420.                     IDAY2 = 1
1421.      C
1422.                     END IF
1423.              END WHILE
1424.      C
1425.      C
1426.      C          THIS IS WHERE TO PRINT IT OUT
1427.      C
1428.      C
1429.      C
1430.               RETURN
1431.      C
1432.               END
1433.      C
1434.      C
1435.      C
1436.      C
1437.      C
1438.      //$DATA
```

# APPENDIX D

The following pages are taken from the MK3 AUTOMATIC TRAFFIC COUNTING EQUIPMENT USER MANUAL. This section describes the procedures to set up interface signals. The user may find more ASCII codes from the IBM System/370 Reference card. Any ASCII character set can be used as EOL and EOF signals. The user simply enters the decimal equivalent of the ASCII character set during the interface signal setting procedure.

## 3.3 Interface

This mode is used to programme the RS-232 interface to be compatible with the computer or off-line terminal which is to be used with the Retriever.

There are 4 functions available to initialise the interface parameters and there is one function available to allow the user to test the interface. These five functions correspond to 5 code numbers. The code number is displayed on the left hand digit of the display. The 6 right hand digits are used as a parameter field. To change the current code number, press the appropriate key in the range 1-5. Pressing a key outside this range will result in Interface 1 being selected.

The parameters displayed for each code are:

| Interface Code | Parameters Displayed | Format |
|---|---|---|
| 1 | Retriever Status; Battery Voltage | C S V.VV |
| 2 | Baud Rate | C BBBBB |
| 3 | No. of End of Line and End of File Characters | C LL FF |
| 4 | Code for Autostop option setting | C A |
| 5 | Code for Status Line option setting | C S |

C = Interface Code Number
S = Status Code
V = Battery Voltage
B = Baud rate
L = End of Line characters
F = End of File characters
A = Autostop Code
S = Status Line Code

Following is a description of the 5 Interface codes.

### 3.3.1 Interface 1 - Display Retriever status and battery voltage;

This function has three characteristics - it displays the present Retriever Status and battery voltage, and it can be instructed to output a test message to a computer or off-line terminal via the RS-232 Interface.

Select Interface 1 by pressing key "1", and the display will show 3 numbers. The left hand digit shows the Interface code selected, in this case 1. The digit in the centre of the display indicates the status of the Retriever. If this digit is 0, the Retriever is O.K., otherwise a fault condition is indicated by a non-zero number. Status is reset when the data files are cleared.

The battery voltage is shown on the right hand side of the display. When the battery is fully charged this should be in excess of 6.30 volts with the charger disconnected. As the battery discharges, this voltage will reduce towards 5.70 volts over a period of 6 to 8 days. At 5.50 volts a cut-out circuit will switch the power to the Retriever off (see section 1.3.1).

WARNING: When the power to the Retriever is switched off, ALL Data will be lost. For this reason it is suggested that the GR 0308 Charger/Interface unit is connected to the Retriever whenever possible. (It should be noted that the

Interface parameters, as described in the following sections, will not be lost when power to the Retriever is switched off).

Once the Interface parameters 2-5 have been initialised, (see the following sections),the user can check the RS-232 interface is matched to the computer or off-line terminal being used.

With Interface 1 selected, and the terminal or computer connected, depress "*" and "#" simultaneously. The Retriever will output repeatedly a test message of numbers and characters. Any End of Line characters (as described in Section 3.3.3) will follow each line, and transmission will stop at the end of each line if the Autostop option has been enabled (see Section 3.3.4) When the "*" and "#" are released, the Retriever will complete output of the current test message and End of Line characters, and it will output the End of File characters (see Section 3.3.3) followed by the End of Line characters again. During this output the Retriever will respond to any status line control as described in Section 3.3.5.

### 3.3.2 Interface 2 - Baud Rate

Select Interface code number 2. The right hand digits show the current Retriever baud rate. To alter the baud rate press the "*" and "#" simultaneously, and when the display shows 5 dashes enter the required baud rate, (with leading zeros), followed by the 'enter' key. The Retriever will confirm the baud rate selected on the display.

The available baud rates are:-

19200 9600 4800 2400 1200 600 300 110

Entry of a baud rate other than shown above will result in the nearest available baud rate being selected.

The average transmission rate will be less than the baud rate, whilst the Retriever calculates data for output. This delay will be approximately 5-500 milliseconds at start of each line of output, proportional to the recording interval of the data file being output. In format 2, longer delays may occur (up to 15 seconds) whilst the output is being compiled.

### 3.3.3 Interface 3 - End of Line and End of File characters

The operator can specify up to 20 characters to be added to the end of each line of output in the Output Data mode, and up to 20 characters to be added at the end of each File in the Output Data mode.

Select Interface Code Number 3; the centre two digits show the present number of characters transmitted at the end of each line, and the right hand two digits show the present number of characters transmitted at the end of each file. To alter the current E.O.L. (End of Line) and E.O.F. (End of File) character strings, depress "*" and "#" simultaneously, and when:

| Display shows | Load into Retriever |
|---|---|
| --- | ASCII code for first EOL character + "ENTER" |
| --- | ASCII code for second EOL character + "ENTER" |
| ..... etc. | |
| --- | ASCII code for last EOL character + "ENTER" |
| --- | 999 + "ENTER" (terminating code) |
| --- | ASCII code for first EOF character + "ENTER" |
| --- | ASCII code for second EOF character + "ENTER" |
| ..... etc. | |
| --- | ASCII code for last EOF character + "ENTER" |
| --- | 999 + "ENTER" (terminating code) |

For example:

For the E.O.L. characters to be CR, LF and six nulls, and the E.O.F. characters to be four asterisks then:

| Display shows | Press Keys | Description |
|---|---|---|
| 1 0 6.40 | 3 | Select Interface 3 function |
| 3 x y | * & # | To change stored characters (x and y are previous values) |
| --- | 013 + ENTER | Load ASCII Carriage Return |
| --- | 010 + ENTER | Load ASCII Line Feed |
| --- | 000 + ENTER | Load ASCII Null |
| --- | 000 + ENTER | Load ASCII Null |
| --- | 000 + ENTER | Load ASCII Null |
| --- | 000 + ENTER | Load ASCII Null |
| --- | 000 + ENTER | Load ASCII Null |
| --- | 000 + ENTER | Load ASCII Null |
| --- | 999 + ENTER | Terminate EOL / Commence EOF |
| --- | 042 + ENTER | Load ASCII * |
| --- | 042 + ENTER | Load ASCII * |
| --- | 042 + ENTER | Load ASCII * |
| --- | 042 + ENTER | Load ASCII * |
| --- | 999 + ENTER | Terminate loading EOF chars |
| 3 8 4 | - Indicates 8 EOL and 4 EOF characters have been accepted. | |

All ASCII codes in the range 000 to 127 are accepted, any code in excess of 127 will terminate that part of the loading. Attempted entry of more than 20 E.O.L. or 20 E.O.F. characters will be ignored, but the first 20 characters loaded will be stored. If no E.O.F. characters are required then enter the terminating code "999" twice after the last E.O.L. character code.

All numbers entered and displayed on the Retriever are in the normal decimal numbering system. If you wish to enter a character code whose value is known in binary, octal, or hexadecimal, conversion to a decimal number must be done before entry of decimal equivalent into the Retriever.

For example:

| | Decimal | Binary | Octal | Hexadecimal |
|---|---|---|---|---|
| ASCII Line Feed | 010 | 0001010 | 012 | 0A |

ASCII Codes for Control Characters

| Code | Name | Description |
|---|---|---|
| 000 | NULL | A null character usually to fill time |
| 001 | SQH | Start of Heading |
| 002 | STX | Start of Text |
| 003 | ETX | End of Text |
| 004 | EOT | End of Transmission |
| 005 | ENQ | Enquiry |
| 006 | ACK | Acknowledge |
| 007 | BELL | Bell |
| 008 | BS | Backspace |
| 009 | HT | Horizontal Tab |
| 010 | LF | Line Feed |
| 011 | VT | Vertical Tab |
| 012 | FF | Form Feed |
| 013 | CR | Carriage Return |
| 016 | DLE | Data Link Escape |
| 017 | DC1 | Device Code 1 |
| 018 | DC2 | Device Code 2 |
| 019 | DC3 | Device Code 3 |
| 020 | DC4 | Device Code 4 |
| 027 | ESC | Escape |
| 127 | DEL | Delete |

Only commonly used control characters have been listed, but the user can input any valid ASCII codes required.

### 3.3.4 Interface 4 - Autostop Option Code

When enabled, this option halts all output from the Retriever to the computer or terminal after completion of each line of data output in the Output Data mode (including programmed end of line characters). The computer or terminal must then send a DC1 code to recommence transmission.

Select Interface code number 4. The right hand digit shows the current option as follows:

| Autostop Option Code | Resulting Operation |
|---|---|
| 0 | Normal operation, no stop at end of line |
| 1 | Stops transmission at end of each line of output in Output Data mode (and Interface 1 test mode) |

To alter the Autostop Option Code, depress "*" and "#" simultaneously, and when the display shows 1 dash, enter the appropriate one digit code.

### 3.3.5 Interface 5 - Status Line Input Option

Some terminals and/or communications devices present a "device ready" signal as an indication that serial data (such as output from the Retriever) can now be sent, or must now be terminated. On an RS232 device this will usually be on pin 20 of the 25 way connecting plug and normally has high or positive level to indicate "device ready" for receipt of data.

Status Line (Pin 20 of 25 way RS232 "D" connector)

| Input Code | Result |
|---|---|
| 0 | Ignore Status Input |
| 1 | Initialize transmission of next character only whilst Status input is high |
| 2 | Initialize transmission of next character only whilst Status input is low |

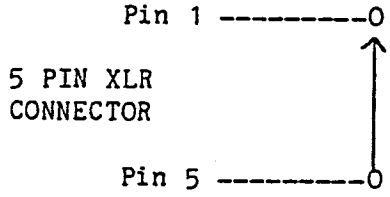
In this context Status Line input is defined as:

-15 to 0 volts - Status Low
5 to +15 volts - Status High

(0 to 5 volts undefined).

Select Interface Code Number "5". The right hand display will show the current value of Status Line Control.

To alter, press "*" and "#" simultaneously and when display shows one dash enter one digit number 0 through 2 as indicated above.
This facility could be used as follows with an external switch:




(Switch open equivalent to Low status input, switch closed equivalent to high status input)
Diagram shows switch closed

Note that if Status Line Input is left disconnected it assumes a low level, i.e. no transmission would occur with Status Line Control = 1, but continuous transmission with 0 or 2.