base. This test also verified that a portion of the TLIGHT routine was working correctly.

6.2.3 Tests on the LINK Routine

The LINK routine is similar to the ELINK routine and the same basic strategy was used in testing. The data routine is more complicated as the constants for manipulating the travelling times need to be calculated. The first part of the simulation routine which moves the vehicles down the street is the same as the ELINK routine and was tested in the same manner. The random allocation of vehicles to lanes was tested using different allocation proportions. The segment of the program which randomly assigns travelling times to vehicles was isolated and put into a loop and the travelling times were examined. This was repeated for different means and standard deviations. While this test does not verify good statistical properties it was considered to be sufficient as the base distribution has been shown to have good statistical properties and the manipulations performed will not alter these properties. These tests showed that the software manipulation was operating as required.

6.2.4 Tests on the TLIGHT Routine

Tests on the data segment of the TLIGHT routine were similar to those on the LINK routine as the methods used for calculating the constants used in both routines are the same. The section of the simulation routine which moves vehicles into queues was tested as described in 6.2.2. The overall operation of the TLIGHT routine was also tested in a similar manner except with the signals held in the green state. The outputs of the TLIGHT were then examined to verify that the correct proportion of vehicles were turning right, left and proceeding straight on. The discharge rate was also tested in this manner. The operation of the signal timing section was tested by

setting breakpoints (i.e. points in the program where execution becomes suspended) before the statements which would change the state or phase of the signal. On encountering a breakpoint the epoch counter was examined to see if the change occurred at the correct time. A check on the generation of acceptance gap times was performed in a similar manner as the test on the travelling times as described in 6.2.3. A second entry link was then added to the intersection to form an opposing queue to right turners from the first entry link. Breakpoints were again used to verify that the gap acceptance subroutine was functioning correctly. All the above tests were repeated for all the lanes and legs of the intersection.

6.2.5 Conclusion

The above tests on the traffic routines showed that the software was functioning as intended. The tests were comprehensive and an attempt was made to cover all the possible cases that might arise in simulating a large network.

6.3 Simulation of a Network with Four Intersections

6.3.1 Introduction

A simulation of four linked intersections was performed in order to demonstrate how the simulator is used and to obtain an estimate of the speed of the simulator. Unfortunately sufficient real data could not be obtained in time so a fictitious network and data were used. The network was chosen in such a way that most of the features of the simulator could be demonstrated. An attempt was also made to design the network so that it would be representative of a section of a (South African) city traffic system.

6.3.2 Description of Simulation

The layout of the network simulated is shown in Fig. 6.1.

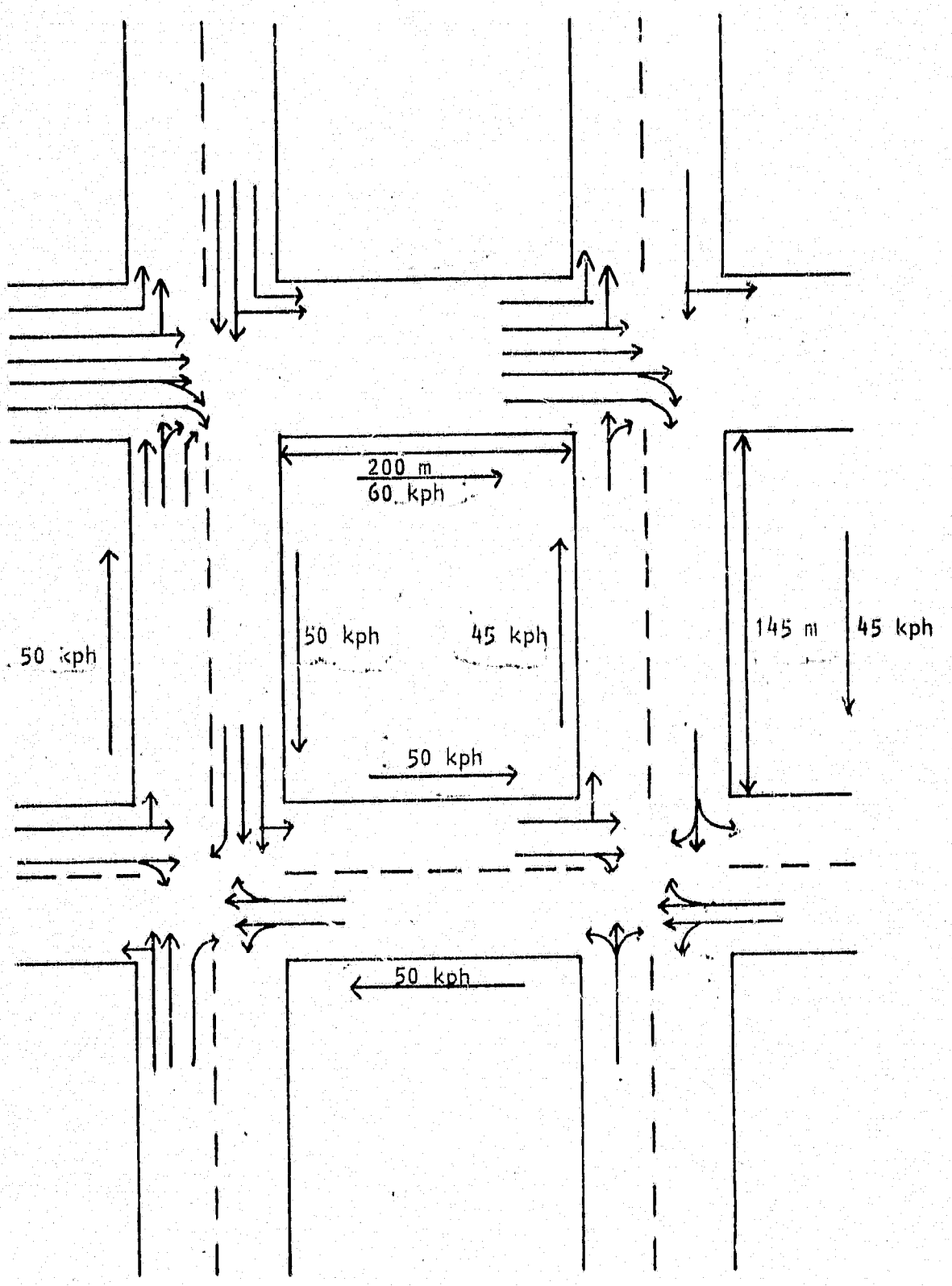

Figure 6.1. Layout of the Network Simulated

The direction and number of lanes for each street are indicated together with mean speeds in the links. The percentages of turning vehicles, allocation of vehicles in links and flow routes in entry links are given in tables 6.1, 6.2 and 6.3. A symbolic representation of the network which shows the interaction of the various simulation elements is given in Fig. 6.2.

The signal timing cycles were arbitrarily chosen but the offsets were calculated to provide free flow for vehicles travelling at the mean speed, from left to right. No data could be found in the literature on the distribution of acceptance gaps, so for this example a normal distribution with a mean of 5 and a standard deviation of 1 second was used.

The signal cycle was split into two equal phases and the queue lengths were sampled after every five changes of phase so that an average of the maximum and minimum queue lengths would be shown. An initialisation period of ten signal phase changes was allowed before the sampling was started so that the network could fill with vehicles and stabilise. The travelling times used for the links were calculated from the chosen mean speed and distances between the intersection. The simulation program was run for a total of 30 000 Epochs which is equivalent to approximately 2 hours and 47 minutes of real time traffic.

6.3.3 Running the Simulation

The program and data for this simulation were written onto two disk files (named TPROG and TDAT respectively) using a standard editor program. . The steps then used in running the program are shown in the printout given. All operator commands or replies are underlined. The first statement, GET, TPROG reads the program from the disk and lists it on the console. The command /E then terminates the program mode. The command GET, TDATA is then entered and the data is read

Table 6.1. Percentage Turning Vehicles at Intersections

Intersection No.	Leg No.	Queue 1		Queue 2		Queue 3		Queue 4		Queue 5	
		%L	%R	%L	%R	%L	%R	%L	%R	%L	%R
1	1	100	0	25	0	0	0	0	10	0	100
1	2	100	0	20	0	0	0	-	-	-	-
1	3	-	-	-	-	-	-	-	-	-	-
1	4	0	0	0	20	0	100	-	-	-	-
2	1	100	0	22	0	0	0	0	15	0	100
2	2	15	0	-	-	-	-	-	-	-	-
2	3	-	-	-	-	-	-	-	-	-	-
2	4	0	20	-	-	-	-	-	-	-	-
3	1	15	0	0	10	-	-	-	-	-	-
3	2	20	0	0	0	0	100	-	-	-	-
3	3	20	0	0	20	-	-	-	-	-	-
3	4	20	0	0	0	0	100	-	-	-	-
4	1	20	0	0	15	-	-	-	-	-	-
4	2	10	15	-	-	-	-	-	-	-	-
4	3	20	0	0	20	-	-	-	-	-	-
4	4	20	20	-	-	-	-	-	-	-	-

Table 6.2. Allocation of Vehicles to Lanes

Link No.	% Lane 1	% Lane 2	% Lane 3	% Lane 4	% Lane 5
1	12	23	35	18	12
2	33	50	17	-	-
3	33	50	17	-	-
4	100	-	-	-	-
5	100	-	-	-	-
6	66	34	-	-	-
7	71	29	-	-	-
8		D U M M Y			

Table 6.3. Entry Link Flow Rates

ELINK No.	Flow Rate Lane 1 VPH	Flow Rate Lane 2 VPH	Flow Rate Lane 3 VPH	Flow Rate Lane 4 VPH	Flow Rate Lane 5 VPH
1	100	200	300	150	100
2	200	300	100	-	-
3	200	-	-	-	-
4	0 (DUMMY)		-	-	-
5	300	150	-	-	-
6	200	300	100	-	-
7	250	-	-	-	-
8	250	100	-	-	-

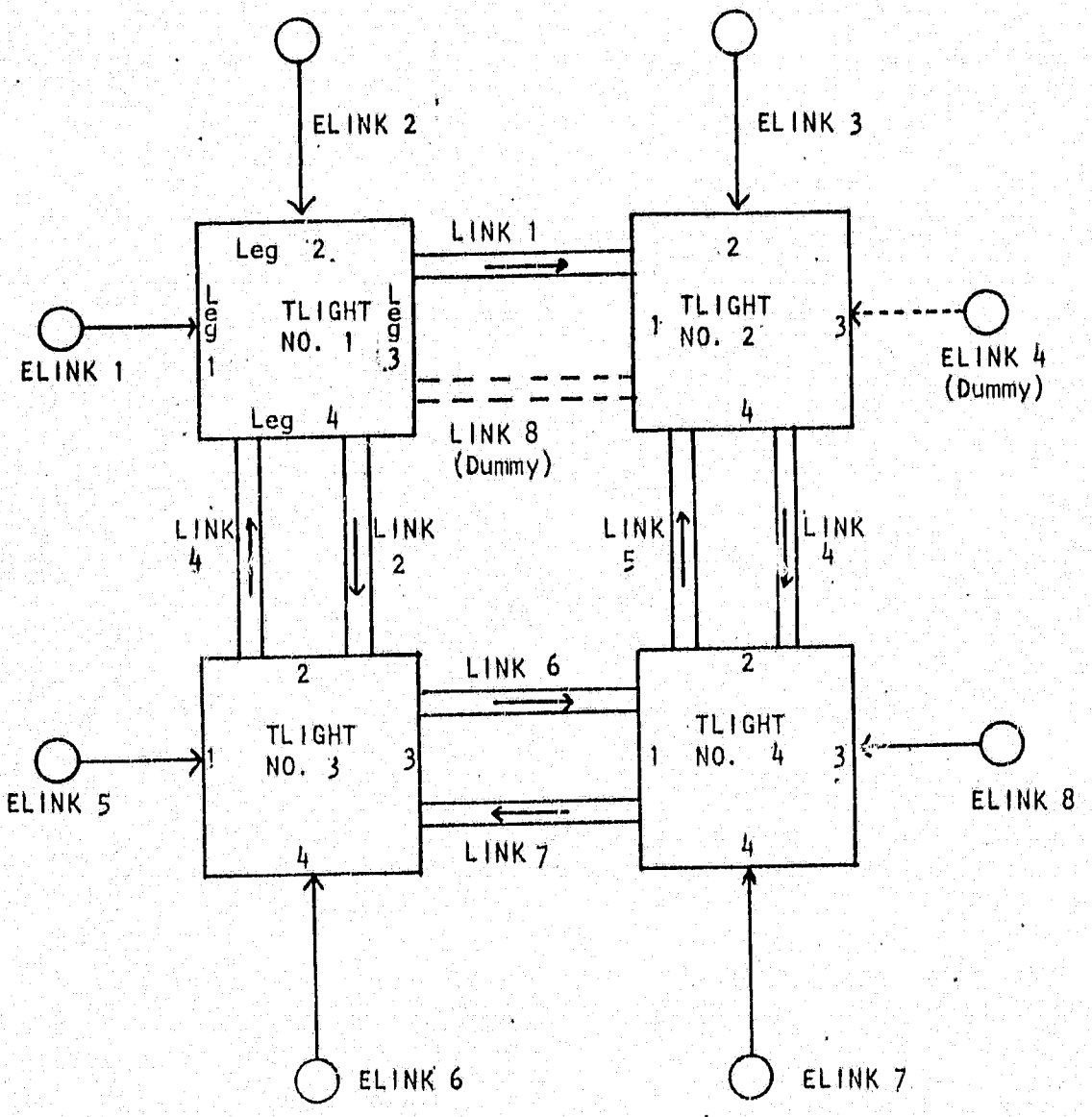

Figure 6.2. Symbolic Representation of the Network

```
TRAF

ENTER PROGRAM
" GET,TPROG

" INITIAL
" IGEN
" IFILE
" DYNAMIC
" ELINK,1
" ELINK,2
" ELINK,3
" ELINK,4
" ELINK,5
" ELINK,6
" ELINK,7
" ELINK,8
" LINK,1,TLIGHT,1,3
" LINK,2,TLIGHT,1,4
" LINK,3,TLIGHT,3,2
" LINK,4,TLIGHT,2,4
" LINK,5,TLIGHT,4,2
" LINK,6,TLIGHT,3,3
" LINK,7,TLIGHT,4,1
" LINK,8,TLIGHT,3,1
" TLIGHT,1,ELINK,1,ELINK,2,LINK,8,LINK,3
" TLIGHT,2,LINK,1,ELINK,3,ELINK,4,LINK,5
" TLIGHT,3,ELINK,5,LINK,2,LINK,7,ELINK,6
" TLIGHT,4,LINK,6,LINK,4,ELINK,8,ELINK,7
" PRINT,1,TLIGHT,1
" PRINT,2,TLIGHT,2
" PRINT,3,TLIGHT,3
" PRINT,4,TLIGHT 4
" WRITE,1,TLIGHT,1
" WRITE,2,TLIGHT,2
" WRITE,3,TLIGHT,3
" WRITE,4,TLIGHT,4
" TERMINAL
" CFILE
"
/E
```

Network Definition Statements

```
* GET,IDATA

* ELINK,1,100,200,300,150,100
* ELINK,2,200,300,100
* ELINK,3,200
* ELINK,4,0
* ELINK,5,300,150
* ELINK,6,200,300,100
* ELINK,7,250
* ELINK,8,250,100
* LINK,1,NORMAL,1100,300,12,23,35,18,12
* LINK,2,NORMAL,1000,400,11,50,17
* LINK,3,NORMAL,1000,400,33,50,17
* LINK,4,NORMAL,1100,400,100
* LINK,5,NORMAL,1100,400,100
* LINK,6,NORMAL,1440,400,45,11
* LINK,7,NORMAL,1440,400,71,29
* TLIGHT,1,1,1000,100,0,27,0,0,0,0,10,0,100
* TLIGHT,1,2,1700,100,0,20,0,0,0
* TLIGHT,1,4,1000,0,0,0,12,0,100
* TLIGHT,2,1,1500,100,0,22,0,0,0,0,15,0,100
* TLIGHT,2,2,1300,17,0
* TLIGHT,2,4,1300,0,20
* TLIGHT,3,1,1500,17,0,0,10
* TLIGHT,3,2,1500,20,0,0,0,0,100
* TLIGHT,3,3,1300,10,0,0,10
* TLIGHT,3,4,1700,20,0,0,0,0,100
* TLIGHT,4,1,1500,10,0,0,17
* TLIGHT,4,2,1300,10,15
* TLIGHT,4,3,1500,10,0,0,10
* TLIGHT,4,4,1300,20,20
* TLIGHT,1,5,0,15,7,2,15,7,2
* TLIGHT,2,5,12,30,7,2,30,7,2
* TLIGHT,3,5,0,30,7,1,30,7,1
* TLIGHT,4,5,14,30,7,2,30,7,2
* TLIGHT,1,6,NORMAL,500,100
* TLIGHT,2,6,NORMAL,500,100
* TLIGHT,3,6,NORMAL,500,100
* TLIGHT,4,6,NORMAL,500,100
* PRINT,1,10000,10000
* PRINT,2,10000,10000
* PRINT,3,10000,10000
* PRINT,4,10000,10000
* WRITE,1,1120,1110
* WRITE,2,1256,1320
* WRITE,3,1170,1170
* WRITE,4,1212,1170
```

Run Time Data Statements

```
*:/I

*:/R
NO OF RUNS = 30000
 10000   1    0  2  2  0  0    0  0  0  0  0    0  0  0  0  0    0  0  0  0  0
 10000   2    0  0  0  0  0    2  0  0  0  0    0  0  0  0  0    2  0  0  0  0
 10000   3    0  0  0  0  0    0  0  3  0  0    1  4  0  0  0    0  2  0  0  0
 10000   4    0  1  0  0  0    3  0  0  0  0    1  0  0  0  0    4  0  0  0  0
 20000   1    0  0  4  0  0    1  1  0  0  0    0  0  0  0  0    1  1  0  0  0
 20000   2    0  0  0  0  0    0  0  0  0  0    0  0  0  0  0    3  0  0  0  0
 20000   3    0  0  0  0  0    2  2  0  0  0    0  0  0  0  0    2  4  0  0  0
 20000   4    0  0  0  0  0    5  0  0  0  0    1  0  0  0  0    2  0  0  0  0

*:COPY
```

Running the Simulation of Four Intersections

QUEUES AT TRAFFIC LIGHT NO. 1

EPOCH	LEG NO 1					LEG NO 2					LEG NO 3					LEG NO 4				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1320	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2640	0	0	4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2	0	0
3960	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	4	2	0	0
5280	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	2	0	0
6600	0	0	0	0	0	2	2	1	0	0	0	0	0	0	0	0	1	0	0	0
7920	0	0	0	1	0	2	2	2	0	0	0	0	0	0	0	1	0	1	0	0
9240	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0
10560	0	0	2	0	0	2	2	0	0	0	0	0	0	0	0	2	2	1	0	0
11880	0	4	4	1	0	0	3	0	0	0	0	0	0	0	0	0	0	3	0	0
13200	2	0	2	1	1	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0
14520	0	4	5	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15840	2	2	1	0	2	0	2	0	0	0	0	0	0	0	0	1	1	0	0	0
17160	1	0	0	1	0	2	1	0	0	0	0	0	0	0	0	0	11	2	0	0
18480	0	2	1	0	0	5	4	0	0	0	0	0	0	0	0	2	5	2	0	0
19800	0	0	0	0	0	5	1	0	0	0	0	0	0	0	0	0	1	0	0	0
21120	0	0	0	0	0	2	2	0	0	0	0	0	0	0	0	1	2	2	0	0
22440	0	0	4	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
23760	2	2	4	0	2	0	1	0	0	0	0	0	0	0	0	0	1	2	0	0
25080	1	4	5	1	0	0	1	0	0	0	0	0	0	0	0	1	4	3	0	0
26400	2	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	3	2	0	0
27720	0	1	1	0	0	0	0	2	0	0	0	0	0	0	0	0	6	2	0	0
29040	1	0	3	0	0	2	1	0	0	0	0	0	0	0	0	4	0	3	0	0

Results of the Simulation of Four Intersections

QUEUES AT TRAFFIC LIGHT NO 2

EPOCH	LEG NO 1					LEG NO 2					LEG NO 3					LEG NO 4				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1356	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2676	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3996	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
5316	2	4	2	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6636	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	7	0	0	0	0
7956	0	3	4	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9276	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10596	3	2	3	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11916	0	0	0	0	1	2	0	0	0	0	0	0	0	0	0	2	0	0	0	0
13236	2	4	1	3	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14556	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	5	0	0	0	0
15876	1	0	4	2	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
17196	0	0	1	0	0	3	0	0	0	0	0	0	0	0	0	5	0	0	0	0
18516	1	2	5	4	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
19836	1	2	0	0	0	1	0	0	0	0	0	0	0	0	0	4	0	0	0	0
21156	1	2	2	3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22476	0	0	0	1	0	6	0	0	0	0	0	0	0	0	0	1	0	0	0	0
23796	2	5	4	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25116	0	0	0	1	0	2	0	0	0	0	0	0	0	0	0	7	0	0	0	0
26436	2	1	5	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27756	1	0	0	0	0	2	0	0	0	0	0	0	0	0	0	2	0	0	0	0
29076	2	2	7	6	1	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0

Results of the Simulation of Four Intersections

QUEUES AT TRAFFIC LIGHT NO 3

EPOCH	LEG NO 1					LEG NO 2					LEG NO 3					LEG NO 4				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1170	5	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
2340	1	0	0	0	0	0	0	0	0	0	4	2	0	0	0	0	1	0	0	0
3510	2	0	0	0	0	0	1	2	0	0	1	0	0	0	0	0	0	0	0	0
4680	1	1	0	0	0	4	2	1	0	0	4	0	0	0	0	2	5	1	0	0
5850	2	1	0	0	0	0	2	0	0	0	0	3	0	0	0	2	3	2	0	0
7020	2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
8190	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	1	1	0	0	0
9360	5	1	0	0	0	0	3	0	0	0	2	1	0	0	0	1	1	0	0	0
10530	4	2	0	0	0	0	0	0	0	0	4	1	0	0	0	0	0	0	0	0
11700	5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
12870	3	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	1	0	0
14040	0	1	0	0	0	2	3	1	0	0	1	0	0	0	0	4	0	4	0	0
15210	0	0	0	0	0	2	2	1	0	0	0	0	0	0	0	2	2	2	0	0
16380	1	2	0	0	5	1	0	1	0	0	0	0	0	0	0	1	2	0	0	0
17550	1	0	0	0	0	1	0	2	0	0	1	0	0	0	0	1	0	0	0	0
18720	2	0	0	0	0	0	0	0	0	0	1	4	0	0	0	0	0	0	0	0
19890	1	4	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
21060	3	1	0	0	0	0	1	0	0	0	4	2	0	0	0	0	0	0	0	0
22230	1	0	0	0	0	0	0	2	0	0	1	1	0	0	0	2	4	0	0	0
23400	0	0	0	0	0	1	2	0	0	0	1	0	0	0	0	4	1	2	0	0
24570	0	0	0	0	0	3	2	2	0	0	1	0	0	0	0	0	2	0	0	0
25740	0	1	0	0	0	0	3	0	0	0	4	0	0	0	0	1	2	0	0	0
26910	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0
28080	4	1	0	0	0	0	0	0	0	0	4	1	0	0	0	0	0	0	0	0
29250	2	4	0	0	0	0	0	4	0	0	2	2	0	0	0	0	0	0	0	0

Results of the Simulation of Four Intersections

QUEUES AT TRAFFIC LIGHT NO 4

EPOCH	LEG NO 1					LEG NO 2					LEG NO 3					LEG NO 4				
	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1212	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0
2382	0	0	0	0	0	1	0	0	0	0	5	0	0	0	0	1	0	0	0	0
3552	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4722	0	0	0	0	0	5	2	0	0	0	0	0	0	0	0	5	0	0	0	0
5892	11	3	0	0	0	1	0	0	0	0	0	0	0	0	0	14	0	0	0	0
7062	5	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
8232	3	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	0	0	0	0
9402	4	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
10572	3	1	0	0	0	1	0	0	0	0	4	0	0	0	0	0	0	0	0	0
11742	1	0	0	0	0	1	0	0	0	0	1	2	0	0	0	0	0	0	0	0
12912	0	1	0	0	0	6	0	0	0	0	0	3	0	0	0	3	0	0	0	0
14082	0	1	0	0	0	2	0	0	0	0	0	0	0	0	0	5	0	0	0	0
15252	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	2	0	0	0	0
16422	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	2	0	0	0	0
17592	2	0	0	0	0	5	0	0	0	0	0	2	0	0	0	4	0	0	0	0
18762	4	4	0	0	0	3	0	0	0	0	3	1	0	0	0	1	0	0	0	0
19932	0	2	0	0	0	2	0	0	0	0	2	1	0	0	0	0	0	0	0	0
21102	3	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0
22272	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
23442	1	0	0	0	0	1	0	0	0	0	1	2	0	0	0	1	0	0	0	0
24612	0	1	0	0	0	3	0	0	0	0	0	0	0	0	0	2	0	0	0	0
25782	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	1	0	0	0	0
26952	4	0	0	0	0	0	0	0	0	0	5	2	0	0	0	4	0	0	0	0
28122	5	0	0	0	0	0	0	0	0	0	2	1	0	0	0	3	0	0	0	0
29292	2	1	0	0	0	5	0	0	0	0	5	1	0	0	0	0	0	0	0	0

Results of the Simulation of Four Intersections

and printed in a similar way. The commands /I and /R the initialise the system and run the simulation. The output that follows is an example of the printout from the PRINT routine giving the epoch number, the intersection number and the queue lengths of each leg of the respective intersections. The final output is printed when the command: COPY is given.

6.3.4 Results of the Simulation

The simulation took 190 seconds to run. This corresponds to a real time to simulation time ratio of 52 for the four intersections. Since the network contained 4 intersections and an average of 2 lanes per approach per intersection a speed of 200 times real time per average intersection can be expected from the simulator. To a first approximation the speed of simulation is inversely proportional to the number of intersections and the number of lanes per intersection.

6.4 Conclusions

It has been shown in this chapter that the simulator is easy to use and is extremely fast. It has been found that the software functions correctly.

As pointed out in 6.2 the simulator has not been validated and this is an area for extensive future work.

CHAPTER 7

DISCUSSION AND CONCLUSIONS

7.1 Introduction

This chapter probes the strengths and weaknesses of the present simulator. The speed of simulation is examined and compared to similar work and validation of the simulator is discussed. Suggestions for future work are given and the use of the FASP executive for situations other than traffic is discussed.

7.2 Speed of Simulation

High speed of simulation was one of the main design criteria in the present work. A simulation speed of approximately 200 times that of real time was obtained (see 6.3.4). This was mainly due to two factors - i.e. efficient software and fast hardware data sources. The question that must be asked is whether the cost of providing hardware data sources is justified by the increase in speed of simulation.

The simulation routines ELINK, LINK and TLIGHT were examined and an estimate of run time was made by counting the number of instructions which would be executed per average epoch. A typical mix of these routines in a simulation was taken to be one ELINK, four TLIGHT and twelve LINK routines and the number of instructions executed per epoch was calculated. This process was repeated assuming that random number generation was implemented in software taking the time for generation of random numbers to be 240 μs which is equivalent to about 100 average instructions [19].

The analysis showed that the use of hardware data sources increased the average speed of simulation by a factor of 2. This figure is a conservative estimate and can be regarded as the minimum increase in speed due to the hardware data sources. Since the cost of the hardware data sources is about a quarter of the cost of the total system, their inclusion is considered to be worthwhile.

The efficiency of the software and use of a minicomputer can be judged by comparing the speed of the present work with that of Grigg and Hartley [8,9]. The main differences between the present work and that of Grigg and Hartley, is that a large time shared computer programmed in FORTRAN is used for the latter work.

A recent simulation [9] of ten intersections, forty links and ten entry links resulted in a simulation time to real time ratio of 8 to 1. Assuming the intersections to have two lanes per approach, a ratio of approximately 1 to 13 would be obtained in the present simulation, i.e. 104 times faster.

It should be noted that a NOVA 1200 minicomputer was used for the present work. This model is outdated and has been superceded by a cheaper, faster model, i.e. the NOVA 3. By using the new model simulation speed would be increased by a factor of about 2.

7.3 Validation of the Simulator

As mentioned in Chapter 6 the present simulator has not been validated. Future research must be devoted to this problem. The simplest method of validating the simulator would be to compare the results of a number of simulations with those obtained using the UTCS 1 [1] micro model which has previously been validated. This method of validation would also show the relative merits of macro and micro modelling and could pinpoint areas where the present macro modelling is sufficiently accurate and where the modelling requires refining.

7.4 Suggestions for Future Work

The most important area for future work is in refining vehicle dynamics and discharge at intersections. This work would be related to validation of the simulator. An investigation of behaviour of turning vehicles is required to ascertain a distribution which can be used for acceptance gaps. Since the data source used can provide random numbers with an arbitrary frequency distribution, it is unnecessary to confine this study to obtaining an acceptance gap distribution which can be represented mathematically.

Dynamic control policies may be simulated by modifying and extending the TLIGHT routine. Vehicle detectors may be simulated by testing the appropriate bit in the LINK word to determine the presence or absence of a vehicle. This information can be passed to the intersection routine to simulate a vehicle actuated control policy.

In testing dynamic control policies, it may be necessary to use alternative indices of performance instead of queue length Indices that may be of interest are number of stops per vehicle, a ge travelling time and average delay per vehicle. This could be provided by adding additional simulation routines or could be calculated from the relationships between these indices and queue lengths given by Gerlough and Wagner [41].

7.5 Use of FASP for Simulation of Stochastic Systems

The FASP executive was designed to be a general purpose simulation executive and may be used as the basis for simulating any discrete stochastic system. At present, studies of lift and pedestrian traffic are being undertaken in the department of Electrical Engineering. It is envisaged that simulation routines for simulating these situations will be written for use with the FASP executive.

7.6 Conclusions

A simulation executive and routines for simulating traffic have been described.

The executive provides a structure for the simulation and simplifies the programming and use of the traffic simulation routines. Being modular and flexible the executive can be used for simulating different situations.

The traffic simulation routines, while not being the last word in traffic simulators, illustrate the advantages of the approach used. Use of a small digital computer programmed in assembly language and hardware data sources produced an interactive simulator whose speed is far greater than any software simulator found in the literature.

Clearly the approach is worthwhile and it is hoped that future work will produce a useful traffic design tool using this approach.

APPENDIX 1

USING THE TRAFFIC SIMULATOR UNDER DDOS 1.6

traffic simulator is loaded from disk by typing: TRAF.

The program will start automatically and print out ENTER PROGRAM to indicate that FASP is in the program mode. The network definition statements may then be entered and the simulation may be run.

APPENDIX 2

FASP CONSOLE PROCEDURES

Input from the console is placed in a buffer 74 characters long. Editing may be performed by use of the rub out and control X keys. The rub out key deletes the last character entered. It may be used repeatedly. The control X key deletes the entire line typed. A space or a tab is echoed back to the console as a space but is not stored in the buffer. Line feeds and other non printing characters are ignored so that the input medium could be paper tape for batch operation. A carriage return denotes the end of a statement or command and control is transferred to the approximate routine in FASP.

APPENDIX 3

FASP ERROR MESSAGES

No.	Meaning	Possible cause
1.	Memory overflow. This error is fatal and FASP must be reloaded after the source of error has been located and rectified	(a) Routine specified too much storage for a particular memory size. (b) Incorrect header format.
2.	User routine undefined.	Routine specified in call statement could not be found.
3.	Zero or negative address specified	The number of a routine was zero, negative or not given.
4.	Address to large	The number of a routine was greater than the number of storage blocks provided for that routine.
5.	Illegal break character	A break character was not, ; or carriage return. Usually an alphabetic character following a number
6.	Illegal command	Unknown executive command or no data routine.
7.	Precision error in PUT.	Precision >5 or <1 specified following PUT pseudo instruction

No.	Meaning	Possible cause
8.	Too many parameters in list	
9.	Too few parameters in list.	
10.	PUTP illegal in Data routine.	
11.	File already exists	
12.	File does not exist.	

APPENDIX 4

CREATING A FASP SYSTEM UNDER DDOS 1.6

To create a simulation system the various modules of FASP and user supplied routines must be loaded by the relocateable loader. This is easily done under DDOS with the following command.

LINK (IMAGEFILE, $TTO, FASP:B, INTERUPT SERV ROUTINE$_1$, ...
. . . . INT SERV ROUTINE$_N$, SIMULATION ROUTINE$_1$, ...
SIMULATION ROUTINE$_N$ FASPEND:B).

INTERRUPT SERV ROUTINE and SIMULATION ROUTINE 1 to N are the names given to the binary object modules of interrupt service routines and simulation routines supplied by the user.

The resultant image on IMAGEFILE may now be loaded and run.

A sample map produced by the loader is shown overleaf.

```
DDOS 1.6 RELOCATING LINKER VER 1.0 - 07/22/75   TRAF

TITLE   NREL    ZREL    START

FASP    000400  000050
ELINK   002377  000117
IGEN    002530  000120
LINK    002573  000121
.MAIN   003163  000122
PRINT   003277  000122
WRITE   003423  000123
TLIGH   004666  000131
FASPE   005604  000140  007164
        023437  000147

START ADDR      007164


SYMBOL TABLE

ACT    000111   AJTAB  000106   AMTAB  000107   C177   000034
C3     000023   C77    000031   COMD   001140   COMP3  007406
COMPS  007307   CRLF   006073   DDOS   000145   DFLG   000110
DIVID  006062   DWRIT  005667   ERET   000053   ERROR  006054
GET    006055   GETAD  006057   GETAS  006065   GETBR  006060
GETC   006071   IEND   000144   IGNOR  001602   INIT   001376
INIT1  007164   INITR  001406   INTR   000105   IPT    000113
ISTRT  000143   MES    006074   MULTI  006063   NONE   100077
OPT    000112   PROG   000116   PUT    006056   PUTC   006070
PUTP   006146   RETUR  002061   SP     000041   TEMP2  000044
TEMP3  000045   TR2    001503   USEND  000140   .BUS   000100
.COMD  000075   .GETC  000066   .PUTC  000067   .READ  000071
.RET   000061   .WRIT  000070   .XEC   000076
```

APPENDIX 5

DESIGN OF A HARDWARE RANDOM PULSE GENERATOR

Chapter 4 described a system which could produce 16 independent random pulse outputs from a single random number generator. A detailed description of the random number generator was given in that chapter and the description of the remaining circuitry is given below.

Design of the Comparator

The comparator must compare two 12 bit numbers and give a 1 at the output if the generated number is less than the number in the memory. Three 4 bit comparators type SN7485 are used by connecting the ripple carry outputs to the carry inputs.

An alternative arrangement can be used to give a faster comparison but the extra complexity was not warranted as the overall speed of the generator was high enough to satisfy the speed of operation condition given in 4.2.2.

Design of the Memory

The 16 word 12 bit memory is made by connecting the address lines and read/write enable inputs of three 16 word - 4 bit memories type SN7489 in parallel.

Additional circuitry is provided so that the memories can be programmed either under computer control or from switches on the front panel. To do this a 16 bit word is required. The four most significant bits contain the memory address and the remaining 12 the data. The basic memory circuit is shown in Fig. A5.1. Open collector gates are used in the buffer to permit wire - OR connections.

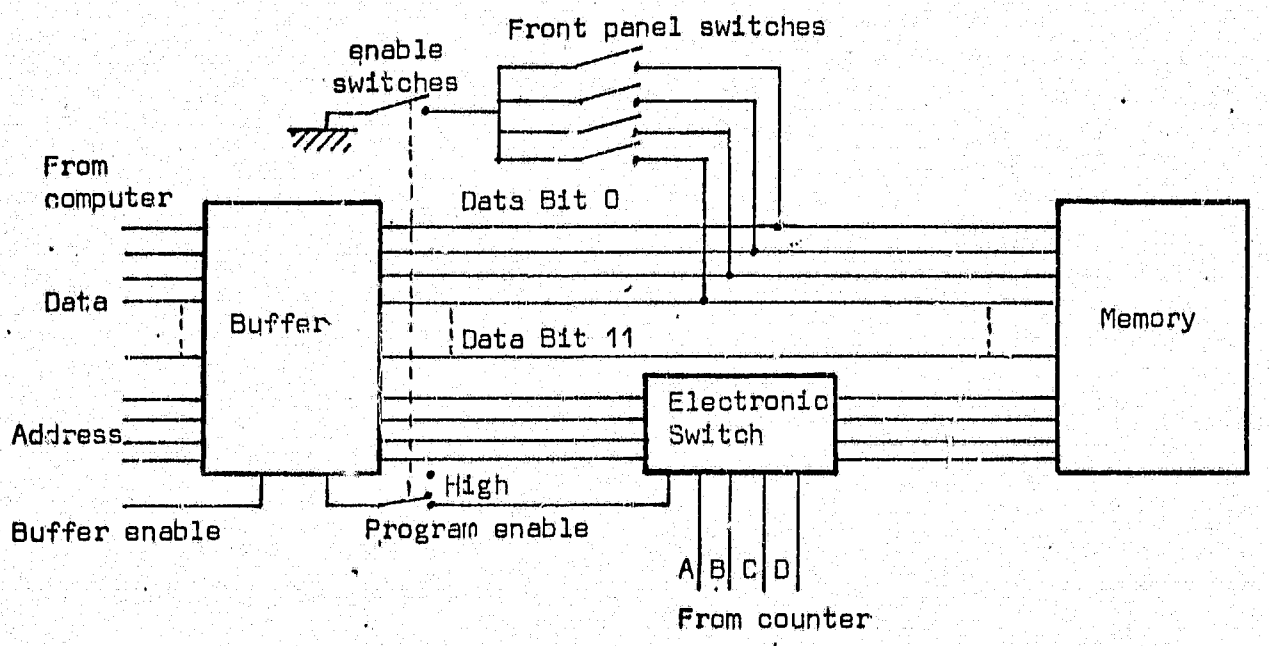

Fig A5.1 Memory Circuit

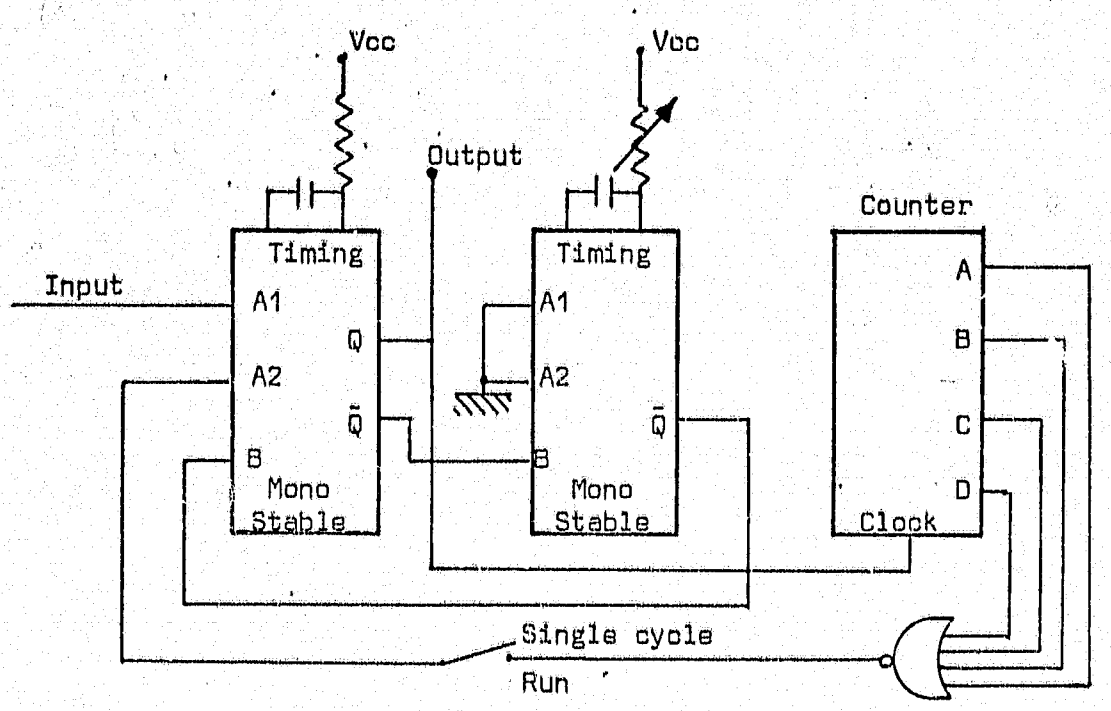

Fig A5.2 Internal Clock

This simplifies the logic and reduces the number of contacts required on the memory board connector.

Design of the Internal Clock

The internal clock provides the timing pulses for the system. When a pulse is applied at the input, the clock starts and after 16 pulses stops again. Thus for each input pulse the system generates 16 random numbers. The circuit uses two cross coupled monostables which comprise a triggered multivibrator. The multivibrator runs for as long as A1 or A2 on IC 1 is held low. The input pulse is applied to A1 and A2 is connected via a NOR gate ot the 4 outputs of the address counter. A negative going pulse at the input starts the clock which clocks the counter. The output of the NOR gate goes low and remains low till the counter reaches the all zero state causing A2 to go high which stops the clock. The circuit used is given in Fig. A5.2.

Features of the Pulse Generator

The following front panel controls are provided on the generator.

(a) Run - single cycle

This switch selects the mode of the internal clock. In the run position the generator goes through the complete cycle and generates 16 random pulses for each input pulse applied. This is the normal mode of operation. The single cycle position disconnects the internal clock and is used for examining the memory contents or for testing every word of the uniform random number generator.

(b) Programming Switches

Sixteen switches are provided for manually programming the memory. The memory address is set up on the first four switches and the data on the remaining 12.

(c) Program - Run

This switch is used to apply the data from the programming switches to the memory inputs.

(d) Program Pulse

This push button pulses the write enable input of the memory and stores the data set up with the programming switches in the memory.

(e) Pulse Button

This push button applies a pulse to the input of the internal clock and is used for checking out the system and examining memory contents in conjunction with switch (a).

(f) Clear Button

This push button resets the generator to the all one condition, sets the address counter to zero and clears the output shift register.

(g) Visual Read Out

Three 16 bit LED displays are provided.

(i) Memory address and data

This is used for examining the contents of the memory.

(ii) Uniform random number

This displays the 16 least significant bits of the uniform random number generator. [illegible] 4 of these bits are not used by the system but are useful for checking the operation of the generator.

(iii) Output register

This displays the states of the 16 independent random pulse channels.

(h) Computer Operation

The system is provided with a suitable interface to connect it to a NOVA computer. The device code is 70.

(i) Buffer A (input)

This buffer contains the 16 independent outputs of the generator which can be read into the computer with the following instruction :

DIA AC,70

(ii) Buffer A (output)

This buffer receives data from the computer to program the memory.

The four most significant bits (i.e. bits 0 - 3) must contain the address and the remaining bits (bits 4 - 15) the data.

The following instruction is used:

DOA AC,70

N.B. For programming via the computer the programming switches (b) must all be in the "one" position.

(iv) Pulse

A P suffix to any input/output instruction applies a pulse to the input of the internal clock and performs the same function as push button (e).

(v) Clear

A C suffix clears the device as described in (F).

System Tests

After constructing the system the output of the uniform random number generator was checked against a software simulation and it produced identical results as expected. The various modes of operation were then checked. The only calibration procedure is to

set the timing of the internal clock to the fastest possible speed. An examination of the system and the manufacturers' specified propagation times revealed that the critical delay path was: address counter, memory and comparator. Typical delay for this combination was given as 120 ns. The delay of the implemented system was found to be approximately 100 ns. The internal clock was set to 120 ns which was the highest speed the internal clock could operate at. This means that 1,92 µs are required to generate the 16 random pulses.

APPENDIX 6

STATISTICAL TESTS ON THE POISONION DISTRIBUTION RANDOM NUMBER GENERATOR

Table A6.1. Frequency Tests on the Uniform Random Number Generator

Bits	Lower Bound Failures	Upper Bound Failures
0 - 3	65	43
4 - 7	54	66
8 - 11	51	58

Table A6.2. Serial Correlation Test on the Uniform Random Number Generator

Number Tested	Bits 0 - 3		Bits 3 - 7		Bits 8 - 1	
	Lower Bound Failures	Upper Bound Failures	Lower Bound Failures	Upper Bound Failures	Lower Bound Failures	Upper Bound Failures
0	13	19	23	16	15	15
1	20	24	19	13	14	16
2	21	22	21	18	21	17
3	15	24	23	17	15	20
4	10	26	18	19	18	30
5	11	21	15	21	13	19
6	16	14	24	19	16	22
7	20	17	15	24	16	14
8	24	19	6	16	18	20
9	19	26	19	17	18	27
10	16	17	14	16	19	27
11	13	17	18	16	18	14
12	15	16	18	24	22	18
13	23	21	23	14	18	16
14	22	21	14	17	19	21
15	19	19	13	16	21	27

Table A6.3. Autocorrelation Test on Bits 8 - 11 (M.S.B.)

Run Number	Mean Value of Coefficients	Standard Deviation of Coefficients	Chi-Square Value	Largest Coefficient	Delay Value
1	+000323	0,05673	1,95957	-0,20525	3
2	-000321	0,07212	0,64642	-0,16478	16
3	-000310	0,06753	1,05047	-0,23400	3
4	-000182	0,06163	0,44439	-0,21077	3
5	+000885	0,06832	6,80813	-0,29230	3
6	-000065	0,08638	2,26261	-0,26174	3
7	-000231	0,06138	5,89904	+0,21418	49
8	-000510	0,06616	1,85856	-0,18378	33
9	+000259	0,05740	1,35351	+0,20513	77
10	+000317	0,06832	2,36361	-0,20258	26
11	-000594	0,05871	5,78903	-0,15264	77
12	+000033	0,05397	0,74743	-0,16572	79
13	+000126	0,05633	2,36361	-0,15162	29
14	+000637	0,05819	2,46464	-0,14782	2
15	+000154	0,05268	0,34338	-0,14899	35
16	+000065	0,05284	2,97981	-0,14098	47
17	+000112	0,05226	2,16160	+0,14271	81
18	+000321	0,05999	3,97981	-0,14499	85
19	-000158	0,05606	4,28285	-0,14113	62
20	-000722	0,05961	0,94946	-0,18400	61

$X^2_{.95} = 0,103$ $X^2_{.05} = 5,991$

Table A6.4. Autocorrelation Test on Bits 0 - 7

Run Number	Mean Value of Coefficients	Standard Deviation of Coefficients	Chi-Square Value	Largest Coefficient	Delay Value
1	+0,00294	0,05439	1,99998	+0,16412	16
2	+0,00502	0,05496	0,79995	-0,13050	46
3	-0,00168	0,05819	0,49996	-0,13762	46
4	-0,00432	0,05645	0,99996	-0,13744	26
5	-0,00222	0,05311	3,80001	+0,14553	32
6	+0,00007	0,05897	3,60000	+0,13363	42
7	+0,00028	0,06150	1,49998	+0,17218	73
8	+0,00245	0,05766	2,50000	-0,15540	2
9	+0,00608	0,04751	5,50004	+0,14337	87
10	+0,00650	0,05659	3,60000	+0,14248	41
11	+0,01779	0,05181	1,69999	+0,14292	11
12	+0,02070	0,04925	6,30006	+0,13299	10
13	+0,00578	0,06236	6,70005	+0,19004	89
14	-0,00383	0,05195	4,30003	-0,15386	56
15	+0,00346	0,05740	2,50000	+0,16703	55
16	-0,00759	0,05766	2,90002	-0,19237	1
17	-0,00190	0,05496	7,80007	-0,15751	97
18	+0,00241	0,06224	5,30004	-0,14560	56
19	+0,00202	0,05766	0,89997	-0,15798	63
20	+0,00230	0,06224	2,80001	+0,16790	66

$X^2_{.95} = 0,103$ $X^2_{.05} = 5,991$

Table A6.5. Frequency Test.

Upper Bound Failures

Channel No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	TOTAL
Flow Rate																	
100	5	6	4	2	5	6	2	6	8	7	8	8	7	8	9	3	94
200	5	3	5	6	5	6	1	8	4	6	5	7	6	4	2	2	75
300	6	5	1	6	5	3	4	5	8	3	1	7	10	7	4	8	83
400	6	8	3	2	4	5	7	5	9	3	6	4	14	7	7	6	96
599	6	6	3	3	7	4	5	7	8	1	3	5	7	1	4	8	78
600	3	4	3	6	7	3	4	8	10	4	7	4	7	5	5	9	96
700	5	6	5	4	8	2	3	6	6	3	4	6	10	7	6	4	85
800	8	2	4	3	6	2	1	8	1	5	4	5	8	4	10	4	75
900	7	4	3	7	7	3	2	5	2	5	6	5	6	4	8	7	81
1000	8	6	3	0	6	2	5	7	3	6	5	4	5	8	7	9	84
1100	5	6	4	1	4	2	4	12	4	4	1	5	4	5	6	7	74
1200	6	5	4	3	3	0	4	3	3	4	3	2	6	6	6	10	68
1300	6	2	6	0	4	1	0	3	3	9	9	4	4	3	6	7	67

Table A6.6. Frequency Test

Lower Bound Failures

Channel No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	TOTAL
Flow Rate																	
100	7	15	14	3	8	7	6	10	6	8	10	10	8	6	17	5	140
200	8	6	16	9	8	13	11	7	9	7	9	11	8	11	7	5	145
300	5	6	8	11	13	9	8	13	10	8	13	10	8	6	8	11	178
400	4	5	9	9	8	8	6	5	14	19	10	7	8	15	7	11	145
500	5	9	12	11	13	9	5	10	12	9	10	6	16	5	8	7	147
600	7	7	8	13	13	9	11	9	6	4	9	6	11	12	6	4	135
700	13	15	10	12	12	10	7	8	12	14	8	6	5	12	10	8	164
800	10	8	9	10	6	11	10	11	7	12	5	9	5	8	6	4	131
900	4	9	7	7	12	10	11	6	7	10	9	6	12	10	10	8	138
1000	8	10	6	13	8	11	5	8	11	9	10	8	8	8	14	10	148
1100	7	11	6	12	10	8	13	16	6	10	12	9	5	10	7	5	147
1200	8	12	12	9	9	10	14	7	9	12	8	14	7	11	8	11	161
1300	7	13	13	13	6	7	13	14	9	8	4	12	6	8	10	6	145

Table A6.7. Gap Test

Upper Bound Failures

Channel No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	TOTAL
Flow Rate																	
100	3	4	2	2	3	2	2	1	0	3	1	2	1	2	29		
200	1	3	1	1	1	2	3	0	3	4	3	1	4	2	0	1	30
300	4	1	2	1	2	4	2	1	4	3	2	0	2	3	1	0	32
400	1	3	1	2	2	1	2	0	1	1	4	3	4	1	0	1	27
500	2	0	1	1	0	2	2	0	1	1	0	1	0	3	3	2	19
600	1	1	1	0	0	3	6	2	5	1	3	1	3	2	1	3	33
700	3	2	2	3	0	0	1	0	0	1	4	1	3	3	2	3	28
800	1	1	1	1	0	2	3	2	2	1	3	0	3	2	0	4	26
900	3	2	3	3	2	2	3	1	1	2	1	1	2	2	2	4	34
1000	2	3	3	0	1	3	2	4	3	1	1	2	3	4	0	4	34
1100	1	1	2	0	4	1	7	2	2	2	2	0	2	6	2	2	36
1200	4	0	1	5	1	4	6	6	1	1	3	3	6	4	3	2	50
1300	2	2	1	2	2	1	4	4	4	3	4	2	5	1	4	2	43

Table A6.8. Gap Test

Lower Bound Failures

Channel No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	TOTAL
Flow Rate																	
100	3	2	3	0	3	1	3	1	1	3	3	1	0	2	4	3	33
200	2	2	3	2	0	5	2	1	3	4	4	4	1	2	3	2	40
300	2	1	1	0	3	3	2	3	2	1	2	0	0	3	3	2	28
400	1	3	2	0	0	2	2	1	3	2	1	5	1	2	3	3	31
500	1	4	8	1	1	5	3	0	3	4	1	1	1	2	3	4	42
600	1	2	0	3	2	3	2	1	3	3	1	0	2	3	5	1	32
700	1	2	2	1	4	3	3	1	3	1	0	1	1	0	4	4	31
800	4	3	2	0	1	5	3	2	1	2	0	2	4	0	0	2	31
900	2	1	0	0	0	4	0	3	1	3	2	3	2	3	0	1	25
1000	3	0	3	2	1	2	2	1	1	2	1	0	2	3	0	0	23
1100	0	3	1	2	0	2	0	1	3	0	2	2	1	0	2	1	20
1200	3	2	2	1	1	3	1	1	2	1	0	0	0	1	1	2	21
1300	0	0	1	0	2	2	0	0	0	1	1	2	1	1	1	0	12

Table A6.9. Serial Correlation Numbers Following Upper Bound Failures

Channel No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	TOTAL
Flow Rate																	
100	2	1	1	0	3	4	1	1	0	1	4	0	1	1	1	6	27
200	0	1	2	4	4	6	4	6	4	2	4	0	2	2	2	4	47
300	5	6	7	7	9	8	7	7	6	7	6	6	5	7	5	4	102
400	6	3	2	5	6	2	6	6	6	3	5	6	8	6	9	6	85
500	2	5	5	4	5	5	4	5	8	4	8	6	7	7	8	5	89
600	5	6	5	6	1	6	6	1	5	7	5	5	6	6	5	4	75
700	5	5	3	6	5	5	4	5	7	6	6	1	8	3	7	2	78
800	4	6	4	5	4	4	6	6	4	5	6	4	4	3	9	1	75
900	4	6	7	3	5	3	4	6	3	3	5	5	4	5	6	0	69
1000	0	2	5	5	5	2	3	4	1	2	7	7	3	6	6	3	61
1100	2	3	5	8	5	2	8	4	5	3	5	3	9	4	6	3	75
1200	4	5	2	5	4	2	1	5	8	5	6	7	9	5	7	6	81
1300	6	2	1	2	6	0	7	4	4	4	4	6	5	6	6	7	70

Table A6.10. Serial Correlation: Numbers Following 1.
Lower Bound Failures.

Channel No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	TOTAL
Flow Rate																	
100	0	11	7	11	8	7	12	9	12	8	9	9	12	7	8	5	144
200	8	11	7	13	8	9	11	15	6	6	11	8	11	11	10	9	154
300	9	4	7	12	7	8	11	12	8	8	13	9	7	4	5	6	130
400	13	9	10	6	7	6	7	8	7	11	14	10	10	6	7	9	140
500	12	9	12	13	6	11	11	7	9	12	11	13	12	10	10	12	169
600	11	8	15	6	12	10	10	14	6	7	8	8	8	6	7	7	141
700	7	4	10	8	3	21	7	11	7	12	8	8	14	10	9	12	151
800	7	8	8	10	8	8	11	11	14	11	8	10	6	6	10	11	147
900	8	7	16	10	15	7	11	14	9	11	7	13	14	11	12	5	110
1000	7	10	10	10	14	8	14	9	15	13	12	13	15	12	11	8	179
1100	9	10	14	8	10	3	6	7	9	10	10	8	12	8	10	6	180
1200	8	3	4	7	8	7	13	7	10	15	6	9	7	11	7	9	131
1300	7	6	11	8	5	7	7	11	10	12	10	7	6	7	13	5	121

Table A6.11. Serial Correlation: Numbers Following 0.

Upper Bound Failures

Channel No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	TOTAL
Flow Rate																	
100	7	3	5	4	6	5	4	6	7	9	8	8	7	4	10	8	10[illegible]
200	6	5	3	6	8	5	4	7	4	8	3	6	5	3	7	4	84
300	5	7	7	6	5	2	8	11	5	5	3	7	10	8	5	2	95
400	7	6	4	8	7	6	5	7	8	4	5	7	11	12	4	8	109
500	10	8	3	8	5	3	6	8	8	4	3	6	8	5	8	8	101
600	7	4	5	3	6	8	5	10	9	5	4	6	5	7	7	5	96
700	7	4	3	4	8	4	6	6	7	3	6	6	7	9	5	7	92
800	8	2	1	7	6	5	4	10	3	5	3	5	9	6	7	5	86
900	10	5	3	7	8	1	6	9	7	7	3	4	13	9	6	10	108
1000	7	7	3	5	3	3	5	1	6	8	5	6	6	7	7	8	87
1100	9	7	3	2	8	5	3	3	5	7	4	4	7	6	6	11	90
1200	10	5	3	2	5	2	3	7	5	7	5	6	7	6	7	6	86
1300	7	4	3	6	4	5	5	4	3	6	2	5	4	3	6	7	74

Table A6.12. Serial Correlation: Numbers Following 0.

Lower Bound Failures

Channel No. / Flow Rate	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	TOTAL
100	9	10	4	9	11	10	11	11	8	11	8	7	8	6	14	5	142
200	8	9	8	8	10	11	8	7	10	8	5	11	14	8	15	6	146
300	7	7	11	16	6	12	13	10	12	9	7	5	5	12	9	8	149
400	11	10	15	4	11	14	11	13	18	10	7	9	5	10	9	8	165
500	7	5	9	7	11	11	14	9	13	9	9	12	13	6	8	9	152
600	9	9	17	8	14	9	11	12	16	8	7	7	10	12	8	7	164
700	14	14	9	10	10	11	10	9	7	11	5	5	13	7	9	4	148
800	9	13	10	12	10	6	14	13	11	16	10	13	6	9	12	12	176
900	4	9	9	5	4	10	8	10	12	3	10	10	7	7	13	7	128
1000	6	11	13	10	10	8	9	8	10	10	10	12	9	10	11	7	154
1100	7	7	7	7	6	11	5	7	13	7	10	5	4	10	10	7	123
1200	8	11	7	11	13	7	9	8	9	12	11	10	6	8	5	9	144
1300	10	12	3	11	11	8	11	10	7	10	14	10	7	4	9	11	148

APPENDIX 7

PROGRAMS FOR FREQUENCY AND SERIAL CORRELATION TESTS

```
20   PRINT "NO. OF RUNS REQUIRED =";
21   INPUT K
22   PRINT
70   PRINT "FREQUENCIES (1) OR CHI↑2 RESULT(0)";
80   INPUT A
90   PRINT
100   IF A=1 GOTO  140
110   PRINT "SUMMARY (0) OR FULL TABLE (2)";
120   INPUT A
130   PRINT
131   PRINT
132   PRINT
133   PRINT "**************"
140  LET C1=3.841
150  LET C2=.004
151  FOR B1= 0 TO 15
152     PRINT "BIT TESTED =";B1
153     PRINT "**************"
154     FOR P=100 TO 1300 STEP 100
155       PRINT "BLOK RATE =";P
156       LET L= 0
157       LET H= 0
160       LET P=P/102400
165       LET S= INT (20/P+.5)
170       LET P1=P*4095
200       LET B9= INT (2↑B1+.5)
210       LET B2= INT (2↑12*B1+.5)
211       LET P9=P1+B2
220       CALL "PROG",P9,B9
261       PRINT
262       PRINT
270       IF A<>2 GOTO  310
271       PRINT "RUN NO.","CHI↑2 VALUE"
310       FOR J=1 TO K
320         CALL "PULSE",S,E
330         LET E=S-E
430         IF A<>1 GOTO  470
440         PRINT "NO.","OBSERVED","EXPECTED"
450         PRINT " ","FREQUENCY","FREQUENCY"
470         LET Y=S*(1-P)
475         LET X=(S-E-Y)*(S-E-Y)/Y
480         IF A<>1 GOTO  500
490         PRINT "0",S-E,Y
500         LET Y=S*P
505         IF A<>1 GOTO  515
510         PRINT "1",E,Y
515         LET X=X+(E-Y)*(E-Y)/Y
520         IF A<>1 GOTO  560
530         PRINT "CHI↑2 VALUE =";X
560         IF X<C1 GOTO  590
570         LET H=H+1
580         GOTO  640
590         IF X>C2 GOTO  620
600         LET L=L+1
620         IF A= 0 GOTO  640
630         PRINT J,X
640       NEXT J
650       PRINT
660       PRINT "UPPER FOUND FAILURES =";H
690       PRINT "LOWER FOUND FAILURES =";L
691       PRINT
692       PRINT
693     NEXT P
694   NEXT B1
700   END
```

```
                        .TITLE  .FREQ
                        .EXTN   .CALL, RETURN, GET, PUT .
        000001          .TXTM   1
                        .NREL
000000'000022'          NEXT
00001'177777            -1
00002'177777            -1
00003'177777            -1
00004'177777            -1
00005'177777            -1
00006'150322            .TXTF   "PR
00007'147707  OG
00010'000000  "
00011'177777            .CALL
00012'177777            GET
00013'065070            LDA     1,70
00014'000011'           .CALL
00015'000012'           GET
00016'044435            STA     1,.C1
00017'060270            NIOC    70
00020'000014'           .CALL
00021'177777            RETURN
        000022'         NEXT=.
00022'000056'           NEXT1
00023'177777            -1
00024'177777            -1
00025'177777            -1
00026'177777            -1
00027'177777            -1
00030'150325            .TXTF   "PU
00031'146323  LS
00032'142400  E"
00033'000020'           .CALL
00034'000015'           GET
00035'044420            STA     1,.SAMP
00036'024415            LDA     1,.C1
00037'100400            SUB     0,0
00040'042414            STA     0,.C2
00041'070770  LOOP:     DIAP    2,70
00042'133404            AND     1,2,SZR
00043'010411            ISZ     .C2
00044'014411            DSZ     .SAMP
00045'000774            JMP     LOOP
00046'024406            LDA     1,.C2
00047'000033'           .CALL
00050'177777            PUT
00051'000047'           .CALL
00052'000021'           RETURN
        000001  .C1:    .BLK    1
        000001  .C2:    .BLK    1
        000001  .SAMP:  .BLK    1
        000056'         NEXT1=.
                        .END
```

```
                        .TITLE SERIAL
                        .EXTN  .CALL, RETURN, GET, PUT
        000061          .TXTM   1
                        .NREL
000000'000022'          NEXT
000001'177777           -1
000002'177777           -1
000003'177777           -1
000004'177777           -1
000005'177777           -1
000006'150322           .TXTF   "PR
000007'147757   OG
000010'000100   "
000011'177777           .CALL
000012'177777           GET
000013'065070           DOA     1,70
000014'000011'          .CALL
000015'000012'          GET
000016'044443           STA     1,.C1
000017'000270           NIOC    70
000020'000014'          .CALL
000021'177777           RETURN
        000022'         NEXT=.
000022'000064'          NEXT1
000023'177777           -1
000024'177777           -1
000025'177777           -1
000026'177777           -1
000027'177777           -1
000030'150325           .TXTF   "PU
000031'146323   LS
000032'142400   E"
000033'000000'          .CALL
000034'000015'          GET
000035'044426           STA     1,.SAMP
000036'002423           LDA     1,.C1
000037'102400           SUB     0,0
000040'141422           STA     0,.C2
000041'077770   LOOP:   DIAP    2,70
000042'133405           AND     1,2,SNR
000043'000776           JMP     LOOP
000044'014417   LP1:    DSZ     .SAMP
000045'000402           JMP     .+2
000046'000406           JMP     F1
000047'077770           DIAP    2,70
000050'133405           AND     1,2,SNR
000051'000770           JMP     LOOP
000052'010410           ISZ     .C2
000053'000771           JMP     LP1
000054'002400   F1:     LDA     1,.C2
000055'000000'          .CALL
000056'177777           PUT
000057'000055'          .CALL
000060'000021'          RETURN
        000001  .C1:   .ILK    1
        000001  .C2:   .ILK    1
        000001  .SAMP: .ILK    1
        000064'         NEXT1=.
                        .END
```

APPENDIX 8

PROGRAM FOR GAP TESTING

```
5  DIM E[10],Z[,P]
6  LET H= 0
7  LET L= 0
10  PRINT "FLOW RATE =";
20  INPUT F
30  PRINT
40  PRINT "NO. OF SAMPLES = ";
50  INPUT S
60  PRINT
70  PRINT "FREQUENCIES (1) OR CHI↑2 RESULT(2)
80  INPUT A
90  PRINT
100  IF A=1 GOTO  160
110  PRINT "SUMMARY (0) OR FULL TABLE (2)";
120  INPUT A
130  PRINT
140  LET C1=16.919
150  LET C2=3.325
160  LET F=F/10800
170  LET F1=F*4095
180  PRINT "TEST WHICH FIT (0-15)";
190  INPUT B1
200  PRINT
205  PRINT "NO. OF RUNS REQUIRED =";
206  INPUT R
207  PRINT
208  LET E9= INT (2↑F1+.5)
210  LET E2= INT (2↑12*F1+.5)
211  LET F9=E1+E2
220  CALL "PLOG",F9,E9
225  LET E[ 0]= 0
230  FOR I=1 TO 9
240    LET E[I]= INT (-1/I* LOG (1-(I*(1- EXP (-F))/10/F))-.5)
250  NEXT I
260  LET E[10]=1000
261  PRINT
262  PRINT
270  IF A=1 GOTO  310
271  PRINT "RUN NO.","CHI↑2 VALUE"
310  FOR J=1 TO R
320    CALL "INIT",S,E[ 1],E[ 2],E[ 3],E[ 4],E[ 5],E[ 6],E[ 7],E[ 8],E[ 9],F[ 10
J
330    CALL "PULSE",A[ 0],A[ 1],A[ 2],A[ 3],A[ 4],A[ 5],A[ 6],A[ 7],A[ 8],A[ 9]
```

```
430     IF A<>1 GOTO  451
440     PRINT "     GAP ","OBSERVED","EXPECTED"
450     PRINT " ","FREQUENCY","FREQUENCY"
451     LET Y1= 0
452     LET X= 0
460     FOR I= 0 TO 9
470       LET Y=S*(P*(1- EXP (-P*E[I+1]-P))/(1- EXP (-P))-Y1)
475       LET Y1=Y1+Y/S
480       IF A<>1 GOTO  500
490       PRINT E[I];E[I+1],Z[I],Y
500       LET X=X+(Z[I]-Y)*(Z[I]-Y)/Y
510     NEXT I
520     IF A<>1 GOTO  560
530     PRINT "CHI-2 VALUE = ";X
560     IF X<C1 GOTO  590
570     LET H=H+1
580     GOTO  630
590     IF X>C2 GOTO  620
600     LET L=L+1
620     IF A= 0 GOTO  640
630     PRINT J,X
640   NEXT J
650   PRINT
680   PRINT "UPPER BOUND FAILURES = ";H
690   PRINT "LOWER BOUND FAILURES = ";L
691   PRINT
692   PRINT
693   GOTO  5
700   END
```

```
                        .TITLE  GAP
                        .EXTN   .CALL, RETURN, GET, PUT
        000001          .TXTM   1
                        .NREL
00000'000023'           NEXT
00001'177777            -1
00002'177777            -1
00003'177777            -1
00004'177777            -1
00005'177777            -1
00006'150322            .TXTF   "PR
00007'147707  OG
00010'000000  "
00011'177777            .CALL
00012'177777            GET
00013'065070            DOA     1,70
00014'000011'           .CALL
00015'000012'           GET
00016'044404            STA     1,.C1
00017'060270            NIOC    70
00020'000014'           .CALL
00021'177777            RETURN
      000001            .C1:    .BLK    1
      000023'           NEXT=.

00023'000071'           NEXT1
00024'177777            -1
00025'177777            -1
00026'177777            -1
00027'177777            -1
00030'177777            -1
00031'144716            .TXTF   "IN
00032'144724  IT
00033'000000  "
00034'030420            LDA     2,.SAMP
00035'000020' LP:       .CALL
00036'000015'           GET
00037'045000            STA     1,0,2
00040'151400            INC     2,2
00041'174014            COM#    3,3,SZR
00042'000773            JMP     LP
00043'102400            SUB     0,0
00044'024411            LDA     1,C10
00045'034472            LDA     3,.RES
00046'041400  LP5:      STA     0,0,3
00047'175400            INC     3,3
00050'125404            INC     1,1,SZR
00051'000775            JMP     LP5
00052'000035'           .CALL
00053'000021'           RETURN

00054'000056' .SAMP:    SAMP
00055'177766  C10:      -10.
      000013  SAMP:     .BLK    11.
      000071'           NEXT1=.

00071'000153'           NEXT2
00072'177777            -1
00073'177777            -1
00074'177777            -1
```

```
---
 00075'177777            -1
 00076'177777            -1
 00077'150325            .TXTF    "PU
 00100'146323  LS
 00101'142400  E"
 00102'030720            LDA      2, .C1
 00103'102400  LOOP:     SUB      0, 0
 00104'064770            DIAP     1, 70
 00105'147404            AND      2, 1, SZR
 00106'000403            JMP      LP2
 00107'101400            INC      0, 0
 00110'000774            JMP      LOOP+1

 00111'034743  LP2:      LDA      3, .SAMP
 00112'175400            INC      3, 3
 00113'025400            LDA      1, 0, 3
 00114'122032            ADC#     1, 0, SZC ;IN RANGE?
 00115'000775            JMP      LP2+1     ; NO
 00116'024422            LDA      1, DIFF
 00117'137000            ADD      1, 3
 00120'025400            LDA      1, 0, 3
 00121'125400            INC      1, 1
 00122'045400            STA      1, 0, 3
 00123'014733            DSZ      SAMP
 00124'000757            JMP      LOOP

 00125'030412            LDA      2, .RES
 00126'102400  RESL:     SUB      0, 0
 00127'025000            LDA      1, 0, 2
 00130'000052'           .CALL
 00131'177777            PUT
 00132'151400            INC      2, 2
 00133'174014            COM#     3, 3, SZR
 00134'000772            JMP      RESL
 00135'000130'           .CALL
 00136'000053'           RETURN

 00137'000141' .RES:     RES
 00140'000062  DIFF:     .RES-.SAMP-1
       000012  RES:      .BLK     10.

       000153'           NEXT2=.
                         .END
```

APPENDIX 9

PROGRAM FOR AUTO AND CROSS CORRELATION TESTS

```
5  DIM E[100]
10  PRINT "NO. OF CODE REQUIRED = ";
20  INPUT K
30  PRINT
31  LET A2=10000
40  PRINT "SUMARY (0) OR FULL TABLE (1) ";
50  INPUT A1
60  PRINT
70  PRINT
80  PRINT
130  PRINT
131  FOR P=200 TO 1300 STEP 200
132    FOR P1=P TO 1300 STEP 200
155      LET P=P/10800
156      LET P3=P1/10800
160      LET P1=P*4095
161      LET P4=P3*4095
162      GOSUB  410
163      FOR B1= 0 TO 15
164        IF B1<>5 GOTO  166
165        GOSUB  581
166        FOR B3=B1+1 TO 15
170          LET B9= INT (2↑B1+.5)
180          LET B2= INT (2↑12*B1+.5)
190          LET P9=P1+B2
195          LET P8=P4+ INT (2↑12*P3+.5)
196          LET B8= INT (2↑B3+.5)
200          CALL "INIT",P9,I9,P8,B8,S
210          LET S2= 0
220          LET C9= 0
222          LET L= 0
223          LET H= 0
230          LET S1= 0
233          LET C9= 0
234          LET P1=P
235          IF B1<>B3 GOTO  240
236          LET P1=P+1
240          FOR I=P1-P TO P1-1
250            CALL "CORR",I,C
251            IF C>K GOTO  253
252            LET L=L+1
253            IF C<>0 GOTO  270
260            LET H=H+1
270            LET S1=S1+C
280            LET E[I]=C
290            IF C<=C9 GOTO  310
300            LET C9=C
310            IF A1= 0 GOTO  330
320            PRINT C,A
330          NEXT I
331          LET S1=S1/P
332          FOR I=P1-P TO P1-1
333            LET S2=S2+(E[I]-S1)*(E[I]-S1)
334          NEXT I
340          LET S2= SQR (S2/(P-1))
360          PRINT B1;"-";B3; TAB 13;S1; TAB 23;S2; TAB 33;L; TAB 50;H
```

```
;
361           PRINT  TAB 63; C9
374         NEXT B3
[illegible]   NEXT B1
[illegible]  NEXT F1
[illegible] NEXT F
[illegible] END
410 LET D=.0 [illegible]
420 LET P5=P*P3
430 LET Q=1-P5
440 LET P6= LOG P5
450 LET Q1= LOG Q
460 LET C5=A2*Q1
466 LET D=D+ EXP.(C5)
467 LET K= 0
468 IF D>=.1 GOTO  530
470 FOR K=1 TO 10000
480   LET C5=C5+ LOG (A2-K+1)- LOG K+P6-Q1
500   LET D=D+ EXP (C5)
510   IF D>=.05 GOTO  530
520 NEXT K
530 LET D1=D
540 FOR J=K+1 TO 10000
550   LET C5=C5+ LOG (A2-J+1)- LOG (J)+P6-Q1
560   LET D1=D1+ EXP (C5)
570   IF D1>=.9 GOTO  581
580 NEXT J
581 PRINT
582 PRINT
583 PRINT
586 PRINT "FLOW RATES =";F;"AND";F1
587 PRINT "**************************"
588 PRINT
590 PRINT "LOWER LIMIT =";K,"UPPER LIMIT =";J
600 LET A=P5*10000
610 PRINT "EXPECTED MEAN =";A,"EXPECTED STD. DEV. =";  SQR (A*Q)
614 PRINT
615 PRINT  TAB 2;"BITS";  TAB 14;"MEAN";  TAB 23;"STD. DEV. ";
616 PRINT  TAB 35;"LOWER 10%";  TAB 47;"UPPER 10%";  TAB 58;
617 PRINT "LARGEST COEF"
619 PRINT
620 RETURN
630 END
```

```
                                .TITLE  AUTO
        000001                  .TXTM   1
                                .EXTN   .CALL, GET, RETURN, PUT
                                .NREL
00000'000101'                   NEXT
00001'177777                    -1
00002'177777                    -1
00003'177777                    -1
00004'177777                    -1
00005'177777                    -1
00006'144716                    .TXTF   "IN
00007'144724    IT
00010'000000    "
00011'177777                    .CALL
00012'177777                    GET
00013'065070                    DOA     1,70
00014'000011'                   .CALL
00015'000012'                   GET
00016'044453                    STA     1,M1
00017'000014'                   .CALL
00020'000015'                   GET
00021'065070                    DOA     1,70
00022'000017'                   .CALL
00023'000020'                   GET
00024'044446                    STA     1,M2
00025'000022'                   .CALL
00026'000023'                   GET
00027'044447                    STA     1,SHIFT
00030'102400                    SUB     0,0
00031'004424                    JSR     SHIF


                                                ;  READ IN SEQUENCE
00032'020435                    LDA     0,C625
00033'040440                    STA     0,LOOPC ;  LOOP COUNTER
00034'034444                    LDA     3,.TABLE ;  START OF TABLE
00035'030434                    LDA     2,M1
00036'024432    LOOP:           LDA     1,C16
00037'044435                    STA     1,P1C ;  LOOP1 COUNTER
00040'126420                    SUBZ    1,1
00041'060770    LOOP1:          DIAP    0,70
00042'125120                    MOVZL   1,1     ;  SHIFT WORD
00043'143404                    AND     2,0,SZR
00044'125400                    INC     1,1
00045'014427                    DSZ     P1C
00046'000773                    JMP     LOOP1
00047'045400                    STA     1,0,3   ;  STORE 16 BITS OF SEQUENCE
00050'175400                    INC     3,3     ;  INCREMENT ADDRESS POINTER
00051'014422                    DSZ     LOOPC
00052'000764                    JMP     LOOP
00053'000025'                   .CALL
00054'177777                    RETURN

00055'060270    SHIF:           NIOC    70
00056'024420                    LDA     1,SHIFT
00057'107005                    ADD     0,1,SNR
00060'001400                    JMP     0,3
00061'044416                    STA     1,.SHIFT
00062'060370    SH:             NIOP    70
00063'014414                    DSZ     .SHIFT
```

```
00064'000776            JMP     SH
00065'001400            JMP     0,3

00066'154360  C10000:   -10000.
00067'001161  C625:     625.
00070'000020  C16:      20
      000001  M1:       .BLK    1
      000001  M2:       .BLK    1
      000001  LOOPC:    .BLK    1
      000001  P1C:      .BLK    1
      000001  AVE:      .BLK    1
      000001  SHIFT:    .BLK    1
      000001  .SHIFT:   .BLK    1
00100'000154' .TABLE:   TABLE

      000101'           NEXT=.

00101'001335'           NEXT1
00102'177777            -1
00103'177777            -1
00104'177777            -1
00105'177777            -1
00106'177777            -1
00107'141717            .TXTF   "CO
00110'151322  RR
00111'000000  "
00112'000053'           .CALL
00113'000026'           GET
00114'121000            MOV     1,0
00115'004740            JSR     SHIF

00116'102400            SUB     0,0     ; COMPUTE       S(AN)*(AN+T)
00117'040432            STA     0,COEF
00120'024752            LDA     1,M2
00121'030746            LDA     2,C625
00122'050430            STA     2,LP1C
00123'034755            LDA     3,.TABLE
00124'030744  LP1:      LDA     2,C16
00125'050426            STA     2,LP2C
00126'031400            LDA     2,0,3
00127'060770  LP2:      DIAP    0,70
00130'123404            AND     1,0,SZR
00131'151133            MOVZL#  2,2,SNC
00132'000402            JMP     .+2
00133'010416            ISZ     COEF
00134'151120            MOVZL   2,2
00135'014416            DSZ     LP2C
00136'000771            JMP     LP2
00137'175400            INC     3,3
00140'014412            DSZ     LP1C
00141'000763            JMP     LP1
00142'102400            SUB     0,0
00143'024406            LDA     1,COEF
00144'000112'           .CALL
00145'177777            PUT
00146'000144'           .CALL
00147'000054'           RETURN


      000001  COEFN:    .BLK    1
```

```
000001  COEF:   .BLK    1
000001  LP1C:   .BLK    1
000001  LP2C:   .BLK    1
001161  TAELE:  .BLK    625.

001335'         NEXT1=.
                .END
```

APPENDIX 10

RESULTS OF AUTOCORRELATION TEST ON THE POISSONION DISTRIBUTION RANDOM NUMBER GENERATOR

FLOW RATE = 200
****

LOWER LIMIT = 1 UPPER LIMIT = 7
EXPECTED MEAN = 3.42935 EXPECTED STD. DEV. = 1.85153

BIT	MEAN	STD. DEV.	LOWER 10%	UPPER 10%	LARGEST COEF
0	3.63	1.80714	10	5	8
1	2.25	1.53987	35	2	8
2	3.08	1.78478	16	4	8
3	2.52	1.64826	28	0	6
4	3.03	1.73178	19	4	8
5	3.54	1.85548	10	6	10
6	3.71	1.84388	12	7	9
7	3.47	1.76643	11	7	8
8	3.41	1.84278	13	4	9
9	3.56	1.7075	11	5	7
10	3.35	1.93519	18	5	9
11	3.74	1.72691	7	6	10
12	3.74	1.81252	8	7	9
13	3.57	1.72477	12	5	8
14	4.32	2.1456	10	17	9
15	4.06	1.81364	7	8	8

FLOW RATE = 300
****

LOWER LIMIT = 4 UPPER LIMIT = 13
EXPECTED MEAN = 7.71605 EXPECTED STD. DEV. = 2.77671

BIT	MEAN	STD. DEV.	LOWER 10%	UPPER 10%	LARGEST COEF
0	7.91	2.71581	12	4	14
1	5.6	2.42462	35	0	11
2	7.66	2.62935	13	4	15
3	7.47	2.38028	11	1	13
4	8.07	2.92068	9	7	13
5	7.46	2.43468	14	1	13
6	7.67	2.34005	7	3	13
7	7.88	2.69034	8	5	15
8	8.42	3.02903	10	11	16
9	8.04	2.56203	4	7	14
10	8.07	2.60634	9	4	17
11	8.39	2.7262	4	9	17
12	9.2	2.97803	3	12	17
13	7.8	3.05504	13	7	15
14	8.68	2.66241	7	7	17
15	8.53	2.50436	5	6	15

FLOW RATE = 400
****

LOWER LIMIT = 8 UPPER LIMIT = 20
EXPECTED MEAN = 13.7174 EXPECTED STD. DEV. = 3.70116

BIT	MEAN	STD. DEV.	LOWER 10%	UPPER 10%	LARGEST COEF
0	14.06	3.20926	3	6	21
1	11.29	3.44742	21	0	19
2	12.17	3.4817	13	2	22
3	12.48	3.20819	8	1	22
4	13.68	3.5416	5	4	25
5	13.85	3.33446	7	5	22
6	14.82	3.63867	3	11	27
7	15.51	3.37995	1	12	23
8	14.99	3.63622	4	12	23
9	15.49	3.42449	1	11	23
10	14.4	3.33636	1	7	24
11	13.65	3.37661	3	4	24
12	16.03	3.81504	2	19	26
13	13.81	3.82097	10	7	22
14	14.72	3.70933	4	9	25
15	15.95	3.68329	2	14	26

FLOW RATE = 500
****

LOWER LIMIT = 14 UPPER LIMIT = 29
EXPECTED MEAN = 21.4335 EXPECTED STD. DEV. = 4.62466

BIT	MEAN	STD. DEV.	LOWER 10%	UPPER 10%	LARGEST COEF
0	21.02	3.9363	5	2	30
1	17.79	4.44334	26	0	28
2	18.63	4.40993	19	2	32
3	19.01	3.74563	14	1	29
4	22.92	4.45556	3	7	35
5	22.4	4.06134	1	7	35
6	22.43	4.45006	3	7	39
7	22.14	4.13904	3	9	32
8	22.85	4.15933	2	8	35
9	25.23	4.28021	0	25	33
10	22.6	4.04763	1	7	34
11	21.54	3.57437	3	1	29
12	23.45	4.69121	1	14	36
13	21.33	4.96259	5	9	33
14	22.92	4.76662	6	13	35
15	25.4	4.52373	0	27	37

FLOW RATE = 600
****

LOWER LIMIT = 22 UPPER LIMIT = 40
EXPECTED MEAN = 30.8642 EXPECTED STD. DEV. = 5.54697

BIT	MEAN	STD. DEV.	LOWER 10%	UPPER 10%	LARGEST COEF
0	30.14	4.81793	5	1	40
1	25.94	5.02482	23	0	37
2	25.84	4.79839	24	1	41
3	28.59	4.70351	11	1	40
4	32.63	4.88587	1	9	44
5	32.42	5.1053	3	7	43
6	32.61	5.15005	3	9	45
7	33.17	5.31255	3	11	46
8	32.25	5.09382	3	10	44
9	36.02	5.02313	0	28	46
10	32.53	5.12559	1	10	44
11	30.42	4.5664	5	2	42
12	34.73	5.28357	0	18	48
13	29.29	5.21127	7	3	44
14	33.44	5.26147	2	11	47
15	34.29	5.1704	0	15	48

FLOW RATE = 700
****

LOWER LIMIT = 32 UPPER LIMIT = 53
EXPECTED MEAN = 42.0096 EXPECTED STD. DEV. = 6.46785

BIT	MEAN	STD. DEV.	LOWER 10%	UPPER 10%	LARGEST COEF
0	42.11	5.64951	5	3	53
1	36.96	6.05343	21	0	52
2	36.71	5.42774	22	0	51
3	40.31	5.96461	9	4	54
4	43.82	5.5055	1	7	53
5	42.54	5.96443	6	6	59
6	45.01	6.25144	2	12	63
7	43.61	6.04003	3	8	56
8	43.02	5.7717	3	6	55
9	45.92	5.3761	0	9	58
10	42.63	6.18347	5	7	57
11	43.21	6.16816	3	9	60
12	47.43	5.83434	0	18	62
13	41.03	6.04139	6	3	56
14	47.37	6.5.472	2	20	65
15	45.7	5.98735	0	13	58

FLOW RATE = 800
****

LOWER LIMIT = 43 UPPER LIMIT = 66
EXPECTED MEAN = 54.8697 EXPECTED STD. DEV. = 7.33705

BIT	MEAN	STD. DEV.	LOWER 10%	UPPER 10%	LARGEST COEF
0	54.25	6.0409	4	1	70
1	49.74	6.53292	21	2	70
2	47.88	6.12541	23	1	72
3	53.67	6.27492	5	3	69
4	55.86	6.41181	2	7	72
5	55.32	6.78809	6	6	71
6	59.64	6.95804	1	18	83
7	54.14	6.37548	5	4	68
8	52.83	6.46365	6	2	68
9	60.17	6.52586	1	18	76
10	55.05	6.67328	5	8	69
11	57.17	7.25027	3	12	75
12	59.17	6.79876	0	19	75
13	52.85	7.35447	11	4	68
14	63.45	7.0099	1	39	82
15	60.96	6.59187	0	25	75

FLOW RATE = 900
****

LOWER LIMIT = 56 UPPER LIMIT = 81
EXPECTED MEAN = 69.4444 EXPECTED STD. DEV. = 8.30434

BIT	MEAN	STD. DEV.	LOWER 10%	UPPER 10%	LARGEST COEF
0	69.34	7.36523	4	7	88
1	62	6.9282	25	0	80
2	61.89	7.04657	21	1	85
3	69.18	6.92874	3	5	85
4	74.8	7.81476	1	27	93
5	67.02	7.07383	9	2	84
6	74.37	7.58605	0	22	100
7	73.01	7.39162	2	13	92
8	69.12	7.28256	2	5	89
9	75.55	7.75425	1	22	97
10	71.39	7.24443	2	14	86
11	69.08	7.27895	5	4	87
12	72.8	7.6673	2	16	94
13	66.78	8.19776	10	1	81
14	80.47	7.64284	1	53	99
15	77.86	6.94569	0	34	92

FLOW RATE = 1000
****

LOWER LIMIT = 71 UPPER LIMIT = 100
EXPECTED MEAN = 85.7338 EXPECTED STD. DEV. = 9.21948

BIT	MEAN	STD. DEV.	LOWER 10%	UPPER 10%	LARGEST COEF
0	84.49	8.27614	6	2	102
1	78.13	7.88508	18	1	103
2	76.25	7.91415	25	0	98
3	84.63	7.7233	7	4	102
4	90.32	8.27417	1	19	106
5	85.63	8.73127	5	8	105
6	91.14	8.37505	0	14	122
7	89.52	7.81668	1	11	109
8	89.21	7.94588	0	9	110
9	91.88	8.24163	1	17	113
10	87.72	7.83165	0	6	108
11	82.57	7.8897	8	2	105
12	90.06	7.92236	1	12	109
13	83.88	9.20063	9	4	105
14	97.61	9.17603	1	46	122
15	99.44	8.27471	0	46	119

FLOW RATE = 1100
****

LOWER LIMIT = 87 UPPER LIMIT = 118
EXPECTED MEAN = 103.738 EXPECTED STD. DEV. = 10.1322

BIT	MEAN	STD. DEV.	LOWER 10%	UPPER 10%	LARGEST COEF
0	103.26	8.45141	4	3	124
1	93.56	8.77012	27	0	115
2	96.71	8.55178	14	1	120
3	99.55	8.19253	10	2	120
4	107.92	9.45554	1	16	129
5	105.41	9.48585	4	14	127
6	107	9.16625	2	10	137
7	108.03	9.05029	1	18	130
8	105.29	8.71001	3	7	128
9	112.08	10.2412	0	24	138
10	103.41	8.36332	4	5	128
11	102.46	9.02119	5	7	127
12	107.22	7.97162	1	10	125
13	101.31	9.99259	10	5	126
14	114.08	9.5776	1	38	137
15	114.95	8.95145	0	38	134

FLOW RATE = 1200
****

LOWER LIMIT = 106 UPPER LIMIT = 141
EXPECTED MEAN = 123.457 EXPECTED STD. DEV. = 11.0423

BIT	MEAN	STD. DEV.	LOWER 10%	UPPER 10%	LARGEST COEF
0	128.82	9.55934	1	4	145
1	114.05	10.0407	23	1	149
2	115.09	8.75604	17	0	136
3	120.88	9.28261	7	3	145
4	129.91	10.697	0	16	154
5	122.54	10.0115	6	3	143
6	125.89	9.66183	1	6	153
7	123.86	9.8504	3	4	151
8	122.46	9.01111	4	2	145
9	131.23	10.8347	1	16	158
10	123.91	10.1971	4	6	152
11	119.2	9.16403	8	1	142
12	128.09	9.85547	0	9	148
13	119.44	10.0366	11	2	144
14	133.31	10.3998	1	24	161
15	134.18	9.82818	0	26	155

FLOW RATE = 1300
****

LOWER LIMIT = 126 UPPER LIMIT = 163
EXPECTED MEAN = 144.89 EXPECTED STD. DEV. = 11.9495

BIT	MEAN	STD. DEV.	LOWER 10%	UPPER 10%	LARGEST COEF
0	145.74	10.5415	3	6	173
1	137.83	11.6541	14	4	176
2	134.9	10.1633	21	0	158
3	142.92	9.95409	5	5	169
4	150.59	11.9562	2	16	178
5	144.8	11.2681	6	6	172
6	142.6	9.33404	2	1	167
7	140.92	10.253	9	2	167
8	143.42	9.72673	1	1	168
9	156.47	11.4525	0	25	181
10	140.53	10.5116	8	3	170
11	141.11	9.94213	7	1	163
12	145.02	9.53564	2	3	167
13	139.58	11.1034	15	1	167
14	154.32	10.7425	1	21	183
15	156.17	9.98992	0	25	178

APPENDIX 11

SAMPLE RESULTS OF CROSSCORRELATION TESTS ON THE POISSONION DISTRIBUTION RANDOM NUMBER GENERATOR

FLOW RATES = 200 AND 200

LOWER LIMIT = 1 UPPER LIMIT = 7
EXPECTED MEAN = 3.42935 EXPECTED STD. DEV. = 1.35158

BITS	MEAN	STD. DEV.	LOWER 10%	UPPER 10%	LARGEST COEF
0 - 1	3.13	1.31677	20	4	9
0 - 2	3.71	1.21623	10	7	9
0 - 3	2.93	1.7767	21	4	8
0 - 4	3.66	1.7269	6	4	8
0 - 5	3.74	1.6132	8	3	8
0 - 6	3.63	1.66153	9	7	9
0 - 7	3.45	2.[illegible]359	21	3	10
0 - 8	3.5	1.39396	14	11	8
0 - 9	3.52	1.76035	10	7	8
0 - 10	3.63	1.71587	11	6	9
0 - 11	3.22	1.66103	13	4	9
0 - 12	3.72	1.93336	12	8	10
0 - 13	3.81	1.65569	8	6	8
0 - 14	3.73	1.95343	12	9	10
0 - 15	3.92	1.75625	12	7	10
1 - 2	2.95	1.74295	23	3	9
1 - 3	2.53	1.49379	24	1	7
1 - 4	2.39	1.75749	22	5	8
1 - 5	2.85	1.67196	22	3	8
1 - 6	3.01	1.76093	19	4	9
1 - 7	2.9	1.73205	22	5	8
1 - 8	2.74	1.73857	23	4	9
1 - 9	3.14	1.35572	21	3	9
1 - 10	2.72	1.62167	25	1	8
1 - 11	2.6	1.34033	21	0	6
1 - 12	3.1	1.76764	15	3	8
1 - 13	3.01	1.76093	21	5	7
1 - 14	3.39	1.69965	15	5	8
1 - 15	3.4	1.[illegible]319	11	3	11
2 - 3	3.01	1.59357	17	2	7
2 - 4	3.45	1.93609	14	3	10
2 - 5	3.56	2.[illegible]779	19	8	10
2 - 6	3.84	2.[illegible]5[illegible]	15	11	9
2 - 7	3.39	1.76893	15	3	8
2 - 8	3.34	1.84346	11	7	9
2 - 9	3.56	1.71289	13	5	8
2 - 10	3.18	1.73976	13	6	10
2 - 11	3.53	1.41746	5	2	8
2 - 12	3.44	1.74341	14	3	9
2 - 13	3.49	1.[illegible]5[illegible]44	15	6	9
2 - 14	3.9	1.[illegible]7756	10	10	8
2 - 15	4	1.83167	5	8	9
3 - 4	3.07	1.[illegible]5715	17	4	8
3 - 5	3.1	1.[illegible]545	18	4	9
3 - 6	3.4	1.83126	17	5	8
3 - 7	2.76	1.57711	25	2	8
3 - 8	2.95	1.73132	20	4	8
3 - 9	3.21	1.92254	14	7	8
3 - 10	2.67	1.71134	23	2	7

3 - 11	3.15	1.833333	17	3	3
3 - 12	3.03	1.833366	16	6	6
3 - 13	3.49	1.93953	15	7	9
3 - 14	3.13	1.61044	16	2	7
3 - 15	3.12	1.59033	14	4	8
4 - 5	3.39	1.65691	14	4	8
4 - 6	3.38	1.81307	13	6	8
4 - 7	3.35	1.84438	16	7	8
4 - 8	3.56	1.81085	11	5	8
4 - 9	3.33	1.83203	16	4	8
4 - 10	3.37	1.93664	14	7	10
4 - 11	3.8	1.64884	13	7	7
4 - 12	3.53	1.89192	12	5	10
4 - 13	3.44	1.74863	14	7	8
4 - 14	3.69	1.90531	11	7	9
4 - 15	3.24	1.83465	8	8	10

FLOW RATES = 200 AND 250

LOWER LIMIT = 1 UPPER LIMIT = 7
EXPECTED MEAN = 3.48935 EXPECTED STD. DEV. = 1.35153

BITS	MEAN	STD. DEV.	LOWER 10%	UPPER 10%	LARGEST COEF
5 - 6	4.03	1.99933	3	10	10
5 - 7	3.5	1.94185	14	6	9
5 - 8	3.62	1.93803	13	[illegible]	9
5 - 9	3.61	1.63112	10	6	7
5 - 10	3.7	1.35592	11	10	8
5 - 11	3.67	1.67666	9	7	7
5 - 12	3.7	1.91485	12	10	8
5 - 13	3.5	1.92583	13	9	8
5 - 14	3.64	1.35057	10	8	9
5 - 15	3.94	1.8635	9	8	9
6 - 7	3.44	1.76566	10	7	8
6 - 8	3.91	1.38176	7	10	9
6 - 9	3.59	1.77579	10	6	9
6 - 10	3.37	1.96255	17	3	13
6 - 11	3.86	2.10343	11	10	10
6 - 12	3.8	1.30347	14	7	9
6 - 13	3.41	1.36999	13	5	8
6 - 14	4.23	1.96474	7	14	10
6 - 15	4.14	1.77536	6	11	10
7 - 8	3.32	1.693	10	4	9
7 - 9	3.81	1.33933	11	8	9
7 - 10	3.21	1.64713	15	2	8
7 - 11	3.13	1.58514	13	2	8
7 - 12	3.13	1.76135	20	3	8
7 - 13	3.15	1.33333	20	5	8
7 - 14	3.69	1.33393	12	9	9
7 - 15	3.93	1.96049	9	8	10
8 - 9	3.6	1.90698	11	6	10
8 - 10	3.28	1.63349	13	3	7
8 - 11	3.9	1.93061	9	8	10
8 - 12	3.65	1.63998	6	8	8
8 - 13	3.45	1.97133	15	6	10
8 - 14	3.83	1.33591	7	8	12
8 - 15	3.75	2.09557	15	10	10
9 - 10	3.65	1.73305	11	6	9
9 - 11	3.56	1.74342	9	6	8
9 - 12	3.55	1.31116	9	7	8
9 - 13	3.47	1.97691	13	9	8
9 - 14	3.79	1.30512	13	6	9
9 - 15	4.05	1.97133	7	14	9
10 - 11	3.22	1.76142	20	3	8
10 - 12	3.69	1.9053	11	8	9
10 - 13	3.51	1.3135	16	5	9
10 - 14	3.47	1.91933	13	6	10
10 - 15	4.01	2.07673	9	9	10
11 - 12	3.43	1.99946	16	6	9
11 - 13	3.43	1.73193	13	8	10
11 - 14	3.35	1.9464	10	7	10
11 - 15	3.78	1.39678	10	9	9
12 - 13	3.55	2.11476	13	3	11
12 - 14	3.96	2.03467	8	15	9
12 - 15	4.15	1.96685	10	11	10
13 - 14	3.83	2.02249	14	10	9
13 - 15	3.56	1.33358	9	7	9
14 - 15	4.04	2.02409	10	15	11

FLOW RATES = 400 AND 800

LOWER LIMIT = 19 UPPER LIMIT = 35
EXPECTED MEAN = 27.4343 EXPECTED STD. DEV. = 5.23064

BITS	MEAN	STD. DEV.	LOWER 10%	UPPER 10%	LARGEST COEF
0 - 1	26.71	4.53553	4	3	43
0 - 2	26.4	4.82836	7	4	38
0 - 3	27	4.17544	2	4	42
0 - 4	28.63	5.45254	6	16	41
0 - 5	23.61	4.69379	2	11	39
0 - 6	29.19	5.11083	1	13	45
0 - 7	27.59	5.15633	2	13	42
0 - 8	27.6	5.43091	9	10	40
0 - 9	29.63	5.26212	5	15	45
0 - 10	29.02	4.41296	2	9	44
0 - 11	28.78	5.25044	3	14	41
0 - 12	29.34	5.23044	3	14	43
0 - 13	27.36	4.42106	1	6	40
0 - 14	30.53	4.70799	6	21	40
0 - 15	29.49	4.92856	1	15	42
1 - 2	24.07	4.37639	17	2	37
1 - 3	24.34	4.1333	9	1	38
1 - 4	25.47	4.64373	11	3	39
1 - 5	25.04	4.61815	11	1	40
1 - 6	26.07	4.93183	8	5	43
1 - 7	24.79	4.74844	11	4	38
1 - 8	24.53	4.36053	11	3	39
1 - 9	25.44	5.1543	13	2	39
1 - 10	25.13	4.39331	12	2	39
1 - 11	25.29	4.34776	12	3	37
1 - 12	26.04	4.33435	6	4	40
1 - 13	23.95	4.71411	18	3	36
1 - 14	27.79	5.21321	2	11	43
1 - 15	26.83	5.15624	7	9	38
2 - 3	25.73	4.73392	13	3	37
2 - 4	26.49	4.49577	4	5	38
2 - 5	26.8	5.38194	5	5	46
2 - 6	27.46	5.4743	3	11	44
2 - 7	25.76	4.70344	11	3	37
2 - 8	25.71	5.05733	12	4	39
2 - 9	27.51	4.92671	6	9	42
2 - 10	26.53	5.16627	6	7	40
2 - 11	26.33	4.96259	6	7	41
2 - 12	26.55	5.28691	7	6	44
2 - 13	25.33	4.69203	11	3	37
2 - 14	27.42	4.30777	4	9	41
2 - 15	23.11	5.01692	3	13	40
3 - 4	26.52	5.1133	7	3	42
3 - 5	26.31	4.35163	11	5	40
3 - 6	27.1	5.27382	6	7	40
3 - 7	25.92	4.43697	7	2	36
3 - 8	25.53	4.77253	10	4	36
3 - 9	27.35	4.91914	3	9	39
3 - 10	25.33	5.16151	12	5	33

3 - 11	26.72	5.30639	8	7	42
3 - 12	26.72	5.07335	11	3	38
3 - 13	25.53	4.14913	4	1	36
3 - 14	27.94	4.25469	2	9	37
3 - 15	26.83	5.16603	5	8	39
4 - 5	28.15	4.63906	4	8	46
4 - 6	28.69	4.28455	8	10	42
4 - 7	28.31	4.82772	5	11	39
4 - 8	28.13	4.95566	6	9	41
4 - 9	30.17	5.00813	1	19	42
4 - 10	28.66	4.39241	2	11	39
4 - 11	28.99	5.61331	4	14	45
4 - 12	29.85	6.00073	4	20	41
4 - 13	27.56	4.74337	5	9	33
4 - 14	29.39	4.30717	2	15	45
4 - 15	29.06	4.76056	1	14	41

FLOW RATES = 400 AND 360

LOWER LIMIT = 19 UPPER LIMIT = 35
EXPECTED MEAN = 27.4343 EXPECTED STD. DEV. = 5.23064

BITS	MEAN	STD. DEV.	LOWER 10%	UPPER 10%	LARGEST CODE
5 - 6	29.16	4.66321	6	13	45
5 - 7	27.54	5.03666	5	3	44
5 - 8	26.33	5.23156	10	5	44
5 - 9	29.12	5.56735	4	16	42
5 - 10	27.37	5.46734	8	10	40
5 - 11	23.7	4.66774	3	11	41
5 - 12	23.44	4.93749	4	13	42
5 - 13	26.99	5.2363	11	5	41
5 - 14	23.32	5.42413	3	20	39
5 - 15	29.53	4.99334	3	19	39
6 - 7	29.05	5.20755	3	12	43
6 - 8	28.63	4.63158	2	12	40
6 - 9	29.65	4.70107	0	13	43
6 - 10	28.64	5.2932	4	16	41
6 - 11	27.94	4.44453	8	11	38
6 - 12	29.7	4.72794	0	17	43
6 - 13	27.18	4.4775	1	5	39
6 - 14	30.36	4.79159	0	24	45
6 - 15	29.3	4.73317	1	15	45
7 - 8	28.17	5.00111	4	12	41
7 - 9	29.59	4.80549	1	16	41
7 - 10	27.92	4.32973	2	11	40
7 - 11	28.19	4.39166	1	7	40
7 - 12	28.94	5.07034	1	14	44
7 - 13	27.28	5.69253	8	10	40
7 - 14	30.72	5.11499	2	20	46
7 - 15	29.3	4.83141	1	17	44
8 - 9	31.39	5.37333	0	24	44
8 - 10	28.92	5.18012	5	13	40
8 - 11	30.73	4.74517	1	24	42
8 - 12	29.14	5.14441	1	18	44
8 - 13	28.03	4.65393	3	7	33
8 - 14	31.17	5.16994	0	25	44
8 - 15	30.03	5.09999	1	22	47
9 - 10	29.53	4.76132	2	15	42
9 - 11	30.37	5.42655	2	21	46
9 - 12	30.32	5.7693	2	23	44
9 - 13	28.95	4.83735	4	11	44
9 - 14	31.2	5.30433	1	31	43
9 - 15	31.35	5.49631	1	29	52
10 - 11	29.46	5.11961	2	13	40
10 - 12	29.33	4.35767	2	14	43
10 - 13	27.66	5.16167	5	11	43
10 - 14	29.66	4.33703	1	16	41
10 - 15	31.13	5.64407	2	21	45
11 - 12	27.99	4.66939	2	11	40
11 - 13	27.62	5.4919	5	9	47
11 - 14	28.67	4.99376	4	13	39
11 - 15	28.56	4.87304	1	9	33
12 - 13	29.77	5.67797	2	17	45
12 - 14	31.57	4.99747	1	31	42
12 - 15	31.19	5.13581	1	28	47
13 - 14	29.11	5.30319	4	13	45
13 - 15	28.41	5.02743	1	12	45
14 - 15	29.74	5.40734	4	22	43

FLOW RATES = 1000 AND 1200
****

LOWER LIMIT = 87 UPPER LIMIT = 117
EXPECTED MEAN = 102.881 EXPECTED STD. DEV. = 10.0907

BITS	MEAN	STD. DEV.	LOWER 10%	UPPER 10%	LARGEST COEF
0 - 1	97.8	8.1265	12	1	119
0 - 2	99.01	9.47136	8	6	129
0 - 3	99.41	7.36917	7	0	116
0 - 4	104.98	9.08208	2	8	128
0 - 5	102.22	8.46092	2	5	129
0 - 6	101	8.39071	5	3	127
0 - 7	100.7	9.41145	7	5	122
0 - 8	102.57	9.17533	3	9	121
0 - 9	104.73	8.99139	2	11	129
0 - 10	103.94	9.03412	4	10	129
0 - 11	98.83	8.93979	9	4	121
0 - 12	103.53	9.97285	7	10	128
0 - 13	98.88	7.82674	8	1	117
0 - 14	104.97	9.19864	1	14	130
0 - 15	107.04	8.7593	0	13	130
1 - 2	93.83	8.97025	24	0	116
1 - 3	97.33	8.55495	16	1	118
1 - 4	101.35	8.88235	71	4	126
1 - 5	97.62	8.87599	11	2	119
1 - 6	98.14	8.9194	11	0	116
1 - 7	96.11	9.05525	16	2	123
1 - 8	98.29	10.1277	15	3	128
1 - 9	99.54	8.94247	7	2	123
1 - 10	98.68	8.92715	9	1	122
1 - 11	95.72	9.07329	20	2	117
1 - 12	99.03	8.87587	7	3	132
1 - 13	94.17	9.63163	21	0	113
1 - 14	101.73	9.21662	4	4	126
1 - 15	101.58	10.6952	8	8	124
2 - 3	95.7	8.991	20	2	119
2 - 4	99.35	8.40558	5	3	124
2 - 5	97.38	8.59333	13	1	125
2 - 6	99.23	8.74384	6	2	124
2 - 7	96.69	9.02933	17	0	115
2 - 8	97.73	9.07743	9	2	119
2 - 9	101.15	9.11693	6	4	122
2 - 10	99.33	8.39795	10	1	120
2 - 11	95.43	8.71807	14	0	116
2 - 12	99.79	9.3163	9	5	123
2 - 13	96.37	9.2645	17	1	122
2 - 14	100.82	8.7973	7	2	123
2 - 15	101.34	8.99063	9	3	121
3 - 4	104.76	9.2399	2	13	125
3 - 5	102.21	7.93443	5	5	123
3 - 6	102.55	8.65191	2	6	126
3 - 7	101.5	9.00449	6	4	122
3 - 8	102.81	9.12232	4	7	129
3 - 9	107.2	8.63179	0	15	129
3 - 10	101.83	9.1883	3	14	125

3 - 11	100.4	9.71199	8	4	129
3 - 12	104.65	8.99984	1	11	130
3 - 13	99.34	9.31839	10	3	127
3 - 14	104.42	9.1034	4	9	126
3 - 15	105.53	10.0052	4	15	27
4 - 5	105.42	8.7596	0	11	132
4 - 6	106.66	8.79073	1	12	128
4 - 7	106.85	9.96696	4	19	130
4 - 8	107.25	8.75754	1	14	129
4 - 9	110.11	9.71605	0	25	137
4 - 10	106.42	8.53414	1	10	133
4 - 11	104.27	9.05409	3	5	122
4 - 12	109.03	9.64148	3	18	131
4 - 13	102.71	8.58502	4	5	122
4 - 14	110.23	8.61739	0	21	123
4 - 15	109.43	9.19646	1	22	136

FLOW RATES = 1000 AND 1200
****

LOWER LIMIT = 87 UPPER LIMIT = 117
EXPECTED MEAN = 102.881 EXPECTED STD. DEV. = 10.0907

BITS	MEAN	STD. DEV.	LOWER 10%	UPPER 10%	LARGEST COEF
5 - 6	105.06	9.08968	2	14	124
5 - 7	103.09	8.44241	3	8	121
5 - 8	102.74	9.95099	5	5	132
5 - 9	106.56	8.9067	1	9	136
5 - 10	105.11	10.0693	4	11	131
5 - 11	101.17	8.94092	8	2	124
5 - 12	105.56	9.93951	2	14	132
5 - 13	100.62	8.97557	9	3	120
5 - 14	106.66	9.10234	1	13	131
5 - 15	103.23	8.70254	1	15	134
6 - 7	106.68	10.1104	1	15	133
6 - 8	105.69	9.32324	1	14	125
6 - 9	108.26	8.68949	0	18	135
6 - 10	105.47	10.0919	3	16	138
6 - 11	103.56	8.97167	4	8	123
6 - 12	108.26	8.52164	1	15	129
6 - 13	102.51	8.63804	2	7	129
6 - 14	110.05	9.04015	0	23	134
6 - 15	110.53	8.66135	0	26	130
7 - 8	104.92	9.56283	4	8	132
7 - 9	109.34	9.10012	0	23	130
7 - 10	105.24	9.08111	2	11	127
7 - 11	102.48	9.21511	4	7	127
7 - 12	105.34	9.64962	2	16	123
7 - 13	102.74	9.70431	5	9	128
7 - 14	107.13	10.4172	4	17	131
7 - 15	109.48	9.11372	1	24	131
8 - 9	110.1	9.05258	0	24	132
8 - 10	106.59	9.18175	1	15	128
8 - 11	104.58	8.67613	1	12	125
8 - 12	106.17	8.46148	2	14	133
8 - 13	104.02	7.43724	1	6	118
8 - 14	110.1	9.28068	1	20	149
8 - 15	108.99	10.6254	1	23	133
9 - 10	108.68	9.57887	0	18	145
9 - 11	105.65	9.17105	3	14	127
9 - 12	107.83	10.3378	2	21	131
9 - 13	104.05	7.8924	4	7	126
9 - 14	110.39	10.1283	2	24	139
9 - 15	109.58	10.792	2	27	141
10 - 11	102.52	9.15353	6	4	129
10 - 12	105.69	9.25532	1	14	130
10 - 13	101.24	10.0344	7	6	133
10 - 14	106.34	8.32206	3	11	123
10 - 15	108.01	9.76334	0	16	130
11 - 12	102.22	9.42548	4	8	125
11 - 13	99.78	8.583	9	1	123
11 - 14	103.71	9.69043	6	8	139
11 - 15	104.33	8.56825	2	8	125
12 - 13	105.17	9.33378	2	9	128
12 - 14	108.82	9.13243	2	18	134
12 - 15	109.74	8.71502	0	21	130
13 - 14	103.11	8.68981	5	7	129
13 - 15	102.82	8.23685	4	5	118
14 - 15	112.03	9.36082	0	30	132

APPENDIX 12

TRAFFIC SIMULATION ROUTINE LISTINGS

```
---
PAGE   1
07/04/75
 1                              .TITL   LINK
 2                    ; TRAFFIC LINK OF UP TO 5 LANES
 3                    ; THE INPUT FROM ANOTHER PARAMETER IS
 4                    ; SPLIT UP AMONGST THE LANES IN A USER SPECIFIED PROPORTION
 5                    ; TRAVELLING  TIMES ARE ASSIGNED ACCORDING TO
 6                    ; A RANDOM DISTRIBUTION SPECIFIED BY THE USER
 7
 8                    ; PROGRAM: LINK,NO,INPUTPARM,NOC,LEG NO]
 9
10                    ; DATA:    LINK,NO,DIST,MEAN*100,STD DEV *100,
11                    ;          %FLOW RATE 1,[% F.R.2,.....% F.R.5]
12
13                           .EXTN   PUTP,GET,GETAD,GETBR,RETURN,MULTIPLY
14                           .EXTN   DIVIDE,ERROR
15                           .HEAD           *LINK 'ROUTINE*
16
17                           .ZREL
18  00000-000151' .LINK:     LINK
19
20                           .NREL
21        000001             .TXTM   1
22
23  00000'000012'            L1                      ; ADD OF VARIABLE ROUTINE
24  00001'000144             100.                    ; NO OF LINKS
25  00002'000010             8.                      ; CONSTANT STORAGE LOCS
26  00003'000144             100.                    ; NO OF LINKS
27  00004'000031             25.                     ; INITIAL STORAGE LOCS
28  00005'000372'            NEXTL
29  00006'046111             .TXT    *LINK*
30  00007'047113
31  00010'000000
32  00011'040416             STA     0,L2    ; VARIABLE ADD
33  00012'044416             STA     1,L3    ; INITIAL ADD
34  00013'177777             GETAD           ; INPUT PARM
35  00014'177777             GETBR           ; GET BREAK
36  00015'174015             COM#    3,3,SNR ; LAST PARM ?
37  00016'000405             JMP     .+5     ; YES
38  00017'124400             NEG     1,1     ; SUBTRACT 1
39  00020'124000             COM     1,1
40  00021'177777             GET             ; GET LEG NO
41  00022'107000             ADD     0,1
42  00023'044406             STA     1,L4    ; INPUT INITIAL ADD
43  00024'177777             PUTP            ; WRITE PROGRAM
44  00025'000011'            L4
45  00026'006000-            JSR     @.LINK
46  00027'000001  L2:        .BLK    1       ; CONST ADD
47  00030'000001  L3:        .BLK    1       ; INITIAL ADD
48  00031'000001  L4:        .BLK    1       ; INPUT ADD
```

```
---
PAGE   2                    *LINK ROUTINE*
07/04/75
 1                  ; CONSTANT ROUTINE
 2
 3  00032'040775  L1:      STA     0,L2     ; CONST ADD
 4  00033'000013'          GETAD            ; DIST ADD
 5  00034'034773           LDA     3,L2
 6  00035'051405           STA     2,5,3    ; DIST ADD
 7  00036'031100           LDA     2,100,2  ; MEAN OF DIST
 8  00037'050771           STA     2,L3
 9  00040'000021'          GET              ; MEAN *100
10  00041'105000           MOV     0,1      ; * 128 *3
11  00042'030504           LDA     2,C600
12  00043'102400           SUB     0,0
13  00044'177777           MULTIPLY
14  00045'030477           LDA     2,C100   ; /100
15  00046'177777           DIVIDE
16  00047'040762           STA     0,L4     ; TEMP STORE MEAN
17  00050'000040'          GET              ; STD DEV *100
18  00051'105000           MOV     0,1      ; *128 * 3
19  00052'030474           LDA     2,C600
20  00053'102400           SUB     0,0
21  00054'000044'          MULTIPLY
22  00055'030467           LDA     2,C100   ; /100
23  00056'000046'          DIVIDE
24  00057'034750           LDA     3,L2
25  00060'031405           LDA     2,5,3    ; ADD OF DIST
26  00061'031101           LDA     2,101,2  ; STD DEV OF DIST
27  00062'126400           SUB     1,1
28  00063'000056'          DIVIDE
29  00064'034743           LDA     3,L2
30  00065'041407           STA     0,7,3    ; STD DEV CONSTANT
31  00066'111000           MOV     0,2
32  00067'102400           SUB     0,0
33  00070'024740           LDA     1,L3
34  00071'000054'          MULTIPLY         ; MEAN*STD DEV
35  00072'024737           LDA     1,L4
36  00073'106400           SUB     0,1
37  00074'030733           LDA     2,L2
38  00075'045006           STA     1,6,2    ; MEAN CONSTANT
39  00076'126400           SUB     1,1
40  00077'044450           STA     1,TEMP   ; ZERO TEMP
41  00100'000050'  L11:    GET              ; % FLOW RATE
42  00101'101004           MOV     0,0,SZR  ; ZERO ?
43  00102'004424           JSR     FLOW     ; NO
44  00103'041000           STA     0,0,2    ; STORE CONSTANT
45  00104'125400           INC     1,1
46  00105'000014'          GETBR
47  00106'174014           COM#    3,3,SZR  ; END OF LIST ?
48  00107'000411           JMP     L12      ; NO
49  00110'176400           SUB     3,3      ; ZERO REMAINING LANES
50  00111'022437           LDA     0,@.C5
51  00112'106433           SUBZ#   0,1,SZC
52  00113'177777           RETURN           ; ALL PROCESSED
53  00114'055001           STA     3,1,2
54  00115'151400           INC     2,2
55  00116'125400           INC     1,1
56  00117'000773           JMP     .-5
57
58  00120'151400  L12:     INC     2,2
```

```
---
PAGE   3                    *LINK ROUTINE*
07/04/75
 1  00121'022427             LDA     0,@.C5
 2  00122'106433             SUBZ#   0,1,SNC ; MORE THAN 5 LANES ?
 3  00123'000755             JMP     L11     ; NO
 4  00124'177777             ERROR           ; ERROR WARNING
 5  00125'000024             20.
 6
 7  00126'044701  FLOW:      STA     1,L2    ; SAVE STATES
 8  00127'050701             STA     2,L3
 9  00130'054701             STA     3,L4
10  00131'024416             LDA     1,TEMP  ; ADD IN PREV TOTAL
11  00132'107000             ADD     0,1
12  00133'044414             STA     1,TEMP
13  00134'102400             SUB     0,0     ; MULTIPLY BY 2^12
14  00135'030410             LDA     2,C4096
15  00136'000071'            MULTIPLY
16  00137'030405             LDA     2,C100  ; DIVIDE BY 100
17  00140'000073'            DIVIDE
18  00141'024666             LDA     1,L2
19  00142'030666             LDA     2,L3
20  00143'002666             JMP     @L4     ; RETURN
21
22  00144'000144  C100:      100.
23  00145'010000  C4096:     4096.
24  00146'000600  C600:      600
25  00147'000001  TEMP:      .BLK    1
26  00150'000341' .C5:       C5
```

```
---
PAGE   4                    *L..'K ROUTINE*
07/04/75
 1                    ; MAIN PROGRAM
 2
 3  00151'054656  LINK:    STA     3,L2
 4  00152'102400           SUB     0,0
 5  00153'031400           LDA     2,0,3    ; CONST ADD
 6  00154'113000           ADD     0,2      ; DISPLACEMENT
 7  00155'025000           LDA     1,0,2    ; % FLOW
 8  00156'125005           MOV     1,1,SNR  ; ZERO ?
 9  00157'000425           JMP     LIN      ; YES, DONT PROCESS
10  00160'031401           LDA     2,1,3    ; GET INITIAL ADD
11  00161'105000           MOV     0,1
12  00162'127120           ADDZL   1,1      ; *4
13  00163'133000           ADD     1,2
14  00164'025005           LDA     1,5,2    ; FIRST WORD OF LANE
15  00165'125120           MOVZL   1,1      ; SHIFT
16  00166'045005           STA     1,5,2
17  00167'025006           LDA     1,6,2    ; 2ND WORD
18  00170'125100           MOVL    1,1
19  00171'045006           STA     1,6,2
20  00172'025007           LDA     1,7,2    ; 3RD WORD
21  00173'125100           MOVL    1,1
22  00174'045007           STA     1,7,2
23  00175'025010           LDA     1,10,2   ; 4TH WORD
24  00176'125100           MOVL    1,1
25  00177'045010           STA     1,10,2
26  00200'126560           SUBCL   1,1      ; SAVE CARRY
27  00201'031401           LDA     2,1,3
28  00202'113000           ADD     0,2
29  00203'045000           STA     1,0,2    ; STORE OUTPUT
30  00204'101400  LIN:     INC     0,0
31  00205'030514           LDA     2,C5
32  00206'112415           SUB#    0,2,SNR  ; FINISHED ?
33  00207'000402           JMP     LINK1    ; YES
34  00210'000743           JMP     LINK+2
35  00211'126400  LINK1:   SUB     1,1
36  00212'044533           STA     1,BLOCK  ; ZERO BLOCKED VEHICLES
37  00213'034614           LDA     3,L2
38  00214'031402           LDA     2,2,3    ; GET INPUT ADD
39  00215'031000           LDA     2,0,2    ; GET INPUT
40  00216'151005           MOV     2,2,SNR  ; ZERO ?
41  00217'000517           JMP     RET
42  00220'060770           DIBP    1,70     ; GET RANDUM NO
43  00221'102400           SUB     0,0
44  00222'031400  LINK2:   LDA     2,0,3    ; VARIABLE ADD
45  00223'151400           INC     2,2
46  00224'035377           LDA     3,-1,2   ; %FLOW
47  00225'101400           INC     0,0
48  00226'166032           ADCZ#   3,1,SZC  ; AC1 >AC3
49  00227'000774           JMP     LINK2+1  ; YES NEXT F.R
50  00230'151400  LINK3:   INC     2,2
51  00231'035377           LDA     3,-1,2   ; NEXT FLOW
52  00232'175004           MOV     3,3,SZR
53  00233'166033           ADCZ#   3,1,SNC  ; AC1 > AC3 ?
54  00234'000403           JMP     .+3      ; NO
55  00235'101400           INC     0,0
56  00236'000772           JMP     LINK3
57  00237'036711           LDA     3,@.C5
58  00240'116433           SUBZ#   0,3,SNC  ; >5 ?
```

```
---
PAGE   5                    *LINK ROUTINE*
07/04/75
  1   00241'161000            MOV     3,0
  2   00242'042505            STA     0,@.L3  ; DISPLACEMENT
  3   00243'036503  LINK4:    LDA     3,@.L2
  4   00244'035400            LDA     3,0,3
  5   00245'021405            LDA     0,5,3   ; GET DIST ADD
  6   00246'061176            DOAS    0,76    ; INITIALISE GENERATOR
  7   00247'025407            LDA     1,7,3   ; ST DEV CONSTANT
  8   00250'066201            DOBC    1,MDV
  9   00251'070776            DIAP    2,76    ; GET RANDUM NO
 10   00252'073301            MUL
 11   00253'060000            NIO     0       ; WAIT
 12   00254'025406            LDA     1,6,3   ; MEAN CONSTANT
 13   00255'061401            DIB     0,MDV
 14   00256'123122            ADDZL   1,0,SZC ; REQUIRED NO
 15   00257'102400            SUB     0,0     ; NEGATIVE
 16   00260'024464            LDA     1,MSK
 17   00261'123700            ANDS    1,0     ; /128
 18   00262'024461            LDA     1,C63
 19   00263'122013            ADCZ#   1,0,SZC ; >63 ?
 20   00264'121000            MOV     1,0     ; YES, SET TO 63
 21   00265'106400            SUB     0,1     ; CONVERT TO TRAVELLING TIME
 22   00266'121000            MOV     1,0
 23   00267'126520            SUBZL   1,1
 24
 25                                           ; RESULT 0<= ACO <= 63
 26   00270'030452            LDA     2,C20   ; 16.
 27   00271'142413            SUBZ#   2,0,SNC ; CURRENT WORD ?
 28   00272'000404            JMP     .+4     ; YES
 29   00273'125400            INC     1,1     ; NEXT WORD
 30   00274'142400            SUB     2,0     ; DECREMENT
 31   00275'000774            JMP     .-4
 32   00276'030453            LDA     2,.TBL  ; TABLE ADD
 33   00277'113000            ADD     0,2
 34   00300'021000            LDA     0,0,2   ; WORD
 35   00301'046447            STA     1,@.L4  ; NO OF STREET WORD
 36   00302'036444            LDA     3,@.L2
 37   00303'031401            LDA     2,1,3   ; INITIAL ADD
 38   00304'026443            LDA     1,@.L3  ; LANE DISP
 39   00305'127120            ADDZL   1,1     ; *4
 40   00306'133000            ADD     1,2
 41   00307'026441            LDA     1,@.L4
 42   00310'133000            ADD     1,2
 43   00311'025000            LDA     1,0,2   ; LANE WORD
 44   00312'107414  LINK6:    AND#    0,1,SZR ; IS POSITION EMPTY ?
 45   00313'000405            JMP     LINK7   ; NO
 46   00314'107000            ADD     0,1
 47   00315'045000            STA     1,0,2   ; STA IN MEMORY
 48   00316'017402            DSZ     @2,3    ; DECREMENT INPUT
 49   00317'000575            JMP     LINK1+3
 50   00320'000416            JMP     RET
 51
 52   00321'101124  LINK7:    MOVZL   0,0,SZR ; TRY NEXT POSITION
 53   00322'000770            JMP     LINK6   ; NOT END OF WORD
 54   00323'016425            DSZ     @.L4    ; END OF LANE ?
 55   00324'000402            JMP     .+2     ; NO
 56   00325'000405            JMP     .+5     ; YES, PUT IN BLOCKED QUEUE
 57   00326'102520            SUBZL   0,0     ; 1 TO BIT 15
 58   00327'150400            NEG     2,2
```

```
---
PAGE   6                    *LINK ROUTINE*
07/04/75
 1  00330'150000                COM     2,2
 2  00331'000760                JMP     LINK6-1
 3  00332'017402                DSZ     @2,3     ; DECREMENT INPUT
 4  00333'000401                JMP     .+1
 5  00334'010411                ISZ     BLOCK
 6  00335'000657                JMP     LINK1+3
 7
 8  00336'020407   RET:         LDA     0,BLOCK ; STORE BLOCKED QUEUE
 9  00337'043402                STA     0,@2,3  ; IN INPUT
10  00340'001403                JMP     3,3     ; RETURN
11
12  00341'000005   C5:          5
13  00342'000020   C20:         20
14  00343'000077   C63:         63.
15  00344'177400   MSK:         177400
16  00345'000001   BLOCK:       .BLK    1
17  00346'000027'  .L2:         L2
18  00347'000010'  .L3:         L3
19  00350'000031'  .L4:         L4
20  00351'000352'  .TBL:        TBL
21
22  00352'000001   TBL:         1B15
23  00353'000002                1B14
24  00354'000004                1B13
25  00355'000010                1B12
26  00356'000020                1B11
27  00357'000040                1B10
28  00360'000100                1B9
29  00361'000200                1B8
30  00362'000400                1B7
31  00363'001000                1B6
32  00364'002000                1B5
33  00365'004000                1B4
34  00366'010000                1B3
35  00367'020000                1B2
36  00370'040000                1B1
37  00371'100000                1B0
38
39          000372'             NEXTL =.
40                              .END
```

```
---
PAGE   1
07/04/75
 1                              .TITL   ELINK
 2
 3                      ; ENTRY LINK OF UP TO 5 LANES
 4                      ; CALLING SEQUENCE:
 5
 6                      ; PROGRAM:-
 7                      ;     ELINK,NO.
 8
 9                      ; DATA:-
10                      ;      ELINK,NO.,FLOW RATE 1,FLOWRATE 2,.....FLOW RATE 5
11
12                              .EXTN   PUTP,GET,RETURN,GETBR,MULTIPLY
13                              .EXTN   DIVIDE,ERROR
14                              .HEAD           *ELINK ROUTINE*
15
16                              .ZREL
17  00000-000064' .ELINK:       ELINK
18
19                              .NREL
20  00000'000020'               EL1             ; ADD OF DATA ROUTINE
21  00001'000100                100             ; 64 ELINKS
22  00002'000005                5               ; CONST STORAGE REQUIRED
23  00003'000100                100
24  00004'000031                25.             ; INITIAL STORAGE REQUIRED
25  00005'000131'               NEXTE           ; POINTER TO NEXT ROUTINE
26        000001                .TXTM   1       ; LEFT TO RIGHT PACKING
27  00006'042514                .TXT    *ELINK* ; NAME OF ROUTINE
28  00007'044516
29  00010'045400
30
31                      ;       PROGRAM AREA
32  00011'040405                STA     0,EL2   ; CONST ADD
33  00012'044405                STA     1,EL3   ; INITIAL ADD
34  00013'177777                PUTP            ; WRITE PROG
35  00014'000017'               EL3
36 .00015'006000-               JSR     @.ELINK
37  00016'000001   EL2:         .BLK    1
38  00017'000001   EL3:         .BLK    1
39
```

```
---
PAGE   2                    *ELINK ROUTINE*
07/04/75
 1                          ; DATA ROUTINE
 2
 3  00020'040776  EL1:      STA     0,EL2
 4  00021'102400            SUB     0,0
 5  00022'040775            STA     0,EL3    ; COUNTER
 6  00023'177777            GET              ; GET FLOW RATE
 7  00024'105000            MOV     0,1      ; SET UP ACS FOR MULTIPLY
 8  00025'102400            SUB     0,0
 9  00026'030434            LDA     2,C10K   ; 2^12
10  00027'177777            MULTIPLY
11  00030'030431            LDA     2,C100
12  00031'177777            DIVIDE
13  00032'030764            LDA     2,EL2
14  00033'024764            LDA     1,EL3    ; DISPLACEMENT
15  00034'133000            ADD     1,2
16  00035'041000            STA     0,0,2    ; STORE FLOW RATE CONSTANT
17  00036'010761            ISZ     EL3      ; BUMP DISPLACEMENT
18  00037'177777            GETBR            ; GET BREAK CODE
19  00040'174014            COM#    3,3,SZR  ; TEST FOR CR (-1)
20  00041'000412            JMP     EL6      ; NO,CONTINUE
21  00042'024755            LDA     1,EL3    ; ZERO REMAINING LANES
22  00043'020420            LDA     0,C5
23  00044'176400            SUB     3,3
24  00045'122415  EL5:      SUB#    1,0,SNR  ; FINISHED ?
25  00046'177777            RETURN           ; YES
26  00047'151400            INC     2,2
27  00050'125400            INC     1,1
28  00051'050000            STA     3,0,2
29  00052'000771            JMP     EL5
30
31  00053'020410  EL6:      LDA     0,C5     ; CHECK FOR MORE THAN 5 PARMS
32  00054'024743            LDA     1,EL3
33  00055'122414            SUB#    1,0,SZR
34  00056'000745            JMP     EL1+3    ; NO
35  00057'177777            ERROR
36  00060'000024            20.              ; MORE THAN 5 LANES
37
38  00061'023060  C100:     10000.
39  00062'010000  C10K:     10000
40  00063'000005  C5:       5
41
42                ;         ACTUAL USER SUBROUTINE
43
44  00064'054731  ELINK:    STA     3,EL2
45  00065'102400            SUB     0,0
46  00066'031400            LDA     2,0,3    ; GET CONSTANT ADD
47  00067'113000            ADD     0,2      ; ADD IN DISPLACEMENT
48  00070'025000            LDA     1,0,2    ; GET FLOW RATE
49  00071'125005            MOV     1,1,SNR  ; IS IT ZERO ?
50  00072'000411            JMP     EL7      ; YES , NEXT LANE
51  00073'071770            DIBP    2,70     ; GET RANDUM NO
52  00074'145020            ADCZ    2,1      ; 1 TO CARRY IF VEHICLE GENERATED
53  00075'126560            SUBCL   1,1
54  00076'031401            LDA     2,1,3    ; GET INITIAL ADD
55  00077'115000            MOV     0,3
56  00100'177120            ADDZL   3,3      ; *4
57  00101'173000            ADD     3,2
58  00102'125201            MOVR    1,1
```

```
---
PAGE   3                        *ELINK ROUTINE*
07/04/75
 1   00103'025003                 LDA     1,5,2   ; GET 1ST LENGTH OF LANE
 2   00104'125100                 MOVL    1,1     ; SHIFT
 3   00105'045003                 STA     1,5,2   ; STORE
 4   00106'025006                 LDA     1,6,2   ; NEXT LENGTH
 5   00107'125100                 MOVL    1,1
 6   00110'045006                 STA     1,6,2
 7   00111'025007                 LDA     1,7,2
 8   00112'125100                 MOVL    1,1
 9   00113'045007                 STA     1,7,2
10   00114'025010                 LDA     1,10,2
11   00115'125100                 MOVL    1,1
12   00116'045010                 STA     1,10,2
13   00117'126360                 SUBCL   1,1     ; SAVE CARRY
14   00120'034676                 LDA     3,EL2
15   00121'031401                 LDA     2,1,3   ; INITIAL ADD
16   00122'113000                 ADD     0,2
17   00123'045000                 STA     1,0,2   ; STORE OUTPUT
18   00124'101400  EL7:           INC     0,0     ; BUMP COUNTER
19   00125'030715                 LDA     2,C5
20   00126'112415                 SUB#    0,2,SNR ; FINISHED ?
21   00127'001402                 JMP     2,3     ; YES RETURN
22   00130'000736                 JMP     ELINK+2 ; CONTINUE
23
24         000131'                NEXTE=.
25
26                                .END
```

```
---
PAGE   1
07/22/75
 1                              .TITL   TLIGHT
 2                      ; TLIGHT IS A CONTROLLED INTERSECTION WITH UP TO
 3                      ; 5 LANES PER APPROACH
 4                      ; FIXED TIME IS USED
 5
 6                      ; PROG: TLIGHT,NO,INPUT PARM 1,NO 1,INPUT PARM 2,  NO 2,.....
 7                      ;       .....INPUT PARM 4,NO 4
 8
 9                      ;DATA:  TLIGHT,NO,LEG NO,SAT FLOW RATE,%LEFT 1,%RIGHT 1
10                      ;       [%LEFT 2,%RIGHT 2.......%LEFT 5,%RIGHT 5]
11
12                      ; DATA: TLIGHT,NO,5,OFFSET,GREEN 1-3,AMBER,DEAD,
13                      ;       GREEN 2-4,AMBER,DEAD
14
15                      ; DATA: TLIGHT,NO,6,DISTRIBUTION,MEAN *100,STD DEV * 100
16
17                              .EXTN   PUTP,GET,GETAD,GETEP,RETURN,MULTIPLY
18                              .EXTN   DIVIDE,ERROR
19                              .HEAD           *TLIGHT ROUTINE*
20                              .ZREL
21  00000-000232'       .TLIGHT: TLIGHT
22  00001-000005        C5:      5
23  00002-000001        RTN:     .BLK    1
24  00003-000001        TEMP3:   .BLK    1
25  00004-000001        TEMP2:   .BLK    1
26  00005-000001        TEMP0:   .BLK    1
27  00006-000001        SATAD:   .BLK    1
28
29        000001                .TXTM   1
30                              .NREL
31  00000'000035'               TT1             ; VARIABLE ROUTINE
32  00001'000050                40.             ; NO OF TLIGHTS
33  00002'000066                66              ; CONSTANT STORAGE LOCS
34  00003'000050                40.             ; NO OF TLIGHTS
35  00004'000033                33              ; INITIAL LOCS
36  00005'000716'               NEXTT
37  00006'052114                .TXT    *TLIGHT*
38  0000C '044507
39  00010'044124
40  00011'000000
41
42                                          ; PROG AREA
43  00012'040415                STA     0,T2    ; CONST ADD
44  00013'044415                STA     1,T3    ; INITIAL ADD
45  00014'177777                GETAD           ; INPUT PARM ADD
46  00015'044414                STA     1,T4
47  00016'000014'               GETAD
48  00017'044413                STA     1,T4+1
49  00020'000016'               GETAD
50  00021'044412                STA     1,T4+2
51  00022'000020'               GETAD
52  00023'044411                STA     1,T4+3
53  00024'177777                PUTP
54  00025'000034'               T4+3
55  00026'005000-               JSR     @.TLIGHT
56  00027'000001        T2:     .BLK    1       ; CONST ADD
57  00030'000001        T3:     .BLK    1       ; INITIAL ADD
58  00031'000004        T4:     .BLK    4       ; INPUT PARM ADDS
```

```
---
PAGE   2                    *TLIGHT ROUTINE*
07/22/75
 1  00035'040772  TT1:     STA     0,T2     ; CONST ADD
 2  00036'111000            MOV     0,2
 3  00037'177777            GET              ; LEG NO
 4  00040'034001-           LDA     3,C5
 5  00041'116415            SUB#    0,3,SNR  ; SIGNAL SETTING ?
 6  00042'000472            JMP     SIG      ; YES
 7  00043'162032            ADCZ#   3,0,SZC  ; DISTRIBUTION ?
 8  00044'000503            JMP     DIST     ; YES
 9  00045'040763            STA     0,T3     ; LEG NO
10  00046'000037'           GET              ; SAT FLOW RATE
11  00047'105000            MOV     0,1      ; PROCESS TO 12 BITS
12  00050'102400            SUB     0,0
13  00051'030550            LDA     2,C10K
14  00052'177777            MULTIPLY
15  00053'030543            LDA     2,C100
16  00054'177777            DIVIDE
17  00055'024753            LDA     1,T3
18  00056'030751            LDA     2,T2
19  00057'133000            ADD     1,2
20  00060'041377            STA     0,-1,2   ; FLOW RATE WORD
21  00061'121000            MOV     1,0      ; * 5
22  00062'127120            ADDZL   1,1
23  00063'107000            ADD     0,1
24  00064'030743            LDA     2,T2
25  00065'150400            NEG     2,2
26  00066'150000            COM     2,2
27  00067'133000            ADD     1,2
28  00070'050744            STA     2,T4+3   ; EFFECTIVE ADD
29  00071'102400            SUB     0,0
30  00072'040736            STA     0,T3     ; COUNTER
31  00073'000046'  TT5:     GET              ; % LEFT TURNS
32  00074'040735            STA     0,T4
33  00075'000073'           GET              ; % RIGHT TURNS
34  00076'040734            STA     0,T4+1
35  00077'030733            LDA     2,T4+1   ; % RIGHT
36  00100'102400            SUB     0,0      ; RUNNING SUM
37  00101'004423            JSR     CONV+1   ; CONVERT TO 12 BITS
38  00102'030732            LDA     2,T4+3
39  00103'041024            STA     0,24,2
40  00104'030725            LDA     2,T4     ; LEFT
41  00105'004415            JSR     CONV
42  00106'042726            STA     0,@T4+3
43  00107'010725            ISZ     T4+3
44  00110'010720            ISZ     T3       ; NEXT LANE
45  00111'177777            GETBR
46  00112'175004            MOV     3,3,SZR  ; BREAK = , ?
47  00113'177777            RETURN           ; NO
48  00114'020714            LDA     0,T3
49  00115'024001-           LDA     1,C5
50  00116'122433            SUBZ#   1,0,SNC  ; MORE THAN 5 LANES ?
51  00117'000754            JMP     TT5      ; NO
52  00120'177777            ERROR            ; ERROR WARNING
53  00121'000024            20.
```

```
---
PAGE   3                    *TLIGHT ROUTINE*
07/22/75
 1   00122'020711  CONV:    LDA     0,T4+2
 2   00123'054003-          STA     3,TEMP3
 3   00124'113000           ADD     0,2
 4   00125'050706           STA     2,T4+2
 5   00126'101400           SUB     0,0
 6   00127'024470           LDA     1,C4096 ; * 2^12
 7   00130'000052'          MULTIPLY
 8   00131'030464           LDA     2,CD100 ; /100
 9   00132'000054'          DIVIDE
10   00133'002003-          JMP     @TEMP3
11
12   00134'024457  SIG:     LDA     1,C7    ; SIGNAL SETTINGS
13   00135'044673           STA     1,T3
14   00136'000075'          GET
15   00137'115120           MOVZL   0,3     ; CONVERT TO EPOCHS
16   00140'163000           ADD     3,0
17   00141'101400           INC     0,0
18   00142'041057           STA     0,57,2
19   00143'151400           INC     2,2
20   00144'0146E4           DSZ     T3
21   00145'000771           JMP     SIG+2
22   00146'000113'          RETURN
23
24   00147'000022'  DIST:    GETAD
25   00150'034657           LDA     3,T2
26   00151'051454           STA     2,54,3  ; DIST ADD
27   00152'031100           LDA     2,100,2 ; MEAN OF DIST
28   00153'050655           STA     2,T3
29   00154'000136'          GET             ; MEAN * 100
30   00155'105000           MOV     0,1     ; * 3*128
31   00156'030442           LDA     2,C600
32   00157'102400           SUB     0,0
33   00160'000130'          MULTIPLY
34   00161'030437'          LDA     2,CD100 ; /100
35   00162'000132'          DIVIDE
36   00163'040646           STA     0,T4    ; TEMP STORE
37   00164'000154'          GET             ; STD * 100
38   00165'105000           MOV     0,1     ; * 128*3
39   00166'030432           LDA     2,C600
40   001E   '02400          SUB     0,0
41   00170'000160'          MULTIPLY
42   00171'030424           LDA     2,CD100
43   00172'000162'          DIVIDE
44   00173'034634           LDA     3,T2
45   00174'031454           LDA     2,54,3  ; ADD OF DIST
46   00175'031101           LDA     2,101,2 ; STD DEV OF DIST
47   00176'126400           SUB     1,1
48   00177'000172'          DIVIDE
49   00200'034627           LDA     3,T2
50   00201'041456           STA     0,56,3  ; STD DEV CONST
51   00202'111000           MOV     0,2
52   00203'102400           SUB     0,0
53   00204'024634           LDA     1,T3
54   00205'000170'          MULTIPLY        ; MEAN * STD DEV
55   00206'034621           LDA     3,T2
56   00207'024632           LDA     1,T4
57   00210'106400           SUB     0,1
58   00211'045455           STA     1,55,3  ; MEAN CONST
```

```
---
PAGE   4                    *TLIGHT ROUTINE*
07/22/75
 1   00212'000146'            RETURN
 2
 3   00213'000007   C7:       7
 4   00214'000012   C10:      10.
 5   00215'000144   CD100:    100.
 6   00216'025060   C108:     10800.
 7   00217'010000   C4096:    4096.
 8   00220'000600   C600:     600
 9   00221'010000   C1 :      10000
```

```
---
PAGE   5                    *TLIGHT ROUTINE*
07/22/75
 1  00222'054002- TLIGHT: STA     3,RTN
 2  00223'102520          SUBZL   0,0
 3                                          ; MOVE VEHICLES INTTO QUEUES
 4  00224'040005-         STA     0,TEMP0 ; COUNTER
 5  00225'031401          LDA     2,1,3   ; INITIAL ADD
 6  00226'050003-         STA     2,TEMP3
 7  00227'117000  TL1:    ADD     0,3
 8  00230'031401          LDA     2,1,3   ; INPUT PARM ADD
 9  00231'024001-         LDA     1,C5
10  00232'147000          ADD     2,1
11  00233'021000  TL2:    LDA     0,0,2   ; OUTPUT OF LINK
12  00234'101005          MOV     0,0,SNR ; EMPTY
13  00235'000403          JMP     .+3     ; NO
14  00236'034003-         LDA     3,TEMP3 ; QUEUE COUNTER
15  00237'011404          ISZ     4,3     ; BUMP QUEUE
16  00240'151400          INC     2,2
17  00241'010003-         ISZ     TEMP3
18  00242'146414          SUB#    2,1,SZR ; FINISHED LEG ?
19  00243'000770          JMP     TL2     ; NO
20  00244'010005-         ISZ     TEMP0   ; LEG COUNTER
21  00245'020005-         LDA     0,TEMP0
22  00246'024001-         LDA     1,C5
23  00247'034002-         LDA     3,RTN
24  00250'106032          ADCZ#   0,1,SZC ; FINISHED ?
25  00251'000756          JMP     TL1     ; NO
26
27  00252'031400          LDA     2,0,3   ; CONST ADD
28  00253'025401          LDA     3,1,3   ; INITIAL ADD
29  00254'126400          SUB     1,1
30  00255'044505          STA     1,AMB
31  00256'021430          LDA     0,30,2  ; EXAMINE SIGNALS
32  00257'101122          MOVZL   0,0,SZC ; PHASE ?
33  00260'024001-         LDA     1,C5    ; PHASE 2-4
34  00261'044500          STA     1,PCNT  ; PHASE COUNTER
35  00262'101220          MOVZR   0,0
36  00263'101205          MOVR    0,0,SNR ; GREEN ?
37  00264'000477          JMP     GREEN
38  00265'101204          MOVR    0,0,SZR ; AMBER ?
39  00266'000405          JMP     UPSIG+3 ; NO , DEAD
40  00267'010473          ISZ     AMB     ; AMBER FLAG
41  00270'000473          JMP     [illegible]N
42
43  00271'034002- UPSIG:  LDA     [illegible]
44  00272'031400          LDA     [illegible]   ; CONST ADD
45  00273'025401          LDA     3,1,3   ; INITIAL ADD
46  00274'011431          ISZ     31,3    ; BUMP SIG COUNTER
47  00275'021432          LDA     0,32,3  ; OFFSET FLAG
48  00276'101005          MOV     0,0,SNR ; OFFSET PERIOD OVER ?
49  00277'000451          JMP     OFSET   ; NO
50  00300'126400          SUB     1,1
51  00301'021430          LDA     0,30,2  ; SIG WORD
52  00302'101135          MOVZL#  0,0,SNR ; GREEN ?
53  00303'000405          JMP     TEST    ; YES
54  00304'125400          INC     1,1
55  00305'101222          MOVZR   0,0,SZC ; WHICH CYCLE ?
56  00306'000402          JMP     TEST
57  00307'000775          JMP     .-3
58
```

```
---
PAGE   6                    *TLIGHT ROUTINE*
07/22/75
 1   00310'102400  TEST:    SUB     0,0
 2   00311'133000            ADD     1,2       ; ADD CYCLE DISP
 3   00312'024447            LDA     1,PCNT    ; PHASE
 4   00313'125004            MOV     1,1,SZR   ; WHICH PHASE ?
 5   00314'101540            INCCL   0,0       ; 2-4 , 3-> AC0
 6   00315'113000            ADD     0,2
 7   00316'025060            LDA     1,60,2    ; GET MAX COUNT
 8   00317'021431            LDA     0,31,3    ; COUNTER
 9   00320'106415            SUB#    0,1,SNR   ; UPDATE ?
10   00321'000403            JMP     .+3       ; YES
11   00322'034002- RET:      LDA     3,RTN
12   00323'001406            JMP     6,3       ; RETURN
13   00324'021430            LDA     0,30,3    ; SIGNAL WORD
14   00325'024433            LDA     1,MSK1
15   00326'107414            AND#    0,1,SZR   ; CHANGE PHASE ?
16   00327'000413            JMP     CHNG      ; YES
17   00330'105000            MOV     0,1       ; PRESERVE BIT 0
18   00331'101135            MOVZL#  0,0,SNR   ; GREEN ?
19   00332'101521            INCZL   0,0,SKP   ; YES
20   00333'103120            ADDZL   0,0       ; SHIFT 2 POSITIONS
21   00334'125120            MOVZL   1,1       ; BIT 1 TO CARRY
22   00335'101200            MOVR    0,0       ; RESTORE BIT 0
23   00336'041430            STA     0,30,3
24   00337'102400            SUB     0,0
25   00340'041431            STA     0,31,3    ; CLEAR COUNTER
26   00341'000761            JMP     RET       ; RETURN
27
28   00342'101120  CHNG:     MOVZL   0,0       ; BIT 0 TO CARRY
29   00343'102460            SUBC    0,0       ; PRESERVE CARRY ,SET TO GREEN
30   00344'041431            STA     0,31,3    ; CLEAR COUNTER
31   00345'101260            MOVCR   0,0       ; COMPLEMENT BIT 0
32   00346'041430            STA     0,30,3
33   00347'000753            JMP     RET
34
35   00350'021057  OFSET.    LDA     0,57,2    ; OFFSET PERIOD
36   00351'025431            LDA     1,31,3    ; COUNTER
37   00352'106414            SUB#    0,1,SZR   ; PERIOD OVER ?
38   00353'000747            JMP     RET       ; NO
39   00354'102400            SUB     0,0
40   00355'041431            STA     0,31,3    ; CLEAR COUNTER
41   00356'011432            ISZ     32,3      ; BUMP FLAG
42   00357'000743            JMP     RET
43
44   00360'000002  MSK1:     2
45   00361'000001  PCNT:     .BLK    1
46   00362'000001  AMB:      .BLK    1
```

```
---
PAGE   7                    *TLIGHT ROUTINE*
07/22/75
 1   00363'141000  GREEN:  MOV     2,0
 2   00364'133000          ADD     1,2      ; PHASE CONST ADD
 3   00365'137000          ADD     1,3      ; PHASE INITIAL ADD
 4   00366'125004          MOV     1,1,SZR  ; WHICH PHASE
 5   00367'101400          INC     0,0      ; 2-4
 6   00370'040006-         STA     0,SATAD  ; SAT FLOW ADD
 7   00371'102400          SUB     0,0
 8   00372'040476          STA     0,QCNT   ; QUEUE COUNTER
 9   00373'101400          INC     0,0
10   00374'125004          MOV     1,1,SZR  ; WHICH LEG ?
11   00375'101400          INC     0,0      ; LEG 2
12   00376'040473          STA     0,LCNT   ; LEG COUNTER
13
14   00377'021404  GR1:    LDA     0,4,'    ; GET QUEUE
15   00400'101005          MOV     0,0,SNR  ; EMPTY ?
16   00401'000424          JMP     GR2      ; YES
17   00402'101112          MOVL#   0,0,SZC  ; BALKED QUEUE
18   00403'000467          JMP     RIGHT    ; YES
19   00404'061770          DIBP    0,70
20   00405'026006-         LDA     1,@SATAD
21   00406'122032          ADCZ#   1,0,SZC  ; DISCHARGE ?
22   00407'000416          JMP     GR2      ; NO
23   00410'061770          DIBP    0,70     ; GET RND NO
24   00411'101005          MOV     0,0,SNR  ; IGNORE 0
25   00412'000776          JMP     .-2
26   00413'025030          LDA     1,30,2   ; RIGHT TURNS CONST
27   00414'122033          ADCZ#   1,0,SNC  ; RIGHT ?
28   00415'000455          JMP     RIGHT
29   00416'024744          LDA     1,AMB
30   00417'125004          MOV     1,1,SZR  ; AMBER ?
31   00420'000405          JMP     GR2      ; YES , ONLY RIGHT TURNS ALLOWED
32   00421'025004          LDA     1,4,2    ; LEFT TURN CONST
33   00422'122033          ADCZ#   1,0,SNC  ; LEFT ?
34   00423'000427          JMP     DISCH    ; YES
35   00424'000425          JMP     DISCH+1  ; STRAIGHT
36
37   00425'010441  GR2:    ISZ     QCNT
38   00426'151400          INC     2,2
39   00427'175400          INC     3,3
40   00430'024440          LDA     1,QCNT
41   00431'020001-         LDA     0,C5
42   00432'106414          SUB#    0,1,SZR  ; FINISHED LEG ?
43   00433'000744          JMP     GR1      ; NO
44   00434'010435          ISZ     LCNT     ; BUMP LEG COUNTER
45   00435'010434          ISZ     LCNT
46   00436'024433          LDA     1,LCNT
47   00437'106432          SUBZ#   0,1,SZC  ; FINISHED BOTH PHASES ?
48   00440'000631          JMP     UPSIG    ; YES
49   00441'117000          ADD     0,3      ; NEXT LEG
50   00442'113000          ADD     0,2
51   00443'125400          SUB     1,1
52   00444'044424          STA     1,QCNT
53   00445'010006-         ISZ     SATAD
54   00446'010006-         ISZ     SATAD
55   00447'000710          JMP     GR1
56
57   00450'126401  DISCH:  SUB     1,1,SKP  ; LEFT
58   00451'126520          SUBZL   1,1
```

```
---
PAGE   8                    *TLIGHT ROUTINE*
07/22/75
 1   00452'015404           DSZ     4,3      ; DECREMENT QUEUE
 2   00453'000401           JMP     .+1
 3   00454'020415           LDA     0,LCNT
 4   00455'123000           ADD     1,0
 5   00456'024510           LDA     1,C4
 6   00457'122432           SUBZ#   1,0,SZC  ; >=4 ?
 7   00460'122400           SUB     1,0      ; YES
 8   00461'054003-          STA     3,TEMP3
 9   00462'034002-          LDA     3,RTN
10   00463'035401           LDA     3,1,3    ; INITIAL ADD
11   00464'117000           ADD     0,3
12   00465'011400           ISZ     0,3      ; BUMP OUTPUT
13   00466'034003-          LDA     3,TEMP3
14   00467'000736           JMP     GR2
15
16   00470'000001  @CNT:    .BLK    1
17   00471'000001  LCNT:    .BLK    1
18
19   00472'050004- RIGHT:   STA     2,TEMP2
20   00473'054003-          STA     3,TEMP3
21   00474'020666           LDA     0,AMB
22   00475'101004           MOV     0,0,SZR  ; AMBER ?
23   00476'000560           JMP     AMBER    ; YES
24   00477'020772           LDA     0,LCNT   ; LEG COUNTER
25   00500'024465           LDA     1,C2
26   00501'122032           ADDZ#   1,0,SZC  ; >2 ?
27   00502'122401           SUB     1,0,SKP  ; YES
28   00503'123000           ADD     1,0
29   00504'040005-          STA     0,TEMP0  ; AC0   = OPPOSING LEG NO
30   00505'034002-          LDA     3,RTN
31   00506'035401           LDA     3,1,3    ; INITIAL ADD
32   00507'105000           MOV     0,1      ; *5
33   00510'102120           ADDZL   0,0
34   00511'123000           ADD     1,0
35   00512'117000           ADD     0,3      ; ADD OF 1ST QUEUE
36   00513'126400           SUB     1,1
37   00514'175400  RT1:     INC     3,3
38   00515'021776           LDA     0,-2,3   ; GET QUEUE
39   00516'101133           MOVZL#  0,0,SNC  ; BALKED
40   00517'101005           MOV     0,0,SNR  ; EMPTY
41   00520'000402           JMP     .+2
42   00521'000506           JMP     BALK     ; CANT TURN
43   00522'125400           INC     1,1
44   00523'020001-          LDA     0,C5
45   00524'122414           SUB#    1,0,SZR  ; FINISHED ?
46   00525'000767           JMP     RT1      ; NO
47                                           ; CHECK FOR GAPS
48   00526'036002-          LDA     3,@RTN   ; CONST ADD
49   00527'021454           LDA     0,54,3   ; DIST ADD
50   00530'061176           DOAS    0,76     ; INITIALISE GENERATOR
51   00531'025456           LDA     1,56,3   ; STD DEV CONST
52   00532'066701           DOBC    1,MDV
53   00533'070776           DIAP    2,76     ; RND NO
54   00534'073301           MUL
55   00535'060000           NIO     0        ; WAIT
56   00536'025455           LDA     1,55,3   ; MEAN CONST
57   00537'061401           DIB     0,MDV
58   00540'123122           ADDZL   1,0,SZC  ; ADD MEAN
```

```
---
PAGE   9                    *TLIGHT ROUTINE*
07/22/75
 1   00541'102400            SUB     0,0       ; NEGATIVE
 2   00542'024526            LDA     1,MSK
 3   00543'123700            ANDS    1,0       ; /128
 4   00544'024525            LDA     1,C63
 5   00545'122032            ADCZ#   1,0,SZC   ; >63 ?
 6   00546'121000            MOV     1,0       ; YES SET TO 63
 7   00547'126520            SUBZL   1,1
 8   00550'044513            STA     1,WDCNT
 9   00551'030521            LDA     2,C20
10   00552'142433            SUBZ#   2,0,SNC   ; CURRENT WORD
11   00553'000405            JMP     RT2       ; YES
12   00554'004413            JSR     CHECK     ; CHECK CURRENT WORD FOR NULL
13   00555'010505            ISZ     WDCNT
14   00556'142400            SUB     2,0
15   00557'000773            JMP     .-5
16
17   00560'034514   RT2:     LDA     3,.TBL
18   00561'117000            ADD     0,3
19   00562'025400            LDA     1,0,3     ; GET POSITION WORD
20   00563'004405            JSR     CHECK+1
21   00564'000432            JMP     TURN      ; SUCESS
22
23   00565'000001   C2:      2
24   00566'000004   C4:      4
```

```
---
PAGE  10                *TLIGHT ROUTINE*
07/22/75
 1  00567'126400  CHECK:  SUB     1,1
 2  00570'054477          STA     3,CHK+3
 3  00571'040473          STA     0,CHK
 4  00572'044473          STA     1,CHK+1
 5  00573'050473          STA     2,CHK+2
 6  00574'020005-         LDA     0,TEMP0 ; OPPOSING LEG NO
 7  00575'034002-         LDA     3,RTN
 8  00576'117000          ADD     0,3
 9  00577'035401          LDA     3,1,3   ; OPPOSING LINK ADD
10  00600'020463          LDA     0,WDCNT ; WORD NO
11  00601'116400          SUB     0,3     ; ADD OF WORD -11
12  00602'152400          SUB     2,2     ; CHECK  ALL LANES
13  00603'145000          MOV     2,1
14  00604'021411  CH1:    LDA     0,11,3  ; GET WORD
15  00605'100000          COM     0,0     ; INCLUSIVE OR
16  00606'107400          AND     0,1
17  00607'106000          ADC     0,1     ; RESULT IN AC1
18  00610'151400          INC     2,2
19  00611'020001-         LDA     0,C5
20  00612'142415          SUB#    2,0,SNR ; FINISHED
21  00613'000404          JMP     .+4     ; YES
22  00614'020752          LDA     0,C4    ; NEXT LANE
23  00615'117000          ADD     0,3
24  00616'000766          JMP     CH1
25  00617'020446          LDA     0,CHK+1 ; POSITION WORD
26  00620'101120          MOVZL   0,0     ; SHIFT 1 POSITION
27  00621'122413          SUBZ#   1,0,SNC ; EMPTY  ENOUGH
28  00622'000405          JMP     BALK    ; NO
29  00623'020441          LDA     0,CHK
30  00624'024441          LDA     1,CHK+1
31  00625'030441          LDA     2,CHK+2
32  00626'002441          JMP     @CHK+3
33
34  00627'030004- BALK:   LDA     2,TEMP2
35  00630'034003-         LDA     3,TEMP3
36  00631'021404          LDA     0,4,3   ; GET QUEUE
37  00632'101120          MOVZL   0,0
38  00633'101240          MOVOR   0,0     ; PUT IN BALK MARK
39  00634'041404          STA     0,4,3
40  00635'002440          JMP     @.GR2   ; RETURN
41
42  00636'034003- TURN:   LDA     3,TEMP3
43  00637'021404          LDA     0,4,3
44  00640'100400          NEG     0,0
45  00641'100100          COML    0,0
46  00642'101220          MOVZR   0,0
47  00643'041404          STA     0,4,3   ; DECREMENT AND DEBALK
48  00644'020625          LDA     0,LCNT
49  00645'100400          NEG     0,0
50  00646'100003          COM     0,0,SNR
51  00647'020717          LDA     0,C4
52  00650'030002-         LDA     2,RTN
53  00651'031001          LDA     2,1,2
54  00652'113000          ADD     0,2
55  00653'011377          ISZ     -1,2    ; BUMP OUTPUT
56  00654'030004-         LDA     2,TEMP2
57  00655'002420          JMP     @.GR2
58
```

```
---
PAGE  11                    *TLIGHT ROUTINE*
07/22/75
 1   00656'061770  AMBER:   DIBP     0,70
 2   00657'026006-          LDA      1,@SATAD
 3   00660'122013           ADCZ#    1,0,SNC ; DISHCHARGE AT SAT RATE
 4   00661'000755           JMP      TURN    ; YES
 5   00662'002413           JMP      @.GR2   ; RETURN
 6
 7   00663'000001  WDCNT:   .BLK     1
 8   00664'000004  CHK:     .BLK     4
 9   00670'177400  MSK:     177400
10   00671'000077  C63:     63.
11   00672'000020  C20:     20
12   00673'100001  C1001:   100001
13   00674'000676' .TBL:    TABLE
14   00675'000425' .GR2:    GR2
15
16   00676'100000  TABLE:   1B0
17   00677'040000           1B1
18   00700'020000           1B2
19   00701'010000           1B3
20   00702'004000           1B4
21   00703'002000           1B5
22   00704'001000           1B6
23   00705'000400           1B7
24   00706'000200           1B8
25   00707'000100           1B9
26   00710'000040           1B10
27   00711'000020           1B11
28   00712'000010           1B12
29   00713'000004           1B13
30   00714'000002           1B14
31   00715'000001           1B15
32
33        000716'           NEXTT = .
34                          .END
```

```
---
PAGE   1
06/25/75
 1                          .TITL   PRINT
 2
 3                  ; PRINTS OUT THE QUEUE LENGTHS AT A TLIGHT
 4
 5                  ; PROG: PRINT,NO.,TLIGHT,NO
 6                  ; DATA: PRINT,NO.,EPOCHS BEFORE PRINT,EPOCHS BETWEEN PRINT
 7
 8                  ; OUTPUT: EPOCH NO,TLIGHT NO,QUEUES LEG 1,LEG 2,LEG3,LEG4
 9
10                          .EXTN   RETURN,GETAD,GET,PUT,CRLF,PUTC,PUTP
11          000001          .TXTM   1
12                          .HEAD           *PRINT ROUTINE*
13                          .ZREL
14  00000-000035' .PRINT:   PRINT
15
16                          .NREL
17  00000'000025'           P1                  ; VARIABLE ROUTINE
18  00001'000050            40.                 ; NO OF PRINTS
19  00002'000003            3                   ; NO OF VARIABLES
20  00003'000050            40.                 ; NO OF PRINTS
21  00004'000002            2                   ; INITIAL STORAGE
22  00005'000124'           NEXTP
23  00006'050122            .TXT    *PRINT*
24  000[illegible] 044515
25  00010'052000
26
27  00011'040411            STA     0,P2        ; VARIABLE ADD
28  00012'044411            STA     1,P3        ; INITIAL ADD
29  00013'177777            GETAD               ; ADD OF TLIGHT
30  00014'044410            STA     1,P4
31  00015'010405            LDA     2,P2
32  00016'055002            STA     3,2,2       ; TLIGHT NO
33  00017'177777            PUTP
34  00020'000024'           P4
35  00021'005000-           JSR     @.PRINT
36  00022'000001    P2:     .BLK    1           ; VAR ADD
37  00023'000001    P3:     .BLK    1           ; INITIAL ADD
38  00024'000001    P4:     .BLK    1           ; TLIGHT INITIAL ADD
39
40  00025'040775    P1:     STA     0,P2        ; CONSTANT ROUTINE
41  00026'044775            STA     1,P3
42  00027'111000            MOV     0,2
43  00030'177777            GET                 ; EPOCHS BEFORE PRINTS
44  00031'041000            STA     0,0,2
45  00032'000030'           GET                 ; EPOCHS BETWEEN PRINTS
46  00033'041001            STA     0,1,2
47  00034'177777            RETURN
```

```
---
PAGE   2                    *PRINT ROUTINE*
06/25/75
 1   00035'054783  PRINT:  STA     3,P2
 2   00036'031401          LDA     2,1,3   ; INITIAL ADD
 3   00037'035400          LDA     3,0,3   ; CONST ADD
 4   00040'021000          LDA     0,0,2   ; FLAG FOR INITIALISATION
 5   00041'101005          MOV     0,0,SNR ; FINISHED ?
 6   00042'000410          JMP     INIT    ; NO
 7   00043'011001          ISZ     1,2     ; BUMP COUNTER
 8   00044'025401          LDA     1,1,3   ; EPOCHS BETWEEN PRINTS
 9   00045'021001          LDA     0,1,2   ; COUNTER
10   00046'106415          SUB#    0,1,SNR ; PRINT ?
11   00047'000411          JMP     PRT     ; YES
12   00050'034752  RET:    LDA     3,P2
13   00051'001403          JMP     3,3
14
15   00052'011001  INIT:   ISZ     1,2     ; BUMP COUNTER
16   00053'020006          LDA     0,6     ; EPOCHS RUN
17   00054'025400          LDA     1,0,3
18   00055'106414          SUB#    0,1,SZR ; FINISHED ?
19   00056'000771          JMP     RET     ; NO
20   00057'011000          ISZ     0,2     ; BUMP  FLAG
21
22   00060'102400  PRT:    SUB     0,0
23   00061'041001          STA     0,1,2   ; CLEAR COUNTER
24   00062'171000          MOV     3,2
25   00063'020006          LDA     0,6
26   00064'177777          PUT
27   00065'000005          5                       ; PRECISION
28   00066'021002          LDA     0,2,2   ; TLIGHT NO
29   00067'000064'         PUT
30   00070'000003          3
31   00071'034731          LDA     3,P2
32   00072'031402          LDA     2,2,3   ; AD OF TLIGHT
33   00073'125400          SUB     1,1
34   00074'044737          STA     1,P3    ; LEG COUNTER
35   00075'020425  LP1:    LDA     0,SP    ; PUT SPACE
36   00076'177777          PUTC
37   00077'125400          INC     1,1
38   00100'151400          INC     2,2
39   00101'021003          LDA     0,3,2   ; GET QUEUE
40   00102'101110          MOVZL   0,0     ; STRIP BIT 0
41   00103'101220          MOVZR   0,0
42   00104'000067'         PUT
43   00105'000002          2
44   00106'020415          LDA     0,C5
45   00107'122414          SUB#    1,0,SZR ; FINISHED LEG
46   00110'000767          JMP     LP1+2   ; NO
47   00111'010712          ISZ     P3      ; BUMP COUNTER
48   00112'024711          LDA     1,P3
49   00113'125400          INC     1,1
50   00114'122011          ADCZ#   1,0,SZC ; FINISHED ?
51   00115'000403          JMP     .+3     ; NO
52   00116'177777          CRLF
53   00117'000731          JMP     RET
54   00120'125400          SUB     1,1
55   00121'000754          JMP     LP1
56
57   00122'000040  SP:     40
58   00123'000005  C5:     5
```

```
---
PAGE   3                    *PRINT ROUTINE*
05/25/75
 1          000124'           NEXTP=.
 2                            .END
```

```
---
PAGE   1
07/04/75
 1                              .TITL   WRITE
 2                      ; WRITES THE QUEUE LENGTHS AT A TLIGHT ONTO DISK
 3                      ; COPY IS THEN USED TO PRINT THE QUEUES
 4
 5                      ; PROG: WRITE,NO,TLIGHT,NO
 6                      ; DATA: WRITE,NO,EPOCHS BEFORE WRITE,EPOCHS BETWEEN WRITES
 7
 8                      ; OUTPUT TO DISK:
 9                      ;    TLIGHT NO,EPOCH NO,QUEUES
10
11                              .EXTN   RETURN,GETAD,GET,PUTP,PUT,CRLF,PU.C,MES
12                              .EXTN   NONE
13                              .EXTD   DDOS,SP.,XEC
14              000001          .TXTM   1
15                              .HEAD           *WRITE ROUTINE*
16                              .ZREL
17  00000-000014' .WRITE:       WRITE
18  00001-000305' .NAME:        NAME
19  00002-000551' .BUFC.        BUFC
20  00003-000001  .STOR:        .BLK    1
21  00004-000005  C5:           5
22
23                              .NREL
24  00000'000026'               W1                  ; VARIABLE ROUTINE
25  00001'000050                40.                 ; NO OF WRITES
26  00002'000003                3                   ; NO OF VARIABLES
27  00003'000050                40.                 ; NO OF WRITES
28  00004'000002                2                   ; INITIAL STORAGE
29  00005'000127'               NEXTW
30  00006'053522                .TXT    *WRITE*
31  00007'044524
32  00010'042400
33
34  00011'040413                STA     0,W2        ; VARIABLE ADD
35  00012'044412                STA     1,W3        ; INITIAL ADD
36  00013'044003-               STA     1,.STOR
37  00014'177777                GETAD               ; ADD OF TLIGHT
38  00015'044410                STA     1,W4
39  00016'030405                LDA     2,W2
40  00017'055002                STA     3,2,2       ; TLIGHT NO
41  00020'177777                PUTP
42  00021'000025'               W4
43  00022'006000-               JSR     @.WRITE
44  00023'000001  W2:           .BLK    1           ; VAR ADD
45  00024'000001  W3:           .BLK    1           ; INITIAL ADD
46  00025'000001  W4:           .BLK    1           ; TLIGHT ADD
47
48  00026'111000  W1:           MOV     0,2
49  00027'177777                GET                 ; EPOCHS BEFORE WRITE
50  00030'041000                STA     0,0,2
51  00031'000027'               GET                 ; EPOCHS BETWEEN WRITES
52  00032'041001                STA     0,1,2
53  00033'177777                RETURN
```

```
---
PAGE   2                    *WRITE ROUTINE*
07/04/75
 1   00034'054767  WRITE:   STA      3,W2
 2   00035'031401           LDA      2,1,3    ; INITIAL ADD
 3   00036'035400           LDA      3,0,3    ; CONST ADD
 4   00037'021000           LDA      0,0,2    ; FLAG FOR INITIALISATION
 5   00040'101005           MOV      0,0,SNR  ; FINISHED ?
 6   00041'000410           JMP      INIT     ; NO
 7   00042'011001           ISZ      1,2      ; BUMP COUNTER
 8   00043'020401           LDA      1,1,3    ; EPOCHS BETWEEN WRITES
 9   00044'021001           LDA      0,1,2    ; COUNTER
10   00045'106415           SUB#     0,1,SNR  ; WRITE ?
11   00046'000411           JMP      WRT      ; YES
12   00047'034754  RET:      LDA      3,W2
13   00050'001403           JMP      3,3
14
15   00051'011001  INIT:     ISZ      1,2      ; BUMP COUNTER
16   00052'020006           LDA      0,6      ; EPOCHS RUN
17   00053'025400           LDA      1,0,3
18   00054'106414           SUB#     0,1,SZR  ; FINISHED ?
19   00055'000772           JMP      RET      ; NO
20   00056'011000           ISZ      0,2
21
22   00057'102400  WRT:      SUB      0,0
23   00060'041001           STA      0,1,2    ; CLEAR COUNTER
24   00061'032742           LDA      2,@W2    ; CONST ADD
25   00062'021002           LDA      0,2,2    ; TLIGHT NO
26   00063'004420           JSR      DWRIT    ; WRITE TLIGHT NO
27   00064'020006           LDA      0,6      ; NO OF EPOCHS
28   00065'004416           JSR      DWRIT
29   00066'034735           LDA      3,W2
30   00067'031402           LDA      2,2,3    ; ADD OF TLIGHT
31   00070'126400           SUB      1,1
32   00071'125400  LP1:      INC      1,1
33   00072'151400           INC      2,2
34   00073'021001           LDA      0,3,2    ; GET QUEUE
35   00074'101120           MOVZL    0,0      ; STRIP BIT 0
36   00075'101120           MOVZR    0,0
37   00076'004405           JSR      DWRIT
38   00077'020427           LDA      0,C20
39   00100'122414           SUB#     1,0,SZR  ; FINISHED
40   00101'000770           JMP      LP1      ; NO
41   00102'000745           JMP      RET
42
43   00103'040416  DWRIT:    STA      0,DWT
44   00104'044416           STA      1,DWT+1
45   00105'050416           STA      2,DWT+2
46   00106'054416           STA      3,DWT+3
47   00107'020416           LDA      0,C2
48   00110'024410           LDA      1,.DWT
49   00111'125120           MOVZL    1,1
50   00112'030002-          LDA      2,.BUFC
51   00113'034001$          LDA      3,DDOS
52   00114'007407           JSR      @D.MOT,3
53   00115'024405           LDA      1,DWT+1
54   00116'030405           LDA      2,DWT+2
55   00117'002405           JMP      @DWT+3
56
57   00120'000123' .DWT:    DWT
58                          .HEAD    *COPY ROUTINE*
```

```
---
PAGE   3          *COPY ROUTINE*
07/04/75
 1  00121'000004  DWT:     .BLK    4
 2  00125'000002  C2:      2
 3  00126'000024  C20:     20.
 4
 5        000127'          NEXTW = .
 6
 7
 8                         ; COPY ROUTINE
 9                 ; COPPIES THE QUEUE LENGTHS OF A TLIGHT FROM DISK TO
10                 ; THE TELETYPE
11                 ; IT CAN BE RUN AS PART OF THE TERMINAL PROCEDURE OR FROM
12                 ; THE KEYBOARD
13
14                 ; PROG: COPY
15                 ; KEYBOARD: COPY
16
17                         .ZREL
18  00005-000146' .COPY:   COPY
19
20                         .NREL
21  00127'000143'          COPY1
22  00130'000000           0                   ; NO STORAGE
23  00131'000000           0
24  00132'000000           0
25  00133'000000           0
26  00134'001163'          NEXTC
27  00135'041517           .TXT    *COPY*
28  00136'050131
29  00137'000000
30
31  00140'000020'          PUTP
32  00141'000142'          .+1
33  00142'006005-          JSR     @.COPY
34
35  00143'020003# COPY1:   LDA     0,.REC
36  00144'040535           STA     0,CRET
37  00145'000402           JMP     COPY+1
38
39  00146'054533  COPY:    STA     3,CRET
40  00147'034001#          LDA     3,DDOS
41  00150'102400           SUB     0,0
42  00151'024002-          LDA     1,.BUFC
43  00152'030001-          LDA     2,.NAME
44  00153'007400           JSR     @D.CYS,3
45  00154'000007           0.OPN
46  00155'102400           SUB     0,0
47  00156'042001-          STA     0,@.STOR
48  00157'101400           INC     0,0
49  00160'040522           STA     0,CC1
50
51                                         ; SET UP TABLE OF TLIGHT NOS
52  00161'006015  COP1:    JSR     @.READ  ; READ LINE
53  00162'000417           JMP     COP     ; EOF
54  00163'030003-          LDA     2,.STOR
55  00164'020515           LDA     0,CC1
56  00165'143000           ADD     2,0     ; LAST ENTRY
57  00166'024524           LDA     1,SBUF  ; GET FIRST WORD
58  00167'035000  COP2:    LDA     3,0,2   ; GET ENTRY IN TABLE
```

```
---
PAGE   4        *COPY ROUTINE*
07/04/75
 1  00170'166415          SUB#    3,1,SNR ; MATCH ?
 2  00171'000770          JMP     COP1    ; YES
 3  00172'151400          INC     2,2
 4  00173'142414          SUB#    2,0,SZR ; END OF TABLE ?
 5  00174'000771          JMP     COP2    ; NO
 6  00175'010505          ISZ     CC1
 7  00176'045377          STA     1,-1,2  ; STORE IN TABLE
 8  00177'045000          STA     1,0,2   ; CLEAR NEXT LOCATION
 9  00200'000761          JMP     COP1    ; LOOP TILL EOF
10
11  00201'020474  COP:    LDA     0,PAGEL
12  00202'040472          STA     0,LNCNT
13  00203'102400          SUB     0,0
14  00204'040477          STA     0,CC2
15  00205'010475  COP3:   ISZ     CC2
16  00206'020475          LDA     0,CC2
17  00207'024473          LDA     1,CC1
18  00210'122415          SUB#    1,0,SNR ; FINISHED ?
19  00211'000456          JMP     RETC    ; YES RETURN
20  00212'006465          JSR     @.HED
21  00213'034001$         LDA     3,DDOS
22  00214'030002-         LDA     2,.BUFC
23  00215'007400          JSR     @D.CYS,3
24  00216'000014          D.REW
25
26  00217'006457  COP4:   JSR     @.READ
27  00220'000765          JMP     COP3    ; EOF
28  00221'030001-         LDA     2,.STOR ; TABLE ADD
29  00222'020461          LDA     0,CC2   ; DISPLACEMENT
30  00223'113000          ADD     0,2
31  00224'021377          LDA     0,-1,2  ; TABLE ENTRY
32  00225'024463          LDA     1,SBUF
33  00226'122414          SUB#    1,0,SZR ; PRINT ?
34  00227'000770          JMP     COP4    ; NO
35
36  00230'020444  PRINT:  LDA     0,LNCNT
37  00231'024444          LDA     1,PAGEL
38  00232'122415          SUB#    1,0,SNR ; NEW PAGE ?
39  00233'006444          JSR     @.HED   ; YES
40  00234'010440          ISZ     LNCNT
41  00235'030443          LDA     2,.SBUF
42  00236'021001          LDA     0,1,2   ; GET EPOCH
43  00237'177777          PUT
44  00240'000005          5
45  00241'125400          SUB     1,1
46  00242'044441          STA     1,CC3   ; LEG COUNTER
47  00243'030435          LDA     2,.SBUF
48  00244'020002$ COP5:   LDA     0,SP    ; PUT SPACE
49  00245'177777          PUTC
50  00246'125400          INC     1,1
51  00247'151400          INC     2,2
52  00250'021001          LDA     0,1,2   ; GET QUEUE
53  00251'000237'         PUT
54  00252'000002          2
55  00253'020004-         LDA     0,C5
56  00254'122414          SUB#    1,0,SZR ; FINISHED LEG ?
57  00255'000771          JMP     COP5+2  ; NO
58  00256'010426          ISZ     CC3     ; BUMP COUNTER
```

```
---
PAGE   5          *COPY ROUTINE*
07/04/75
 1   00257'024425            LDA     1,CC3
 2   00260'125400            INC     1,1
 3   00261'122033            ADCZ#   1,0,SNC ; FINISHED ?
 4   00262'000403            JMP     .+3      ; YES
 5   00263'126400            SUB     1,1
 6   00264'000760            JMP     COP5
 7   00265'177777            CRLF
 8   00266'000731            JMP     COP4
 9
10   00267'034001$ RETC:     LDA     3,DDOS
11   00270'030002-           LDA     2,.BUFC
12   00271'007400            JSR     @D.CYS,3
13   00272'000010            0.CLO
14   00273'002406            JMP     @CRET
15
16   00274'000001 LNCNT:     .BLK    1
17   00275'000070 PAGEL:     56.
18   00276'000365' .READ:    READ
19   00277'0004     HED:     HEAD
20   00300'0005     SBUF:    SBUF
21   00301'000001 CRET:      .BLK    1
22   00302'000001 CC1:       .BLK    1
23   00303'000001 CC2:       .BLK    1
24   00304'000001 CC3:       .BLK    1
25   00305'027457 NAME:      .TXT    *///TRDAT/*
26   00306'027524
27   00307'051104
28   00310'040524
29   00311'027400
30   00312'000054 SBUF:      .BLK    44.
```

```
---
PAGE   6        *COPY ROUTINE*
07/04/75
 1        000000            .TXTM   0
 2  00365'054414  READ:     STA     3,R1
 3  00367'034001$           LDA     3,DDOS
 4  00370'020410            LDA     0,C44
 5  00371'024707            LDA     1,.SBUF
 6  00372'125120            MOVZL   1,1
 7  00373'030002-           LDA     2,.BUFC
 8  00374'007403            JSR     @D.WIN,3
 9  00375'002405            JMP     @R1       ; EOF
10  00376'010404            ISZ     R1
11  00377'002402            JMP     @R1
12
13  00400'000054  C44:      44.
14  00401'000025  C25:      25
15  00402'000001  R1:       .BLK    1
16
17  00403'054544  HEAD:     STA     3,HRET
18  00404'020670            LDA     0,LNCNT
19  00405'024670            LDA     1,PAGEL
20  00406'106405            SUB     0,1,SNR
21  00407'000405            JMP     .+5
22  00410'044664            STA     1,LNCNT
23  00411'000265'           CRLF
24  00412'014662            DSZ     LNCNT
25  00413'000775            JMP     .-2
26  00414'102400            SUB     0,0       ; CLEAR LINE COUNT
27  00415'040657            STA     0,LNCNT
28  00416'000411'           CRLF
29  00417'000416'           CRLF
30  00420'000417'           CRLF
31  00421'177777            MES
32  00422'026455            .TXT    *---*
33  00423'000055
34  00424'000420'           CRLF
35  00425'000424'           CRLF
36  00426'020002$           LDA     0,SP
37  00427'024752            LDA     1,C25
38  00430'152400            SUB     2,2
39  00431'000245'           PUTC
40  00432'151400            INC     2,2
41  00433'146414            SUB#    2,1,SZR ; FINISHED ?
42  00434'000775            JMP     .-3       ; NO
43  00435'000421'           MES
44  00436'052521            .TXT    *QUEUES AT TRAFFIC LIGHT NO.*
45  00437'052505
46  00440'051505
47  00441'040440
48  00442'020124
49  00443'051124
50  00444'043101
51  00445'044505
52  00446'020103
53  00447'044514
54  00450'044107
55  00451'020124
56  00452'047516
57  00453'000056
58  00454'030003-           LDA     2,.STOR
```

```
---
PAGE   7        *COPY ROUTINE*
07/04/75
 1   00455'020626          LDA     0,CC2
 2   00456'113000          ADD     0,2
 3   00457'021377          LDA     0,-1,2
 4   00460'000251'         PUT             ; TLIGHT NO
 5   00461'000003          3
 6   00462'000425'         CRLF
 7   00463'000462'         CRLF
 8   00464'000435'         MES
 9   00465'042440          .TXT    * EPOCH *
10   00466'047520
11   00467'044103
12   00470'000040
13   00471'126400          SUB     1,1
14   00472'044456          STA     1,HH1
15   00473'010455  H1:     ISZ     HH1
16   00474'000464'         MES
17   00475'020040          .TXT    *     LEG NO*
18   00476'020040
19   00477'046040
20   00500'043505
21   00501'047040
22   00502'000117
23   00503'020445          LDA     0,HH1
24   00504'111000          MOV     0,2
25   00505'000460'         PUT
26   00506'000001          1
27   00507'000474'         MES
28   00510'020040          .TXT    *   *
29   00511'000040
30   00512'030436          LDA     2,HH1
31   00513'024436          LDA     1,C4
32   00514'132414          SUB#    1,2,SZR ; FINISHED
33   00515'000765          JMP     H1      ; NO
34   00516'000463'         CRLF
35   00517'024433          LDA     1,C7
36   00520'044430          STA     1,HH1
37   00521'020002$         LDA     0,JP
38   00522'000431'         PUTC
39   00523'014425          DSZ     HH1
40   00524'000776          JMP     .-2
41   00525'152400          SUB     2,2
42   00526'151400  H2:     INC     2,2
43   00527'141000          MOV     2,0
44   00530'000505'         PUT
45   00531'000002          2
46   00532'024004-         LDA     1,C5
47   00533'132414          SUB#    1,2,SZR ; END OF LEG
48   00534'000772          JMP     H2
49   00535'010411          ISZ     HH1
50   00536'020002$         LDA     0,SP
51   00537'000522'         PUTC
52   00540'020410          LDA     0,HH1
53   00541'034410          LDA     3,C4
54   00542'162414          SUB#    3,0,SZR ; FINISHED ?
55   00543'000762          JMP     H2-1    ; NO
56   00544'000715'         CRLF
57   00545'000544'         CRLF
58   00546'002401          JMP     @HRET
```

```
---
PAGE   8          *COPY ROUTINE*
07/04/75
 1
 2  00547' 000001  HRET:    .BLK    1
 3  00550' 000001  HH1:     .BLK    1
 4  00551' 000004  C4:      4
 5  00552' 000007  C7:      7
 6  00553' 000410  BUFC:    .BLK    264.
 7
 8        001163'           NEXTC = .
```

```
---
PAGE   9          *COPY ROUTINE*
07/04/75
 1                  ; IFILE
 2                  ; CREATES FILE//:TRDAT/ AND OPENS IT
 3
 4         000001           .TXTM   1
 5   01163'177777           NONE                ; NO VARIABLE ROUTINE
 6   01164'000000           0
 7   01165'000000           0
 8   01166'000000           0
 9   01167'000000           0
10   01170'001222'          NEXTI
11   01171'044506           .TXT    *IFILE*
12   01172'044514
13   01173'042400
14
15   01174'034001$          LDA     3,DDOS
16   01175'030001-          LDA     2,.NAME
17   01176'102400           SUB     0,0
18   01177'105000           MOV     0,1
19   01200'007400           JSR     @D.CYS,3
20   01201'000011           0.CRT
21   01202'102400           SUB     0,0
22   01203'024002-          LDA     1,.BUFC
23   01204'030001-          LDA     2,.NAME
24   01205'007400           JSR     @D.CYS,3
25   01206'000007           0.OPN
26   01207'000140'          PUTP
27   01210'001221'          IF1
28   01211'034001$          LDA     3,DDOS
29   01212'020001-          LDA     2,.BUFC
30   01213'007400           JSR     @D.CYS,3
31   01214'000010           0.CLO
32   01215'102400           SUB     0,0
33   01216'024002-          LDA     1,.BUFC
34   01217'030001-          LDA     2,.NAME
35   01220'007400           JSR     @D.CYS,3
36   01221'000007  IF1:     0.OPN
37         001222'          NEXTI=.
38
39                    ; CFILE
40                    ; CLOSES FILE ///TRDAT/
41                    ; CFILE MUST BE CALLED BEFORE COPY
42
43   01222'001163'          NONE
44   01223'000000           0
45   01224'000000           0
46   01225'000000           0
47   01226'000000           0
48   01227'001243'          NEXTF
49   01230'041506           .TXT    *CFILE*
50   01231'044514
51   01232'042400
52
53   01233'001207'          PUTP
54   01234'001242'          CF1
55   01235'034001$          LDA     3,DDOS
56   01236'030002-          LDA     2,.BUFC
57   01237'007400           JSR     @D.CYS,3
58   01240'000013           0.EOF
```

```
---
PAGE   1
06/25/75
 1                              .TITL   IGEN
 2
 3                      ; INITIALISES GENERATORS
 4
 5                              .EXTN RETURN,GETAD,PUTP,GET
 6
 7                              .ZREL
 8  00000-000014' .IGEN:        IGEN
 9
10                              .NREL
11
12  00000'000033'               IG1             ; CONSTANT ROUTINE
13  00001'000000                0               ; NO STORAGE
14  00002'000000                0
15  00003'000000                0
16  00004'000000                0
17  00005'000043'               NEXTG
18        000001                .TXTM   1
19  00006'044507                .TXT    *IGEN*
20  00007'042516
21  00010'000000
22
23  00011'177777                PUTP            ; PROGRAM
24  00012'000013'               .+1
25  00013'006000-               JSR     @.IGEN
26
27  00014'102400   IGEN:        SUB     0,0
28  00015'063076                DOC     0,76    ; SET UP MODE 0
29  00016'060270                NIOC    70      ; CLEAR GEN 2
30  00017'020471                LDA     0,C7    ; MODE 7
31  00020'060070                NIO     0
32  00021'063076                DOC     0,76
33  00022'020413                LDA     0,C77
34  00023'062076                DOB     0,75    ; MASK 6 BITS
35  00024'020416                LDA     0,LP+1
36  00025'040414                STA     0,LP
37  00026'060270                NIOP    70      ; LOOP TO SHIFT
38  00027'060376                NIOP    76      ; INITIAL CONDITIONS
39  00030'014411                DSZ     LP
40  00031'000775                JMP     .-3
41  00032'001400                JMP     0,3
42
43  00033'177777   IG1:         GET             ; NO.  OF SHIFTS
44  00034'101400                INC     0,0
45  00035'040405                STA     0,LP+1
46  00036'177777                RETURN
47
48  00037'000077   C77:         77
49  00040'000007   C7:          7
50  00041'000001   LP:          .BLK    1
51  00042'000001                1
52
53        000043'               NEXTG=.
54                              .END
```

```
---
PAGE   1
06/25/75
 1
 2                      .EXTN NONE
 3                      .NREL
 4              000001  .TXTM 1
 5  00000' 177777       NONE
 6  00001' 000000       0
 7  00002' 000000       0
 8  00003' 000000       0
 9  00004' 000000       0
10  00005' 000114'      NEXT
11  00006' 047117       .TXT *NORMAL*
12  00007' 051115
13  00010' 040514
14  00011' 000000
15  00012' 020011       020011
16  00013' 017411       017411
17  00014' 017010       017010
18  00015' 016407       016407
19  00016' 016307       016307
20  00017' 015746       015746
21  00020' 015005       015005
22  00021' 020205       020205
23  00022' 023444       023444
24  00023' 020644       020644
25  00024' 021043       021043
26  00025' 021641       021641
27  00026' 024022       024022
28  00027' 022202       022202
29  00030' 024401       024401
30  00031' 022521       022521
31  00032' 025261       025261
32  00033' 023020       023020
33  00034' 025540       025540
34  00035' 017660       017660
35  00036' 012017       012017
36  00037' 020220       020220
37  00040' 023620       023620
38  00041' 017300       017300
39  00042' 014017       014017
40  00043' 020360       020360
41  00044' 021260       021260
42  00045' 021700       021700
43  00046' 012340       012340
44  00047' 016417       016417
45  00050  017017       017017
46  00051' 014340       014340
47  00052' 023760       023760
48  00053' 017720       017720
49  00054' 021017       021017
50  00055' 017360       017360
51  00056' 026360       026360
52  00057' 021360       021360
53  00060' 022360       022360
54  00061' 021417       021417
55  00062' 026340       026340
56  00063' 016740       016740
57  00064' 024300       024300
58  00065' 014720       014720
```

```
---
PAGE   2
06/25/75
 1   00066' 026017  026017
 2   00067' 014660  014660
 3   00070' 020540  020540
 4   00071' 012420  012420
 5   00072' 015261  015261
 6   00073' 013121  013121
 7   00074' 015401  015401
 8   00075' 013602  013502
 9   00076' 016022  016022
10   00077' 014243  014243
11   00100' 016443  016443
12   00101' 017244  017244
13   00102' 017444  017444
14   00103' 014605  014605
15   00104' 023005  023005
16   00105' 022546  022546
17   00106' 022307  022307
18   00107' 021407  021407
19   00110' 021010  021010
20   00111' 020411  020411
21   00112' 000040  000040
22   00113' 000010  000010
23          000114' NEXT =
24                  .END
```

APPENDIX 13

SIMULATION ROUTINE STORAGE MAPS

Table A13.1. Constant Storage Map for ELINK

Word No. (OCTAL)	CONTENTS
0	Flow Rate Lane 1
1	Flow Rate Lane 2
2	Flow Rate Lane 3
3	Flow Rate Lane 4
4	Flow Rate Lane 5

Table A13.2. Initial Storage Map for ELINK and LINK

Word No. (OCTAL)	Contents		
0	OUTPUT	LANE 1	
1	OUTPUT	LANE 2	
2	OUTPUT	LANE 3	
3	OUTPUT	LANE 4	
4	OUTPUT	LANE 5	
5	WORD 1	LANE 1	(START OF STREET)
6	WORD 2	LANE 1	
7	WORD 3	LANE 1	
10	WORD 4	LANE 1	(END OF STREET)
11	WORD 1	LANE 2	
14	WORD 4	LANE 2	
15	WORD 1	LANE 3	
20	WORD 4	LANE 3	
21	WORD 1	LANE 4	
24	WORD 4	LANE 4	
25	WORD 1	LANE 5	
30	WORD 4	LANE 5	

Table A13.3. TLIGHT Initial Storage Map

Word No. (OCTAL)	Function
0	OUTPUT FROM LEG 1
1	OUTPUT FROM LEG 2
2	OUTPUT FROM LEG 3
3	OUTPUT FROM LEG 4
4	QUEUE 1 LEG 1
5	QUEUE 2 LEG 1
6	QUEUE 3 LEG 1
7	QUEUE 4 LEG 1
10	QUEUE 5 LEG 1
11	QUEUE 1 LEG 2
12	QUEUE 2 LEG 2
13	QUEUE 3 LEG 2
14	QUEUE 4 LEG 2
15	QUEUE 5 LEG 2
16	QUEUE 1 LEG 3
17	QUEUE 2 LEG 3
20	QUEUE 3 LEG 3
21	QUEUE 4 LEG 3
22	QUEUE 5 LEG 3
23	QUEUE 1 LEG 4
24	QUEUE 2 LEG 4
25	QUEUE 3 LEG 4
26	QUEUE 4 LEG 4
27	QUEUE 5 LEG 4
30	SIGNAL WORD
31	SIGNAL COUNTER
32	OFFSET FLAG

Table A13.3. TLIGHT Constant Storage Map

Word No. (OCTAL)	Function		
0	Saturation Flow leg 1		
11	Saturation Flow leg 2		
2	Saturation Flow leg 3		
3	Saturation Flow leg 4		
4	LEFT TURN CONST.	QUEUE 1	LEG 1
5	LEFT TURN CONST.	QUEUE 2	LEG 1
6	LEFT TURN CONST.	QUEUE 3	LEG 1
7	LEFT TURN CONST.	QUEUE 4	LEG 1
10	LEFT TURN CONST.	QUEUE 5	LEG 1
11	LEFT TURN CONST.	QUEUE 1	LEG 2
⋮	⋮	⋮	⋮
16	LEFT TURN CONST.	QUEUE 1	LEG 3
⋮	⋮	⋮	⋮
23	LEFT TURN CONST.	QUEUE 1	LEG 4
⋮	⋮	⋮	⋮
30	RIGHT TURN CONST.	QUEUE 1	LEG 1
31	RIGHT TURN CONST.	QUEUE 2	LEG 1
32	RIGHT TURN CONST.	QUEUE 3	LEG 1
33	RIGHT TURN CONST.	QUEUE 4	LEG 1
34	RIGHT TURN CONST.	QUEUE 5	LEG 1
35	RIGHT TURN CONST.	QUEUE 1	LEG 2
⋮	⋮	⋮	⋮
41	RIGHT TURN CONST.	QUEUE 1	LEG 3
⋮	⋮	⋮	⋮
46	RIGHT TURN CONST.	QUEUE 1	LEG 4
⋮	⋮	⋮	⋮
54	ACCEPTANCE GAP DISTRIBUTION ADDRESS		
55	ACCEPTANCE GAP DISTRIBUTION MEAN CONSTANT		
56	ACCEPTANCE GAP DISTRIBUTION STD. DEV. CONSTANT		
57	OFFSET PERIOD LENGTH		
60	PHASE 1 - 3 GREEN PERIOD LENGTH		
61	PHASE 1 - 3 AMBER PERIOD LENGTH		
62	PHASE 1 - 3 DEAD PERIOD LENGTH		
63	PHASE 2 - 4 GREEN PERIOD LENGTH		
64	PHASE 2 - 4 AMBER PERIOD LENGTH		
65	PHASE 1 - 3 DEAD PERIOD LENGTH		

APPENDIX 14

PROGRAMS FOR GENERATING CONSTANTS FOR ARBITRARY DISTRIBUTIONS

BASIC Program DISTB

```
 10 REM USER ROUTINE MUST START AT LINE 270
 20 REM AND END WITH A RETURN
 30 REM THE NUMBER IS PASSED IN N  AND THE RELATIVE FREQUENCY
 40 REM MUST BE PLACED IN R
 50 REM N AND A9 MUST NOT BE MODIFIED BY USER ROUTINE
 60 PRINT "NO. OF NUMBERS IN SET = ";
 70 INPUT A9
 80 PRINT
 90 PRINT "TURN ON PUNCH AND HIT ESCAPE"
100 ESCAPE 1
110 IF ESC(3) THEN 120
120 GOTO 110
130 FOR N=1 TO A9
140   GOSUB 270
150   LET R=R*100000
160   IF (N/5-INT(N/5))<>0 THEN 230
170   IF ESC(0) THEN 190
180   GOTO 170
190   PRINT
200   PRINT "PUNCH ON"
210   IF ESC(0) THEN 210
220   GOTO 210
230   PRINT  USING "######", R;
240   PRINT " ";
250 NEXT N
260 END
270 LET N1=(N-1)/9
280 LET P=3.0R(3*ATN(1))
290 LET R=EXP(-N1*N1/2)/P*.125
300 RETURN
```

```
---
PAGE   1
06/12/75
 1                              .TITL   DIST
 2                      ; PROGRAM TO COMPUTE THE F AND N NUMBERS
 3                      ; FOR THE RANDOM NUMBER GENERATOR
 4                      ; OUTPUT IS TO DISK IN THE FORM
 5                      ; OF SOURCE PROGRAM WITH FORMAT FOR FASP
 6
 7                              .EXTN   .DSUB,.DADD,.DDIV,.BIND,.SIND
 8                              .EXTN   .DBIN,.DDB
 9              000001          .TXTM   1
10                              .HEAD           *DIST PROGRAM*
11
12              000002  .LOC    2
13  00002 002004                JMP     @.+2
14  00003 002005                JMP     @.+2
15  00004 000000'               STRT
16  00005 000000'               STRT1
17
18              000040  .LOC    40
19  00040 000545' READ:  .GETC
20  00041 000555' PRINT: .PUTC
21  00042 000536' DSK:   .DSK
22  00043 000555' PUTC:  .PUTC
23  00044 000001 TEXT:   .BLK    1
24  00045 001014' RTN1:  .RTN1
25  00046 000001 RET1:   .BLK    1
26  00047 100000 IND:    @0
27
28                              .ZREL
29
30  00000-177777 DSUB:   .DSUB
31  00001-177777 DADD:   .DADD
32  00002-177777 DDIV:   .DDIV
33  00003-177777 BIND:   .BIND
34  00004-177777 SIND:   .SIND
35  00005-177777 DDB:    .DDB
36  00006-177777 DBIN:   .DBIN
37  00007-000015 CR:     15
38  00010-000012 LF      12
39  00011-077400 MSK:    77400
40  00012-000001 DCOS:   .BLK    1
41  00013-000015 .DTX1:  D.TX1
42  00014-000001 BUFP:   .BLK    1
43  00015-000024= ENAME:  2*.ENAME
44  00016-015215   ENAME. "**400-15
45  00017-000742' NAME:   .NAME
46  00020-001714" PROG:   2*.PROG
47  00021-002010" EPROG:  2*.EPROG
48  00022-000001 CNT1:   .BLK    1
49  00023-000032 CNT2:   .EPROG-.PROG
50  00024-000010 CNT3:   .RTN1-.EPROG
51  00025-000177 C177:   177
52  00026-000001 C105:   1
53  00027-103240         103240
54  00030-001145" STM53:  2*.ST53
55  00031-001405" STM54:  2*.ST54
56  00032-001440" STM55:  2*.ST55
57  00033-001472" STM56:  2*.ST56
58  00034-001514" STM57:  2*.ST57
```

```
---
PAGE   2                    *DIST PROGRAM*
06/12/75
 1   00035-001ECE" STM58:   2*.ST58
 2   00036-001540" STM59:   2*.ST59
 3   00037-001544" STM60:   2*.ST60
 4   00040-001576" STM61:   2*.ST61
 5   00041-001620" STM62:   2*.ST62
 6   00042-001552" STM63:   2*.ST63
 7   00043-000001  MEAN:    .BLK    1
 8   00044-000001  STD:     .BLK    1
 9   00045-001011' N:       .N
10   00046-001023' F:       .F
11   00047-001024' RF:      .RF
12   00050-000057- U:       .G10
13   00051-000075- XN:      .XN
14   00052-000100- XP;      .XP
15
16   00053-000000  .F10:    0
17   00054-000000  .F11:    0
18   00055-000000  .F12:    0
19   00056-000000  .F13:    0
20   00057-000001  .G10:    .BLK    2
21   00061-000004  .H10:    .BLK    4
22   00065-000004  .H11:    .BLK    4
23   00071-000000  C0:      0
24   00072-000001  C1:      1
25   00073-000002  C2:      2
26   00074-000017  C17:     17
27   00075-000002  .XN:     .BLK    2
28   00077-000001  .YN:     .BLK    1
29   00100-000002  .XP:     .BLK    2
30   00102-000001  .YP:     .BLK    1
31   00103-170000  MSK17:   170000
32   00104-000065- A01:    .H11
```

```
---
PAGE   3                        *DIST PROGRAM*
06/12/75
 1                                      .NREL
 2                              ; START OF PROGRAM
 3
 4  00000' 0E4012- STRT:    STA     3,DDOS
 5  00001' 024047           LDA     1,IND
 6  00002' 137020           ADC     1,3
 7  00003' 024013-          LDA     1,.DTX1
 8  00004' 167000           ADD     3,1
 9  00005' 044044           STA     1,TEXT
10
11  00006' 00E04E  STRT1:   JSR     @RTN1
12  00007' 020037-          LDA     2,STM60
13  00010' 00E044           JSR     @TEXT
14  00011' 030017-          LDA     2,NAME
15  00012' 1E1120           MOVZL   2,2
16  00013' 050022-          STA     2,CNT1
17  00014' 00E040  LP1:     JSR     @READ
18  00015' 024007-          LDA     1,CR
19  00016' 10E41E           SUB#    0,1,SNR
20  00017' 000414           JMP     FIN
21  00020' 1E3122           MOVZR   2,3,SZC
22  00021' 000407           JMP     LP2
23  00022' 024011-          LDA     1,MSK
24  00023' 101300           MOVS    0,0
25  00024' 123400           AND     1,0
26  00025' 041400           STA     0,0,3
27  00026' 151400           INC     2,2
28  00027' 000765           JMP     LP1
29  00030' 025400  LP2:     LDA     1,0,3
30  00031' 123000           ADD     1,0
31  00032' 000773           JMP     .-5
32
33  00033' 020011- FIN:     LDA     0,CNT1
34  00034' 145000           MOV     2,1
35  00035' 106400           SUB     0,1
36  00036' 044022-          STA     1,CNT1
37  00037' 102400           SUB     0,0
38  00040' 155223           MOVZR   2,3,SNC
39  00041' 000404           JMP     .+4
40  00042' 025400           LDA     1,0,3
41  00043' 020011-          LDA     0,MSK
42  00044' 123400           AND     1,0
43  00045' 041400           STA     0,0,3
44  00046' 005045           JSR     @RTN1
45  00047' 0E4012-          LDA     3,DDOS
46  00050' 030017-          LDA     2,NAME
47  00051' 102400           SUB     0,0     ; PROPS
48  00052' 024073-          LDA     1,C2    ; ASCII
49  00053' 007400           JSR     @0.CYS,3
50  00054' 000011           0.CRT
51  00055' 102400           SUB     0,0
52  00056' 105240           MOVOR   0,1
53  00057' 030017-          LDA     2,NAME
54  00060' 007400           JSR     @0.CYS,3
55  00061' 000007           0.OPN
56  00062' 020023-          LDA     0,CNT2
57  00063' 101110           MOVZL   0,0
58  00064' 024020-          LDA     1,PROG
```

```
---
PAGE   4                        *DIST PROGRAM*
05/12/75
 1   00065' 007407                JSR     @D.MOT,3
 2   00066' 024017-               LDA     1,NAME
 3   00067' 12E120                MOVZL   1,1
 4   00070' 020022-               LDA     0,CNT1
 5   00071' 007407                JSR     @D.MOT,3
 6   00072' 024015-               LDA     1,ENAME
 7   00073' 020001-               LDA     0,C2
 8   00074' 007407                JSR     @D.MOT,3
 9   00075' 050014-               STA     2,BUFP
10
11   00076' 006045                JSR     @RTN1
12   00077' 030042-               LDA     2,STM63
13   00100' 006044                JSR     @TEXT
14   00101' 005006-               JSR     @DBIN
15   00102' 044041-               STA     1,MEAN
16   00103' 005045                JSR     @RTN1
17   00104' 030041-               LDA     2,STM62
18   00105' 006044                JSR     @TEXT
19   00106' 005006-               JSR     @DBIN
20   00107' 044044-               STA     1,STD
```

```
---
PAGE   5                *DIST PROGRAM*
05/12/75
 1   00110'006047'        .D00:   JSR @RTN1       ;NO. OF MEMBERS IN SET7
 2   00111'030030-                LDA 2,STM53
 3   00112'003044                 JSR @TEXT
 4   00113'005005-                JSR @DDB
 5   00114'062212         NIOC    TTI
 6   00115'050053-                STA 2,.F10      ;STORE AT .F10
 7   00116'151210                 MOVZR 2,2
 8
 9   00117'101410         .D01:   SUBZR 0,0       ;COMPUTE U
10   00120'151222                 MOVZR 2,2 SZC
11   00121'000401                 JMP .D010
12   00122'101220                 MOVZR 0,0
13   00123'000775                 JMP .D01+1
14
15   00124'040062-        .D010:  STA 0,.G10+1    ;STORE U
16   00125'024071-                LDA 1,C0
17   00126'044057-                STA 1,.G10
18
19
20   00127'034045-        .D04:   LDA 3,N         ;COMPUTE STORAGE AREAS
21   00130'020053-                LDA 0,.F10
22   00131'117000                 ADD 0,3
23   00132'054046-                STA 3,F
24   00133'101120                 MOVZL 0,0
25   00134'117000                 ADD 0,3
26   00135'054047-                STA 3,RF
27
28   00136'020063         .D041:  LDA 0,.F10      ;INITIALISE N AND F
29   00137'040061-                STA 0,.H10
30
31   00140'020071-                LDA 0,C0
32   00141'105000                 MOV 0,1
33   00142'030046-                LDA 2,F
34   00143'034045-                LDA 3,N
35
36   00144'045400         .D042:  STA 1,0,3
37   00145'041000                 STA 0,0,2
38   00146'041001                 STA 0,1,2
39
40   00147'125400                 INC 1,1
41   00150'175400                 INC 3,3
42   00151'121400                 INC 2,2
43   00152'121400                 INC 2,2
44
45   00153'014061-                DSZ .H10
46   00154'000770                 JMP .D042
```

```
---
PAGE   6                    *DIST PROGRAM*
06/12/75
 1
 2
 3  00155' 006045          .D05:   JSR @RTN1       ;OUTPUT TABLE HEADING
 4  00156' 030031-                 LDA 2,STM54
 5  00157' 006044                  JSR @TEXT
 6  00160' 006045                  JSR @RTN1
 7  00161' 030031-                 LDA 2,STM55
 8  00162' 006044                  JSR @TEXT
 9  00163' 006045                  JSR @RTN1
10  00164' 030033-                 LDA 2,STM56
11  00165' 006044                  JSR @TEXT
12  00166' 006045                  JSR @RTN1
13  00167' 030031-                 LDA 2,STM54
14  00170' 006044                  JSR @TEXT
15  00171' 006045                  JSR @RTN1
16
17  00172' 020035-         .D06:   LDA 0,.F10      ;LOOP TO INPUT RELATIVE FREQUE
NCIES
18  00173' 024071-                 LDA 1,C0
19  00174' 030047-                 LDA 2,RF
20
21  00175' 040041-                 STA 0,.H10
22  00176' 044062-                 STA 1,.H10+1
23  00177' 050063-                 STA 2,.H10+2
24  00200' 060110          NIOS    TTI
25
26  00201' 030034-         .D07:   LDA 2,STM57
27  00202' 006044                  JSR @TEXT
28  00203' 024063-                 LDA 1,.H10+1
29  00204' 006003-                 JSR @BIND
30
31  00205' 030035-                 LDA 2,STM58
32  00206' 006044                  JSR @TEXT
33  00207' 006037-                 JSR @DCB
34  00210' 044056-                 STA 1,.H11+1
35  00211' 050057-                 STA 2,.H11+2
36  00212' 020071-                 LDA 0,C0
37  00213' 040060-                 STA 0,.H11
38  00214' 040070-                 STA 0,.H11+3
39
40  00215' 020036-                 LDA 0,C105      ;DIVIDE BY 10**5
41  00216' 024027-                 LDA 1,C105+1
42  00217' 030104-                 LDA 2,A0I1
43  00220' 006002-                 JSR @DDIV
44  00221' 030063-                 LDA 3,.H10+2
45
46  00222' 041400          .D08:   STA 0,0,3
47  00223' 045401                  STA 1,1,3
48
49  00224' 030032-                 LDA 2,STM59
50  00225' 006044                  JSR @TEXT
51  00226' 006045                  JSR @RTN1
```

```
PAGE   7                    *DIST PROGRAM*
06/12/75
 1
 2  00227'010061-          .D09:   ISZ .H10+1          ;INC. NUMBER COUNT
 3
 4  00230'020061-                  LDA 0,.H10+2
 5  00231'024073-                  LDA 1,C2
 6  00232'123000                   ADD 1,0
 7  00233'040063-                  STA 0,.H10+2        ;ADDRESS COUNT INCREMENTED
 8
 9  00234'014061-                  DSZ .H10
10  00235'000744                   JMP .D07
11
12  00236'030031-                  LDA 2,STMT4         ;RULE OFF TABLE
13  00237'005044                   JSR @TEXT
14  00240'006045                   JSR @RTN1
15  00241'034042           LDA     3,DSK
16  00242'054041           STA     3,PRINT
17
18  00243'020061-          .D10:   LDA 0,.F10          ;LOOP TO COMPUTE RF=RF-U
19  00244'024047-                  LDA 1,RF
20  00245'040061-                  STA 0,.H10
21  00246'044062-                  STA 1,.H10+1
22
23  00247'034061-          .D11:   LDA 3,.H10+1        ;LOAD RF
24  00250'021400                   LDA 0,0,3
25  00251'025401                   LDA 1,1,3
26  00252'005000-                  JSR @DSUB
27  00253'000057-                  .G10                ;RF-U
28  00254'034062-                  LDA 3,.H10+1
29  00255'041400                   STA 0,0,3
30  00256'045401                   STA 1,1,3
31
32  00257'024073-          .D12:   LDA 1,C2            ;INCREMENT RF ADDRESS
33  00260'137000                   ADD 1,3
34  00261'054061-                  STA 3,.H10+1
35
36  00262'014061-                  DSZ .H10
37  00263'000754                   JMP .D11
38
39  00264'020061-          .D13:   LDA 0,.F10          ;OUTER LOOP TO GET N AND F
40  00265'040061-                  STA 0,.H10
41
42  00266'020071-          .D14:   LDA 0,C0            ;XN AND XP TO 0
43  00267'040075-                  STA 0,.XN
44  00270'040076-                  STA 0,.XN+1
45  00271'040100-                  STA 0,.XP
46  00272'040101-                  STA 0,.XP+1
47
48  00273'020061-          .D15:   LDA 0,.F10          ;INNER LOOP
49  00274'024047-                  LDA 1,RF
50  00275'021071-                  LDA 2,C0
51
52  00276'040061-                  STA 0,.H10+1
53  00277'044063-                  STA 1,.H10+2
54  00300'050064-                  STA 2,.H10+3
```

```
PAGE   7                    *DIST PROGRAM*
06/12/75
 1
 2  00227'010061-          .D09:   ISZ .H10+1          ;INC. NUMBER COUNT
 3
 4  00230'020061-                  LDA 0,.H10+2
 5  00231'024073-                  LDA 1,C2
 6  00232'123000                   ADD 1,0
 7  00233'040063-                  STA 0,.H10+2        ;ADDRESS COUNT INCREMENTED
 8
 9  00234'014061-                  DSZ .H10
10  00235'000744                   JMP .D07
11
12  00236'030031-                  LDA 2,STMT4         ;RULE OFF TABLE
13  00237'006044                   JSR @TEXT
14  00240'006045                   JSR @RTN1
15  00241'034042           LDA     3,DSK
16  00242'054041           STA     3,PRINT
17
18  00243'020061-          .D10:   LDA 0,.F10          ;LOOP TO COMPUTE RF=RF-U
19  00244'024047-                  LDA 1,RF
20  00245'040061-                  STA 0,.H10
21  00246'044062-                  STA 1,.H10+1
22
23  00247'034061-          .D11:   LDA 3,.H10+1        ;LOAD RF
24  00250'021400                   LDA 0,0,3
25  00251'022401                   LDA 1,1,3
26  00252'005000-                  JSR @DSUB
27  00253'000057-                  .G10                ;RF-U
28  00254'034062-                  LDA 3,.H10+1
29  00255'041400                   STA 0,0,3
30  00256'045401                   STA 1,1,3
31
32  00257'024073-          .D12:   LDA 1,C2            ;INCREMENT RF ADDRESS
33  00260'137000                   ADD 1,3
34  00261'054061-                  STA 3,.H10+1
35
36  00262'014061-                  DSZ .H10
37  00263'000764                   JMP .D11
38
39  00264'020061-          .D13:   LDA 0,.F10          ;OUTER LOOP TO GET N AND F
40  00265'040061-                  STA 0,.H10
41
42  00266'020071-          .D14:   LDA 0,C0            ;XN AND XP TO 0
43  00267'040075-                  STA 0,.XN
44  00270'040076-                  STA 0,.XN+1
45  00271'040100-                  STA 0,.XP
46  00272'040101-                  STA 0,.XP+1
47
48  00273'020061-          .D15:   LDA 0,.F10          ;INNER LOOP
49  00274'024047-                  LDA 1,RF
50  00275'030071-                  LDA 2,C0
51
52  00276'040062-                  STA 0,.H10+1
53  00277'044063-                  STA 1,.H10+2
54  00300'050064-                  STA 2,.H10+3
```

```
---                                  *DIST PROGRAM*
PAGE   8
06/12/75
 1
 2  00301'014400-          .D16:   LDA 3,.H10+2    ;COMPUTE RF-.XN
 3  00302'021400                   LDA 0,0,3
 4  00303'025401                   LDA 1,1,3
 5  00304'006000-                  JSR @DSUB
 6  00305'000073-                  .XN
 7  00306'101113           .D17:   MOVL# 0,0 SNC   ;GREATER THAN OR EQUAL TO 0 ?
 8  00307'000411                   JMP .D18        ;+VE - JUMP
 9  00310'034063-          .D19:   LDA 3,.H10+2
10  00311'021400                   LDA 0,0,3
11  00312'025401                   LDA 1,1,3
12  00313'040073-                  STA 0,.XN
13  00314'044076-                  STA 1,.XN+1
14  00315'030064-                  LDA 2,.H10+3
15  00316'050077-                  STA 2,.YN
16  00317'000421                   JMP .D22
17
18  00320'014063-          .D18:   LDA 3,.H10+2    ;COMPUTE RF-XP
19  00321'021400                   LDA 0,0,3
20  00322'025401                   LDA 1,1,3
21  00323'006000-                  JSR @DSUB
22  00324'000100-                  .XP
23  00325'101112                   MOVL# 0,0 SZC   ;GREATER  THAN 0?
24  00326'000414                   JMP .D22        ;JUMP - RESULT -VE
25  00327'101014                   MOV# 0,0 SZR    ;GREATER THAN OR = TO 0?
26  00330'000431                   JMP .D21        ;RESULT > 0
27  00331'125015                   MOV# 1,1 SNR    ;
28  00332'000410                   JMP .D22        ;RESULT 0
29
30  00333'014063-          .D21:   LDA 3,.H10+2    ;XP=RF
31  00334'021400                   LDA 0,0,3
32  00335'025401                   LDA 1,1,3
33  00336'040100-                  STA 0,.XP
34  00337'044101-                  STA 1,.XP+1
35  00340'030064-                  LDA 2,.H10+3    ;YP=J
36  00341'050101-                  STA 2,.YP
37
38  00342'010064-          .D22:   ISZ .H10+3      ;INC. COUNT
39
40  00343'014063-                  LDA 3,.H10+2    ;INC. ADDRESS
41  00344'030072-                  LDA 2,C2
42  00345'157000                   ADD 2,1
43  00346'054063-                  STA 3,.H10+2
44
45  00347'014061-                  DSZ .H10+1
46  00350'000731                   JMP .D16
47
48  00351'020075-                  LDA 0,.XN
49  00352'101005                   MOV 0,0 SNR
50  00353'000431                   JMP .D24
51  00354'020101-                  LDA 0,.XP+1
52  00355'101002                   MOV 0,0 SNR
53  00356'000447                   JMP .D24
```

```
---
PAGE   9                    *DIST PROGRAM*
06/12/75
 1
 2  00357'020077-      .D23:   LDA 0,.YN       ;RF(YN)=0
 3  00360'101120               MOVZL 0,0
 4  00361'034047-              LDA 3,RF
 5  00362'117000               ADD 0,3
 6  00363'024071-              LDA 1,C0
 7  00364'045400               STA 1,0,3
 8  00365'045401               STA 1,1,3
 9
10  00366'020075-      .D231:  LDA 0,.XN       ;RF(YP)=XN+XP
11  00367'024076-              LDA 1,.XN+1
12  00370'006001-              JSR @DADD
13  00371'000100-              .XP
14  00372'020102-              LDA 2,.YP       ;XN+XP
15  00373'151120               MOVZL 2,2
16  00374'034047-              LDA 3,RF
17  00375'157000               ADD 2,3
18  00376'041400               STA 0,0,3
19  00377'045401               STA 1,1,3
20
21  00400'020077-      .D232:  LDA 0,.YN       ;N(YN)=YP
22  00401'034045-              LDA 3,N
23  00402'117000               ADD 0,3
24  00403'024102-              LDA 1,.YP
25  00404'041400               STA 1,0,3
26
27  00405'020075-      .D234:  LDA 0,.XN       ;F(YN)=U+XN
28  00406'024076-              LDA 1,.XN+1
29  00407'006001-              JSR @DADD
30  00410'000057-              .G10
31  00411'034046-              LDA 3,F
32  00412'030077-              LDA 2,.YN
33  00413'151120               MOVZL 2,2
34  00414'157000               ADD 2,3
35
36  00415'020053-      .D235:  LDA 2,.F10      ;MULTIPLY BY 1/U
37
38  00416'151221       .D236:  MOVZR 2,2 SZC
39  00417'000404               JMP .D237
40  00420'125120               MOVZL 1,1
41  00421'101100               MOVL 0,0
42  00422'000774               JMP .D236
43
44  00423'041400       .D237:  STA 0,0,3
45  00424'045401               STA 1,1,3
46
47  00425'014051-      .D24:   DSZ .H10
48  00426'000640               JMP .D14
49
50  00427'020053-      .D25:   LDA 0,.F10      ;LOOP TO OBTAIN F IN F POINT F
ORM
51  00430'024046-              LDA 1,F
52  00431'030071-              LDA 2,C0
53
54  00432'040051-              STA 0,.H10
55  00433'044052-              STA 1,.H10+1
56  00434'050053-              STA 2,.H10+2
57
```

PAGE 10
06/12/75

```
*DIST PROGRAM*

 1  00435'151400          .D26:   .SUB 2,2          ;CLEAR ACC. 2
 2  00436'034062-                 LDA 3,.H10+1      ;GET F VALUE
 3  00437'025401                  LDA 1,1,3
 4
 5  00440'014074-                 LDA 3,C17
 6  00441'054064-                 STA 3,.H10+3
 7
 8  00442'000401          .D27.   JMP .+1           ;LEFT SHIFT ACC. 1
 9  00443'125122          .D271:  MOVZL 1,1 SZC
10  00444'000404                  JMP .D29          ;1 IN CARRY BIT
11
12  00445'151400          .D28:   INC 2,2
13  00446'014064-                 DSZ .H10+3
14  00447'000771                  JMP .D27
15
16  00450'024161-         .D29:   LDA 0,MSK17       ;GET MASK
17  00451'123700                  ANDS 1,0          ;RESULT IS M
18
19  00452'100000          .D30:   COM 0,0
20  00453'150000                  COM 2,2
21  00454'143400                  AND 2,0
22
23  00455'014045-         .D31:   LDA 3,N           ;GET N NUMBER
24  00456'024067                  LDA 1,.H10+2
25  00457'137000                  ADD 1,3
26  00460'025400                  LDA 1,0,3         ;N IN ACC. 1
27  00461'125100                  MOVS 1,1
28
29  00462'134000          .D32:   COM 1,1           ;OR N AND F
30  00463'123400                  AND 1,0
31  00464'100000                  COM 0,0
32
33  00465'135000                  MOV 0,1
34  00466'006004-                 JSR @BINO
35  00467'024057-                 LDA 0,CR
36  00470'004446                  JSR .DSK
37  00471'010051-         .D35:   ISZ .H10+2
38  00472'030062-                 LDA 2,.H10+1
39  00473'024071-                 LDA 1,C2
40  00474'133000                  ADD 1,2
41  00475'050051-                 STA 3,.H10+1
42  00476'014061-                 DSZ .H10
43  00477'000715                  JMP .D26
44
```

```
---                                  *DIST PROGRAM*
PAGE  10
06/12/75
 1  00435'151400            .D26:   .SUB 2,2          ;CLEAR ACC. 2
 2  00436'034062-                   LDA 3,.H10+1      ;GET F VALUE
 3  00437'025401                    LDA 1,1,3
 4
 5  00440'014074-                   LDA 3,C17
 6  00441'054064-                   STA 3,.H10+3
 7
 8  00442'000401            .D27:   JMP .+1           ;LEFT SHIFT ACC. 1
 9  00443'125122            .D271:  MOVZL 1,1 SZC
10  00444'000404                    JMP .D29          ;1 IN CARRY BIT
11
12  00445'151400            .D28:   INC 2,2
13  00446'014064-                   DSZ .H10+3
14  00447'000771                    JMP .D27
15
16  00450'020163-           .D29:   LDA 0,MSK17       ;GET MASK
17  00451'123700                    ANDS 1,0          ;RESULT IS M
18
19  00452'100000            .D30:   COM 0,0
20  00453'150000                    COM 2,2
21  00454'143400                    AND 2,0
22
23  00455'014045-           .D31:   LDA 3,N           ;GET N NUMBER
24  00456'024063-                   LDA 1,.H10+2
25  00457'137000                    ADD 1,3
26  00460'025400                    LDA 1,0,3         ;N IN ACC. 1
27  00461'125100                    MOVS 1,1
28
29  00462'124000            .D32:   COM 1,1           ;OR N AND F
30  00463'123400                    AND 1,0
31  00464'100000                    COM 0,0
32
33  00465'105000                    MOV 0,1
34  00466'006004-                   JSR @BIN0
35  00467'020007-                   LDA 0,CR
36  00470'004446                    JSR .DSK
37  00471'010061-           .D35:   ISZ .H10+2
38  00472'030062-                   LDA 2,.H10+1
39  00473'020077-                   LDA 1,C2
40  00474'133000                    ADD 1,2
41  00475'050061-                   STA 2,.H10+1
42  00476'014061-                   DSZ .H10
43  00477'000715                    JMP .D26
44
```

```
---
PAGE  11                        *DIST PROGRAM*
05/12/75
 1   00500' 024041-              LDA     1,MEAN
 2   00501' 006004-              JSR     @BINO
 3   00502' 020007-              LDA     0,CR
 4   00503' 004433               JSR     .DSK
 5   00504' 024044-              LDA     1,STD
 6   00505' 006004-              JSR     @BINO
 7   00506' 020007-              LDA     0,CR
 8   00507' 004427               JSR     .DSK
 9   00510' 210014-              LDA     2,BUFP
10   00511' 020024-              LDA     0,CNT3
11   00512' 101120               MOVZL   0,0
12   00513' 024021-              LDA     1,EPROG
13   00514' 034011-              LDA     3,DDOS
14   00515' 007407               JSR     @D.MOT,3
15   00516' 007400               JSR     @D.CYS,3
16   00517' 000013               D.EOF
17   00520' 007400               JSR     @D.CYS,3
18   00521' 000010               D.CLO
19   00522' 020041               LDA     0,PUTC
20   00523' 040041               STA     0,PRINT
21   00524' 006045               JSR     @RTN1
22   00525' 020040-              LDA     2,STM61
23   00526' 006044               JSR     @TEXT
24   00527' 020017-              LDA     2,NAME
25   00530' 121120               MOVZL   2,2
26   00531' 006044               JSR     @TEXT
27   00532' 006045               JSR     @RTN1
28   00533' 006045               JSR     @RTN1
29   00534' 034011-              LDA     3,DDOS
30   00535' 001442               JMP     D.HOM,3
31   00536' 054046    .DSK:      STA     3,RET1
32   00537' 044406               STA     1,DSK1
33   00540' 210014-              LDA     2,BUFP
34   00541' 034012-              LDA     3,DDOS
35   00542' 007410               JSR     @D.OUT,3
36   00543' 024402               LDA     1,DSK1
37   00544' 006045               JMP     @RET1
38   00545' 000001    DSK1:      .BLK    1
39   00546' 044042    .GETC:     STA     1,RET1
40   00547' 063610               SKPDN   TTI
41   00550' 000777               JMP     .-1
42   00551' 060510               DIAS    0,TTI
43   00552' 101003               MOV     0,0,SNR
44   00553' 000774               JMP     .-4
45   00554' 024017-              LDA     1,C177
46   00555' 123400               AND     1,0
47   00556' 024042               LDA     1,RET1
48
49   00557' 063511    .PUTC:     SKPBZ   TTO
50   00560' 000777               JMP     .-1
51   00561' 061111               DOAS    0,TTO
52   00562' 001400               JMP     0,3
```

```
---
PAGE  12                    *DIST PROGRAM*
06/12/75
 1   00563'020040   .ST53:   .TXT @  HOW MANY NUMBERS IN THE SET @
 2   00564'044117
 3   00565'051440
 4   00566'046501
 5   00567'047111
 6   00570'020116
 7   00571'072515
 8   00572'041105
 9   00573'051111
10   00574'020111
11   00575'047040
12   00576'052110
13   00577'041440
14   00600'051505
15   00601'053040
16   00602'000000
17   00603'020040   .ST54:   .TXT @  **********************@
18   00604'025052
19   00605'027071
20   00606'025052
21   00607'025071
22   00610'025052
23   00611'025071
24   00612'025052
25   00613'027071
26   00614'025052
27   00615'027071
28   00616'025052
29   00617'000000
30   00620'020040   .ST55:   .TXT @  * NUMBER*  EXPECTED  *@
31   00621'025040
32   00622'047111
33   00623'046502
34   00624'042511
35   00625'025040
36   00626'020107
37   00627'054120
38   00630'042501
39   00631'052105
40   00632'041040
41   00633'020052
42   00634'000000
43   00635'020040   .ST56:   .TXT @  *         * FREQUENCY  *@
44   00636'025040
45   00637'020040
46   00640'020040
47   00641'020040
48   00642'025040
49   00643'041111
50   00644'042521
51   00645'051507
52   00646'047103
53   00647'051440
54   00650'020052
55   00651'000000
56   00652'020040   .ST57:   .TXT @  * @
57   00653'025040
58   00654'000000
```

```
---
PAGE  13                    *DIST PROGRAM*
06/12/75
 1  00650' 020040   .ST58:   .TXT @*    @
 2  00651' 020040
 3  00652' 020000
 4  00653' 020052   .ST59:   .TXT @ *@
 5  00654' 000000
 6  00655' 020040   .ST60:   .TXT @  NAME OF DISTRIBUTION  @
 7  00656' 047101
 8  00657' 046505
 9  00660' 020117
10  00661' 041040
11  00662' 042111
12  00663' 051524
13  00664' 051111
14  00665' 041103
15  00666' 052111
16  00667' 047516
17  00670' 020040
18  00671' 000000
19  00672' 020040   .ST61:   .TXT @  OUTPUT ON FILE @
20  00673' 047525
21  00674' 051110
22  00675' 052524
23  00676' 020117
24  00677' 047040
25  00700' 041111
26  00701' 046105
27  00702' 020000
28  00703' 020040   .ST62:   .TXT @  STANDARD DEVIATION    @
29  00704' 051524
30  00705' 046516
31  00706' 042101
32  00707' 051104
33  00710' 020104
34  00711' 042516
35  00712' 044501
36  00713' 051111
37  00714' 047516
38  00715' 020040
39  00716' 020040
40  00717' 000000
41  00720' 020040   .ST63:   .TXT @  MEAN                @
42  00721' 046505
43  00722' 046516
44  00723' 020040
45  00724' 020040
46  00725' 020040
47  00726' 020040
48  00727' 020040
49  00730' 020040
50  00731' 020040
51  00732' 020040
52  00733' 020040
53  00734' 000000
54  00735' 000010   .NAME:   .BLK     10
55  00745' 004450   .PROG:   .TXT @<15>.ENTN NONE<15> NPEL<15>.TXTM 1<15>NONE<15>0:
    15>0:15>0:15>NEXT<15>.TXT *@
56  00746' 042516
57  00747' 052116
58  00750' 020114
```

```
---
PAGE  14                    *DIST PROGRAM*
06/12/75
 1   00756' 047515
 2   00757' 042415
 3   00760' 027116
 4   00761' 051105
 5   00762' 046015
 6   00763' 027124
 7   00764' 054114
 8   00765' 045440
 9   00766' 020415
10   00767' 047117
11   00770' 047105
12   00771' 006460
13   00772' 006463
14   00773' 006460
15   00774' 006462
16   00775' 006516
17   00776' 042520
18   00777' 052015
19   01000' 027114
20   01001' 054124
21   01002' 020071
22   01003' 000000
23   01004' 047107   .EPROG: .TXT @NEXT = .<15>.END<15>0
24   01005' 054124
25   01006' 020075
26   01007' 020056
27   01010' 006475
28   01011' 042516
29   01012' 042015
30   01013' 000000
31   01014' 054044   .RTN1:  STA     3,RET1
32   01015' 020007-         LDA     0,CR
33   01016' 006041          JSR     @PRINT
34   01017' 020010-         LDA     0,LF
35   01020' 006041          JSR     @PRINT
36   01021' 002046          JMP     @RET1
37   01022' 000000   .N:    0
38   01023' 000000   .F:    0
39   01024' 000000   .FF:   0
40
41        001000           .BLK    1000    ; STORAGE FOR FF
42        000000'          .END    STRT
```

APPENDIX 15

FASP LISTING

```
---
PAGE   1
07/04/75
 1                          .TITL   FASP
 2                  ; FLEXIBLE ASSEMBLER SIMULATION PACKAGE
 3
 4                  ; REVISION HISTORY:
 5                  ;       8/6/75
 6
 7                  ;                               D. YAMEY
 8                  ;                               UNIVERSITY OF THE WITWATERSRAND
 9                  ;                               JOHANNESBURG
10
11                          .ENT    C3,C177,C77,SP,AJTAB,AMTAB,.BUS,OPT,TR2
12                          .ENT    IPT,ACT,PROG,.RET,ERET,.XEC,COMD,IGNORE
13                          .ENT    TEMP2,TEMP3,DFLG,.READ,.WRITE,.PUTC
14                          .ENT    ERROR,GETAD,GETAB,MES,CRLF,INITR,GET
15                          .ENT    MULTIPLY,DIVIDE,RETURN,NONE,INTR,PUTC
16                          .ENT    INIT,.COMD,GETC,GETBR,.GETC,PUT
17                          .EXTD   USEND,ISTRT,IEND,DDOS
18                          .EXTD   COMP3,COMP5,DWRITE
19                          .EXTN   INIT1
20                          .HEAD           ****FASP****
21
22          000001          .LOC    1       ; PAGE ZERO STUFF
23  00001 '011121'          ISR             ; ADD OF INTERRUPT ROUTINE
24  00002 002004            JMP     @4      ; START
25  00003 002005            JMP     @5      ; RESTART
26  00004 177777            INIT1           ; COMPILER ADDRESS
27  00005 100026-           @.XEC
28
29          000022          .LOC    22      ; PAGE ZERO CONSTANTS
30  00022 000001    C1:     1
31  00023 000003    C3:     3
32  00024 000007    C7:     7
33  00025 000030    C30:    30              ; ASCII CAN
34  00026 000052    C52:    52              ; "*"
35  00027 000054    C54:    54              ; ","
36  00030 000073    C73:    73
37  00031 000077    C77:    77              ; MASK
38  00032 000101    C101:   101             ; "A"
39  00033 000132    C132:   132             ; "Z"
40  00034 000177    C177:   177             ; MASK
41  00035 000377    C377:   377             ; MASK
42  00036 177400    C177400: 177400
43  00037 000015    CR:     15              ; CARRIAGE RETURN
44  00040 000012    LF:     12              ; LINE FEED
45  00041 000040    SP:     40              ; SPACE
46
47  00042 000001    TEMP0:  .BLK    1       ; TEMPORARY STORAGE
48  00043 000001    TEMP1:  .BLK    1
49  00044 000001    TEMP2:  .BLK    1
50  00045 000001    TEMP3:  .BLK    1
51
52                          .ZREL
53  00000-000001    RT1:    .BLK    1       ; RETURN STACK
54  00001-000001    RT2:    .BLK    1
55  00002-000001    RT3:    .BLK    1
56
57  00003-000005$   ERET:   COMP3           ; ERROR RETURN ADD
58  00004-000052'   .ERO?:  ERROR           ; INDIRECT ADDRESSES
```

```
PAGE   2                    ****FASP****
07/04/75
 1   0000     0254' .OCBN:   OCBN
 2   00006-   0360' .BINO:   BINO
 3   00007-000110' .GTAD:   GTAD
 4   00010-000210' .GETB:   GETB
 5   00011-000502' .RET:    RET
 6   00012-000473' .DIVI:   DIVI
 7   00013-000463' .MULT:   MULT
 8   00014-000335' .GT1:    GT1
 9   00015-000222' .GTS:    GTS
10   00016-000523' .GETC:   GETC1
11   00017-001617' .PUTC:   PUTC1
12   00020-001617' .WRITE:  PUTC1
13   00021-000523' .READ:   GETC1
14   00022-000007$ .DWRITE: DWRITE
15   00023-000531' .CRLF:   CRLF1
16   00024-000510' .MES:    MESS
17   00025-000540' .COMD:   COMD
18   00026-000724' .XEC:    EXEC
19   00027-000753' .ER6:    ER6
20   00030-001521' .BUS:    BUS
21   00031-001050' .TRAP:   TRAP1
22   00032-001055' .TR2:    TRAP2
23   00033-001657' .TBUF:   TBUF
24   00034-001777' .TEND:   TEND
25   00035-001160' INTR:    DISMIS
26   00036-001421' AJTAB:   JTAB-1
27   00037-001322' AMTAB:   MTAB-1
28
29                                        ; PAGE ZERO STORAGE
30   00040-000000  DFLG:    0
31   00041-000000  ACT:     0
32   00042-000001  OPT:     .BLK    1
33   00043-000001  IPT:     .BLK    1
34   00044-000001  TOPT:    .BLK    1
35   00045-000001  TIPT:    .BLK    1
36   00046-000001  PROG:    .BLK    1
37
38                                        ; SYMBOL DEFINITION
39
40         006004-          ERROR     = JSR @.ERDR
41         006005-          GET       = JSR @.OCBN
42         006015-          GETAS     = JSR @.GTS
43         006006-          PUT       = JSR @.BINO
44         006007-          GETAD     = JSR @.GTAD
45         006010-          GETBR     = JSR @.GETB
46         002011-          RETUR     = JMP @.RET
47         006013-          MULTIPLY  = JSR @.MULT
48         006012-          DIVIDE    = JSR @.DIVI
49         006020-          PUTC      = JSR @.WRITE
50         006021-          GETC      = JSR @.READ
51         006023-          CRLF      = JSR @.CRLF
52         006024-          MES       = JSR @.MES
53         100027-          NONE      = @.ER6
54         006031-          TRAP      = JSR @.TRAP
```

```
---
PAGE   3                ****FASP****
07/04/75
 1                  ; SUBROUTINE SRCH    ,
 2                  ; SEARCHES FOR A USER ROUTINE
 3
 4
 5                  ; OUTPUT: ADDRESS OF USER SUBROUTINE IN AC3
 6                  ;          ADDRESS OF USER PROG IN AC2
 7
 8                  ; CALLING SEQUENCE:
 9                  ;       JSR @.SRCH
10
11                  ; DESTROYED: AC2,AC3
12
13                          .NREL
14  00000'054001- SRCH:     STA     3,RT2
15  00001'044456            STA     1,SR1
16  00002'006014-           JSR     @.GT1   ; GET FIRST 2 BYTES
17  00003'105300            MOVS    0,1
18  00004'006014-           JSR     @.GT1
19  00005'030042-           LDA     2,OPT
20  00006'050452            STA     2,SR2
21  00007'123000            ADD     1,0
22  00010'034030-           LDA     3,.BUS  ; START OF USER ROUTINES
23  00011'025406  SCH:      LDA     1,6,3   ; GET FIRST 2 BYTES
24  00012'122435            SUBZ#   1,0,SNR ; MATCH ?
25  00013'000407            JMP     SRCH2   ; YES
26  00014'035405  SRCH1:    LDA     3,5,3   ; GET NEXT ROUTINE
27  00015'024001#           LDA     1,USEND
28  00016'136433            SUBZ#   1,3,SNC ; END OF USER PROGS ?
29  00017'000772            JMP     SCH     ; NO
30  00020'006004-           ERROR
31  00021'000002            2               ; USER CODE UNDEFINED
32
33  00022'054437  SRCH2:    STA     3,SR3
34  00023'040433            STA     0,SR0
35  00024'030024            LDA     2,C7
36  00025'173120            ADDZL   3,2     ; AC2 = 2 * ADD OF NAME
37
38  00026'006014- SRCH3:    JSR     @.GT1   ; GET A BYTE
39  00027'155220            MOVZR   2,3
40  00030'025400            LDA     1,0,3   ; GET USER NAME
41  00031'151403            INC     2,2,SNC ; 1ST BYTE ?
42  00032'125300            MOVS    1,1     ; YES
43  00033'034034            LDA     3,C177
44  00034'167400            AND     3,1
45  00035'004534            JSR     GTBRK   ; TEST FOR BREAK
46  00036'000402            JMP     .+2     ; NOT A BREAK
47  00037'000410            JMP     SRCH4
48  00040'106415            SUB#    0,1,SNR ; MATCH ?
49  00041'000765            JMP     SRCH3   ; YES
50  00042'020414  SRCH5:    LDA     0,SR0   ; RESTORE STATES
51  00043'030415            LDA     2,SR2
52  00044'050042-           STA     2,OPT
53  00045'034414            LDA     3,SR3
54  00046'000746            JMP     SRCH1
55
56  00047'125004  SRCH4:    MOV     1,1,SZR   END OF NAME ?
57  00050'000772            JMP     SRCH5   ; NO FAIL
58  00051'151122            MOVZR   2,2,SZC ; SET UP ADD IN AC2
```

```
---
PAGE   4                    ****FASP****
07/04/75
 1   00052'151400            INC     2,2
 2   00053'034406            LDA     3,SR3
 3   00054'024401            LDA     1,SR1
 4   00055'002001-           JMP     @RT2
 5
 6   00056'000001  SR0:      .BLK    1
 7   00057'000001  SR1:      .BLK    1
 8   00060'000001  SR2:      .BLK    1
 9   00061'000001  SR3:      .BLK    1
10
11                 ; SUBROUTINE ERROR
12                 ; PROCESSES ERRORS
13
14                 ; CALLING SEQUENCE :
15                 ; JSR  @.ERROR = ERROR
16                 ; NUMERIC ERROR CODE
17
18                 ; FATAL ERRORS HAVE BIT 0 OF
19                 ; ERROR CODE SET AND CAUSE PROGRAM TO HALT
20
21   00062'054045  ERROR:  STA     3,TEMP3
22   00063'020020-         LDA     0,.WRITE ; ENABLE CONSOLE
23   00064'040042          STA     0,TEMP0
24   00065'020017-         LDA     0,.PUTC
25   00066'040020-         STA     0,.WRITE
26   00067'000024-         MES
27   00070'051105          .TXT    *ERROR*
28   00071'047522
29   00072'000122
30   00073'030045          LDA     2,TEMP3
31   00074'021000          LDA     0,0,2    ; GET CODE
32   00075'101100          MOVL    0,0      ; STRIP BIT 0
33   00076'101220          MOVZR   0,0
34   00077'006006-         PUT              ; PRINT CODE
35   00100'000002          2                ; 2 DIGIT FORMAT
36   00101'006023-         CRLF
37   00102'025000          LDA     1,0,2
38   00103'125132          MOVZL#  1,1,SZC ;  FATAL ?
39   00104'063077          HALT             ; YES YOU LOSE !!!
40   00105'020042          LDA     0,TEMP0 ; RE ENABLE
41   00106'040020-         STA     0,.WRITE
42   00107'002003-         JMP     @ERET    ; NO, RETURN
43
```

```
---
PAGE   5                    ****PASP****
07/04/75
 1                  ; SUBROUTINE GETAD
 2                  ; GETS A STORAGE ADDRESS GIVEN
 3                  ; A CODE AND A NUMBER IN THE COMMAND BUFFER
 4
 5                  ; CALLING SEQUENCE
 6                  ; GETAD ( = JSR @.GTAD )
 7
 8                  ; OUTPUT : ADDRESS OF CONSTANT STORAGE IN AC0
 9                  ;          ADDRESS OF INITIAL STORAGE IN AC1
10                  ;          ADDRESS OF PROGRAM AREA IN AC2
11                  ;          NUMBER OF ROUTINE IN AC3
12                  ;       ADD OF USER VARIABLE ROUTINE IN TEMP3
13                  ;          OUTPUT POINTER POINTS TO
14                  ;          BYTE AFTER BREAK
15                  ; DESTROYED: ALL ACCUMULATORS
16
17  00110'054000-  GTAD:   STA     3,RT1    ; SAVE RETURN
18  00111'004667           JSR     SRCH     ; SEARCH FOR ROUTINE
19  00112'050044           STA     2,TEMP2  ; ADD OF USER PROGRAM
20  00113'054045           STA     3,TEMP3  ; ADD OF ROUTINE
21  00114'021402           LDA     0,2,3    ; CHECK FOR ZERO STORAGE
22  00115'025404           LDA     1,4,3
23  00116'123005           ADD     1,0,SNR  ; ZERO ?
24  00117'002000-          JMP     @RT1     ; YES
25  00120'006057-          JSR     @.OCBN   ; GET NUMBER
26  00121'030022           LDA     2,C1
27  00122'142412           SUBZ#   2,0,SZC  ; < 1 ?
28  00123'000403           JMP     .+3      ; NO
29  00124'006004-          ERROR
30  00125'000003           3                ; ZERO OR -VE ADDRESS
31  00126'040042           STA     0,TEMP0  ; SAVE ROUTINE NO
32  00127'126400           SUB     1,1      ; DISPLACEMENT
33  00130'044043           STA     1,TEMP1
34  00131'034045   GTAD1:  LDA     3,TEMP3  ; ADDRESS OF ROUTINE
35  00132'024043           LDA     1,TEMP1  ; DISPLACEMENT
36  00133'137000           ADD     1,3
37  00134'030042           LDA     2,TEMP0
38  00135'150400           NEG     2,2
39  00136'150000           COM     2,2
40  00137'025402           LDA     1,2,3    ; NO OF LOCATIONS
41  00140'021401           LDA     0,1,3    ; START OF ROUTINES STORAGE
42  00141'006013-          MULTIPLY
43  00142'034045           LDA     3,TEMP3
44  00143'035405           LDA     3,5,3    ; ADD OF NEXT ROUTINE
45  00144'024043           LDA     1,TEMP1  ; DISPLACEMENT
46  00145'137000           ADD     1,3
47  00146'035401           LDA     3,1,3    ; START OF NEXT STORAGE
48  00147'162433           SUBZ#   3,0,SNC  ; OVERLAP ?
49  00150'000402           JMP     .+3      ; NO
50  00151'006004-          ERROR
51  00152'000004           4                ; ADDRESS TOO LARGE
52  00153'024043           LDA     1,TEMP1
53  00154'152400           SUB     2,2
54  00155'132414           SUB#    1,2,SZR  ; FIRST TIME ?
55  00156'000405           JMP     GARET    ; NO, RETURN
56  00157'040411           STA     0,CADD
57  00160'010043           ISZ     TEMP1
58  00161'010043           ISZ     TEMP1
```

```
---
PAGE   6                    ****FASP****
07/04/75
 1  00162'000747            JMP     GTAD1
 2
 3  00163'105000  GARET:    MOV     0,1
 4  00164'020404            LDA     0,CADD  ; CONSTANT STORAGE ADDRESS
 5  00165'030044            LDA     2,TEMP2 ; USER PROG AREA
 6  00166'034042            LDA     3,TEMP0 ; NUMBER OF ROUTINE
 7  00167'002000-           JMP     @RT1
 8
 9  00170'000001  CADD:     .BLK    1
```

```
---
PAGE   7                    ****FASP****
07/04/75
 1                  ; SUBROUTINE GTBRK
 2                  ; CHECKS FOR A BREAK CHARACTER
 3
 4                  ; INPUT : CHAR IN AC0 RIGHT JUSTIFIED
 5
 6                  ; OUTPUT : CODE IN AC3
 7
 8                  ; CODE:
 9                  ;        -1  LAST PARAMETER
10                  ;         0  BREAK WAS ","
11                  ;         1  BREAK WAS ";"
12
13                  ; CALLING SEQUENCE :
14                  ;  JSR  GTBRK
15                  ;  RETURN IF NOT A BREAK CHAR
16                  ;  NORMAL RETURN
17
18  00171'054002- GTBRK:  STA     3,RT3
19  00172'010002-         ISZ     RT3
20  00173'034027          LDA     3,C54    ; ","
21  00174'116405          SUB     0,3,SNR  ; IS IT ,
22  00175'002001-         JMP     @RT3     ; YES
23  00176'034037          LDA     3,CR
24  00177'116415          SUB#    0,3,SNR  ; IS IT CR ?
25  00200'176001          ADC     3,3,SKP  ; -1 TO AC3
26  00201'000403          JMP     .+2
27  00202'002002-         JMP     @RT3     ; YES
28  00203'034030          LDA     3,C73
29  00204'116424          SUBZ    0,3,SZR  ; IS IT ;
30  00205'014002-         DSZ     RT3      ; NO
31  00206'175400          INC     3,3
32  00207'002002-         JMP     @RT3
33
34
35
36                  ;SUBROUTINE GETB
37                  ; RETURNS A CODE FOR THE LAST BREAK CHARACTER
38
39                  ; OUTPUT: CODE IN AC3
40
41                  ; CALLING SEQUENCE :
42                  ;         GETBR
43
44  00210'054000- GETB:   STA     3,RT1    ; SAVE STATES
45  00211'050044          STA     2,TEMP2
46  00212'040043          STA     0,TEMP0
47  00213'014042-         DSZ     OPT
48  00214'004531          JSR     GT1      ; GET BYTE FROM BUFFER
49  00215'004754          JSR     GTBRK
50  00216'000401          JMP     .+1      ; FILLER
51  00217'020042          LDA     0,TEMP0
52  00220'010044          LDA     2,TEMP2
53  00221'002000-         JMP     @RT1     ; RETURN
```

```
---
PAGE   8                ****FASP****
07/04/75
 1                  ; SUBROUTINE GETAS
 2                  ; GETS AN ASCII STRING FROM THE BUFFER
 3
 4                  ; INPUT:
 5                  ; AC0 MUST CONTAIN THE ADDRESS WHICH IS TO RECEIVE
 6                  ; THE FIRST BYTE OF THE STRING
 7
 8                  ; OUTPUT:
 9                  ; STRING STARTING AT LOCATION SPECIFIED AND
10                  ; ENDING IN A NULL WORD OR BYTE
11                  ; AC1 = NO OF WORDS TRANSFERRED
12                  ; AC0, AC2 UNCHANGED
13
14                  ; CALLING SEQUENCE:
15                  ;       LDA     0,"START ADDRESS"
16                  ;       GETAS
17
18  00222'054000- GTS:    STA     3,RT1
19  00223'040042          STA     0,TEMP0
20  00224'050044          STA     2,TEMP2
21  00225'111000          MOV     0,2       ; ADDRESS
22  00226'006014- GTS1:   JSR     @.GT1     ; GET A BYTE
23  00227'004742          JSR     GTBRK     ; BREAK ?
24  00230'000401          JMP     .+2       ; NO
25  00231'000412          JMP     GTS2      ; YES PUT IN NULLS
26  00232'105300          MOVS    0,1
27  00233'006014-         JSR     @.GT1     ; 2ND BYTE
28  00234'004735          JSR     GTBRK
29  00235'000402          JMP     .+2
30  00236'000406          JMP     GTS2+1    ; SINGLE NULL
31  00237'123000          ADD     1,0
32  00240'041000          STA     0,0,2
33  00241'151400          INC     2,2
34  00242'000764          JMP     GTS1
35
36  00243'102401  GTS2:   SUB     0,0,SKP   ; FIRST BYTE
37  00244'020036          LDA     0,C177400
38  00245'107400          AND     0,1
39  00246'045000          STA     1,0,2
40  00247'020042          LDA     0,TEMP0
41  00250'145000          MOV     2,1
42  00251'106400          SUB     0,1       ; NO OF WORDS
43  00252'030044          LDA     2,TEMP2
44  00253'002000-         JMP     @RT1
```

```
---
PAGE   9                    ****FASP*****
07/04/75
 1
 2                  ; CONVERT AN ASCII CHARACTER STRING TO A SINGLE
 3                  ; PRECISION BINARY NUMBER
 4                  ; CONVERTS AN ASCII DECIMAL CHARACTER STRING TO A
 5                  ;       TWO'S COMPLEMENT, FIXED POINT, BINARY NUMBER
 6
 7                  ; INPUT:          OPT MUST POINT TO THE BYTE CONTAINING
 8                  ;                 THE FIRST NUMBER IN BUFFER
 9
10                  ; OUTPUT:         AC0 CONTAINS THE BINARY INTEGER
11                  ;                 AC3 CONTAINS THE BREAK CODE
12
13                  ; CALLING SEQUENCE:
14                  ;       GET  =JSR       A.OCBN
15                  ;       RETURN
16
17
18                  ; DESTROYED:      AC0,AC3
19                  ; UNCHANGED:      AC2,AC1
20
21
22
23
24  00254'054452  OCBN:   STA 3,.EC03     ; SAVE AC3
25  00255'050450          STA 2,.EC02     ; SAVE AC2
26  00256'044456          STA 1,.EC01     ; SAVE AC1
27  00257'102400          SUB 0,0
28  00260'040447          STA 0,.EC10     ; CLEAR SIGN WORD
29  00261'040447          STA 0,.EC11     ; CLEAR SUM WORD
30  00262'004451          JSR GT1         ; GET A CHARACTER
31  00263'024446          LDA 1,.EC20     ; TEST FOR "+"
32  00264'106405          SUB 0,1,SNR
33  00265'000405          JMP .EC97       ; YES
34  00266'024444          LDA 1,.EC21     ; NO, TEST FOR "-"
35  00267'106404          SUB 0,1,SZR
36  00270'000401          JMP .EC96       ; NO EXPLICHT SIGN
37  00271'010436          ISZ .EC10       ; SET FLAG WORD FOR NEGATIVE
38                                        ; NUMBER
39  00272'004443  .EC97:  JSR GT1         ; GET ANOTHER CHARACTER
40  00273'024440  .EC96:  LDA 1,.EC22     ; ASCII "0"
41  00274'030440          LDA 2,.EC23     ; ASCII "9"
42  00275'142033          ADDZ# 2,0,SNC   ; SKIP IF > 9
43  00276'106032          ADDZ# 0,1,SZC   ; SKIP IF >= 0
44  00277'000406          JMP .EC95       ; NOT A DIGIT, THERFORE A BREAK
45                                        ; CHARACTER
46  00300'122400          SUB 1,0         ; REDUCE DIGIT TO 0-9 BINARY
47                                        ; RANGE
48  00301'024427          LDA 1,.EC11     ; SUM WORD
49  00302'004415          JSR .EC50       ; MULTIPLY BY 10 AND ADD
50  00303'044425          STA 1,.EC11     ; SAVE SUM
51  00304'000766          JMP .EC97       ; GET NEXT CHARACTER
52
53
54
55  00305'004654  .EC95:  JSR GTBRK
56  00306'000446          JMP ER5         ; ILLEGAL BREAK CHAR
57  00307'020421          LDA 0,.EC11     ; RESULT TO AC1
58  00310'101120          MOVZL 0,0
```

```
---
PAGE  10                    ****FASP****
07/04/75
 1  00311'014416                DSZ .EC10          ; TEST SIGN
 2  00312'101221                MOVZR 0,0,SKP      ; POSITIVE
 3  00313'100540                NEGOR 0,0          ; NEGATIVE
 4  00314'030411                LDA 2,.EC02        ; RESTORE AC2
 5  00315'024527                LDA 1,.ED01        ; RESTORE AC1
 6  00316'002410                JMP @.EC03
 7
 8
 9                      ; ROUTINE TO MULTIPLY AC1 BY 10 AND ADD AC0
10
11
12  00317'131120  .EC50:  MOVZL 1,2          ; N*2
13  00320'151120          MOVZL 2,2          ; N*4
14  00321'147000          ADD 2,1            ; N*5
15  00322'125120          MOVZL 1,1          ; N*5*2 = N*10
16  00323'107000          ADD 0,1            ; ADD AC0
17  00324'001400          JMP 0,3            ; SUCCESS RETURN
18
19
20
21  00325'000000  .EC02:  0                  ; SAVE AC2
22  00326'000000  .EC03:  0                  ; SAVE AC3
23
24
25  00327'000000  .EC10:  0                  ; FLAG WORD FOR SIGN OF RESULT
26  00330'000000  .EC11:  0                  ; RUNNING SUM WORD
27
28  00331'000053  .EC20:  "+                 ; ASCII "+"
29  00332'000055  .EC21:  "-                 ; ASCII "-"
30  00333'000060  .EC22:  "0                 ; ASCII "0"
31  00334'000071  .EC23:  "9                 ; ASCII "9"
32
33  00335'050416  GT1:    STA     2,GT2
34  00336'030042-         LDA     2,OPT      ; BYTE POINTER
35  00337'020041-         LDA     0,IPT
36  00340'112432          SUBZ#   0,2,SZC
37  00341'000415          JMP     ER9
38  00342'151220          MOVZR   2,2
39  00343'021000          LDA     0,0,2
40  00344'175003          MOV     3,3,SNC    ; WHICH BYTE ?
41  00345'101399          MOVS    0,0        ; 1ST
42  00346'030034          LDA     2,C177
43  00347'143400          AND     2,0        ; MASK
44  00350'010042-         ISZ     OPT        ; BUMP POINTER
45  00351'010401          LDA     2,GT2
46  00352'001400          JMP     0,3
47
48  00353'000001  GT2:    .BLK    1
49  00354'006004- ER5:    ERROR
50  00355'000005          5                  ; ILLEGAL BREAK CHAR
51  00356'006004- ER9:    ERROR
52  00357'000011          9.                 ; TOO FEW PARMS IN LIST
```

```
---
PAGE  11                    ****FASP****
07/04/75
 1
 2                  ; BINARY TO DECIMAL ASCII CONVERT
 3                  ; CONVERTS A SINGLE PRECISION, TWO'S COMPLEMENT
 4                  ;       NUMBER TO AN ASCII CHARACTER STRING
 5
 6                  ; INPUT:        N IN AC0
 7
 8                  ; OUTPUT:       ASCII CHARACTER STRING, TERMINATED BY A
 9                  ;               NULL WORD
10
11                  ; CALLING SEQUENCE:
12                  ;       PUT =   JSR     @.BINO
13                  ;       NO. OF DECIMAL DIGITS
14                  ;       RETURN
15
16
17                  ; DESTROYED:    AC0, AC3, CARRY
18                  ; UNCHANGED:    AC1, AC2
19
20
21  00360'054466  BINO:   STA 3,.ED03     ; SAVE RETURN
22  00361'050464          STA 2,.ED02     ; SAVE AC2
23  00362'044461          STA 1,.ED00     ; SAVE AC1
24  00363'105000          MOV 0,1
25  00364'035400          LDA 3,0,3       ; GET PRECISION
26  00365'020474          LDA 0,C5
27  00366'116012          ADC2# 0,3,SZC   ; PRECISION TOO LARGE ;
28  00367'000403          JMP .+3         ; YES
29  00370'175112          MOVL# 3,3,SNC   ; -VE PRECISION ?
30  00371'000403          JMP .+3         ; NO
31  00372'006004-         ERROR
32  00373'000007          7
33  00374'030456          LDA 2,.ED30     ; ADDRESS OF POWER OF TEN TABLE
34  00375'172400          SUB 3,2
35  00376'050457          STA 2,.ED10     ; INITIALIZE POINTER
36  00377'152400          SUB 2,2
37  00400'050457          STA 2,FLG       ; CLEAR FLAG
38  00401'020731          LDA 0,.EC21     ; ASSUME NEGATIVE
39  00402'125112          MOVL# 1,1,SZC
40  00403'124401          NEG 1,1,SKP
41  00404'020452          LDA 0,.ED21     ; NO, IT IS POSITIVE: GET PLUS
42  00405'006020-  .ED97: PUTC            ; PUT OUT SIGN OR DIGIT
43  00406'036447          LDA 3,@.ED10    ; GET CURRENT  POWER OF TEN
44  00407'010446          ISZ .ED10       ; BUMP POINTER
45  00410'161005          MOV 3,0,SNR
46  00411'000425          JMP .ED98       ; PUT OUT NULL
47  00412'020721          LDA 0,.EC22     ; GET ASCII "0"
48  00413'156422  .ED99:  SUBZ 3,1,SZC    ; DOES POWER OF TEN GO IN?
49  00414'101401          INC 0,0,SKP     ; YES, BUMP RESULT DIGIT
50  00415'157001          ADD 3,1,SKP     ; NO, RESTORE PREVIOUS VALUE
51  00416'000775          JMP .ED99       ; CONTINUE SUBTRACTING
52  00417'030715          LDA 2,.EC23     ; "9"
53  00420'142032          ADC2# 2,0,SZC   ; FORMAT OVERFLOW ?
54  00421'020437          LDA 0,DOLL      ; YES PUT OUT DOLLAR
55  00422'020711          LDA 2,.EC22     ; "0"
56  00423'112435          SUBZ# 0,2,SN?   ; IS IT ZERO ?
57  00424'000403          JMP .+3         ; YES
58  00425'010432          ISZ FLG         ; NO, BUMP FLAG
```

```
---
PAGE  12                    ****FASP****
07/04/75
 1  00426'000757                JMP .ED97          ; PUT OUT DIGIT
 2  00427'032426                LDA 2,@.ED10
 3  00430'151005                MOV 2,2,SNR        ; LAST DIGIT ?
 4  00431'010426                ISZ FLG            ; YES
 5  00432'030425                LDA 2,FLG          ; GET FLAG
 6  00433'151005                MOV 2,2,SNR        ; LEADING ZERO ?
 7  00434'020422                LDA 0,.ED21        ; YES, PUT OUT SPACE
 8  00435'000750                JMP .ED97          ; PUT OUT CHAR
 9
10  00436'005020-  .ED98:       PUTC               ; PUT OUT NULL WORD
11  00437'024404                LDA 1,.ED00        ; RESTORE AC1
12  00440'030405                LDA 2,.ED02        ; RESTORE AC2
13  00441'010405                ISZ .ED03
14  00442'002404                JMP @.ED03         ; RETURN
15
16
17
18  00443'000000  .ED00:        0                  ; SAVE AC0
19  00444'000000  .ED01:        0                  ; SAVE AC1
20  00445'000000  .ED02:        0                  ; SAVE AC2
21  00446'000000  .ED03:        0                  ; SAVE AC3
22
23        000012                .RDX 10
24  00447'023420                10000              ; POWER OF TEN TABLE
25  00450'001750                1000               ; 10**3
26  00451'000144                100                ; 10**2
27  00452'000012                10                 ; 10**1
28  00453'000001                1                  ; 10**0
29  00454'000000  .ED05:        0                  ; END OF TABLE INDICATION
30        000010                .RDX 8
31
32  00455'000000  .ED__:        0                  ; ADDRESS OF CURRENT POWER OF
33                                                 ; TEN ENTRY
34  00456'000040  .ED21:        40                 ; ASCII SPACE
35  00457'000001  FLG:          .BLK 1             ; FLAG FOR ZERO SUPPRESSION
36  00460'000044  DOLL:         "$
37  00461'000005  C5:           5
38
39  00462'000454' .ED30:        .ED05              ; ADDRESS OF POWER OF TEN TABLE
40
```

```
PAGE  13                    ****FASP****
07/04/75
 1                  ; SUBROUTINE MULT
 2                  ; MULTIPLIES 2 SINGLE PRECISION NUMBERS TO FORM A
 3                  ; DOUBLE PRECISION RESULT
 4                  ; THE RESULT IS ADDED TO THE NUMBER IN AC0
 5
 6                  ; INPUT: NUMBERS TO BE MULTIPLIED IN AC1 AND AC2
 7                  ;        NUMBER TO BE ADDED IN AC0
 8
 9                  ; OUTPUT: HIGH ORDER WORD IN AC1
10                  ;         LOW ORDER WORD IN AC0
11
12                  ; CALLING SEQUENCE:
13                  ;       MULTIPLY
14
15                  ; UNCHANGED: AC2
16
17  00463'061001    MULT:   DOA     0,MDV
18  00464'066001            DOB     1,MDV
19  00465'073301            MUL
20  00466'060000            NIO     0       ; WAIT
21  00467'060030            NIO     0
22  00470'064401            DIA     1,MDV
23  00471'061401            DIB     0,MDV
24  00472'001400            JMP     0,3
25
26
27                  ; SUBROUTINE DIVI
28                  ; DIVIDES A DOUBLE PRECISION WORD BY A SINGLE
29                  ; PRECISION WORD TO GIVE A SINGLE
30                  ; PRECISION QUOTIENT
31
32                  ; INPUT: HIGH ORDER WORD IN AC1
33                  ;        LOW ORDER WORD IN AC0
34                  ;        DIVISOR IN AC2
35
36                  ; OUTPUT: QUOTIENT IN AC0
37                  ;         REMAINDER IN AC1
38
39                  ; UNCHANGED: AC2
40
41  00473'065001    DIVI:   DOA     1,MDV
42  00474'062001            DOB     0,MDV
43  00475'073101            DIV
44  00476'060000            NIO     0
45  00477'064401            DIA     1,MDV
46  00500'061401            DIB     0,MDV
47  00501'001400            JMP     0,3
48
```

```
---
PAGE  14                    ****FASP****
07/04/75
 1                  ; SUBROUTINE RET
 2                  ; CHECKS THAT ALL PARAMETERS IN COMMAND STRING
 3                  ; HAVE BEEN USED
 4
 5  00502'020043-   RET:    LDA     0,IPT
 6  00503'024042-           LDA     1,OPT
 7  00504'106415            SUB#    0,1,SNR
 8  00505'002003-           JMP     @ERET   ; NORMAL RETURN
 9  00506'006004-           ERROR
10  00507'000010            8.              ; TOO MANY PARMS  OR
11                                          ; ILLEGAL OPERATION
12
13
14                  ; ROUTINE TO TYPE OUT MESSAGES
15                  ; CALLING SEQUENCE:
16                  ;       MES
17                  ;       REQUIRED MESSAGE PACKED R-L
18                  ;       AND TERMINATED IN A NULL
19
20                  ;RETURN: STATEMENT FOLLOWING MESSAGE
21                  ; DESTROYED : ALL ACS
22
23  00510'165120    MES:    MOVZL   3,1     ; AC1 = 2 * AC3
24  00511'131220    MES1:   MOVZR   1,2
25  00512'021000            LDA     0,0,2   ; GET WORD
26  00513'125400            INC     1,1     ; BUMP POINTER
27  00514'125213            MOVR#   1,1,SNC ; WHICH HALF ?
28  00515'101300            MOVS    0,0     ; LEFT HALF
29  00516'006020-           PUTC            ; PRINT
30  00517'034035            LDA     3,C377
31  00520'163404            AND     3,0,SZR ; TERMINATE ON NULL
32  00521'000770            JMP     MES1
33  00522'001001            JMP     1,2     ; RETURN
34
35  00523'063610    GETC1:  SKPDN   TTI
36  00524'000777            JMP     .-1
37  00525'060610            DIAC    0,TTI
38  00526'024034            LDA     1,C177
39  00527'123400            AND     1,0
40  00530'001400            JMP     0,3
41
42  00531'034405    CRLF1:  STA     3,RT4
43  00532'020037            LDA     0,CR
44  00533'006020-           PUTC
45  00534'020040            LDA     0,LF
46  00535'006020-           PUTC
47  00536'002401            JMP     @RT4
48  00537'000001    RT4:    .BLK    1
```

```
---
PAGE  15                    ****FASP****
07/04/75
 1                      ; COMMAND GET ROUTINE
 2                      ; READS A COMMAND INTO A 74 CHAR BUFFER
 3                      ; EDITING MAY BE PERFORMED BY
 4                      ; RUBOUT AND ^X
 5                      ; TABS AND SPACES ARE ECHOED AS A SPACE
 6                      ; BUT ARE IGNORED
 7                      ; OTHER NON PRINTING CHARACTERS
 8                      ; EXCEPT CARRIAGE RETURN ARE IGNORED
 9
10                      ; CALLING SEQUENCE:
11                      ;  JSR  @.COMD
12
13                      ; OUTPUT:  AC0 = FIRST TWO CHARS
14                      ;          AC1 = NO OF BYTES IN BUFFER
15                      ; CHARS STORED IN BUFFER L-R
16                      ; OPT POINTS TO FIRST WORD IN BUFFER
17  00540'054000- COMD:   STA     3,RT1
18  00541'060277  QUES:   INTDS
19  00542'030514          LDA     2,BUF    ; BUFFER START
20  00543'151120          MOVZL   2,2
21  00544'050043-         STA     2,IPT    ; INPUT POINTER
22  00545'050042-         STA     2,OPT    ; OUTPUT POINTER
23  00546'060110          NIOS    TTI
24  00547'006021- LP1:    GETC             ; GET CHAR
25  00550'024003          LDA     1,CAN
26  00551'106415          SUB#    0,1,SNR  ; WAS IT ^X ?
27  00552'000456          JMP     CAN1     ; YES
28  00553'024001          LDA     1,TAB    ; DONT STORE TAB
29  00554'122434          SUBZ#   1,0,SZR
30  00555'024041          LDA     1,SP     ; DONT STORE SPACE
31  00556'122434          SUBZ#   1,0,SZR
32  00557'000403          JMP     .+3
33  00560'006020-         PUTC             ; PRINT SPACE
34  00561'000765          JMP     LP1-1
35  00562'122012          ADCZ#   1,0,SZC  ; NON PRINTING ?
36  00563'000404          JMP     .+4      ; NO
37  00564'024017          LDA     1,CR
38  00565'122414          SUB#    1,0,SZR  ; CR ?
39  00566'000750          JMP     LP1-1    ; NO
40  00567'030043-         LDA     2,IPT
41  00570'024014          LDA     1,C177
42  00571'106435          SUBZ#   0,1,SNR  ; WAS IT A RUBOUT
43  00572'000427          JMP     RUBO     ; YES
44  00573'151222          MOVCR   2,2,SZC  ; WHICH BYTE ?
45  00574'004412          JSR     GETWD    ; 2ND
46  00575'105300          MOVS    0,1
47  00576'043000          STA     1,0,2    ; STORE IN BUFFER
48  00577 006020-         PUTC             ; ECHO CHAR
49  00600'010041-         ISZ     IPT      ; BUMP POINTER
50  00601'024037          LDA     1,CR
51  00602'106415          SUBZ#   0,1,SNR  ; END OF COMMAND ?
52  00603'000431          JMP     TTRET    ; YES
53  00604'060110          NIOS    TTI      ; RESTART READER
54  00605'000742          JMP     LP1
55
56  00606'175400  GETWD:   INC     3,3
57  00607'054001-         STA     3,R12
58  00610'025000          LDA     1,0,2    ; GET FIRST BYTE
```

```
---
PAGE  16                ****FASP****
07/04/75
 1  00611'034444          LDA     3,C77400        ; MASK
 2  00612'167400          AND     3,1
 3  00613'107000          ADD     0,1
 4  00614'002001-         JMP     @RT2
 5
 6  00615'014043- RUBO:   DSZ     IPT
 7  00616'030043          LDA     2,IPT
 8  00617'034042-         LDA     3,OPT   ; CHECK FOR END OF COMMAND
 9  00620'156032          ADCZ#   2,3,SZC
10  00621'000407          JMP     CAN1    ; YES
11  00622'151220          MOVZR   2,2
12  00623'021000          LDA     0,0,2
13  00624'101003          MOV     0,0,SNC ; WHICH BYTE ?
14  00625'101100          MOVS    0,0     ; 1ST
15  00626'006020-         PUTC            ; ECHO DELETED CHAR
16  00627'000717          JMP     LP1-1
17
18  00630'020711  CAN1:   LDA     0,QUES
19  00631'006020-         PUTC            ; PRINT ?
20  00632'004677          JSR     CRLF1
21  00633'000707          JMP     QUES+1
22
23  00634'004675  T'RET:  JSR     CRLF1
24  00635'030042-         LDA     2,OPT
25  00636'151220          MOVZR   2,2
26  00637'021000          LDA     0,0,2
27  00640'024043-         LDA     1,IPT   ; IGNORE SINGLE CR
28  00641'030042-         LDA     2,OPT
29  00642'151400          INC     2,2
30  00643'146415          SUB#    2,1,SNR
31  00644'000676          JMP     QUES+1
32  00645'146400          SUB     2,1     ; NO OF BYTES
33  00646'125400          INC     1,1
34  00647'034040-         LDA     3,DFLG  ; DISK WRITE FLAG
35  00650'175004          MOV     3,3,SZR
36  00651'006022-         JSR     @.DWRITE
37  00652'002000-         JMP     @RT1
38
39  00653'000030  CAN:    30
40  00654'000011  TAB:    11
41  00655'077400  C77400: 77400
42  00656'000657' BUF:    BUFS
43  00657'000045  BUFS:   .BLK    37.
```

```
---
PAGE  17                ****FASP****
07/04/75
 1                                      ; EXECUTIVE
 2                                      ; PERFORMS THE COMMAND INTERPRETING
 3                                      ; AND DISPATCHES TO THE APPROPRIATE
 4                                      ; ROUTINE
 5
 6  00724'006031- EXEC:   CRLF
 7  00725'020026          LDA     0,C52   ; "*"
 8  00726'006030-         PUTC
 9  00727'020041          LDA     0,SP
10  00730'006030-         PUTC
11  00731'006025-         JSR     @.COMD  ; GET ' 'MAND
12  00732'115100          MOVS    0,3
13  00733'024034          LDA     1,C177  ; MASK
14  00734'167400          AND     3,1
15  00735'034512          LDA     3,SLASH
16  00736'136435          SUBZ#   1,3,SNR ; IS IT AN EXECUTIVE COMMAND ?
17  00737'000404          JMP     .+4     ; YES
18  00740'006067-         GETAD           ; SEARCH USER ROUTINES
19  00741'034045          LDA     3,TEMP3
20  00742'001400          JMP     @0,3    ; JMP TO USER VARIABLE ROUTINE
21  00743'030042-         LDA     2,OPT   ; CHECK FOR MORE THAN
22  00744'034023          LDA     3,C3    ; 3 CHARS IN EXECUTIVE COMMAND
23  00745'173000          ADD     3,2
24  00746'034041-         LDA     3,IPT
25  00747'156032          ADDZ#   2,3,SZC ; ERROR ?
26  00750'000405          JMP     ER6
27  00751'074413          LDA     3,.DISP ; START OF DISP TABLE
28  00752'025400  LE1:    LDA     1,0,3   ; GET COMMAND
29  00753'125004          MOV     1,1,SZR ; END OF TABLE ?
30  00754'000401          JMP     .+3     ; NO
31  00755'006004- ER6:    ERROR
32  00756'000005          6               ; ILLEGAL COMMAND
33  00757'106415          SUB#    0,1,SNR ; MATCH ?
34  00760'001401          JMP     @1,3    ; YES
35  00761'175400          INC     3,3     ; BUMP POINTER
36  00762'175400          INC     3,3
37  00763'000767          JMP     LE1
```