General Purpose System Simulation

The system to be simulated in GPSS is described as a block diagram in which block represents the activities and lines joining the blocks indicate the sequence in which the activities can be executed. Where there is choice of activities more than one line leaves a block and the condition for the choice is stated at the block. Each block must give precise meaning. There are 48 specific blocks. Each of which represents a characteristic action of systems. The program should be written using these block diagrams.

Entities of the system depend upon the nature of the system. For eg in simulation of a communication system movement of messages, in road transportation moving of vehicles are entities. These entities are called transactions. The sequence of events in real time is reflected in movement of transactions from block to block in simulated time.

Transactions starts from generate block and stops at terminate block. Transaction is hold in block and most blocks can hold many transactions simultaneously. Transfer of transaction from one block to another occurs instantaneously at a specific time or when some change of system condition occurs.

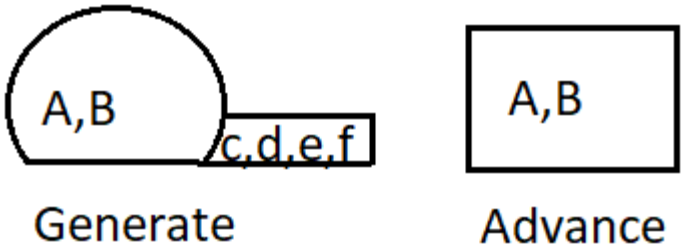
Action Times

Clock time is represented by an integral number with the interval of real time corresponding to a unit of time chosen by the program user. **ADVANCE** blocks represents the expenditure of time. The program computes an interval of time called action time for each transactions as it enters an ADVANCE block, and the transaction remain at this block for this interval of simulated time before attempting to proceed.

Another block that represents action time is **GENERATE** block that creates transaction. The action time at this block controls the interval between successive arrivals of transactions.

Action time may be fixed interval or a random variable and it can depend upon the conditions in the system in various ways. An action time is defined by giving a mean and modifier as A and B fields for the block. If modifier is zero, the action time is a constant equal to mean. If modifier is a positive number greater than or equals to mean, the action time is an integer random variable chosen from the range **mean± modifier** with equal probability of occurrence given to each number in the range.

Functions can be introduced to relate input and output variable. By specifying modifier at Advance or Generate block to be a function, the value of function controls the action time. The action time is derived by multiplying the mean by value of function.



Field A and field B in both blocks are mean and modifier. The fields c, d, e, f in generate block indicates:

c- offset (arrival time of first entity)

d- count (number of arrivals of entity)

e- priority (low and high priority of entity)

f- parameters (gives data types)

Succession of Events

The program maintains records of when each transaction in the system is due to move. It proceeds by completing all movements that are scheduled for execution at a particular instant of time. When there is more than one transactions to move, the program processes transactions in order of priority with first-come-first serve basis.

Transactions do not spend time at any block other than at an ADVANCE block. The program begins moving transaction through the block diagram until one of several circumstances arises. The transaction may enter on ADVANCE block with a non-zero action time, the program will turn its attention to other transactions in the system and return to the transaction when the action time has been expanded.

Secondly – A condition arises that the transaction is attempting to execute by entering block cannot be performed at the current time. The transaction is said to be blocked and it remains at the block it last entered. The program will automatically detect when the blocking condition is removed and will start the transaction at that time.

A third possibility is that transaction enters TERMINATE block in which case it is removed from simulation.

Fourth possibility is that a transaction may be put on a chain. When the program has moved one transaction as far it can go, it turns to other transactions that is to move at same time instant. If all such movements are complete, the program advances the clock to the time of next most imminent event and repeats the process of executing events.

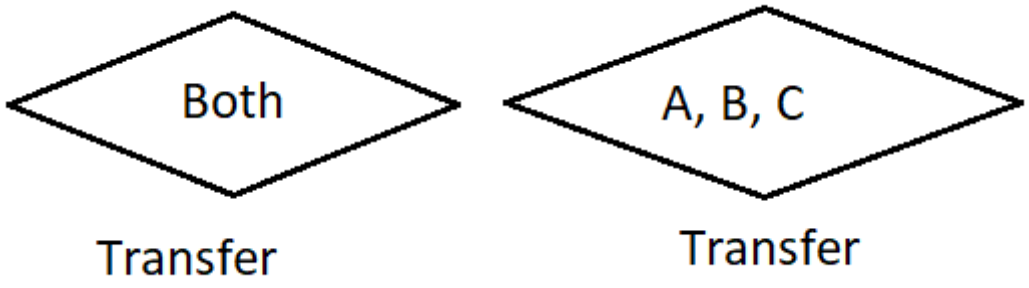
Choice of path

TRANSFER block

The TRANSFER block is used for choice of path .It allows to select the location other than the next sequential block choice is between two blocks. Selection factor in field A gives the choice to next block in field B and C. If there is no choice selection factor is left blank. An unconditional transfer is then made to next block A.

Two modes of choice

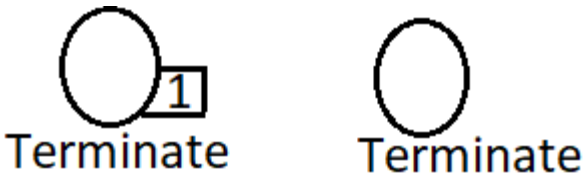
1) A random choice can be made by setting the selection factor, S. The probability of going to next block A, is then 1-S and to C is S.
2) A conditional mode, indicated by setting field A to BOTH, allows a transaction to select an alternate path depending upon existing conditions. The transaction moves to next block B if this move is possible, or to C if it is not possible. If both are impossible, transaction waits for first to become possible, giving preference to B.



Terminate Block

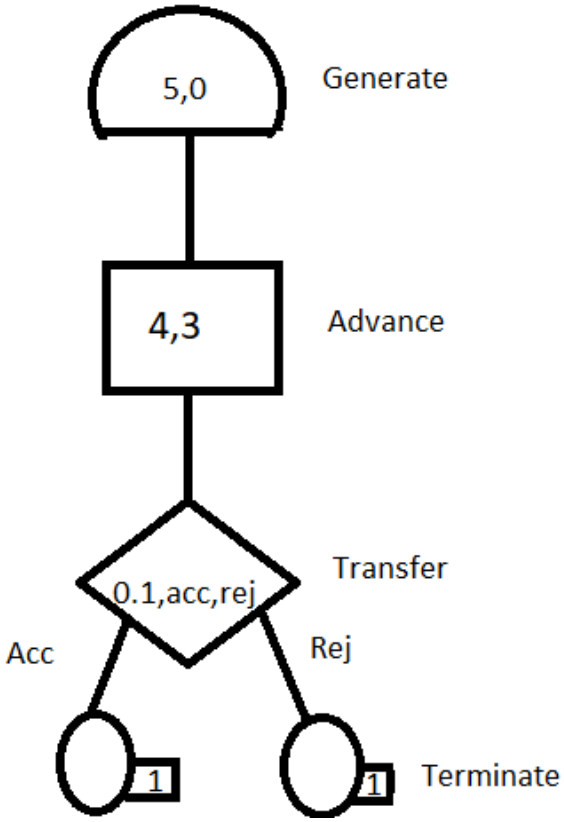
Terminate block is used to end the simulation in GPSS. There may be multiple terminate block in a program but there must be at least one terminate block with nonzero field.

Terminate block is circular shape with a tail on it with field 1.



Example

A machine tool in a manufacturing shop is turning parts at the rate of one every 5 minutes. As they are finished, the parts go to an inspector who takes 4±3 minutes to examine each one and rejects about 10% of the parts. Each part will be represented by one transaction and the time unit selected for the problem will be minute. Simulate it for 1000 parts.



Manufacturing shop model-1

Here, GENERATE block is used to represent the output of the machine by creating one transaction every five units of time. An ADVANCE block with a mean of 4 and modifier of 3 is used to represent inspection. The time spent on inspection will therefore be any one of the values 1, 2, 3, 4, 5, 6 or 7 with equal probability given to each value. Upon completion of inspection, transaction go to a TRANSFER block with

a selection factor of 0.1, so that 90% of the parts go to next location called ACC to represent accepted parts and 10% go to another location called REJ to represent rejected parts. There is no history thus both locations are to TERMINATE block.

Location							Operation										Operation Field											Comments								
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39

An * in column 1 results in the statement being printed in output only. A field from 2 to 6 contains location of blocks where it is necessary. The GPSS program will automatically assign sequential location as it reads the statements, so it is not necessary to assign locations. The transfer block needs to make reference to TERMINATE blocks to which it sends transactions.

The second section of coding from columns 8 to 18 contains the block type name, which must begin in column 8. From 19, a series of fields may be present each separated by commas and having no blanks. Any line following the first blank is treated as comment. The meaning of field depends upon block type.

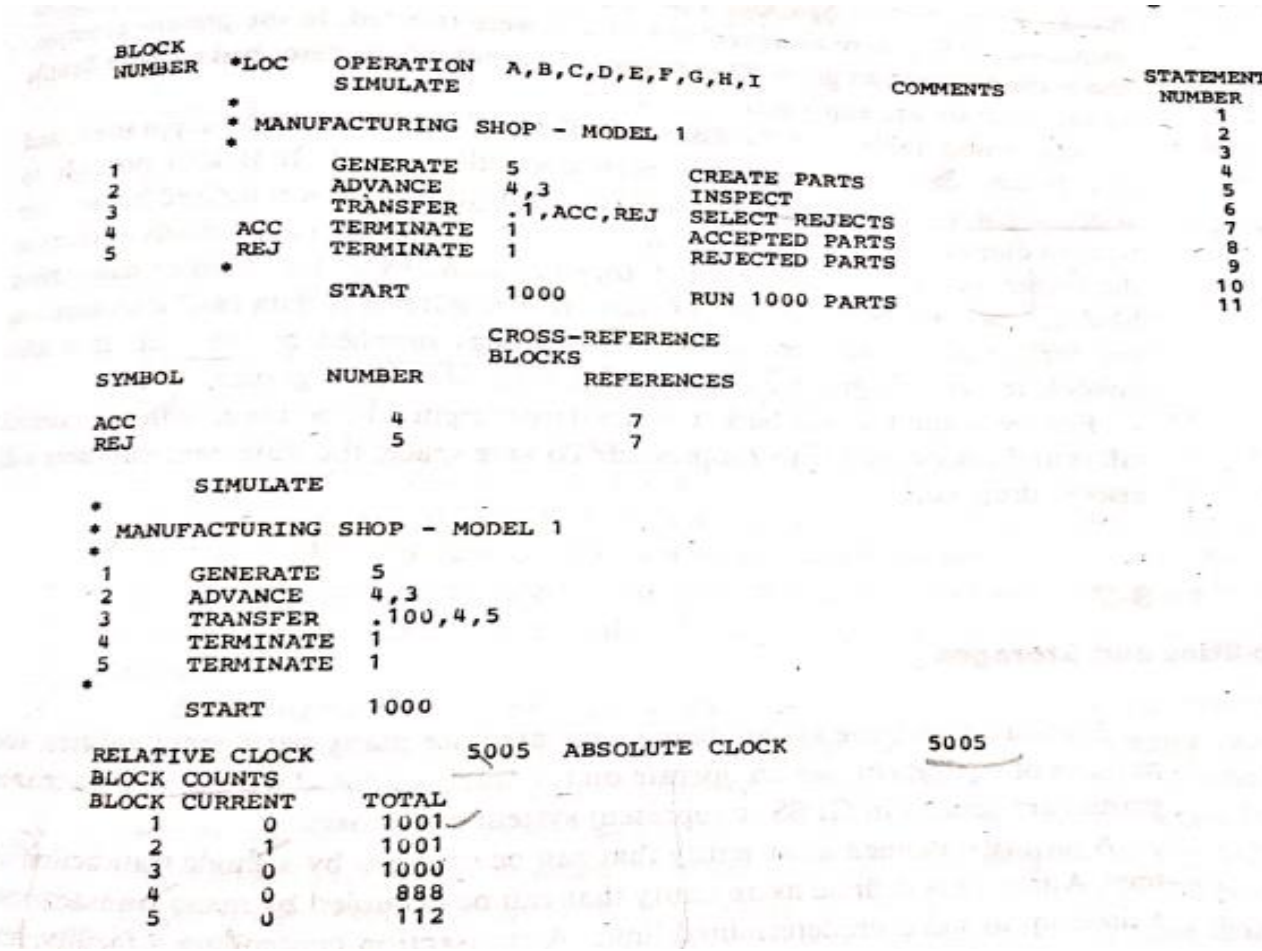
The program accepts input in a free format in which the location field, if used, begins in column 1, but none of other fields has a fixed starting points. A single block marks the transaction from the location field to the operation field and from operation to operand field. If no location is specified an initial blank is used. The free format is very convenient hen entering input from a terminal, since it minimizes the number of characters that have to be entered. The fixed format is easier to use.

For TRANSFER block, the first field is selection factor B and C fields are exit 1 and 2 respectively. In this case exit 1 is the next sequential block, ACC can be omitted from TRANFER and TERMINATE block. Comma must be used to show that field B is missing. So TRANSFER block would be coded as TRANSFER 1, , REJ. If there is unconditional transfer, then also a comma should be included to indicate the field i.e. TRANSFER, REJ.

Program runs until a certain count is reached terminating transactions. TERMINATE block has field A that carries a number indicating by how much termination count is incremented. The number should be

positive or zero. But there must be at least one non zero TERMINATE block. In our example there is one in both block thus 1 should be added in both bad and good inspection.

In last line of coding there is a control statement START. Here the START statement is set to 1000. When START is read, the program begins execution. When simulation is completed, the program prints an output report.



```
BLOCK
NUMBER   *LOC    OPERATION  A,B,C,D,E,F,G,H,I          COMMENTS                STATEMENT
                SIMULATE                                                        NUMBER
                *                                                                  1
                * MANUFACTURING SHOP - MODEL 1                                    2
                *                                                                  3
1                       GENERATE   5                 CREATE PARTS               4
2                       ADVANCE    4,3               INSPECT                    5
3                       TRANSFER   .1,ACC,REJ        SELECT REJECTS             6
4                ACC    TERMINATE  1                 ACCEPTED PARTS             7
5                REJ    TERMINATE  1                 REJECTED PARTS             8
                *                                                                  9
                        START      1000              RUN 1000 PARTS             10
                                                                                  11

                                CROSS-REFERENCE
                                BLOCKS
SYMBOL                  NUMBER          REFERENCES

ACC                        4                7
REJ                        5                7

        SIMULATE
*
* MANUFACTURING SHOP - MODEL 1
*
 1      GENERATE    5
 2      ADVANCE     4,3
 3      TRANSFER    .100,4,5
 4      TERMINATE   1
 5      TERMINATE   1
*
        START       1000

RELATIVE CLOCK              5005  ABSOLUTE CLOCK          5005
BLOCK COUNTS
BLOCK  CURRENT      TOTAL
    1        0       1001
    2        1       1001
    3        0       1000
    4        0        888
    5        0        112
```

Problem input is printed first, with the locations listed from left and a sequential statement number on right. Then table of symbolic location to each symbol.

The first line of output following the listings gives the time at which simulation stop. Time is followed by listing block counts. Two numbers are shown for each block. On the left is count of how many transactions were in the block at simulation stop time and on right is a figure showing total number of transactions entering the block during simulation.

The results show that the counts at block 4 and 5 were 888 and 112 respectively showing 1000 parts inspected i.e. 88.8% accepted and 11.2% rejected.

Facilities and Storages

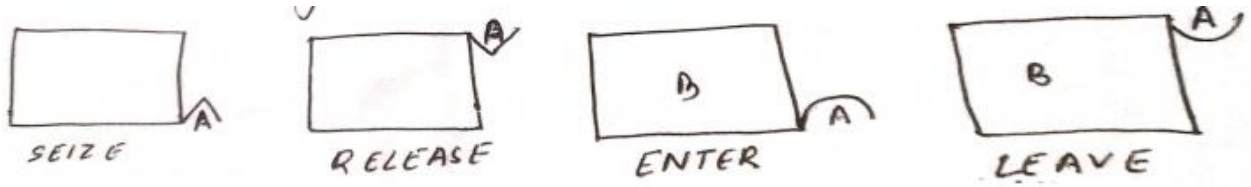
A facility is defined as an entity that can be engaged by a single transaction at a time. A storage is defined as an entity that can be occupied by many transactions at a time, up to some predetermined limit. A transaction controlling a facility can be interrupted or preempted by another transaction.

There can be many instances of each type of entity to a limit set by the program. Entities are identified by number. 0 cannot be assigned. User may assign numbers in any order or symbolic names can also be assigned.

Eg

Types of System	Transaction	Facility	Storage
Communications	Message	Switch	Trunk
Transportation	Car	Tollbooth	Road
Data processing	Record	Key Punch	Computer Memory

SIEZE, RELEASE, ENTER and LEAVE are used for facilities and storages.

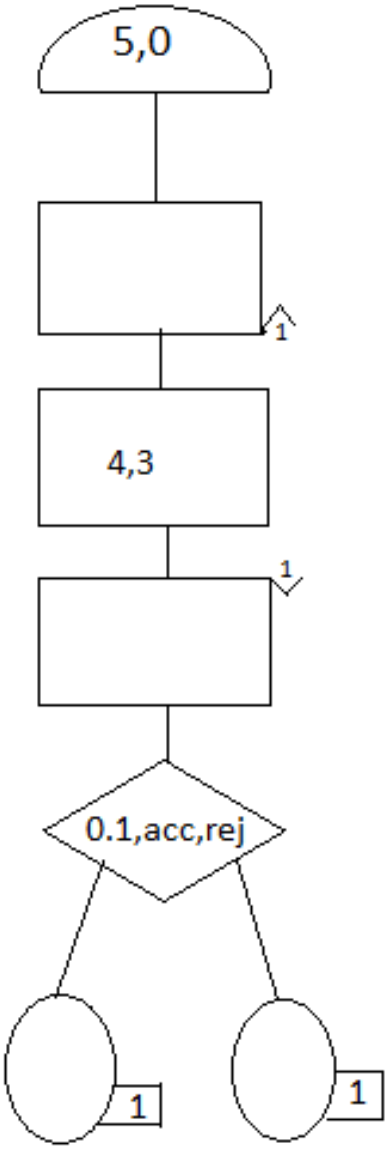


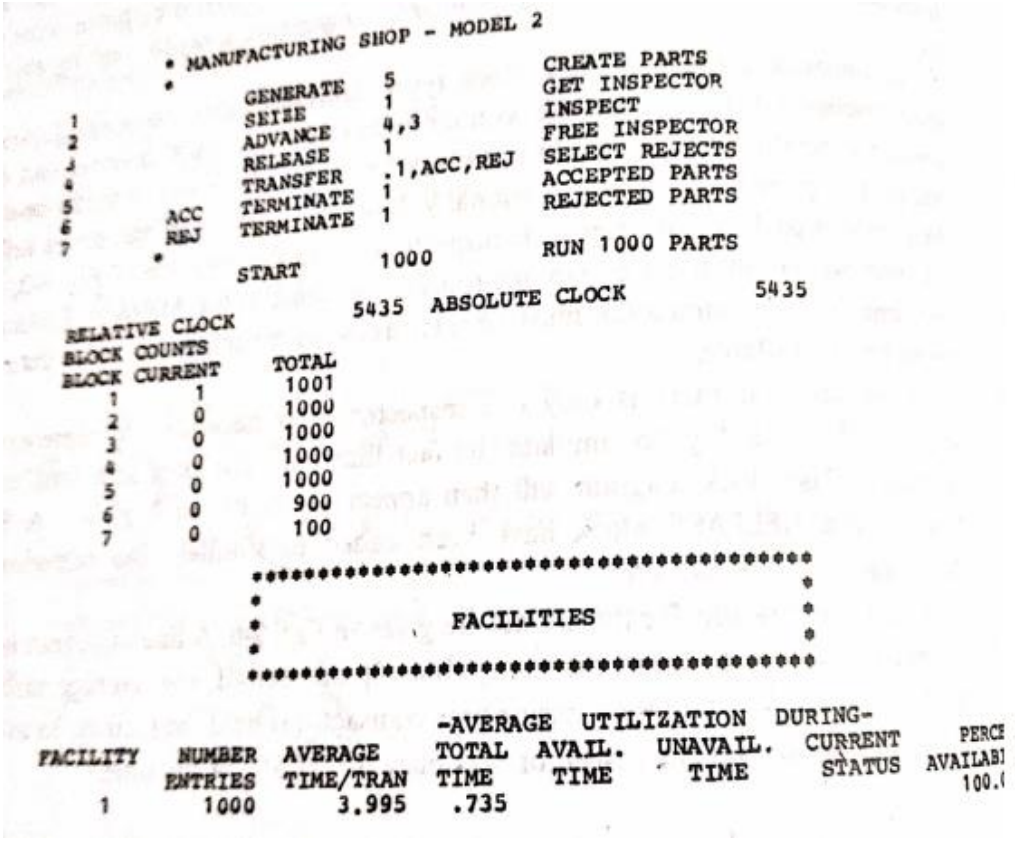
Field A in each case indicates which facility or storage is intended, and the choice is usually marked in the flag attached to the symbols of the blocks. The SEIZE block allows a transaction to engage a facility if it is available. The RELEASE block allows the transaction to disengage facility.

Enter block allows a transaction to occupy space in a storage if available and LEAVE block allows it to give up the space. If field B of the ENTER and LEAVE block are blank, the storage content are changed by 1. If there is a number ≥1, then the contents change by that value. Number of blocks can be used between the points where facility is seized and released to simulate the actions. Similar is the case for storage.

Example of manufacturing shop has average inspection time 4 and average generate rates of parts 5 minutes. There will be normally one part inspected at a time. A new part can arrive before the inspection

of previous one. Thus there will be more than one transaction in ADVANCE block at one time. Assuming for only one inspector and the inspector should be represented by facility.





```
                * MANUFACTURING SHOP - MODEL 2
                *
                          GENERATE    5              CREATE PARTS
                          SEIZE       1              GET INSPECTOR
1                         ADVANCE     4,3            INSPECT
2                         RELEASE     1              FREE INSPECTOR
3                         TRANSFER    .1,ACC,REJ     SELECT REJECTS
4                         TERMINATE   1              ACCEPTED PARTS
5                   ACC   TERMINATE   1              REJECTED PARTS
6                   REJ
7                *
                          START       1000           RUN 1000 PARTS


                                5435  ABSOLUTE CLOCK          5435

RELATIVE CLOCK
BLOCK COUNTS
BLOCK CURRENT      TOTAL
    1       1       1001
    2       0       1000
    3       0       1000
    4       0       1000
    5       0       1000
    6       0        900
    7       0        100
```

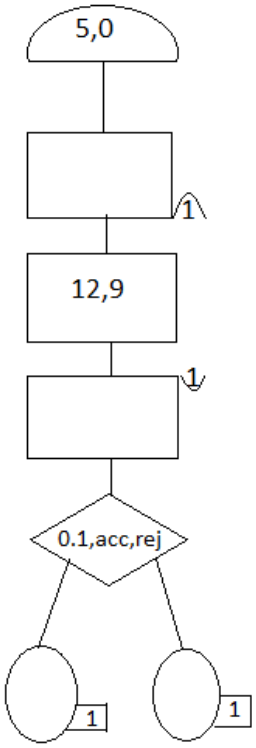


```
                FACILITIES
```

```
                                    -AVERAGE  UTILIZATION  DURING-
FACILITY   NUMBER   AVERAGE    TOTAL   AVAIL.   UNAVAIL.   CURRENT    PERCE
           ENTRIES  TIME/TRAN  TIME     TIME      TIME      STATUS   AVAILABI
                                                                       100.(
    1        1000      3.995    .735
```

The result show inspector was busy for 73.5% of his time. If more than one inspector is available, they can be represented by storage with capacity equal to no. of inspector. SEIZE and RELEASE in the figure should be replaced by ENTER and LEAVE.

Eg Inspection time were 3 times as long as before, and there is 3 inspector, then storage should be added. STORAGE statement has location field which identifies it and operation field A has the capacity greater than or equal to 2. The difference between ENTER, LEAVE and SIEZE, RELEASE is that in facility transaction that grabs entity should free the entity but is not necessary all transactions should free storage. Entering and leaving is different actions in storage.




```
      * MANUFACTURING SHOP - MODEL 3
      *
            GENERATE    5           CREATE PARTS
            ENTER       1           GET AN INSPECTOR
            ADVANCE     12,9        INSPECT
            LEAVE       1           FREE INSPECTOR
            TRANSFER    .1,ACC,REJ  SELECT REJECTS
      ACC   TERMINATE   1           ACCEPTED PARTS
      REJ   TERMINATE   1           REJECTED PARTS
      *
       1    STORAGE     3           NUMBER OF INSPECTORS
      *
            START       1000        RUN 1000 PARTS
```

RELATIVE CLOCK 5355 ABSOLUTE CLOCK 5355

BLOCK	CURRENT	TOTAL
1	1	1003
2	0	1002
3	2	1002
4	0	1000
5	0	1000
6	0	893
7	0	107

STORAGES

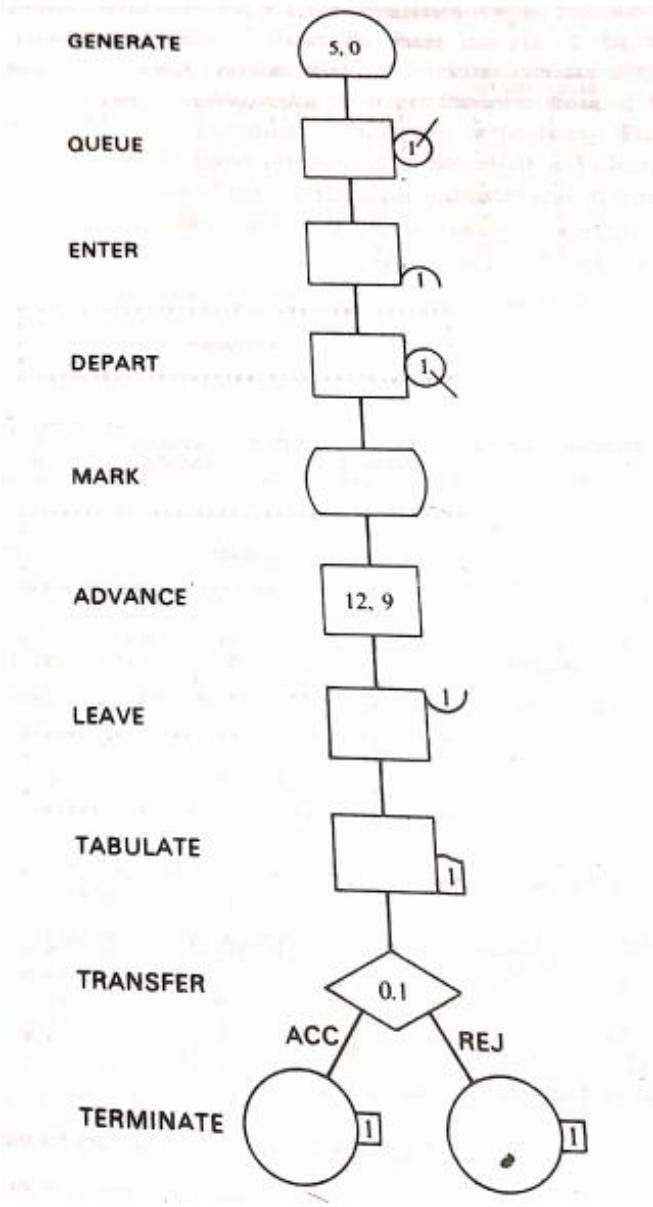
STORAGE	CAPACITY	AVERAGE CONTENTS	ENTRIES	AVERAGE TIME/UNIT	-AVERAGE TOTAL TIME	UTILIZATION AVAIL. TIME	DURING- CURRENT CONTENTS	MAXIMUM CONTENTS
1	3	2.256	1002	12.059	.752		2	3

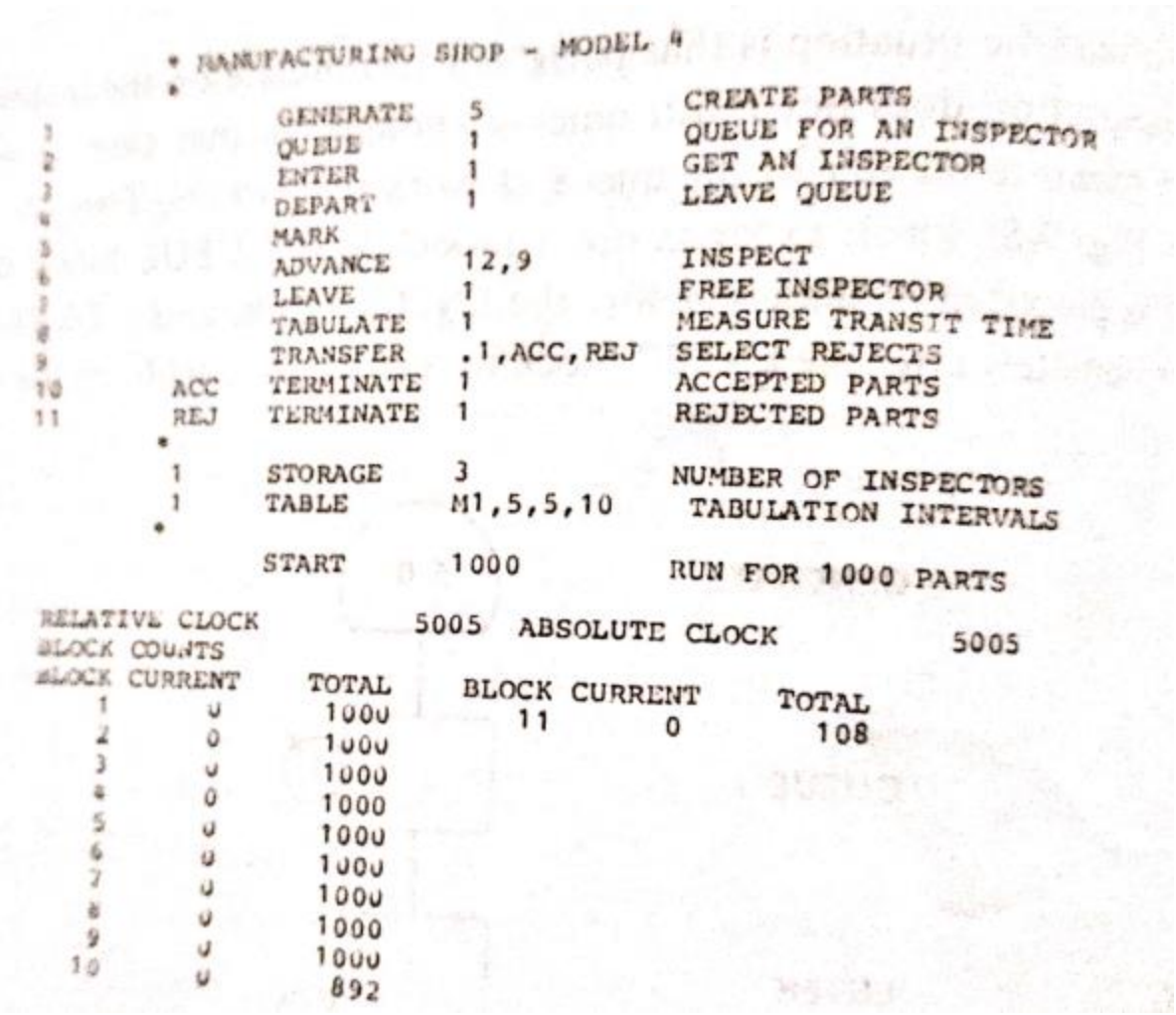
Gathering statistics

The **QUEUE, DEPART, MARK, TABULATE** block are used for gathering statistics.

When the conditions for advancing a transaction are not satisfied several transactions may be kept waiting at a block. When the conditions are favorable they are moved with first-in, first-out basis. The **QUEUE** block increases and **DEPART** block decreases the queue numbered in field A. If field B is blank the change is a unit change otherwise value of B is ≥ 1.

MARK and **TABULATE** gives the length of time taken by transactions to move through the system. The MARK block notes the time of arrival. The TABULATE block subtracts the time noted by MARK block from the time of arrival at TABULATE block. The transit time is entered in table whose number or name is indicated in field A of TABULATE. If transaction entering TABULATE block is not passed through MARK block, the transit time is derived by using as a base the time at which transaction was created.



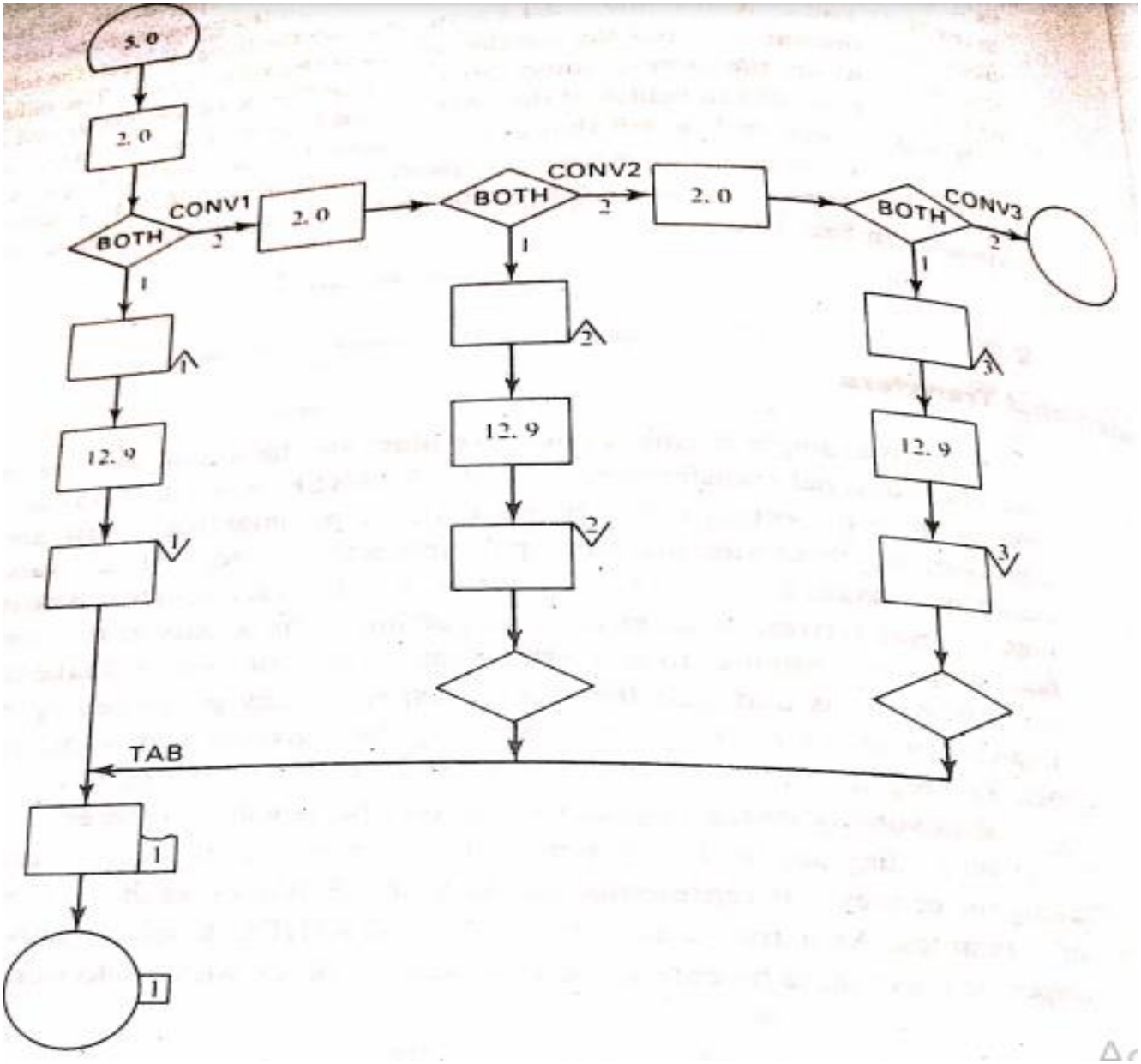


```
          * MANUFACTURING SHOP - MODEL 4
          *
                    GENERATE    5              CREATE PARTS
1                   QUEUE       1              QUEUE FOR AN INSPECTOR
2                   ENTER       1              GET AN INSPECTOR
3                   DEPART      1              LEAVE QUEUE
4                   MARK
5                   ADVANCE     12,9           INSPECT
6                   LEAVE       1              FREE INSPECTOR
7                   TABULATE    1              MEASURE TRANSIT TIME
8                   TRANSFER    .1,ACC,REJ     SELECT REJECTS
9
10        ACC       TERMINATE   1              ACCEPTED PARTS
11        REJ       TERMINATE   1              REJECTED PARTS
          *
          1         STORAGE     3              NUMBER OF INSPECTORS
          1         TABLE       M1,5,5,10       TABULATION INTERVALS
          *
                    START       1000           RUN FOR 1000 PARTS

RELATIVE CLOCK          5005  ABSOLUTE CLOCK          5005
BLOCK COUNTS
BLOCK  CURRENT   TOTAL      BLOCK  CURRENT   TOTAL
    1        0    1000         11        0     108
    2        0    1000
    3        0    1000
    4        0    1000
    5        0    1000
    6        0    1000
    7        0    1000
    8        0    1000
    9        0    1000
   10        0     892
```

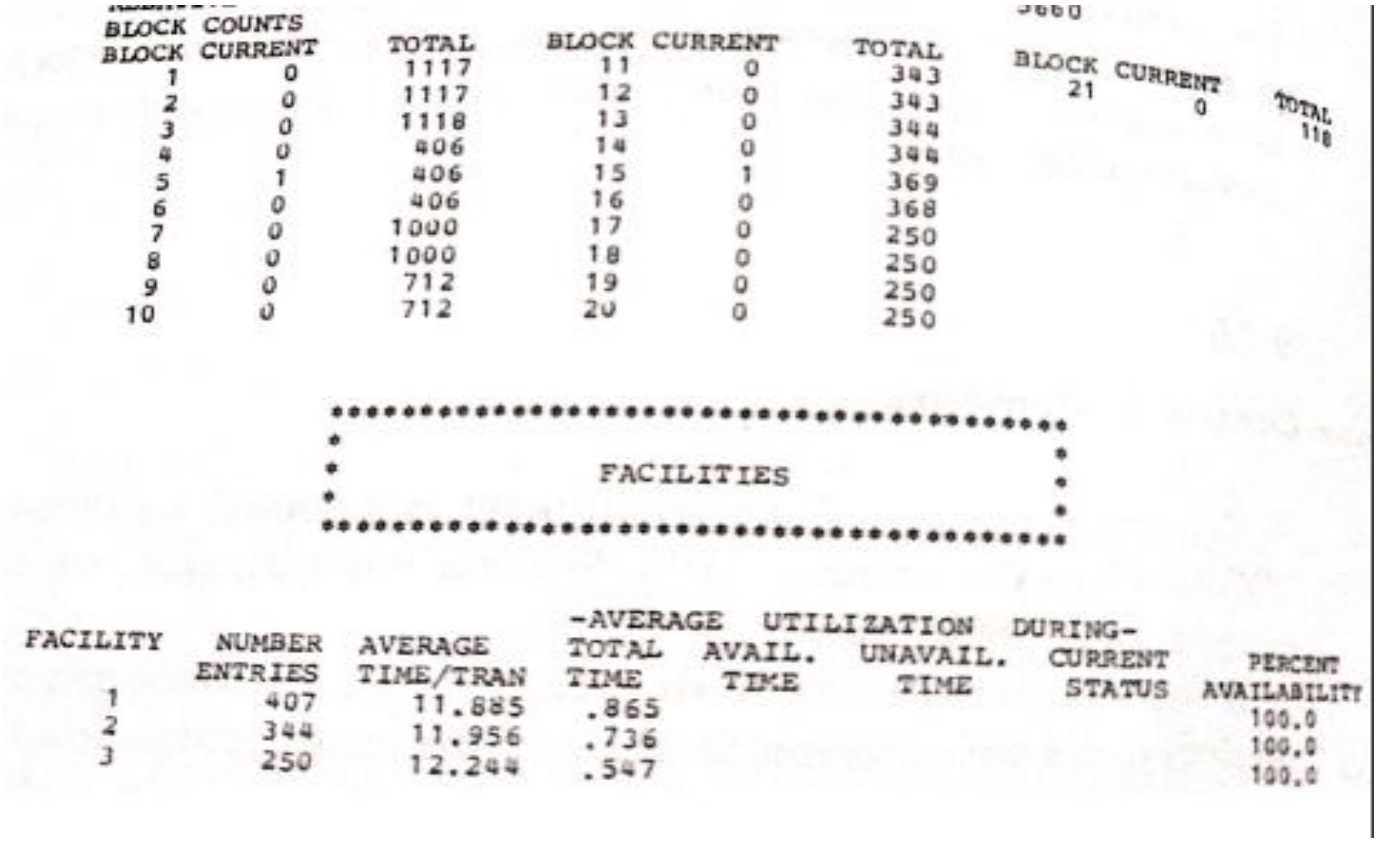
Conditional Transfers

Again consider the case of three inspectors and the manufactured parts are put on a conveyor which carries the parts to the inspector at intervals along the conveyor. It take 2 minutes for a part to reach to the first inspector; if it is free it takes the part for inspection else it takes further 2 minutes to reach to second inspector who will take it if its free else it passes to the part to third inspector which again takes 2 minutes otherwise they are lost. To keep the model small only the transit time of the parts are recorded and the possibility of inspector rejecting parts are ignored.


5.0
2.0
BOTH
CONV1
2
2.0
BOTH
CONV2
2
2.0
BOTH
CONV3
2
1
1
12.9
1
2
12.9
2
3
12.9
3
TAB
1
1

```
                * MANUFA...
                        GENERATE    5               CREATE PARTS
                *       ADVANCE     2               PLACE ON CONVEYOR
1                       TRANSFER    BOTH,,CONV1     MOVE TO FIRST INSPECTOR
2                       SEIZE       1               GET FIRST INSPECTOR
3                       ADVANCE     12,9            INSPECT
4                       RELEASE     1               FREE INSPECTOR
5                       TABULATE    1               MEASURE TRANSIT TIME
6               TAB     TERMINATE   1
7
8               *
                CONV1   ADVANCE     2               PLACE ON CONVEYOR
9                       TRANSFER    BOTH,,CONV2     MOVE TO SECOND INSPECTOR
10                      SEIZE       2               GET SECOND INSPECTOR
11                      ADVANCE     12,9            INSPECT
12                      RELEASE     2               FREE INSPECTOR
13                      TRANSFER    ,TAB
14
                *
                CONV2   ADVANCE     2               PLACE ON CONVEYOR
15                      TRANSFER    BOTH,,CONV2     MOVE TO THIRD INSPECTOR
16                      SEIZE       3               GET THIRD INSPECTOR
17                      ADVANCE     12,9            INSPECT
18                      RELEASE     3               FREE INSPECTOR
19                      TRANSFER    ,TAB
20
                *
                CONV3   TERMINATE
21              *
                1       TABLE       M1,5,5,10       TABULATION INTERVALS
                *
                        START       10,NP           INITIALIZE WITH TEN PARTS
                        RESET
                        START       1000            MAIN RUN

RELATIVE CLOCK          5588  ABSOLUTE CLOCK              5660
```



BLOCK COUNTS

BLOCK	CURRENT	TOTAL
1	0	1117
2	0	1117
3	0	1118
4	0	406
5	1	406
6	0	406
7	0	1000
8	0	1000
9	0	712
10	0	712

BLOCK	CURRENT	TOTAL
11	0	343
12	0	343
13	0	344
14	0	344
15	1	369
16	0	368
17	0	250
18	0	250
19	0	250
20	0	250

BLOCK	CURRENT	TOTAL
21	0	118

FACILITIES

FACILITY	NUMBER ENTRIES	AVERAGE TIME/TRAN	-AVERAGE UTILIZATION DURING- TOTAL TIME	AVAIL. TIME	UNAVAIL. TIME	CURRENT STATUS	PERCENT AVAILABILITY
1	407	11.885	.865				100.0
2	344	11.956	.736				100.0
3	250	12.244	.547				100.0

Program Control Statement

The first statement of GPSS input is a control statement SIMULATE in operation field. Without this statement problem will be assembled but not in use.

Simulation run can be restart or stop and can also repeat changing value to some extent. When GPSS simulation run is finished, the program does not destroy the model immediately. It looks input following the START statements. Input following the START statement can change the model. A STORAGE statement can be inserted giving new value. The model can also be changed by changing existing block or adding new blocks. When the desired changes is made the model reruns with another START statement.

RESET is another control statement. It is used to wipe out all the statistics gathered. It will leave the system loaded with transactions to gather statistics in second run. The output of first run is then not of interest and is used as NP in field B of START. One start statement has run the model for 10 completed transactions. A RESET statement has wipe out the statistics in that run. The second START statement has restarted the simulation from the point of 10^{th} transaction and has continued for 100 transactions.

The RESET statement also sets relative clock to zero. Absolute clock gives the time since the run began. Relative clock gives the time since the last reset statement. If reset is not used both time are same.

CLEAR statement wipes out the statistics and transactions in the system, so that rerun starts the simulation from the beginning. The CLEAR statements return the model to its initial state. It does not reset the random number generator seeds. The sequence of statements

START

CLEAR

START

would run the same problem twice but second run would use a different sets of random numbers.

JOB wipes out the entire model preceding the statement and proceed with following problem.

END terminates all simulation. SIMULATE statement appears only once even if there are multiple jobs.