



**General Purpose Simulation System/360  
Application Description**

This manual discusses the role of GPSS in system simulation, briefly describes the operation of the GPSS program, and points out its application scope. A final section describes the new entities, block types, expanded features, and system requirements of GPSS/360.

## CONTENTS

Introduction . . . . .	1
Computer Simulation Defined . . . . .	1
General Purpose Simulation System . . . . .	1
Operating Highlights . . . . .	2
Application Scope . . . . .	7
GPSS/360 Assembly Program Extensions . . . . .	9
New Entities, New and Extended Block Types, and Other Expanded Features . . . . .	10
Output Editor . . . . .	14
System Requirements . . . . .	16
Bibliography. . . . .	18

This edition, H20-0186-2, is a major revision of, and obsoletes, H20-0186-1. TNL N20-0064-0 has been incorporated in this revision, and hence the TNL is also obsoleted.

Copies of this and other IBM publications can be obtained through IBM branch offices. Address comments concerning the contents of this publication to IBM, Technical Publications Department, 112 East Post Road, White Plains, N.Y. 10601

## INTRODUCTION

The IBM General Purpose Simulation System (GPSS) is a computer program for conducting evaluations and experiments of systems, methods, processes, and designs. In addition to providing all the features and capabilities available in GPSS III, GPSS/360 introduces new entities, block types, and expanded features to increase the versatility and ease of use of GPSS.

### COMPUTER SIMULATION DEFINED

Because of the complex nature of modern business systems, data processing aids are increasingly required to assist the intuition and judgment of management in the evaluation of new methods, concepts, or designs. The practice of experimenting directly on a business and implementing a system before it is fully understood inevitably causes disruptions of normal operations, hasty last-minute corrections, and often personnel or customer resentment. To avoid costly mistakes, the consequences of change must be anticipated before actually implementing a program, and all alternatives should be thoroughly explored.

Computer simulation is a technique that provides an effective means of testing and evaluating a proposed system under various conditions in a laboratory environment. The system's behavior is modeled by a computer program, which reacts to various operating conditions in a manner quantitatively similar to the system itself. Several hours or weeks, or sometimes even years, of simulated activity can be examined on a computer in a matter of minutes. Results help to gain insights, test hypotheses, demonstrate or verify new ideas, establish feasibility, compare alternatives, design systems, or train personnel.

It is appropriate to any discussion on simulation to add a few words of both caution and encouragement. Computer simulation, like any simulation, is not a precise analog of an actual system. What is studied is the behavior of a representation of an actual system. Therefore, careful judgment must still be exercised by the user, both in setting up a good model and in interpreting the results from the simulation.

On the other hand, computer simulation frequently permits measurements which would be impossible to obtain in any other way, and allows the study of environmental situations of a scope far beyond the practicability of experimenting with an actual system. Such abilities as these immeasurably enhance the value of computer simulation in its role as an engineering and management-science tool.

### GENERAL PURPOSE SIMULATION SYSTEM

Computer simulation is recognized as a valuable tool for business managers, systems engineers, and functional specialists alike. Writing simulation programs from scratch, however, is a difficult, time-consuming task, requiring complex and extensive programming. To be most

useful, a simulation must be carried out quickly and be adaptable to change as the work proceeds. The General Purpose Simulation System greatly simplifies this task, and offers substantial additional values to the user. It is easy to apply, and no machine programming is required, nor is typical computer programming experience or training necessary. It is applicable to the study of a wide variety of situations ranging from bank teller queues, supermarket service, and job shop organizations to vehicular flow patterns, message-switching systems, etc. The program features a simple flowchart language for describing the problem or system to be simulated. When this description is transferred to punched cards and presented as input to the computer, the program automatically carries out the simulation of the system.

## Operating Highlights

To understand the operation and range of application of GPSS, one can begin with the familiar process of systems analysis.

The first step in the analysis of any particular system is to isolate the system's elements and formulate the logical rules governing their interaction. The resulting description is known as a model of the system. The model is limited to those aspects of the system which are of interest or appear to be pertinent to the analysis.

The progress of systems studies is greatly enhanced by the introduction of a concise systems language. To illustrate this, we consider two apparently unrelated systems.

In the first, ships arrive at a small port with a known arrival pattern. While in port, the ships unload some of their cargo, taking a certain amount of time, and then proceed on their voyage. There is only one pier, and if a ship arrives while another is unloading, it must wait. If several ships are waiting, the one that arrived first will be unloaded first. Of interest here is the total amount of time that a ship will spend in port, including the time spent waiting for the pier to become available.

In the second system, requests from retail outlets arrive at a warehouse where there is only one clerk to fill them. If requests occur too close together, a backlog builds up. These requests are processed in the order in which they arrive. The question here is: How long does it take a request to clear the warehouse?

Considering these two systems, several similarities can be seen. Both are characterized by units of "traffic" (ships, requests) arriving at a facility (pier, warehouse) requiring service. The facility can handle only one unit of traffic at a time, and if this facility is busy when new arrivals occur, these units must wait and form a queue or waiting line. Thus, three general elements are common to both systems: units of traffic, a facility, and a queue.

Also, the underlying logic of the two systems is identical. This may be demonstrated by means of a flowchart displaying system action. Figure 1 presents the simple harbor system described above, laid out in flowchart format. This shows the interaction of the pier and arriving ships. Figure 2 presents the simple warehouse system, showing interaction between the clerk and arriving requests. By replacing the terminology of harbors and ships with warehouses and requests (also changing specific time values), it is readily seen that Figure 1 also describes the logical operation of Figure 2, and vice versa.

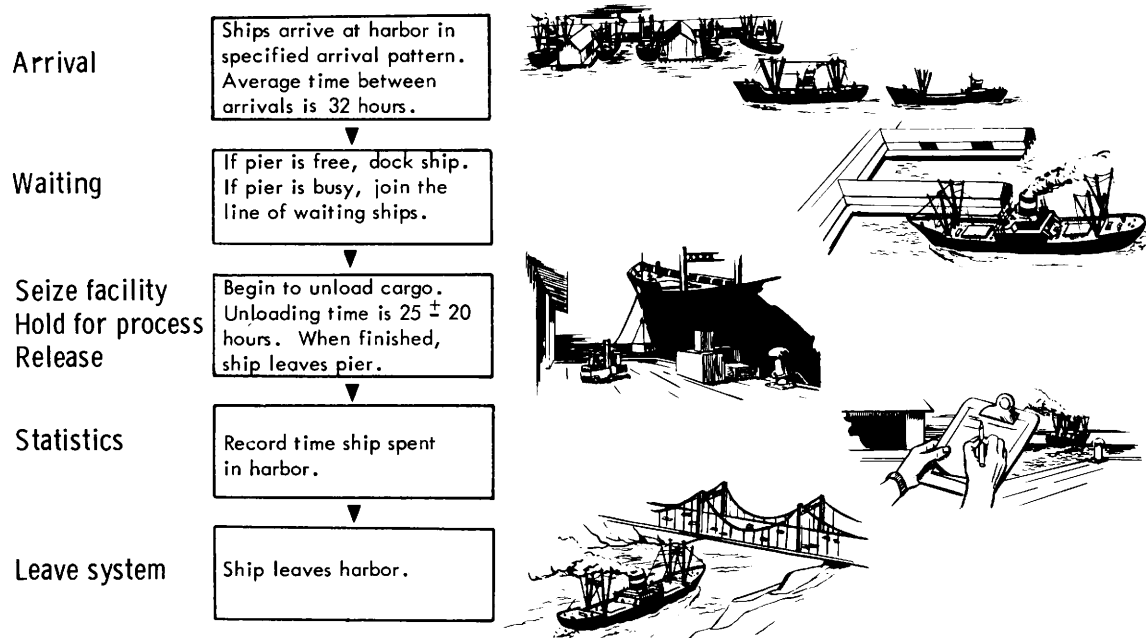


Figure 1. General flowchart for a simple harbor system

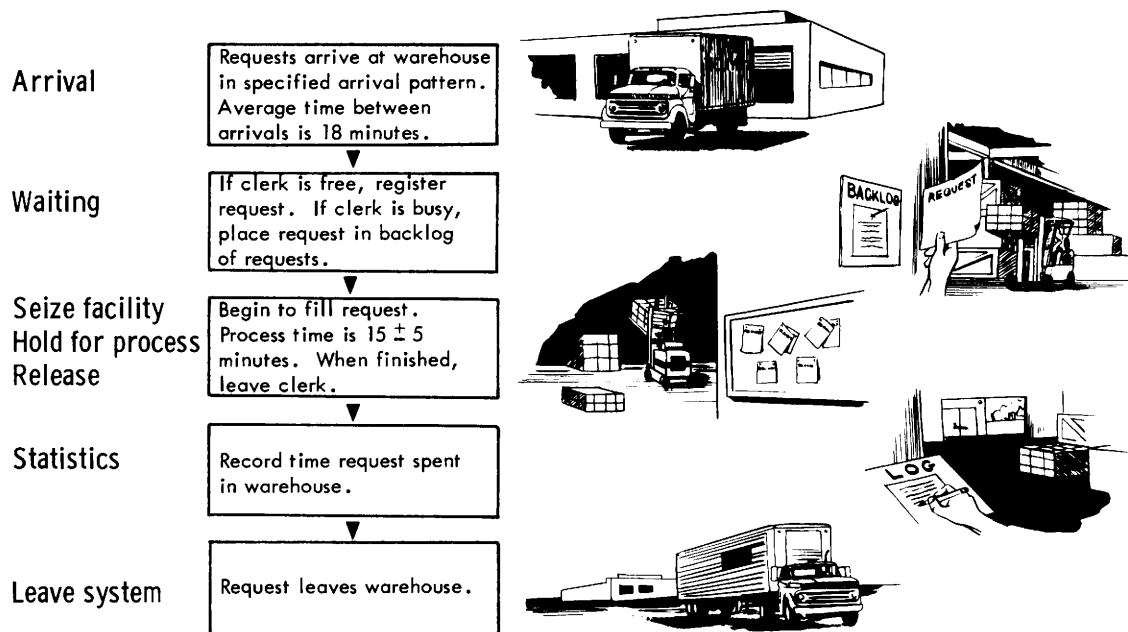


Figure 2. General flowchart for a simple warehouse system

After examining several such diverse and much more complex system models, it becomes evident that many generalizations concerning them can be made. The system elements of each, which appear so different on the surface, may be logically replaced by a small set of abstract elements called "entities". Likewise, the logical rules may be reduced to a common set of simple operations. Thus, a systems language can be developed containing abstract entities and operations involving these entities. By identifying these entities and operations with specific elements and logical rules in a particular system, a model of that system may be constructed in the general language.

The GPSS program provides such a general systems language. It is built around a set of simple entities, divided into four classes: dynamic, equipment, statistical, and operational.

The dynamic entities in GPSS are called "transactions". These represent the units of traffic, such as ships or requests in the previous examples. They are "created" and "destroyed" as required during the simulation run, and can be thought of as moving through the system causing actions to occur. Associated with each transaction are a number of parameters,

which can be assigned values by the user to represent characteristics of the transaction. For example, a transaction representing a ship might carry the amount of cargo it is to unload in a parameter. This number could then be used in the simulator logic to determine how long the unloading operation would take.

Entities of the second class represent elements of system equipment that are acted upon by transactions. These include facilities, stores, and logic switches. A facility can handle only one transaction at a time, and could represent the simple pier or warehouse in the examples given. It represents a potential bottleneck. A store can handle several transactions concurrently, and could be used to represent a parking lot or a typing pool. A logic switch is a two-state indicator which can be set by one transaction to modify the flow of other transactions. It could model a traffic light or the "next window" sign of a bank teller.

In order to measure system behavior, two types of statistical entities are defined: queues and tables. Each queue maintains a list of transactions delayed at one or more points in the system, and keeps a record of the average number of transactions delayed and the length of these delays. A table may be used to collect any sort of frequency distribution desired. These two entities provide a major portion of GPSS output.

The operational entities, called "blocks", constitute the fourth and final class. Like the blocks of a diagram, they provide the logic of a system, instructing the transactions where to go and what to do next. These blocks, in conjunction with the other three classes of entities identified above, constitute the language of GPSS.

As an example of this language, the simple harbor system outlined in Figure 1 is diagrammed, using conventional GPSS symbols as shown in Figure 3. Each box represents a specific GPSS block, with its name and usually the number of a referenced entity.

To provide input for the simulation, control and definition cards are prepared from a flowchart of the system. This constitutes the model in GPSS language. Once the system model is loaded, the GPSS program generates and moves transactions from block to block according to timing information and logical rules incorporated in the blocks themselves. Each movement is designated to occur at some particular point in time. The program automatically maintains a record of these times, and executes the movements in their correct time sequence. Where actions cannot be performed at the originally scheduled time — for example, when a required facility is already in use — processing temporarily ceases for that transaction. The program automatically maintains a status of the condition causing the delay, and as soon as it changes, the transaction is activated again.

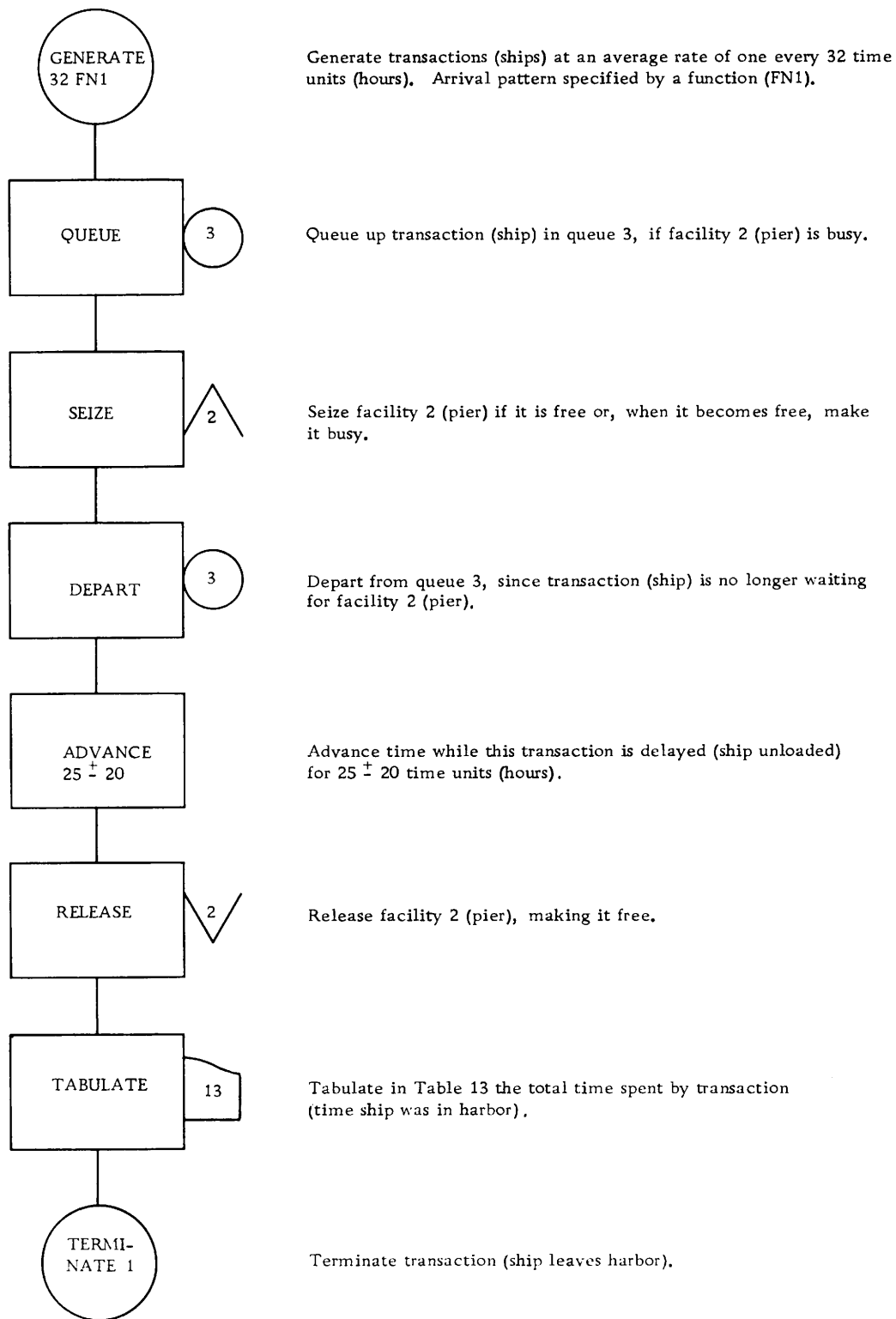


Figure 3. GPSS flowchart for the simple harbor system

This sequence of events is controlled by a simulated clock that records the current time reached in the modeled system. Values shown by this clock are referred to as clock times. The unit of simulator clock time representing a unit of system time is designated by the user. For example, in Figure 3 the unit of clock time equals one hour.

Many more system complexities can be modeled than are illustrated by the example. Priorities can be assigned to selected transactions, and complex logical decisions may be made throughout a simulation. Probability distributions of input variables may be introduced into the model, and provision is made to gather statistical output with ease.

Output from the program provides information on:

- The amount of transaction traffic flowing through the complete system and/or any of its parts
- The average time for transactions to pass through the complete system or between selected points, and the distribution probability of this passage time
- The degree to which each item of equipment in the system is loaded, together with the distribution of storage occupancy
- The maximum and average lengths of queues occurring at various points, as well as their distribution

## APPLICATION SCOPE

GPSS/360 can be applied in most industries to solve a variety of problem situations. In general, these problems have one characteristic in common: transactions, people, or equipment are competing for services of other people or equipment, and it is of interest how well the service organization will respond to the demands. GPSS is designed to offer direct assistance in the investigation of this type of problem. Some representative examples of use are:

1. Telephone company — design of a system to automatically intercept and service telephone calls that cannot be put through because new numbers have been assigned, or because units have been disconnected.
2. Brokerage firm — evaluation of message handling between dispersed local offices and the main office. Different system networks are produced by varying the allocation of offices to circuits.
3. Scientific computer centers — synthesizing of alternate computer configurations, supervisory monitor programs, and operating procedures to handle problems in a central computer complex arriving from multiple sources.

4. Supermarket — appraisal of traffic flows throughout the store, and analysis of queues at checkout counters.
5. Manufacturing shop — evaluation of the effect of alternate scheduling decision rules in the sequencing of parts and assemblies through a discrete product shop.
6. Automobile traffic — simulation of vehicle flow patterns along roads and through intersections and toll gates to determine properly sequenced lane segments.
7. Banks — analysis of the dynamics of arrivals and departures at teller windows.
8. General business — examination of the flow of information through a business organization, and its impact on operating performance.
9. Steel mill — simulation of integrated physical and information systems, and examination of useful alternative decision procedures applied to the control of steel production.
10. Hotels — handling of multiple hotel reservations, registration, and billing under a real-time data processing system.
11. Warehouses and distribution facilities — study of truck arrivals and departures in a large warehouse complex, and analysis of product composition and loads being distributed to remote reception points.
12. Product design — evaluation of design problems for electro-mechanical devices.
13. Railroads — analysis of diesel power requirements in terms of load characteristics and the time when loads become available to be moved.
14. Teleprocessing — design of control systems for remote-computing and message-switching networks to handle large volumes of randomly occurring data efficiently and quickly.

In this section, which lists and describes some of the additional features provided by GPSS/360, a basic knowledge of GPSS is assumed. Since GPSS/360 provides the features and capabilities available in GPSS III, the reader is referred to the Bibliography for further detail. The additional features are categorized into three areas:

1. GPSS/360 Assembly Program Extensions
2. New Entities, New and Extended Block types and other Expanded Features
3. Output Editor

## GPSS/360 Assembly Program Extensions

GPSS/360 Assembly Program provisions have been extended to include symbolic addressing of all entities, a cross-referencing dictionary of symbolic block and entity assignments, a MACRO operation, and a system UPDATE feature. These extensions facilitate coding and debugging of simulation models, relieve the user of much unnecessary bookkeeping, and considerably enhance the user's ability to make modifications to a simulation model. These feature capabilities are described below.

### SYMBOLIC ADDRESSING OF ENTITIES

In general, wherever a specific entity type is implied by a block field, a symbol may now be used to reference the entity. In fields where the entity type is not implied, the symbol must be preceded by a character or characters that identify the entity type and a \$. The Assembly Program assigns unique numbers to each entity.

### CROSS-REFERENCE DICTIONARY

In the GPSS/360 Assembly Program output, the block symbol table now includes a cross-reference dictionary. Following each symbol is a list of the blocks in which the symbol is referenced. The output also includes a table of entities that were defined symbolically within the model. The symbols are grouped according to entity type, and the numerical value assigned to the symbol by the Assembly Program appears following the symbol itself.

### MACRO OPERATIONS

A GPSS/360 Assembly Program MACRO operation is provided which permits parallel or nearly identical parts of programs to be coded only once and then used at any point in the model by inserting a card calling the desired macro definition. Each macro instruction may have up to 10 arguments associated with it. This enables the user to vary the function of the macro each time it is called.

### SYSTEM UPDATING

An UPDATE feature is provided that enables the user to place symbolic model(s) on tape and subsequently make modifications to the model by updating the master tape. This eliminates the need to maintain large symbolic decks and submit these decks each time a simulation run is desired.

### ERROR CHECKING

More extensive error checking than in previous GPSS versions is now done in the assembly phase. Many errors which were formerly discovered in execution are now detected earlier. Most important among these are the detection of nonexistent entities, such as SEIZE 400 (where core allocation does not allow for 400 facilities). Reference to an undefined FUNCTION or VARIABLE will also be detected before execution.

## New Entities, New and Extended Block Types, and other Expanded Features

### EXTENSION OF PARAMETER MAGNITUDES

Each of the parameters of a transaction may now have positive or negative integer values. In addition, the values of all parameters specified in the creation of a transaction at a GENERATE block may have the conventional magnitude limit of  $2^{15}-1$  or the expanded maximum magnitude of  $2^{31}-1$ . The parameter limit is specified in the GENERATE block. Since these parametric values are used extensively in simulation models to give a multi-faceted identity to a transaction, the ability to assign values as large as  $\pm 2, 147, 483, 647$  to these parameters greatly increases the flexibility of GPSS.

### INDEPENDENT RANDOM NUMBER GENERATORS

Eight random number generators completely independent of each other are provided. Since the user may reference any of these sources, this feature allows him to relate independent random sequences to various sections of a model.

### USER CHAIN STATISTICS

Standard output statistics are provided for each referenced User Chain. The interpretation of these statistics depends on the model. For example, if a User Chain were being used to represent inventory, the statistics would represent maximum number of items in inventory, average number of items in inventory, total number of items in inventory over a period of time, and the average time an item was in inventory.

### PREEMPT BLOCK PRIORITIES

The operation of the PREEMPT block has been extended to allow PREEMPTing based on the priority of the transaction entering the PREEMPT block. If the PREEMPTing on priority option is not specified by the PREEMPT block, its operation will be the same as the operation of the PREEMPT block of GPSS III. GPSS/360 provides 128 priority levels. This simplifies the modeling of systems such as computer systems, message-processing systems, manufacturing, etc., where the ability to obtain an item of equipment is determined solely by the priority associated with the request.

Another new feature of the PREEMPT block is the ability to obtain access to the PREEMPTed transaction so that alternate processing may be used.

### MULTIPLE QUEUE STATISTICS

A transaction is now allowed to enter multiple QUEUE blocks. With this ability, statistics for multiple queues may be obtained by passing a transaction through a sequence of QUEUE blocks and subsequent DEPART blocks. ADVANCE blocks will also be allowed between QUEUE and DEPART blocks without destroying Queue statistics.

## VARIABLE STATEMENTS

To increase the logical power and capabilities of GPSS, Boolean Variable statements are provided. This makes it possible to make decisions at a single GPSS block based on the status and value of many GPSS entities. The elements making up the Boolean Variable are interpreted as a 1 if nonzero and 0 if zero. Conditional statements are allowed as elements of the Boolean Variable, such as (X10 G 500), which is interpreted as a 1 if the contents of SAVEVALUE 10 is greater than 500 and a zero if less than or equal to 500. Parentheses, indirect addressing, and the Boolean operations AND(\*) and OR(+) are allowed in Boolean Variable statements.

The standard arithmetic VARIABLE statement now permits the use of parentheses to group terms or to denote multiplication. Furthermore, the user has the option of defining a VARIABLE statement as "floating point", in which the elements are not truncated before use. Likewise, the results of intermediate arithmetic computations are not truncated in a floating-point VARIABLE.

## THE GROUP ENTITY

To establish a means of grouping transactions or other entities, a new entity type, Group, is provided. A Group is a list of numbers. The meaning of a Group depends on what elements constitute a Group and how the analyst creates, manipulates, and removes members of a Group within the model.

A Group and reference to members of a Group is completely independent of the status of the members comprising the Group. Transactions that make up a Group can all be referenced regardless of where they are in the system. For example, policemen (Transactions) may be categorized by the foreign languages they speak, with all Spanish speaking policemen in one Group, all Russian speaking policemen in another Group, etc. Regardless of where Russian speaking policemen are in the system — that is, no matter how their residence is split among Future, Current, Interrupt, or User Chains — they may all be referenced as a single entity.

In addition to Transactions, other entities such as Facilities, Storages, etc., may also be classified in Groups. This classification may be made before and during execution of the model. For example, machines scheduled for preventive maintenance could be initially specified as members of a given Group, subsequently removed from this Group, and then associated with some other Group. Also, entities may be members of several Groups simultaneously.

## MATRIX SAVEVALUES

Matrix Savevalues provide the ability for the user to associate additional attributes with GPSS entities such as Facilities, Storages, Logic Switches, User Chains, etc. For example, if a facility represented a machine, certain characteristics (such as number of failures, number of different job types processed, time the machine was last used, job type that the

machine is currently processing, time machine is scheduled for preventive maintenance, length of time current job is scheduled on the machine, etc.) might be of interest to the user.

A Savevalue may be a fullword (32 bits) or a halfword (16 bits). The Matrix Savevalue will be arranged, as specified by the user, in an M x N matrix, where M is the number of rows and N is the number of columns. In the above example, we may designate the row as the machine number and use the columns to represent machine attributes. Another use of Matrix Savevalues is the specification of operation sequences as in a job shop model. Each row may represent an operation, and the first entry in a row may represent the name of the operation. Subsequent entries in that row could be used to represent operational parameters.

#### COUNT AND SELECT

Two new block types provide a means for COUNTing and SELECTing based on the status of GPSS entities. This eliminates the need for complicated block sequences formerly required to perform this function. The COUNT block has the ability to count the quantity of a given entity type that meets a specified condition; for example, the number of facilities not in use. The SELECT block would test the same entities and status as those specified for the COUNT block, but would select the first entity that meets the specified status condition.

#### SELECTIVE RESET AND CLEAR

The RESET and CLEAR cards may now specify entity statistics that are not to be RESET or CLEARed. Options in the RESET card include statistics for facilities, storages, queues, tables and user chains. The CLEAR card may specify SAVEVALUE locations that are not to be CLEARed.

#### NEW SYSTEM NUMERICAL ATTRIBUTES

To provide further convenience and versatility, the following additional statistical System Numerical Attributes (SNA's) are provided:

##### Transactions:

PR Priority of the transaction currently active.

##### Facilities:

FRn Utilization of facility n in parts per thousand. For example, if the utilization were .88, the value of FRn would be 880.

FCn Number of entries for Facility n.

FTn Average time each transaction used Facility n. When referenced, the computed value will be truncated. For example, if FTn were 1.23, the computed value would be 1.

Storages

- SRn Utilization of Storage n in parts per thousand. For example, if the utilization were .65, the value of SRn would be 650.
- SA<sub>n</sub> Average contents of Storage n (truncated).
- SM<sub>n</sub> Maximum contents of Storage n.
- SC<sub>n</sub> Number of entries for Storage n.
- ST<sub>n</sub> Average time each transaction used Storage n. When referenced, the computed value will be truncated.

Groups:

- G<sub>n</sub> Number of items in Group n.

User Chains:

- CA<sub>n</sub> Average number of transactions on User Chain n (truncated).
- CH<sub>n</sub> Current number of transactions on User Chain n.
- CM<sub>n</sub> Maximum number of transactions on User Chain n at any one time.
- CC<sub>n</sub> Number of transactions (entries) that were on User Chain n.
- CT<sub>n</sub> Average time each transaction was on User Chain n. When referenced, the computed time will be truncated.

Queues:

- QA<sub>n</sub> Average contents of Queue n (truncated).
- QM<sub>n</sub> Maximum contents of Queue n.
- QC<sub>n</sub> Number of entries in Queue n.
- QZ<sub>n</sub> Number of zero entries in Queue n.
- QT<sub>n</sub> Average time each transaction was on Queue n (including zero entries). When referenced, the computed time will be truncated.
- QX<sub>n</sub> Average time each transaction was on Queue n (including zero entries). When referenced, the computed time will be truncated.

Tables:

TCn        Number of entries in Table n.

TDn        Standard deviation of Table n.

Matrix Savevalues:

MXk (m, n)    Value at row m, column n, of Matrix Savevalue k.  
Each of the entries m, n, and k may be indirectly  
addressed.

## Output Editor

The GPSS/360 output editor provides a means of editing and specifying a simulation output format that will be most meaningful to the user. The editor will provide three different output options. Plotting capabilities are available with any of these options. The choices of output are:

- Standard output
- Standard output with user-specified titles and/or deletions
- User-specified format, titles, and content

Under all three options all entities defined by symbolics, except BLOCKS, will have the symbolic name listed in the GPSS/360 output rather than the numeric value assigned by the GPSS/360 Assembly Program.

The standard output is similar to that provided by GPSS III. The second option allows the user to substitute or add titles in various portions of the output and also to select those entity attributes that should be listed. In the third option, all alphameric information and entity statistics are explicitly specified or requested by the user, allowing him complete flexibility in format and contents.

The user may also request graphic output that provides a pictorial representation of GPSS/360 entity statistics. Each graph consists of a 60-row by 132-column matrix. By use of the appropriate control cards the user will have the ability to specify the:

1. Origin for the X and Y axes.
2. Entity SNA to be plotted.
3. Range of the entity type — for example, Facility 1 through Facility 10, etc.
4. Scale factors to be assigned to the X and Y axes.
5. Graph type — continuous or bar type.
6. Titles, units, etc. — a list of alphameric information and its location on the graph.



## SYSTEM REQUIREMENTS

Minimum System Requirements for OS/360: A 2030 Model F with Universal Instruction Set; 1442, 2501, 2520, or 2540 Card Reader Punch; 1403, 1404, 1443 or 1445 Printer; and two 2311 Disk Storage Drives.

Minimum System Requirements for DOS/360: A 2030 Model F with Universal Instruction Set; 1442, 2501, 2520, or 2540 Card Reader Punch; 1403, 1404, 1443 or 1445 Printer; and either two 2311 Disk Storage Drives or one 2311 and four 2400 series Magnetic Tape Units.

Additional core storage will permit the construction of larger, more complex models. Additional logical devices are required if the WRITE-JOBTAPE, UPDATE or READ-SAVE features of GPSS/360 are used.

The normal quantity of GPSS/360 entities provided may be selected by the user by means of a control card submitted with the input deck. The normal quantity of GPSS/360 entities provided for main core storage of 64K, 128K, 256K and upward is given in the following table.

### NORMAL QUANTITY OF GPSS/360 ENTITIES

<u>Entity Type</u>	<u>64K</u>	<u>128K</u>	<u>256K and upward</u>
Transaction	200	600	1200
Blocks	120	500	1000
Facilities	35	150	300
Storages	35	150	300
Queues	70	150	300
Logic switches	200	400	1000
Tables	15	30	100
Functions	20	50	200
Variables	20	50	200
Savevalues (fullword)	100	400	1000
Savevalues (halfword)	50	200	500
User chains	20	40	100
Groups	5	10	25
Boolean Variable	5	10	25
Matrix Savevalue (full)	5	10	25
Matrix Savevalue (half)	5	10	25

The estimated number of S/360 bytes allocated GPSS/360 entities, Program and Common are as follows:

MAIN CORE STORAGE (BYTES)	<u>64K</u>	<u>128K</u>	<u>256K</u>
GPSS/360 Storage Allocation	44K	88K	162K

The automatic reallocate feature of GPSS/360 provides the user the ability to reallocate the normal set of entities in a manner similar to the method used in GPSS III to conform to individual model requirements.

For example, additional blocks may be obtained by reducing the available number of unused entities, so that relatively large models may be constructed using the smaller core sizes. Furthermore, by means of this feature, core storage of 512K and upward may be fully utilized for GPSS models.

GPSS/360 is written to run as a normal batch job under control of OS/360 or DOS/360, requiring no operator intervention. All diagnostics are executed automatically when required. A general system chart for the program is given in Figure 5.

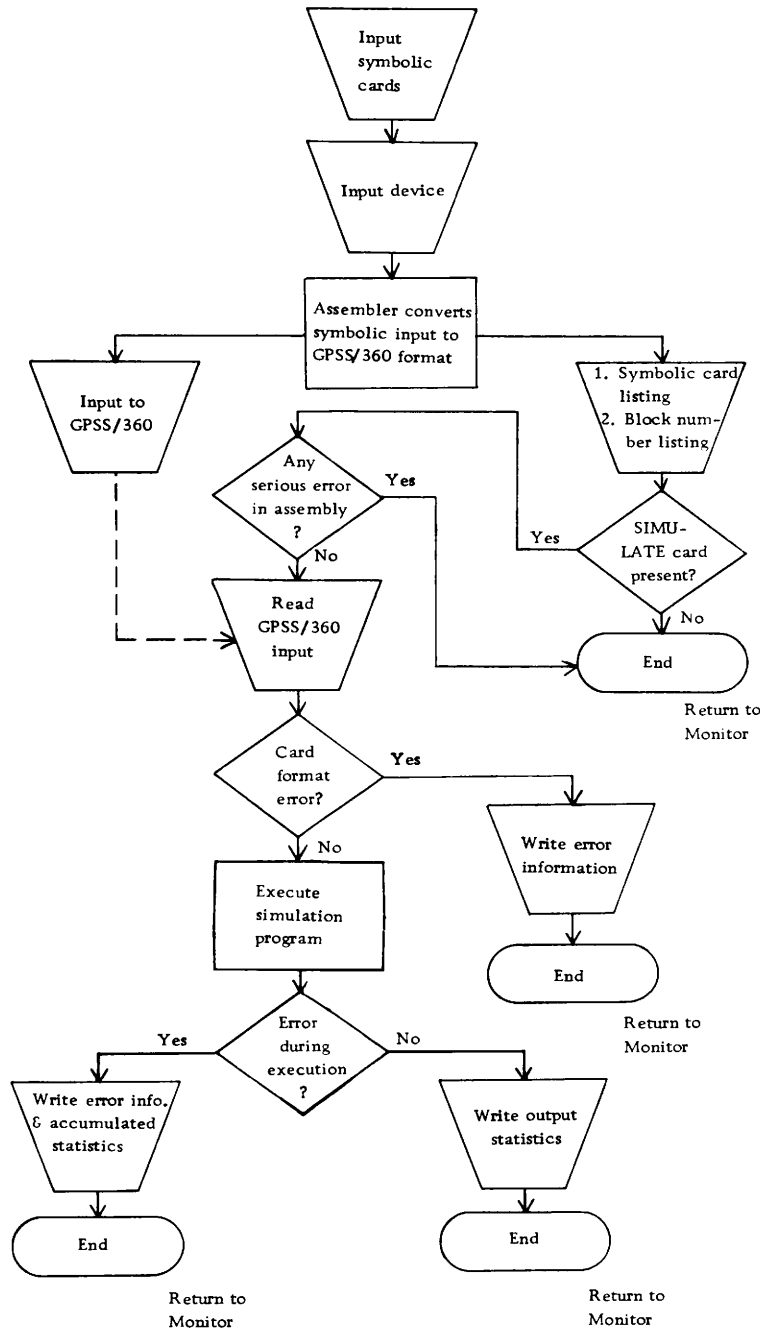


Figure 5. GPSS/360 general system chart

## BIBLIOGRAPHY

Gordon, G., "A General Purpose Systems Simulation Program", Proceedings of the Eastern Joint Computer Conference. (Washington, D.C.), December 1961, The Macmillan Company.

Gordon, G., "A General Purpose Systems Simulator", IBM Systems Journal, Volume One, September 1962. (Also reprinted by IBM under form number 320-1714.)

Herscovitch, H., and Schneider, T.H., "GPSS III — An expanded General Purpose Simulator", IBM Systems Journal, Volume Four, Number Three, 1965.

General Purpose Simulation System/360, Introductory User's Manual, IBM manual (H20-0304).

General Purpose Simulation System/360, User's Manual, IBM manual (H20-0326).

**IBM**

International Business Machines Corporation  
Data Processing Division  
112 East Post Road, White Plains, N.Y. 10601  
(USA Only)

IBM World Trade Corporation  
821 United Nations Plaza, New York, New York 10017  
(International)