S3, THE SYSTEM AND SOFTWARE SIMULATOR

ABSTRACT

By Leo J. Cohen
MetaSystems, Inc.
Trenton, New Jersey

The System and Software Simulator (S3) is a computer program written entirely in Fortran IV and capable of execution on any computer having that compiler available. Thus far S3 has been applied to computer system simulation problems while executing on the Univac 1108 and the IBM system 360, models 50, 65, and 75.

The purpose of S3 is threefold; in the first place it provides a convenient computer oriented language for the specification of the total hardware/software environment of a computer system. Secondly, it generates an executable digital event simulation model of the specified system, and thirdly, operates the generated model for some specified length of simulated time, producing performance statistics for all hardware and software aspects of the simulated system.

S3 was developed originally for the Army Computer System Evaluation Command, Fort Meyer, Virginia, and has undergone extensive evolutionary development since its acceptance. At this time S3 provides an English-like computer oriented specification language for describing all software and hardware aspects of the computer system to be simulated. The hardware description is in four categories: peripheral devices, controllers and channels, memories, CPUs. In each of these categories there is both a performance and a connectivity description. Thus, for example, a magnetic tape is described in terms of its data storage unit (BIT, BYTE, WORD, etc.), the inter-record gap, its stop/start time, and if applicable, the maximum allowable time interval between successive I/O references after which the stop/start time must be invoked. Rewinding and mounting times may also be made part of a magnetic tape specification. For a printer, the number of characters per line, number of lines per second, line and page spacing time, stop/start time, etc., are specified. In the case of a random access device the specification calls for such items as storage unit, latency time, seek time if applicable, and so on. It is to this degree of detail then, that each peripheral equipment of the computer system under simulation is specified. In addition, each such specification designates the controller to which it is connected, and each controller designates the channel or channels with which it is in turn connected.

The specification of a channel is again, in terms of the data unit transmitted and the rate of that transmission. Channels are designated as selector or multiplexor, and in the latter case, specify both a selector and burst mode performance. Selector channels may be simplex, half duplex or full duplex, and the specification for all channels designates the CPUs by which they may be operated.

CPU performance is specified in terms of cycle rate, average instruction time (via Gibson Mix), and add, multiply, and divide time for any selection of decimal, fixed point and floating arithmetic. The CPU specification, in addition to designating channel connections, also indicates to which of the simulated core memories it has access.

Core memory performance is specified in terms of access rate, units of data storage, data access and instruction access, and total size. The CPU/memory connections are also specified.

With the foregoing specification of the hardware environment of the system, single processors, multiprocessors, multicomputers, mixtures of these and networks of computers may be

specified for a simulation. Three configuration examples are shown in figures 1, 2 and 3. As in a real system, however, in addition to the hardware it is necessary to have an operating system specification. In order to provide a capability for specifying the operating system, or systems, of the simulated computer, S3 makes available a formal "programming language" for expressing the flow chart logic of the operating systems involved. Thus, for example, there is a PLACE statement for inserting a program ID into a specified queue. This is one of a number of queue manipulation statements. There are five basic algorithms for core memory allocation available in S3, and statements for testing, packing, allocating, and deallocating the simulated memory allow for all but the most exotic allocation procedures. Other statements in the S3 operating system "programming language" provide for the setting, testing, and resetting of local and global switches, conditional and unconditional transfer of control, testing for end of file, loop control, and so on. In addition, a CPU may generate interrupts for itself, or interrupts for any other specified CPU of the system. Finally, there is a "programming mechanism" for the representation of all interrupts to each CPU, and the consequent actions taken on the occurrence of such interrupts. This technique allows for the representation of such actions as OPEN/CLOSE, READ/WRITE, PRINT, SEEK, CLOCK, and so on, plus other, arbitrarily assigned interrupt conditions. Figure 4 shows some of these statements.

The simulated operating system may be separated into submodules which are then overlayed and/or multiprogrammed as required by the actual system under simulation. Thus, for example, the simulated operating system may include a program for job request interpretation which is assumed to be generally nonresident. This program can then be called for by a simulated external interrupt, or by the simulated operating system, loaded into the simulated memory (if not already present) and executed according to the program selection algorithm of the operating system itself.

With a generated simulation model in hand, the next step is to input representations of the application programs whose simulation is to be executed. Again, S3 provides a "programming language" for the representation of application program flow charts. This provides a comprehensive set of input/output statements for file opening and closing, reading and writing, printing, seeking, etc. Subprograms may be called, jobs may be selected and initiated, and the parallel processing of subprograms and jobs may be initiated and controlled. Switches may be manipulated, conditional and unconditional transfers made, and so on. S3 also provides a "job control language" for application programs which describes the files that will be used, the distribution of these files on peripheral equipments (for random access devices, their specific distribution within such devices), the structure of such files, and their local program names. Files are designated for input or output, and the degree of buffering, if any, is specified.

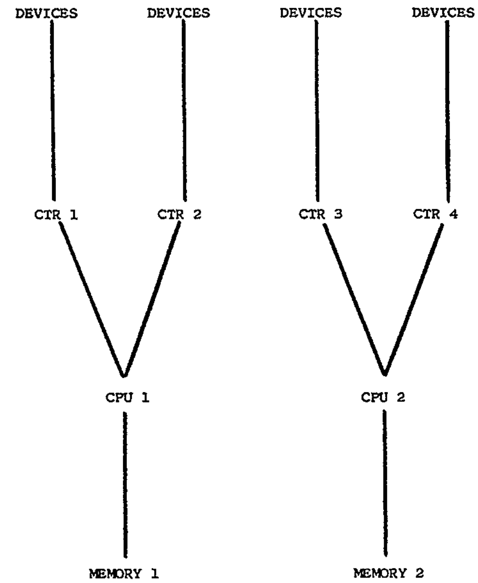
The "job control language" also designates the amount of core space required by the application program for instructions, constants, and data separately. During the course of its execution an application program may increase or decrease its allocated space, may terminate at any selected point, or create the call for some other job at a point in future simulated time. The "job control language" also specifies whether the application program is reentrant, and finally, designates the frequency of the job's appearance in the simulated computer system, with such intervals either fixed or determined statistically.

The models of the application programs are input in the S3 "programming language" to the generated computer system simulation program. This program is then executed for a specified amount of simulated time and results in a wide range of statistics covering the performance of each and every facility of the hardware/operating system environment. This includes statistics for device, controller, channel, CPU and memory performance, the time/space performance of all specified queues, a statistical summary of operating system idle time, cycle time, overhead, and application program load, and a complete, detailed summary of all interrupt activities, for each interrupt, by CPU. The hardware statistics include detailed information concerning the specified memories in conjunction with the allocation schemes employed by the operating system. Thus, time/space statistics on memory utilization are made available, the number of times that memory allocation was deferred, the number of times memory had to be repacked (if that is part of the operating system algorithm), and the average amount of remainder memory capacity. Figure 5 is a summary of the overall S3 process.

For each application program whose execution is simulated, S3 provides both turnaround time and throughput figures. These figures break down the total amount of simulated time into that allocated for CPU, I/O, and queue delay for each application program. The minimum, maximum and average times in these categories for all replications of a given application program is shown, along with the minimum, maximum and average number of file references made by all instances of this program's execution for each file referred to by the program. A sample output for an application program is shown in figure 6.

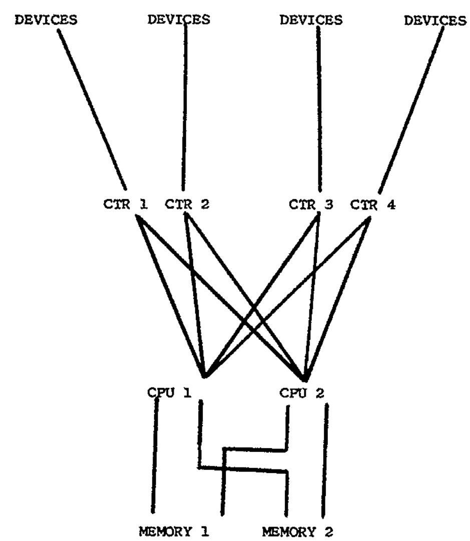
S3, the System and Software Simulator, has been effectively applied in a number of design and analysis situations. It has been used to design two real-time airborne hardware/operating system configurations for a very large multiprocessor. It has also been employed in the design of an operating system for an already existing process control computer. This operating system was developed to provide a multiprogramming control over the process functions and at the same time to make available the remainder capacity of the computer for the execution of general, or non-process control, application programs.

S3 has also been used in the analysis of performance of existing and proposed systems. In particular, it has been applied to the IBM 4Pi, the Burroughs D84T, and the Litton 3050M computers. A model of the GE 625 using GECOS2 has also been developed and successfully run, and at the present time efforts are underway to develop a System 360 model employing the MVT operating system.



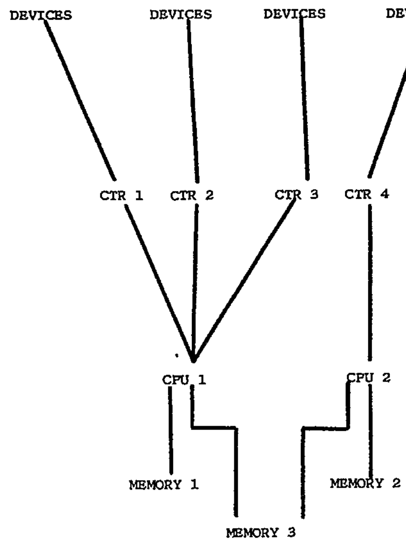
MULTICOMPUTER

FIGURE 1



MULTIPROCESSOR

FIGURE 2



MIXED SYSTEM

FIGURE 3

THE S3 PROCESS

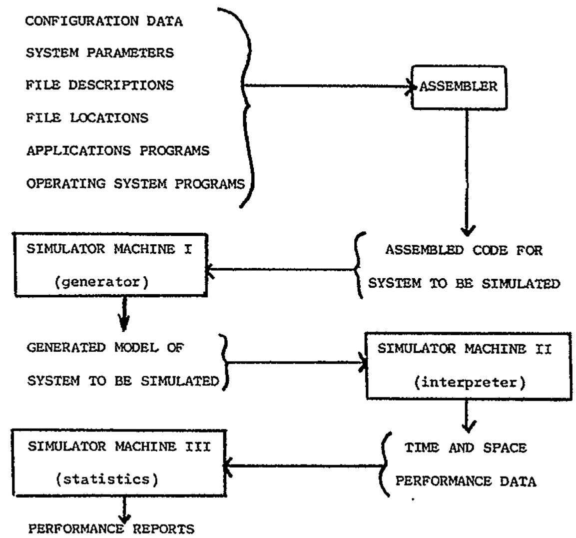


FIGURE 5

MEMORY	A,B,C	used to query the current state of the memory map. The parameters specify conditions for the query
ALLOCATE	A,B	used to assign memory and modify the memory map
DEALLOCATE	A,B	reverse the functions of ALLOCATE
PACK	A,B	to modify the memory map without allocation
EXAMINE	A	examine queue A
PLACE	A,B	place program or I/O call A into queue B
SELECT	A,B	select program or I/O call A from queue B
BUFFER		used to control buffering
SEEK	A	used to locate a named file on its device

FIGURE 4

WORKER ROUTINE NAME: PERTRT

NO. OF GENERATED TRANSACTIONS: 8

NO. OF CALLED TRANSACTIONS: 12

FREE RUNNING TIME: 9.116787

	MINIMUM	AVERAGE	MAXIMUM
TURNAROUND TIME	9.863741	17.168611	56.001099
NORMALIZED	1.082	1.883	6.143
CPU TIME	6.526128	7.512069	8.116787
T/I QUEUE TIME	.498839	5.325928	18.044619
IOT QUEUE TIME	.019146	2.007327	22.130671
NO. OF REFERENCES ORDINAL FILE #			
1	112	123	196
2	224	246	392
3	14	14	12
4	16	17	26

FIGURE 6