International Conference
on
Information Processing
and
Auto-math 59

June 1959

R/W 40 Data Processing System

S. Rothman

RAMO-WOOLDRIDGE
a division of Thompson Ramo Wooldridge Inc.
5500 West El Segundo Boulevard
P. O. Box 90534 Airport Station
Los Angeles 45, California

I. INTRODUCTION

It is a pleasure and privilege to address this distinguished group, and an honor for Thompson Ramo Wooldridge, Inc., to be represented here today. The purpose of this presentation is to describe a particular data processing problem and our solution to it in the hope that the analyses and design solutions may be of assistance to others facing these or similar problems.

Before I begin, I would like to acknowledge the technical and financial support of the Rome Air Development Center, Air Research and Development Command of the United States Air Force. A system such as this represents the efforts of almost one hundred men, all of whom deserve specific mention. In particular I am grateful for the guidance of E. E. Bolles and D. L. Drukey, and the fine technical contributions of L. Amdahl, R. Perkins, R. Ryle, A. Scarbrough, M. Shiowitz, and G. West.

In this talk, I will discuss the overall systems problem and system solution before discussing the RW-40 design. Then I will present some of the supporting analysis. Finally, some of the applications will be covered.

II. SYSTEMS PROBLEM AND SOLUTION

The systems design problem was a reasonably complex one. A data processing system had to be designed with the capability of handling an extremely large amount of data on a continuous basis with very short throughput times. It was known that the overall volume of data to be processed increased with time and that the time allowed for processing decreased. Both the high-speed scientific type of problem and the commercial variety had to be solved, and they were sufficiently interrelated to require access to the same data. A single existing computer could not meet the time requirements on this number of different problems. In fact, seven or eight of the biggest, fastest computers available would have to be modified to meet this requirement. Such a modification would cost approximately 30 million dollars. It was quite apparent that parallel capacity was required for simultaneous execution of multiple problems. Finally, the performance requirements were such that the data processing system had to provide real time service to the individual operators on a request basis.

Both cost and time limitations necessitated that the system be developed using current techniques. However, the system had to be capable of expanding so that it could grow nondisruptively and include, as required, new developments in the field.

The basic requirements could be met satisfactorily through the use of multiple computing modules and a new and general organization of these modules. The organization problem was this: it was necessary to control the multiple computers, interconnect them for problem solution, let the divorced groups operate independently, and then disconnect them for reallocation. More specifically, the system would incorporate: (1) Multiple modules of the various basic types, (2) an expandable Central Exchange

for interconnecting these modules for data transfer, (3) a flexible alert capability, i. e., the control communication required for the type of organization, (4) a Master Control Program, and (5) a special buffered input-output system.

The RW-40, therefore, employs a number of small, self-contained,low cost modules capable of high-speed processing. In general these modules are memory, processing, and buffering types. Multiples of each of these are used in the system. These modules are interconnected through a Central Exchange into the various configurations needed to solve each problem. This Central Exchange determines the number of working modules that can be used in the system, and since the Central Exchange is itself expandable, it allows for the addition of modules into the system as the needs dictate. The Central Exchange can be controlled automatically by the Master Computer and the processing modules.

Thus, the RW-40 can allocate its own capacity to handle problems under the Master Control Program.

The use of multiple modules contributes toward the reliability necessary for continuous operation. System reliability is enhanced because the problems assigned to a malfunctioning module can readily and automatically be reassigned to other identical modules, and the faulty module replaced or repaired on an off-line basis. Thus continuous operation is possible.

The buffered input-output system provides each operator with access to the data processing system. The Master Computer, in conjunction with the interrupt feature and Central Exchange, sets up the processing configuration required to service the request. Therefore it is possible for the operator, through his input-output device, to access the computing system and obtain information or display within the real time of his needs.

In summary, the five features of the system are: (1) Multiple modules of the various basic types, (2) an expandable Central Exchange for interconnecting these modules for data transfer, (3) a flexible alert capability, i. e. , the control communication required for the organization, (4) a Master Control element, and (5) a special buffered input-output system.

III. DATA PROCESSOR DESIGN

A. GENERAL

A description will be given now of the RW-40 system (shown in Figure 1) and its modules with special emphasis on unique aspects of the modules. There are three principal elements of the system: the input-output devices, the input-output buffering, and the processing system itself.

B. BRIEF DESCRIPTION

The input-output devices include the general purpose Flexowriter, a rather flexible console for communication with the RW-40; a printer for high-speed output; and a plotter. The console has keys which can activate particular programs and a number of cathode-ray tubes for display purposes which are refreshed by a magnetic drum, designated the Display Buffer. The printer is a 900-line per minute Analex printer, and the plotter is an Electronics Associates plotter.

The input-output buffer, the Peripheral Buffer, is a magnetic drum plus control circuitry which handles up to 32 input-output stations at 30 characters a second. There is no limit to the number of Peripheral Buffers which may be used in the system. The Peripheral Buffers produce alert signals when in use. The Display Buffer, as previously mentioned, refreshes the information on the cathode-ray tubes at the console at the rate of 15 frames per second. An intermediate speed Peripheral Buffer, ranging over 30 to 10,000 characters per second in speed, is being considered. This is, of course, a magnetic core device.

The Computer Module is a general purpose computer that uses the 26-bit word, and contains 1024 words of core memory. Its add time is 40 microseconds. It has a two-address order structure with a hold

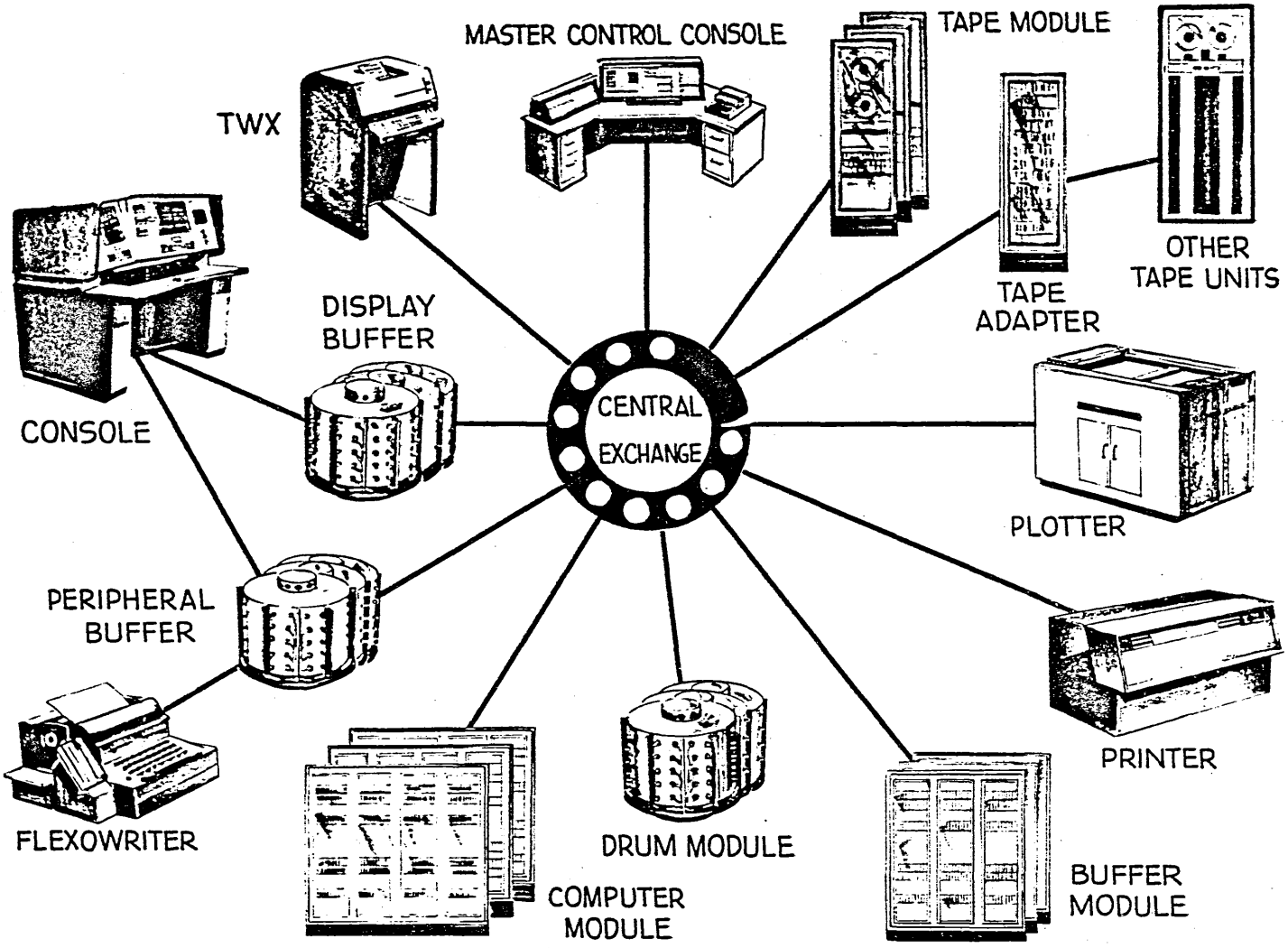

MASTER CONTROL CONSOLE
TAPE MODULE
TWX
OTHER
TAPE UNITS
TAPE
ADAPTER
DISPLAY
BUFFER
CONSOLE
CENTRAL
EXCHANGE
PLOTTER
PERIPHERAL
BUFFER
PRINTER
FLEXOWRITER
DRUM MODULE
COMPUTER
MODULE
BUFFER
MODULE

FIGURE 1

or replace mode. The Computer Module has a 13-bit interrupt register which, if masked appropriately, sends the control unit to an appropriate address in the memory, which in turn can be programmed to send the computer to an appropriate sub-routine. The computer issues interrupts through the data communication system; i. e., the Central Exchange. The master controlling element in the system is the Computer Module that is working on the Master Control Program. This computer is designated the Master Computer. The Master Control Program will be discussed in detail below.

The Buffer Module contains two 1024-word core memory devices which are capable of either independent or cooperative action. Both of these devices, which we will call "the Single Buffer," are internally programmed machines which perform logical operations, but do not perform arithmetic. The function of the Buffer Module is to perform off-line tape and drum processing and to serve as a high-speed messenger between the Master Computer and the Slave Computers. The Buffer Module, when connected to the Computer Module, serves as additional core memory for the Computer Module. It can issue interrupt messages and can be interrupted by being disconnected from the Central Exchange.

The Drum Modules have 8192 words and an average access time of eight milliseconds. They serve principally as back-up memory to the Computer Module.

The Tape Modules handle one-inch tapes of 16 channels, their speed is 150 inches per second, and they contain 200 bits per inch. The inter-record gap is 1-1/4 inch, and they can read backwards. The read-write heads are designed to use Mylar coated tapes.

A Tape Adapter Module will permit the control of commercial tape units. Thereby, tape-to-card and card-to-tape equipment is made available.

The Central Exchange, discussed in detail below, is used for data transfer between any two units in the system. It operates under the control of the Master Computer and, in a lesser way, under the control of any of the Buffer or Computer Modules.

A human supervisor sits at the Master Control Console. He can interrupt the Master Computer to make scheduling decisions that exceed the subtlety of the Master Control Program. In case of failure of the Master Computer, the human supervisor chooses and manually switches control to the new Master Computer. The real time clock is included in the Master Control Console for the requirements of the Master Control Program. In general, the human supervisor of the system manages it by exceptions, in that unusual circumstances are printed out by the Master Control Program and periodic reports on problem and equipment status are produced for the supervisor's review.

C. DETAILED DESCRIPTION

There are a number of elements that are necessary to the implementation of a multiple computer system such as this. These are the Master Control Program, the Central Exchange, the Alert Communication, and the Buffer Module.

1. The Master Control Program

The Master Control Program coordinates the functions of the data processing system. Some of its duties in the master slave(s) computer, real-time mode of operation are:

1) Detection of input requests for service
2) Interpretation and classification of requests
3) Assignment of requests to the proper area for solution
4) Assignment of Slave Computers to problem lists
5) Supervision of internal operation of the system
6) Supervision of the "assignment" rules of the Central Exchange
7) Monitoring progress
8) Supervision of handling of queues internal to the system
9) Supervision of the handling of system malfunctions
10) Dissemination of results
11) Liaison with human operators
12) Simulation

Each of these duties is discussed in detail below.

a. Detection of Input Requests for Service. (See Figure 2.) The Master Computer monitors all input stations and thus serves as a communication link between human operators and automatic input devices and the Data Processing System. Input requests are arranged, with assistance from the Master Computer, for interpretation.

b. Interpretation and Classification of Requests. (See Figure 3.) The input requests are classified as to type of problem and priority. This classification is determined in part from the address of the input device.

c. Assignment of Requests to the Proper Area for Solution (Scheduling Function.) (See Figure 4.) The request or problem is tagged with the time of arrival and identification. It is then inserted into the proper problem list in a common memory location (e.g., Drum Module). This list is ordered according to priority and consists of problems of the same type. The incoming problem may actually replace a problem in process and relegate it to the waiting list if (a) the priority warrants this, (b) the problem in process is not sufficiently close to completion, and (c) the incoming problem would not otherwise be served soon enough.

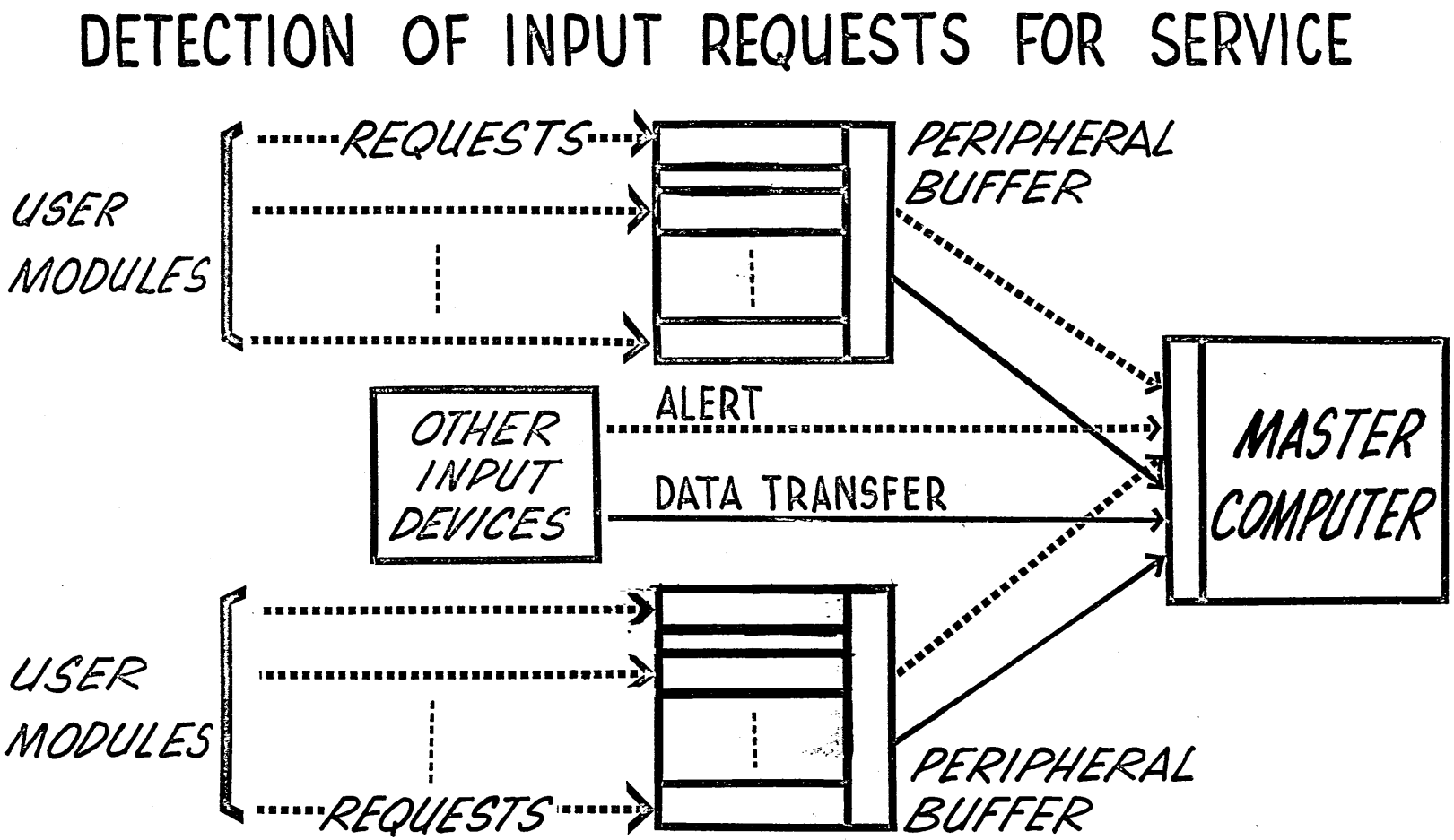

DETECTION OF INPUT REQUESTS FOR SERVICE
REQUESTS
USER MODULES
PERIPHERAL BUFFER
OTHER INPUT DEVICES
ALERT
DATA TRANSFER
MASTER COMPUTER
USER MODULES
REQUESTS
PERIPHERAL BUFFER

FIGURE 2

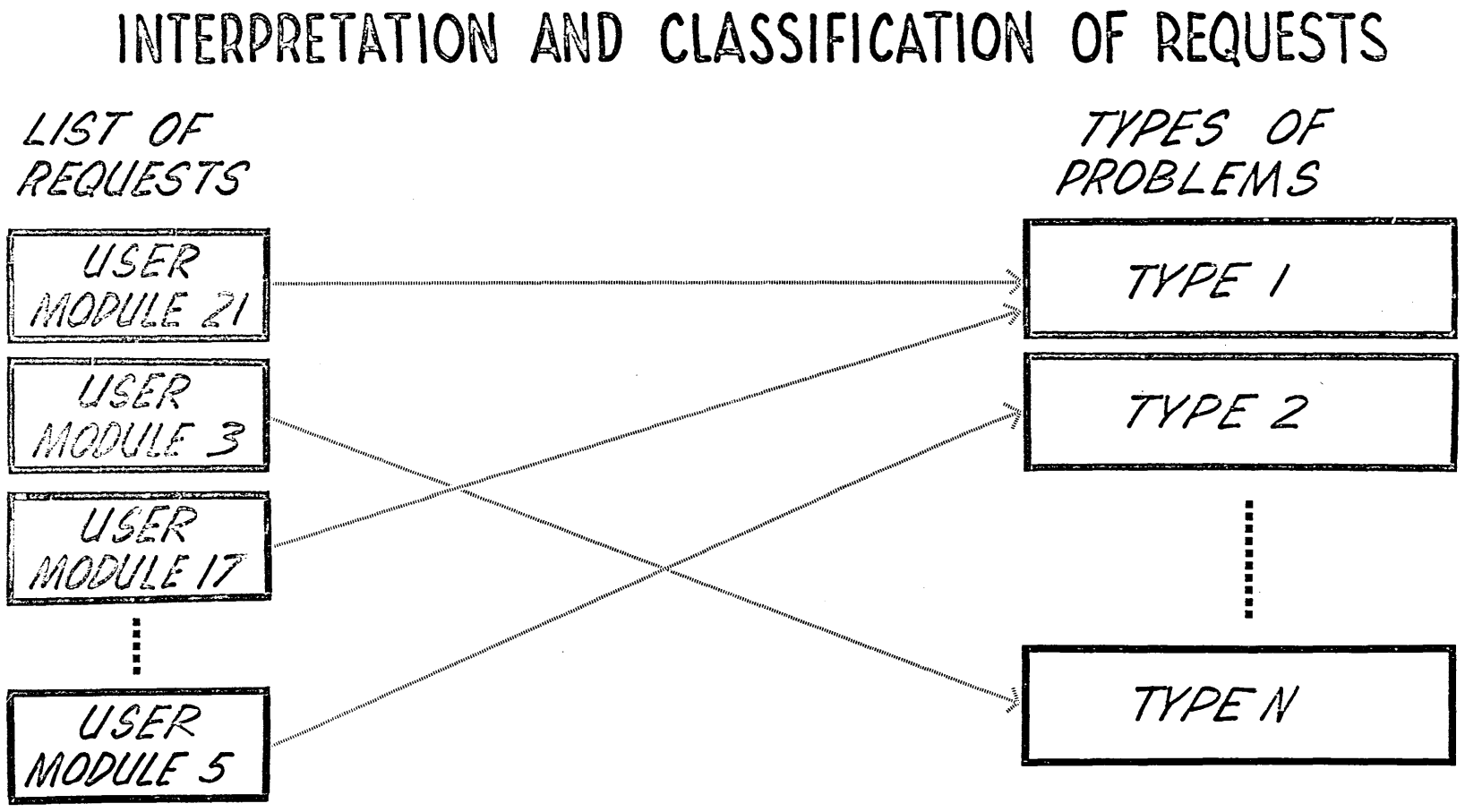

INTERPRETATION AND CLASSIFICATION OF REQUESTS
LIST OF REQUESTS
TYPES OF PROBLEMS
USER MODULE 21
USER MODULE 3
USER MODULE 17
USER MODULE 5
TYPE 1
TYPE 2
TYPE N

FIGURE 3

ASSIGNMENT OF REQUESTS TO THE PROPER AREA FOR IDENTIFICATION

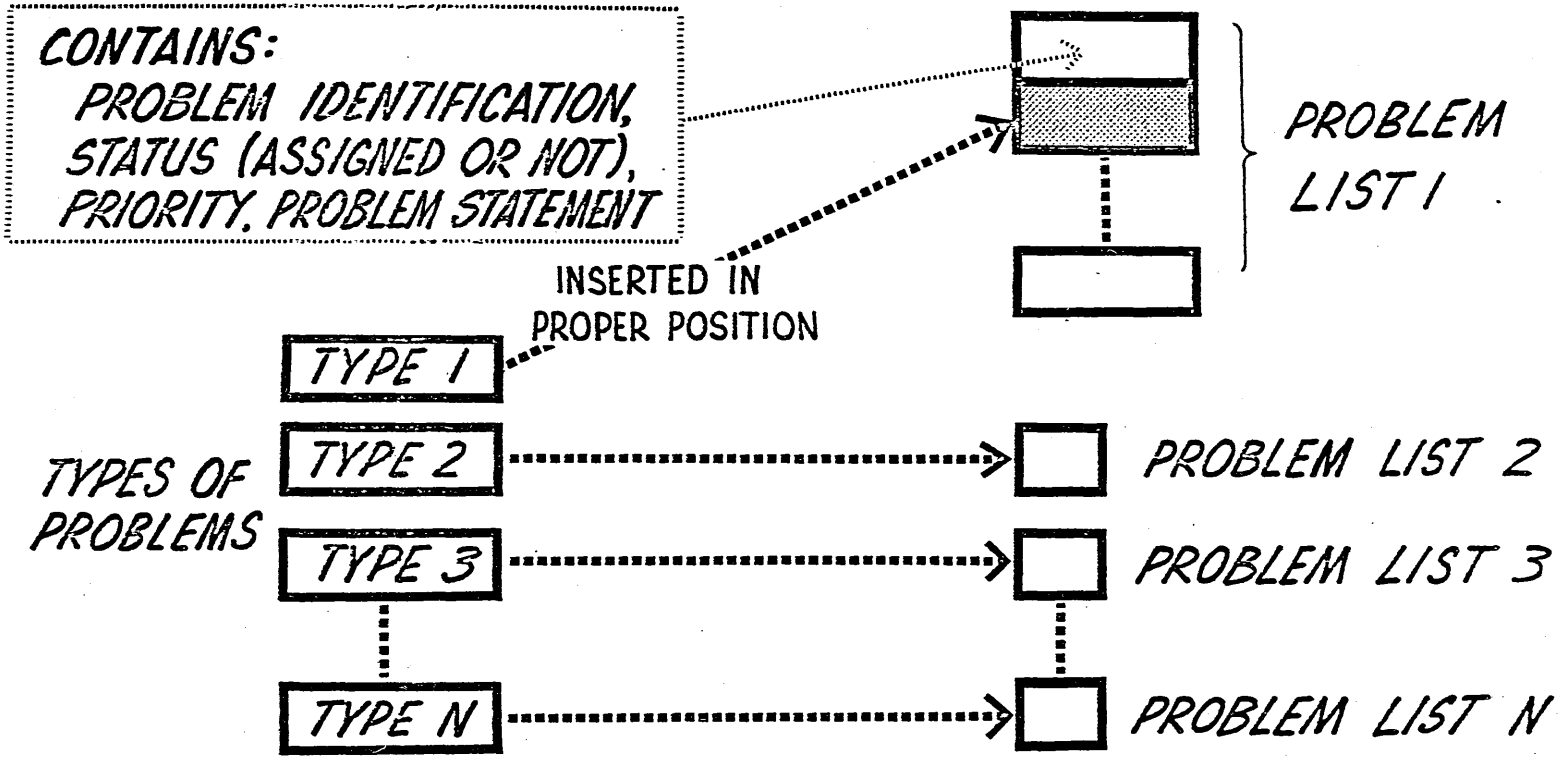


FIGURE 4

The priorities of problems in the lists are periodically upgraded in proportion to the length of time they have waited.

A problem may be retained in its entirety, or it may be divided into sub-problems for assignment to different problem lists.

d. Assignment of Slave Computers to Problem Lists. (See Figure 5.) One or more slave computers is assigned to a given problem list. When more than one computer is being used, a programmed lockout scheme prevents interference between computers when obtaining assignments. An indication is provided of the next problem in the list to be assigned. The assignment of computers to problem lists is required only initially and whenever changes are made. The Master Computer signals the Slave Computer via interrupt communication facilities (described in the next section). In connection with this function, the Master Computer maintains a table of Slave Computer status and assignment.

The Slave Computer associates itself with other functional modules (e.g., drums and tapes) via the Central Exchange as required by the problem, within the limits of the overall assignment rules enforced by the Master Computer. Since a given complex of modules is working on the same type of problem, this minimizes the number of changes in configurations required as different problems enter the system.

e. Supervision of Internal Operation of the System. The above steps have, in a general way, traced the handling of an input request by the system. In these and subsequent steps there arises a number of interactions internal to the system. To aid in the performance of these actions, there is available a system of alert communication, by means of which computers can be alerted. The alert system has two modes: In the sense mode, the Computer Module seeks the alerts in a manner of its own choosing; in the interrupt mode, the Computer Module is actually forced to drop what it is doing and immediately act upon the alert.

ASSIGNMENT OF SLAVE COMPUTERS TO PROBLEM LISTS

COMPLEXES OF MODULES

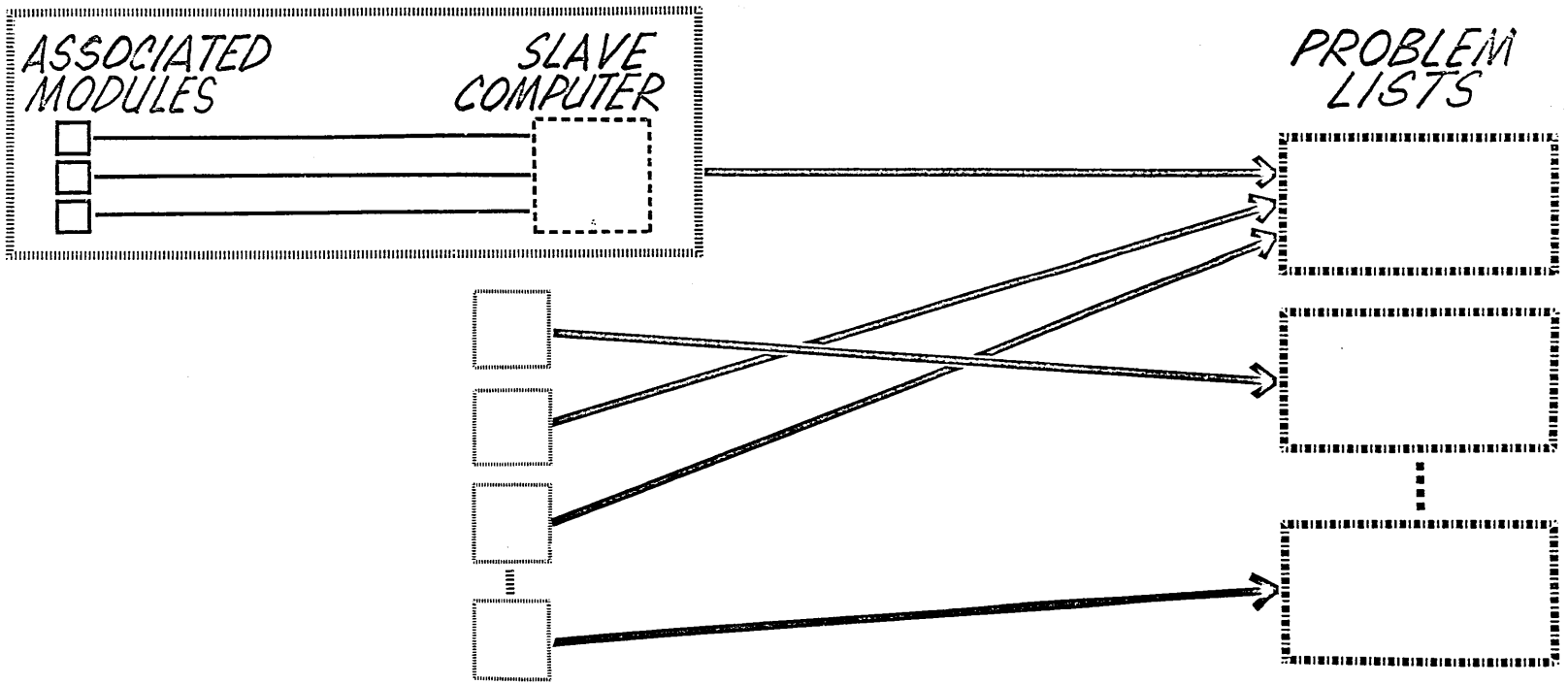


FIGURE 5

The sense-interrupt communication is one of the means by which the Master Computer serves the needs and coordinates the actions of the other modules of the system. Signaling occurs between Slave Computers and the Master Computer in the process of normal communications as well as when trouble conditions exist.

Subsequent sections give examples of internal operations of the system under the jurisdiction of the Master Computer.

The Master Computer, with assistance in some cases from the Master Control Console, has the ability to select the best mode of operation considering the nature and volume of the input requests to the system and the internal status of the system.

When the volume of requests is large, it operates in the sense rather than the interrupt mode. That is, it periodically scans the input alert indicators. Therefore, it would only rarely be necessary to interrupt its scan cycle--due to an infrequent, high-priority input.

On the other hand, when the input traffic is light, it would be inefficient for the Master Computer to periodically scan indicators which are rarely active. Therefore, it would then put itself in the interrupt mode and go about doing some useful work such as culling and updating files or participating in preventive maintenance routines.

Any input request or appropriate signal internal to the system would automatically interrupt the Master Computer, which would then terminate its present activity and serve the request.

When the nature of the workload becomes such that the Master Computer is overloaded but a Slave Computer is idle, certain of the less important Master Computer functions are relegated to the Slave Computer. In large systems, it may be desirable or necessary to employ a hierarchy of supervisory computers.

For the type of problem where a string of related requests is coming from a given input station, the Master Computer may temporarily relegate the supervising of the input-output functions associated with this station to the Slave Computer assigned to this station.

f. Supervision of the "Assignment" Rules of the Central Exchange. (See Figure 6.) The Central Exchange provides a flexible means of communicating data between modules. As a result of the use of modules to form various equipment complexes, the availability of certain modules is limited to a single complex or in some cases to select complexes. The Master Computer dictates the allowable assignment conditions, which may vary with the situation at hand. The control of assignments of particular modules by the Master Computer prevents conflicting requests for the same modules and disruption of problems in process in case of failure.

g. Monitoring of Progress.

1. Problem Lists. The Master Computer monitors the progress made on the various problem lists. Where progress is uneven there will be a discrepancy between the priority of unsolved problems in one list relative to those of other lists. Computers are then reassigned to more nearly serve requests in order of priority. Again, complexes of working modules are formed as required.

2. Sub-problems. Where several computers are working on sections of a single problem, the Master Computer schedules the operation so that the proper results arrive at the proper places at the required times. The interrupt communication channels provide a means for signaling the occurrence of significant events in the process and thus aid in keeping the various phases of the solution in proper synchronism.

SUPERVISION OF ASSIGNMENT RULES FOR THE CENTRAL EXCHANGE
(TYPICAL CONNECTIONS SHOWN)

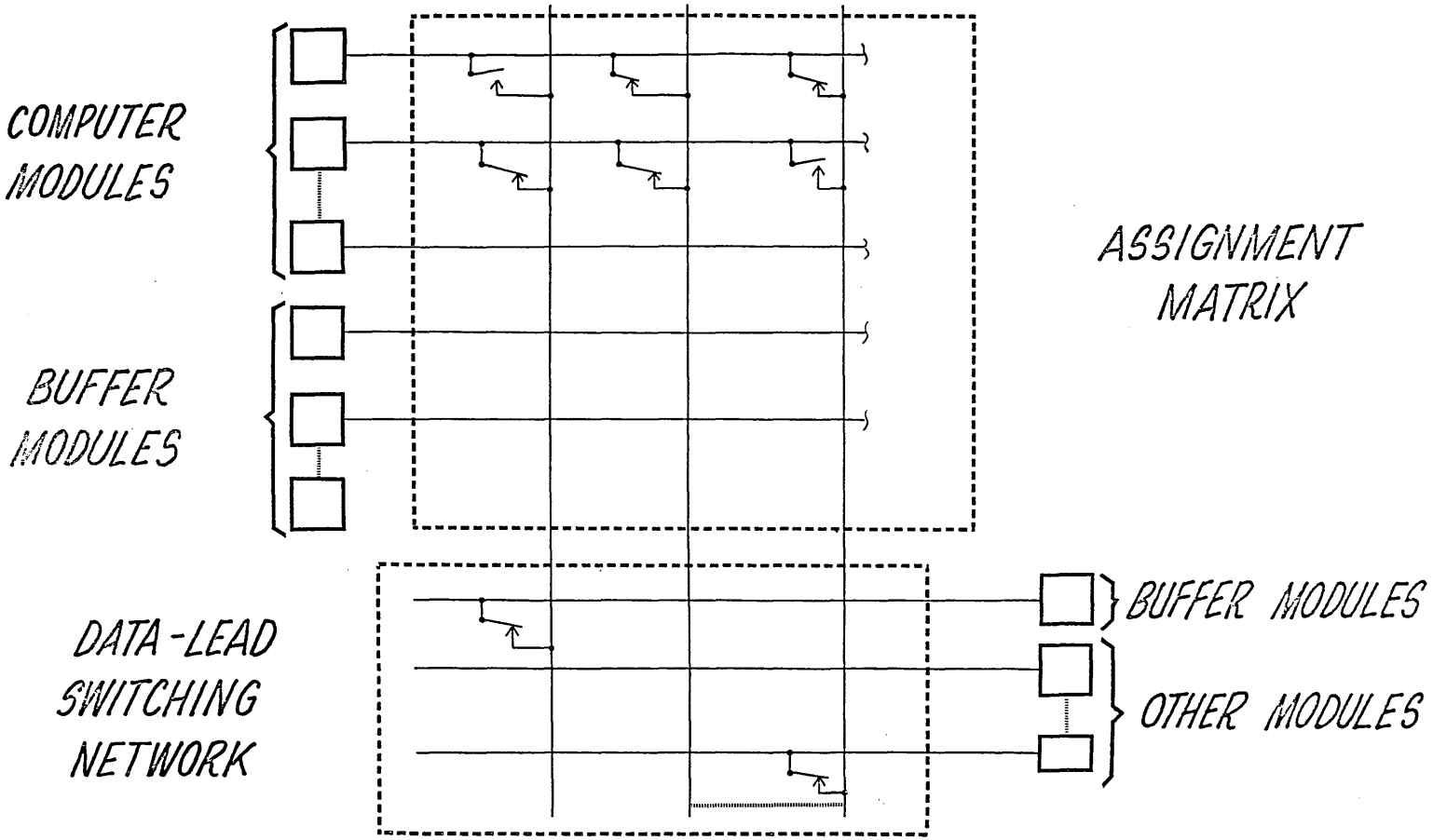


FIGURE 6

h. Supervision of Handling of Queues Internal to the System. (See Figure 7.) During periods of peak load conditions, situations develop where the Master Computer must schedule the accessibility of common files to the various equipment complexes. In this mode of operation, only the Master Computer can control the connection of Slave Computers or Buffer Modules to certain portions of the common memory. The modules which require access to the common files place their request with the Master Computer, which in turn enters this request in the queue in accordance with its relative priority. The Master Computer then signals (by means of interrupt communications) the next-in-line requesting module when the common files are available to it. This module then can either accept or temporarily reject the right-to-access depending upon its present status. If it rejects the action, the Master Computer interrupts the next-in-line requesting module. The previous requesting module is then placed at a lower position in the list.

i. Supervision of the Handling of System Malfunctions. When a malfunction occurs in a module it is reported to the Master Computer via the interrupt communications path. If the identity of the faulty module is not directly known, the Master Computer directs a diagnostic routine to find and identify the module in question. Then, the Master Computer determines the most efficient modus operandi in view of the existing work-load and the available modules. It causes the faulty module to be switched out of its working complex, causes a replacement (if available) to be switched in its place, and reports the location and nature of the trouble to the operator at the Master Control Console. In rare cases where the Master Computer is unable to determine the best course of action, it passes on the problem to the operator at the Master Control Console.

The Slave Computers will periodically store intermediate results. In case a new problem-to-computer assignment must be made due to a

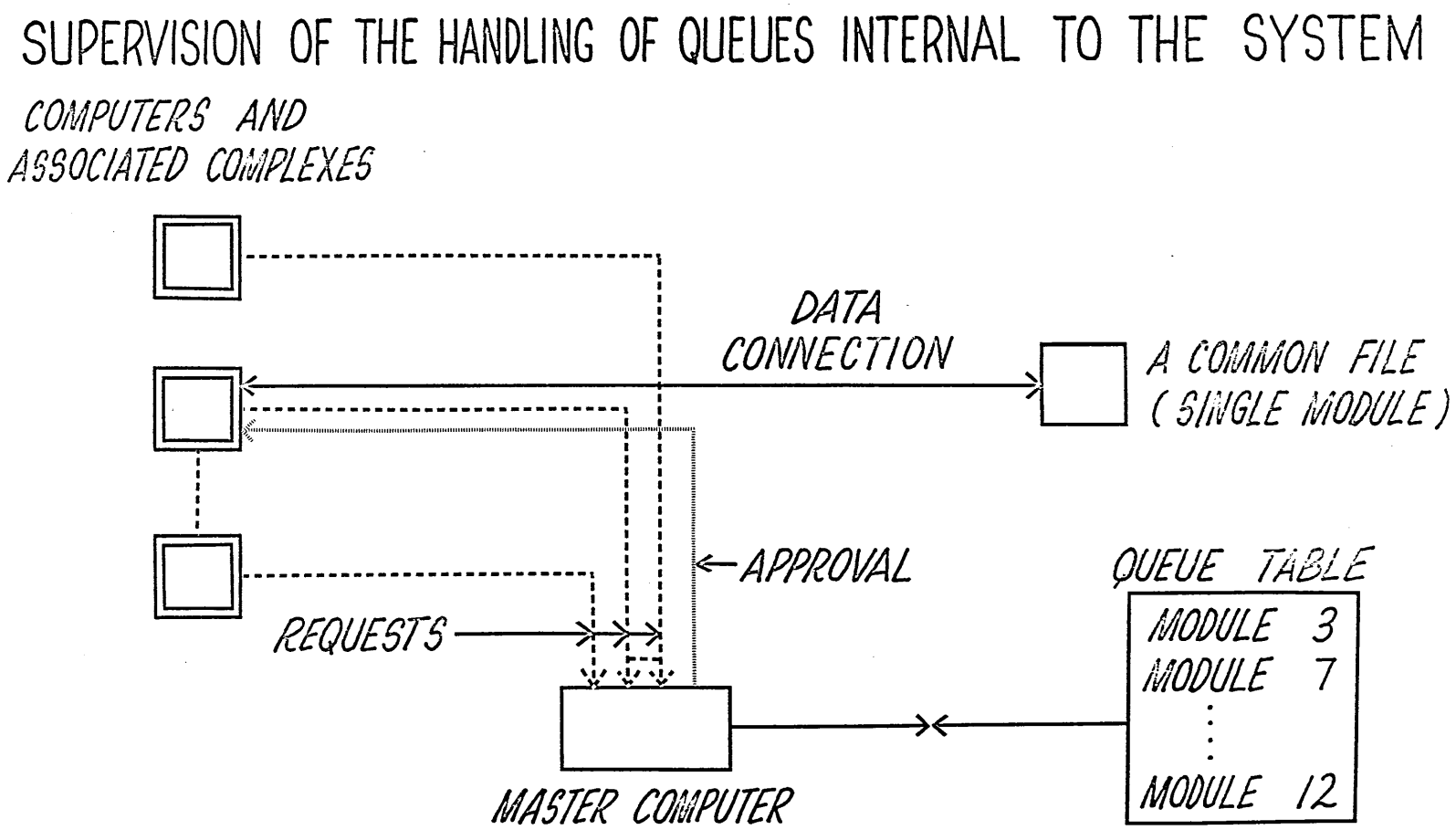

SUPERVISION OF THE HANDLING OF QUEUES INTERNAL TO THE SYSTEM
COMPUTERS AND
ASSOCIATED COMPLEXES
DATA
CONNECTION
A COMMON FILE
(SINGLE MODULE)
APPROVAL
QUEUE TABLE
MODULE 3
MODULE 7
MODULE 12
REQUESTS
MASTER COMPUTER

FIGURE 7

malfunction or change of priorities, a recovery program is employed whereby intermediate results can be stored for later retrieval, thus avoiding the necessity of completely reworking the problem.

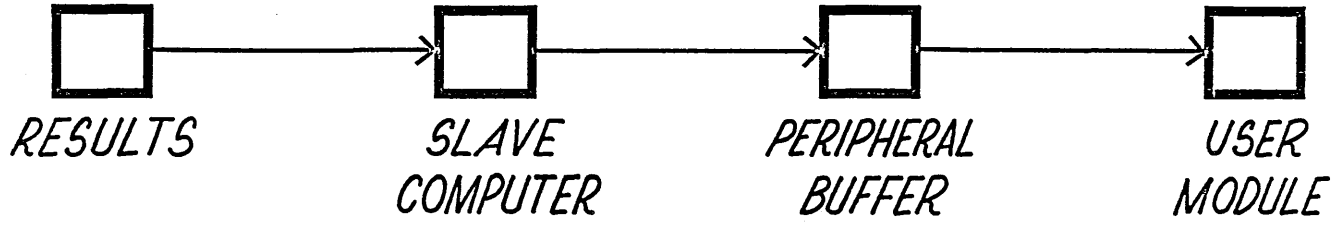
j. Dissemination of Results. (See Figure 8.) When a Slave Computer has finished a problem, the Master Computer supervises the transfer of the requested data to the proper output area(s).

The problem lists are continuously culled as problems leave the system. Records must be kept indicating system performance, status and progress of work loads. If this imposes too much of a delay to perform in serial, the notification of the effort will suffice.

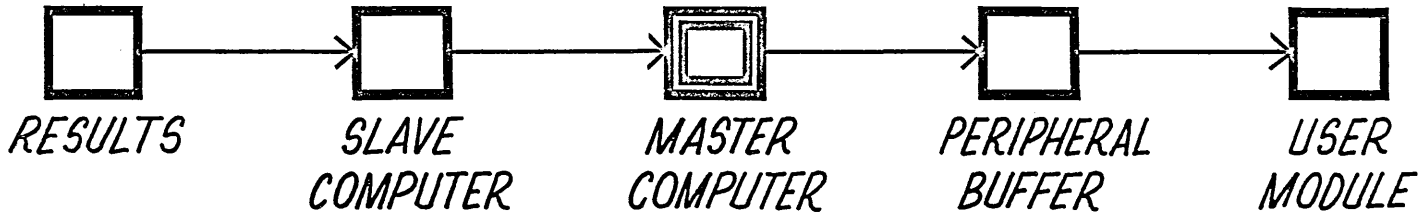
k. Liaison with Human Operators. In general it is desirable to delegate as much responsibility as possible to the Master Computer. Thus, interference by human operators occurs only on an exception basis. However, when requirements dictate that certain functions should be performed by humans, the Master Control Program serves as the intermediary in the man-machine relationship. This would happen (for example) when the subtlety of the decision-making mechanism for priority scheduling is exceeded.

l. Simulation. A Monte Carlo simulation of this system is in process at present. It is anticipated, as the work progresses, that varying levels of delegation of authority from the Master Computer to Slave Computers will be studied to determine the system effectiveness. Of principle interest will be the delays introduced by the control function. In detail, it will be possible to study the pay-off from periodic dumps and the size of dedicated memories needed for problem lists.

DISSEMINATION OF RESULTS



METHOD 1 – SLAVE COMPUTER SUPERVISES THE TRANSFER OF OUTPUT DATA



METHOD 2 – MASTER COMPUTER SUPERVISES THE TRANSFER OF OUTPUT DATA

FIGURE 8

2. Central Exchange

The purpose of the Central Exchange (see Figure 9) is to allow the Master Computer to allocate and reapportion equipment to problems. The ability to add modules to the system is achieved by using an expandable Central Exchange rather than a time shared buss. The Computer Module and Buffer Module can only talk to one element at a time. All of the usual switching built into computers has been centralized. A cost saving is effected by centralizing all of the switching in the system rather than incorporating it in each of the Computer Modules and Buffer Modules.

The time properties of this Central Exchange must be compatible with the overall system time requirement. In particular the problem load contains many short duration problems. Furthermore, the switching of a Buffer Module from a Tape Module to a Computer Module must be fast enough to allow the Computer Module to process the tape-extracted information and then return the Buffer Module to the Tape Module all in the inter-record gap. The switching must be electronic. The speed of switching to effect the connection is 65 microseconds. The Central Exchange will handle any data rate up to 200 KC. Although in general it is told by the Master Computer which module can talk to which other modules, this assignment can be automatically changed by the Master Computer. The Computer Module and Buffer Modules in general can connect themselves to any unit that has been assigned to them. Only the Master Computer can effect a connection between any two units in the system.

Because a Computer Module might want to be interrupted by a Buffer Module while working with a Drum Module, the alert or interrupt communication has to be independent of the Central Exchange.

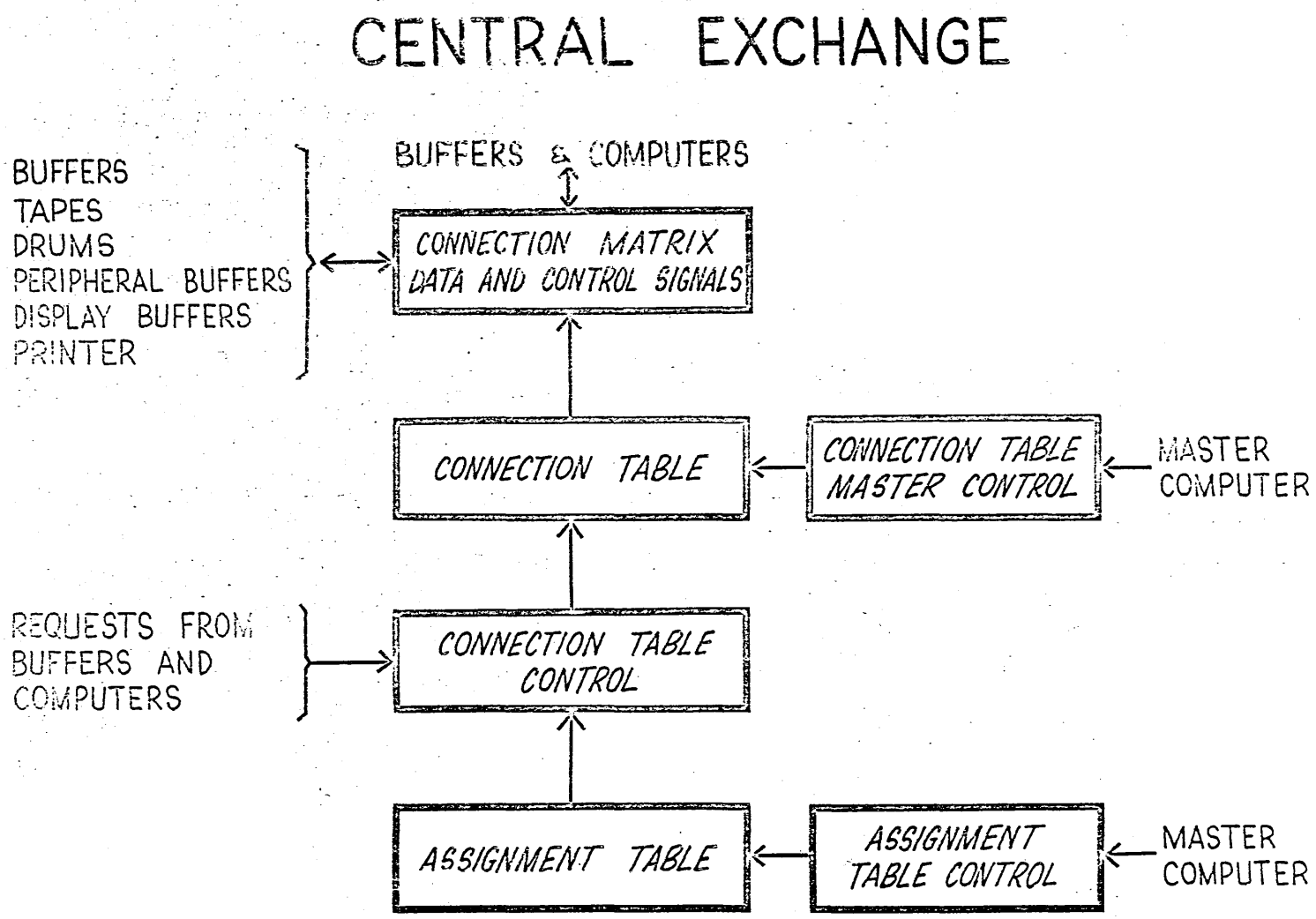

CENTRAL EXCHANGE
BUFFERS & COMPUTERS
BUFFERS
TAPES
DRUMS
PERIPHERAL BUFFERS
DISPLAY BUFFERS
PRINTER
CONNECTION MATRIX
DATA AND CONTROL SIGNALS
CONNECTION TABLE
CONNECTION TABLE
MASTER CONTROL
MASTER
COMPUTER
REQUESTS FROM
BUFFERS AND
COMPUTERS
CONNECTION TABLE
CONTROL
ASSIGNMENT TABLE
ASSIGNMENT
TABLE CONTROL
MASTER
COMPUTER

FIGURE 9

Standardized transmission has been established between all units of the system. Thirty-six bit transmission is accomplished, 18 bits each way. Of those 18 bits, 13 bits are data, 1 is a parity bit, 3 are control bits, and 1 is a clock bit. The switching element to be used for both the tables and the matrices is the Transfluxor.

The requests come in with a specific address which has been supplied to the requesting Computer Module or Buffer Module by the Master Computer. The connection table control first quizzes the assignment table to see if this is a possible request, and secondly, quizzes the connection table to see whether the element is already busy. If so, this message is sent back to the requesting Computer or Buffer Module.

3. Alert Communications

In order to have a multiple computer system in which the data processing capacity can be rapidly reallocated and multiple computers can work on many problems in parallel, an alert signal capacity has to exist. The alert capacity is useful where the timing of a given operation is critical or where the control condition being reported on is so improbable that periodic testing for the condition by the more usual sense type instructions would waste computer time.

The RW-40 Computer Module can respond to alert signals in two different ways. The system alert condition may be sensed continuously by masking an appropriate pattern of bits when the selected or masked alert lines are activated. This masking occurs in a special register called the S-register. The computer program is forced to jump to a designated address that depends upon the bit position in which the alert signal was received. This forced program jump can be used to link to a program which responds appropriately to the control signal sensed. The alert becomes effective at the completion of the instruction currently being executed, so that control via alert communication is much more rapid than would be possible

with interspersed program test or sense type instructions. If the particular bit position in the S-register is not masked, then the Computer Module can use this in the more usual sense mode. The masking used depends upon the program.

A separate communications network is necessary for alert signals since the Computer Module requires this type of control when its peripheral devices such as Buffer Modules are operating independently with, say, Tape Modules. The Computer Module must be in communication with these devices while not actually connected to them through the Central Exchange. Although a completely general switching network for alert signals could be provided by supplying a separate alert switch, this would approach the Central Exchange in cost and complexity. However, many of the system modules do not require alert control. Control by programmed use of test or sense type instructions is more appropriate in some cases. Of these modules that can employ this system alert type of control, a single alert signal from the devices is usually adequate. The Computer Module then questions the source of the alert signal. In these cases, elaborate alert messages are not required since, once alerted, the Computer Module can obtain additional information rapidly by the use of test or sense type of control through the Central Exchange.

In general, the design of the communications of the alert signals are as follows: (See Figure 10.) Several 13-bit flip-flop registers, called "alert registers," are provided in the system. By a flexible form of plugboard-mixing, an appropriate combination of single interrupts from Buffer, Tape, Drum and Peripheral Buffer Modules can be brought to these alert registers. Also, by having an output "hub" on the Central Exchange, perfectly general 13-bit words can be sent to any one of these alert registers by the Computer Modules or the Buffer Modules. These, of course, get mixed with the allocation of single alerts from Buffer, Tape, Drum, and Peripheral Buffer Modules. The allocation of Tape,

ALERT SYSTEM

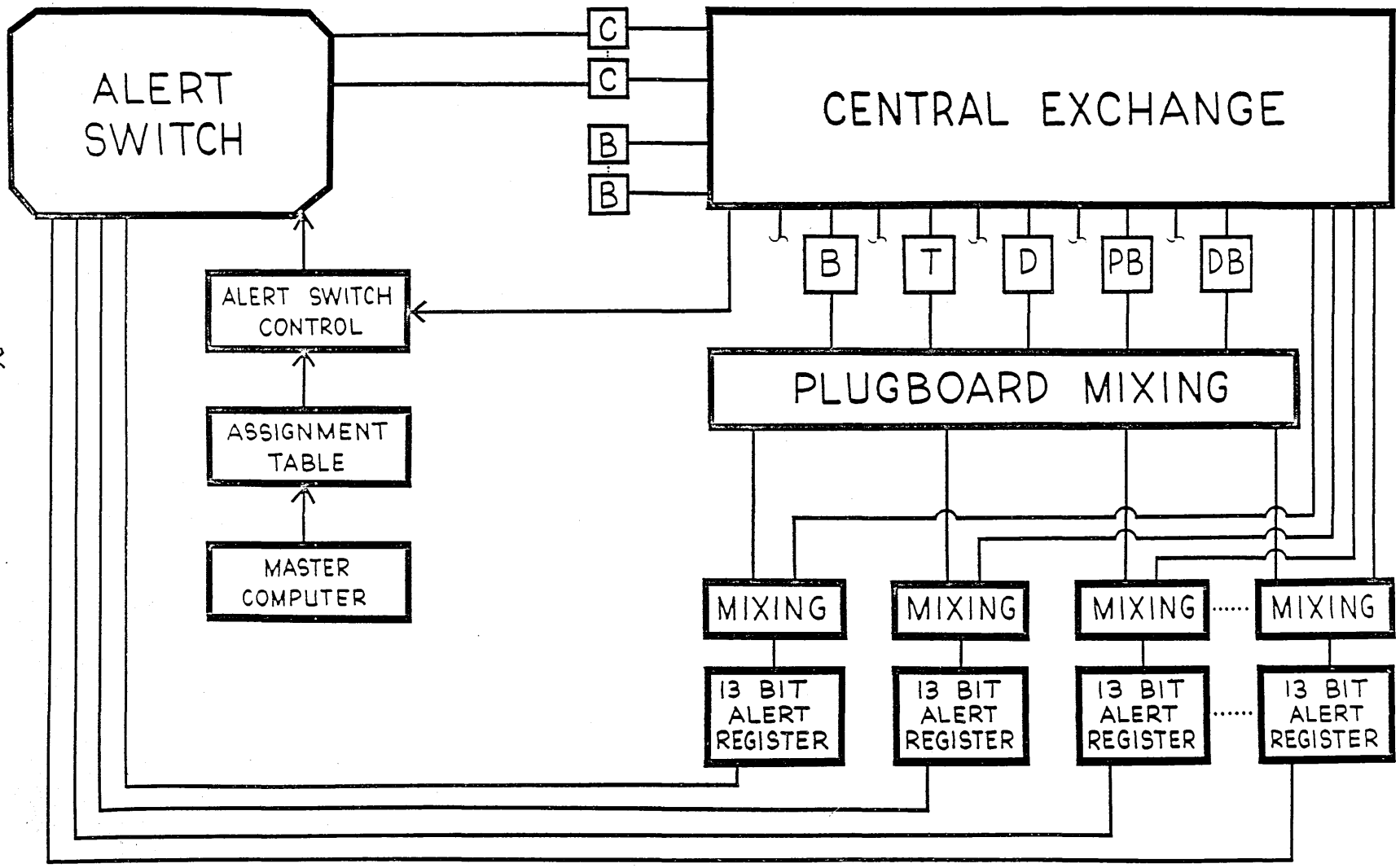


FIGURE 10

Drum, and Peripheral Buffer Modules that is made to a particular alert register is one that is appropriate to a particular problem to be solved; that is, there is a correspondence between the alert register and the type of problem to be solved by the system. Each of these alert registers, then, has an input to the alert switching system that can connect it to any one of the Computer Modules. Each of the Computer Modules has a 13-bit S-register for receiving the alert signals. The alert switch is under the control of the Master Computer. It can be controlled by addressing control words through the Central Exchange from the Computer Module. With this system, any Computer or Buffer Module can alert the appropriate Computer Module by placing a word in the appropriate one-word alert register. This is accomplished by first setting the Central Exchange and then reading out the appropriate word. This requires, of course, that the originating modules know which alert register is being used by which Computer Module; the Buffer Module need only be alerted if it is busy and this can be accomplished by breaking the connection at the Central Exchange.

4. Buffer Module

The Buffer Module is a switchable, internally programmed memory storage device that is necessary in a multiple computer system as a high-speed messenger. The Master Computer can load up a Buffer Module at a 100 KC word rate and then send it off-line to unload to a Slave Computer Module without waiting for the availability of the Slave Computer. Furthermore, this Buffer Module can unload to, or load from, a Tape or a Drum Module at a lower word transfer rate without typing up the Computer Module.

As the concept of switchable core memory began to take shape it became clear that a flexible, switchable allocation of core memory and index registers to those core memories could be of use in many parts of the system as an off-line tape processing device for search and retrieval. It would also be useful for functions such as high-speed buffering for

output to a printer. We believe that a system employing switchable core memories that can be used for many functions is potentially better than a system that shares the memory cycle of one large core in a very complicated interlaced way.

The Buffer Module itself has 1024 words of memory. It can be either internally programmed, or externally programmed: when it is connected to the Computer Module that is, it can be operated one instruction at a time or it can be put into the self-instruction mode by the Computer Module and operated on internally stored programs. It can be taken out of the self-instruction mode either by the Computer Module or by the Buffer Module itself.

This module can perform the types of orders that are required for it to control a Tape Module, a Drum Module or the Central Exchange, as well as data read-in and read-out orders. There are jump instructions which take place on catastrophic failures, internal and external parity checking, and unaccepted commands. If the Central Exchange makes a mistake and connects the Buffer Module to a Drum Module when it should be connected to a Tape Module, the Buffer Module can be programmed to jump to an order that will cause an appropriate interrupt to be issued. Thus the Master Computer is notified that the Central Exchange has made a mistake.

When connected to the Computer Module, this Buffer Module memory can be accessed serially at the same speed that the Computer Module can refer to its own memory.

IV. SUPPORTING ANALYSIS

A. QUEUING ANALYSIS

A queuing analysis of the system was performed in order to determine the number of computers required to satisfy the throughput requirements as well as to provide a mechanism for studying detailed effects on performance of occurances such as:

1) Computer breakdown
2) Priority pre-emption rules
3) Roll-back penalties
4) Delegation of authority from the Master Computer to Slave Computers

The analytic approach provided by Jacob Bricker gave a gross relationship between numbers of computers and probability of satisfaction of throughput requirements.

The analysis is summarized in the following illustrations. Figure 11 shows our assumptions, Figure 12 our definitions, Figure 13 our calculations and Figure 14 our evaluations. Figure 15 shows the gross relationship between probability of service and numbers of computers.

B. MONTE CARLO SIMULATION

The queuing analysis was limited in that it assumed system stability, allowed no pre-emption of problems and lumped reliability, maintenance and repair time into an overall reduction in numbers of operations per second.

ASSUMPTIONS

1. QUEUE DISCIPLINE - FIRST COME FIRST SERVED
2. RELIABILITY, REPAIR, PREVENTATIVE MAINTENANCE LUMPED INTO A FLAT REDUCTION OF AVAILABLE COMPUTER TIME
3. POISSON ARRIVAL RATES FOR PROBLEMS
4. PROBLEM-SERVICE TIME AND COMPUTER DOWN TIME AND UP TIME HAVE NEGATIVE EXPONENTIAL DISTRIBUTIONS

FIGURE 11

DEFINITIONS

1. K NUMBER OF PROBLEM CLASSES
2. π TIME OF PROBLEM COMPLETION
3. T_i THRUPUT TIME REQUIREMENT FOR PROBLEM CLASS i
4. C NUMBER OF COMPUTERS
5. μ_i MEAN SERVICE RATE OF EACH COMPUTER WITH RESPECT TO PROBLEM CLASS i
6. Π MEAN RATE OF OCCURANCE OF RESTORED UPTIME
7. b MEAN RATE OF DOWN TIME
8. λ_i MEAN ARRIVAL RATE OF CLASS i PROBLEMS

FIGURE 12

CALCULATIONS

1. $$\mu'_i = \frac{r\ \mu_i}{b + r}$$

2. $$\lambda = \sum_{i=1}^{K} \lambda_i$$

3. $$\mu = \frac{\lambda}{\sum_{i=1}^{K} \frac{\lambda_i}{\mu'_i}}$$

4. $$\rho = \frac{\lambda}{c\,\mu}$$

5. $$P_0 = \frac{1}{\sum_{n=0}^{c-1} \frac{(c\rho)^n}{n} + \frac{(c\rho)^c}{c!\,(1-\rho)}}$$

FIGURE 13

EVALUATION

1. $P(>0) = \frac{(c\rho)^c}{c!(1-\rho)} P_0$

(PROBABILITY THAT ANY PROBLEM WAITS UNTIL IT ENTERS SERVICE)

2. $\Pi(T < T_i) = P(>0) \left\{ \frac{\mu'_i [1 - e^{c\mu T_i (1-\rho)}] - c\mu(1-\rho)(1-e^{-\mu'_i T_i})}{\mu'_i - c\mu(1-\rho)} \right\}$

$$+ [1 - P(>0)] (1 - e^{-\mu'_i T_i})$$

(PROBABILITY THAT PROBLEMS OF CLASS i WAIT ON LINE AND IN SERVICE FOR A TOTAL TIME OF LESS THAN T_i

FIGURE 14

QUEUING ANALYSIS

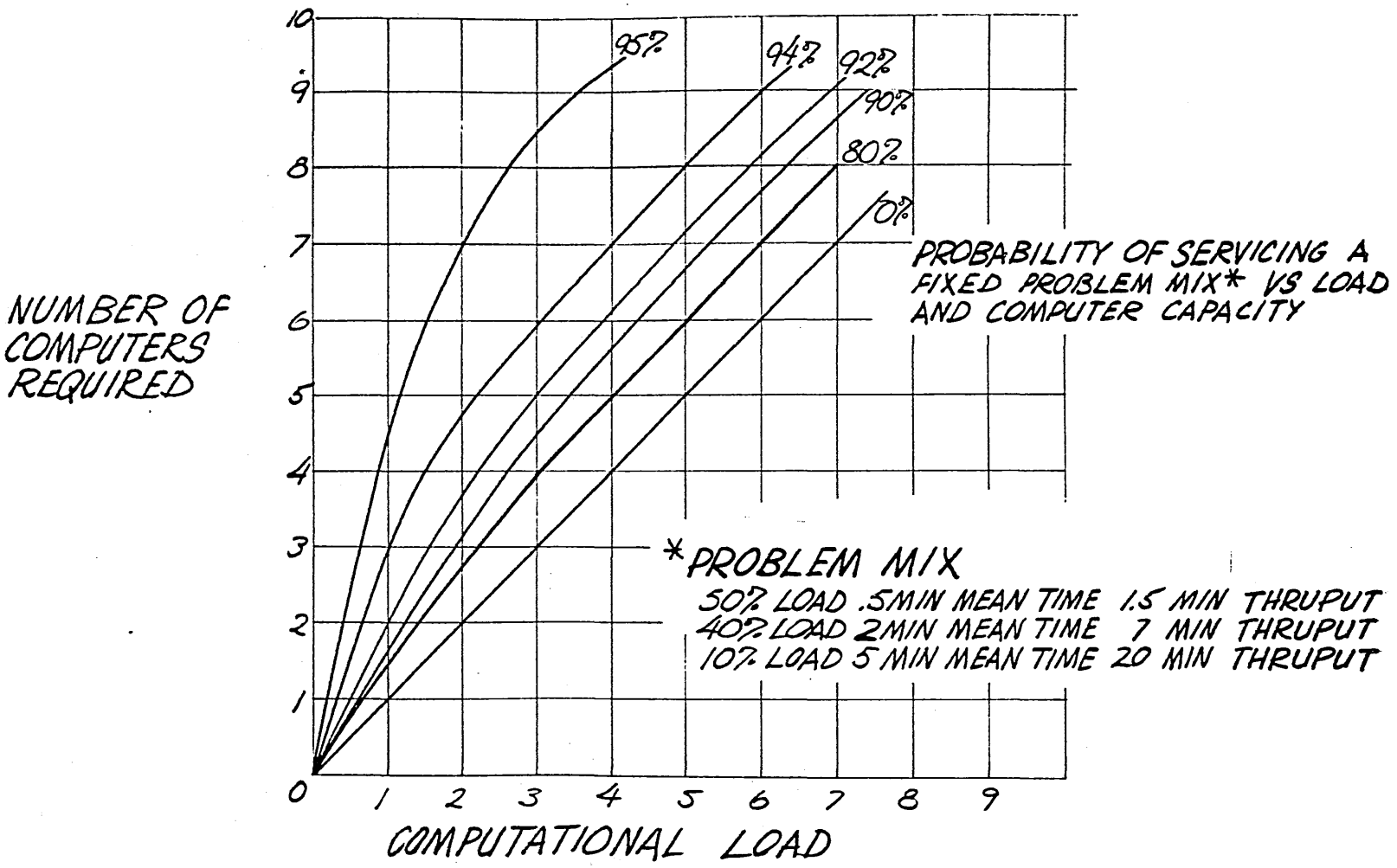


FIGURE 15

As a consequence a more detailed Monte Carlo simulation was started. It can be described here only in part as it is by no means complete. Figure 16 shows a gross block diagram of the Monte Carlo simulation. The inputs and outputs of the Monte Carlo simulator are discussed below.

1. Inputs

In order to provide maximum utility, the simulator is programmed to permit considerable latitude in the parameters which bound its behavior. The following items are specified as initial conditions (and can therefore be modified for each run):

1) The number of computers in the system to be simulated
2) The number of problem classes
3) The characteristics of each problem class
4) The characteristics of computer down-time
5) The penalty for pre-empting
6) The frequency of simulator result outputs

Some of these parameters require additional comments. Each problem class is defined by:

1) Its relative priority
2) The statistical distribution of service time for the members of the class
3) The statistical distribution of arrivals for the members of the class

The occurrence of computer "down-time" is simulated by treating down-time as a pseudo problem class with top priority (a non-functioning computer can obviously not be used for anything until repaired). Service time in this instance is, of course, repair time; and arrival rate distribution is the occurrence of malfunctions.

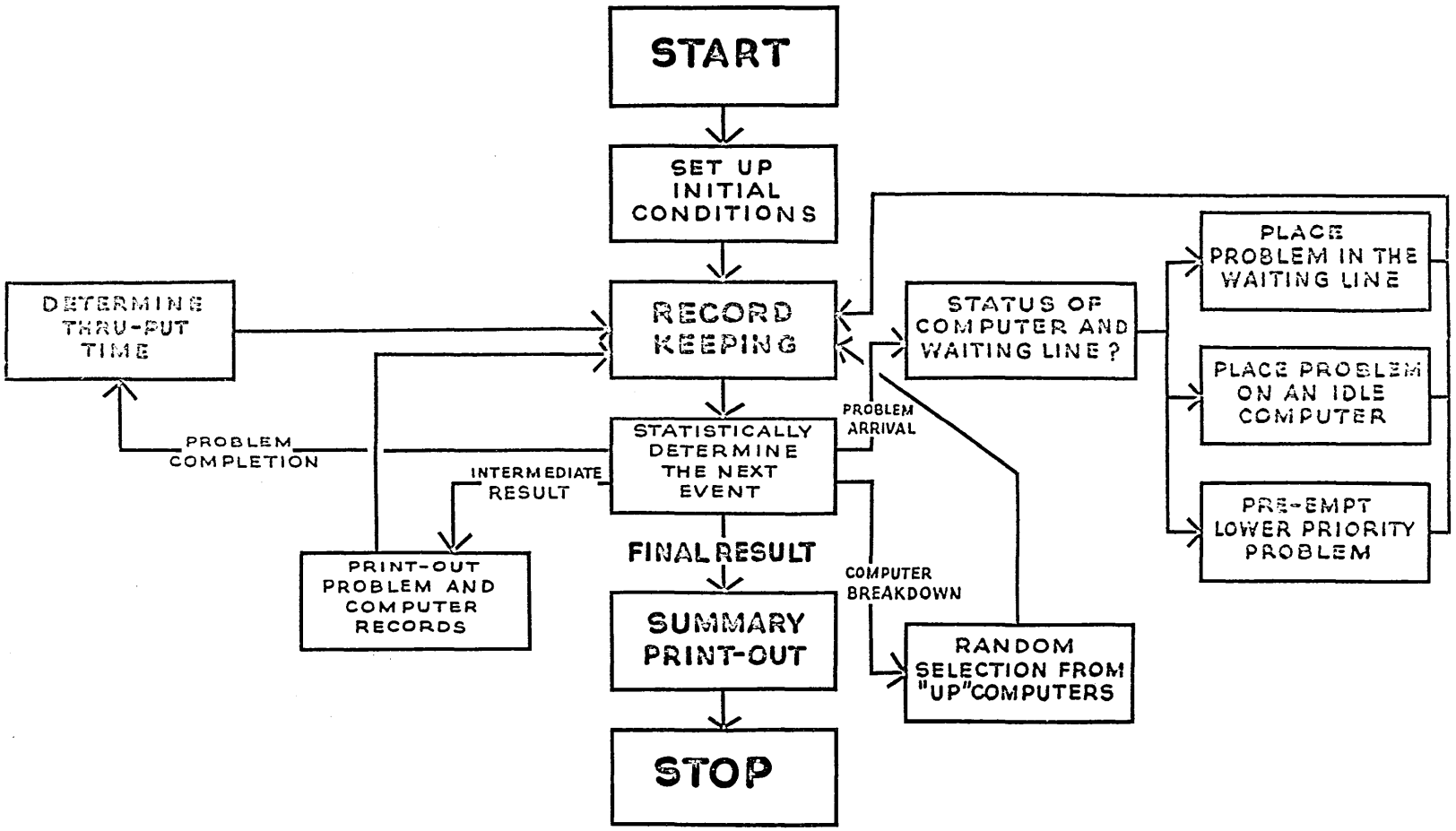

START
SET UP INITIAL CONDITIONS
RECORD KEEPING
DETERMINE THRU-PUT TIME
STATUS OF COMPUTER AND WAITING LINE ?
PLACE PROBLEM IN THE WAITING LINE
PLACE PROBLEM ON AN IDLE COMPUTER
PRE-EMPT LOWER PRIORITY PROBLEM
PROBLEM ARRIVAL
PROBLEM COMPLETION
STATISTICALLY DETERMINE THE NEXT EVENT
INTERMEDIATE RESULT
PRINT-OUT PROBLEM AND COMPUTER RECORDS
FINAL RESULT
COMPUTER BREAKDOWN
SUMMARY PRINT-OUT
RANDOM SELECTION FROM "UP" COMPUTERS
STOP

FIGURE 16

Pre-empting defines a condition under which a higher priority problem dislodges a lower priority problem currently being serviced. Upon returning to the lower priority problem, the Computer Module cannot be restarted at the exact point of interruption, but must be rolled back to the nearest control point. The time lost by this roll-back is called the pre-empting penalty.

2. Outputs

As mentioned above, one of the inputs specifies the frequency of simulator outputs, thus permitting a printout at any arbitrary interval of simulated "real time." Each printout consists of the following:

1) Total elapsed simulated time since start of run (in seconds)
2) Total accumulated "idle" time, and number of occurrences of "idle" time. (Accumulated idle time is sum of idle time of each computer.)
3) Total accumulated "down" time, and number of occurrences
4) Number of problems received, but not yet serviced (length of the waiting line)
5) For each problem class: total service time expended on members of the class, and number of problems serviced
6) For each problem class: the number of problems handled in each group of the problem class, where a "group" is defined as containing all problems whose service time lies between some (defined) minimum and maximum time values.

NOTE

For output, "service time" is the sum of actual service (processing) time plus any waiting time; i. e., "throughput time."

Figures 17 thru 21 describe the basic problem mix used in the simulation. The tables shown in Figures 22 and 23 give the probability of satisfaction of problem class 2 to 5 within the stated time. These 8 runs required the following amount of time:

Run No.	Number of Days
1	5
2	16
3	8
4	3
5	6
6	10
7	10
8	10

Figures 24 to 26 describe the transient behaviour under these circumstances. As might be expected, the thruput time distribution asymtotically approaches the service time distribution as the number of computers increases.

BASIC PROBLEM MIX

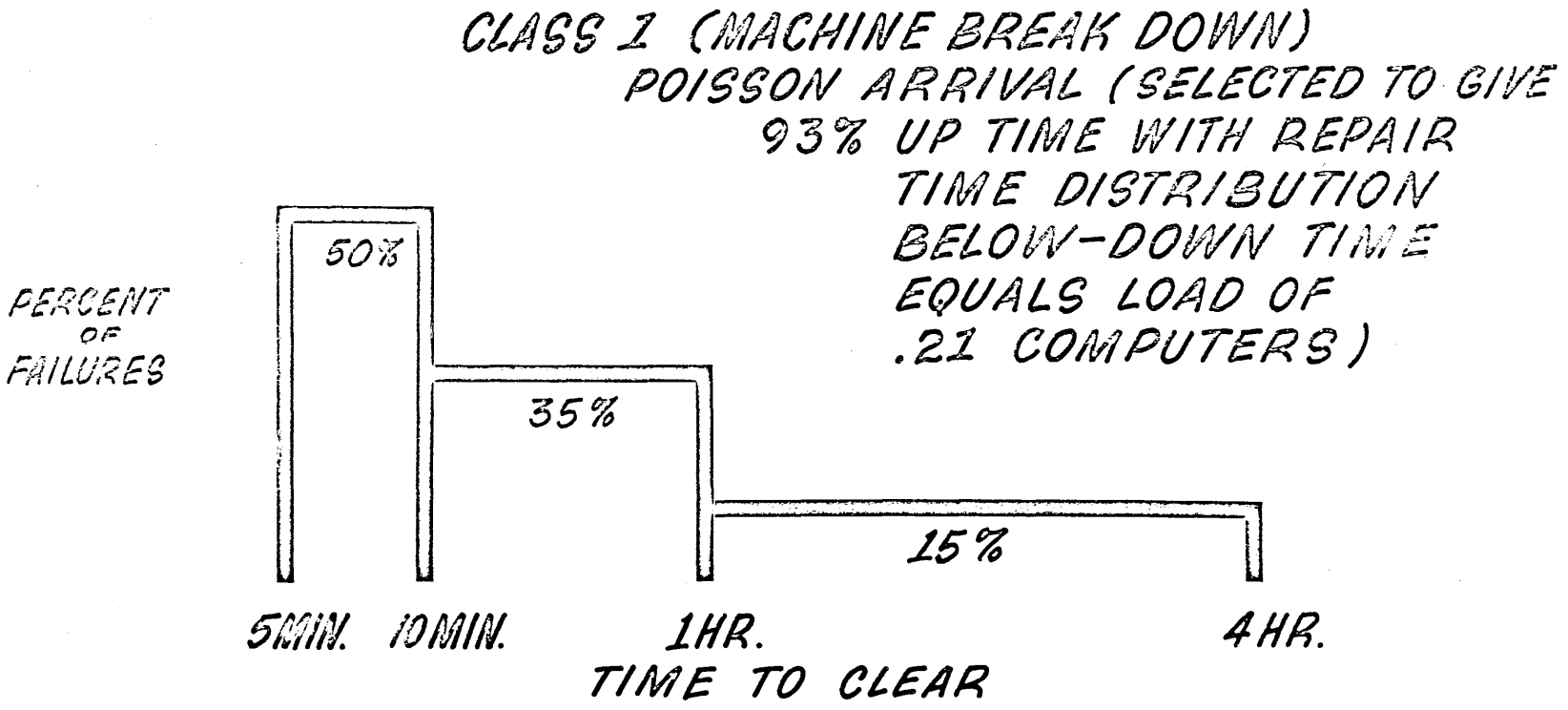


FIGURE 17

BASIC PROBLEM MIX

CLASS 2 PROBLEMS

POISSON ARRIVAL (SELECTED TO GIVE LOAD OF 1 COMPUTER)

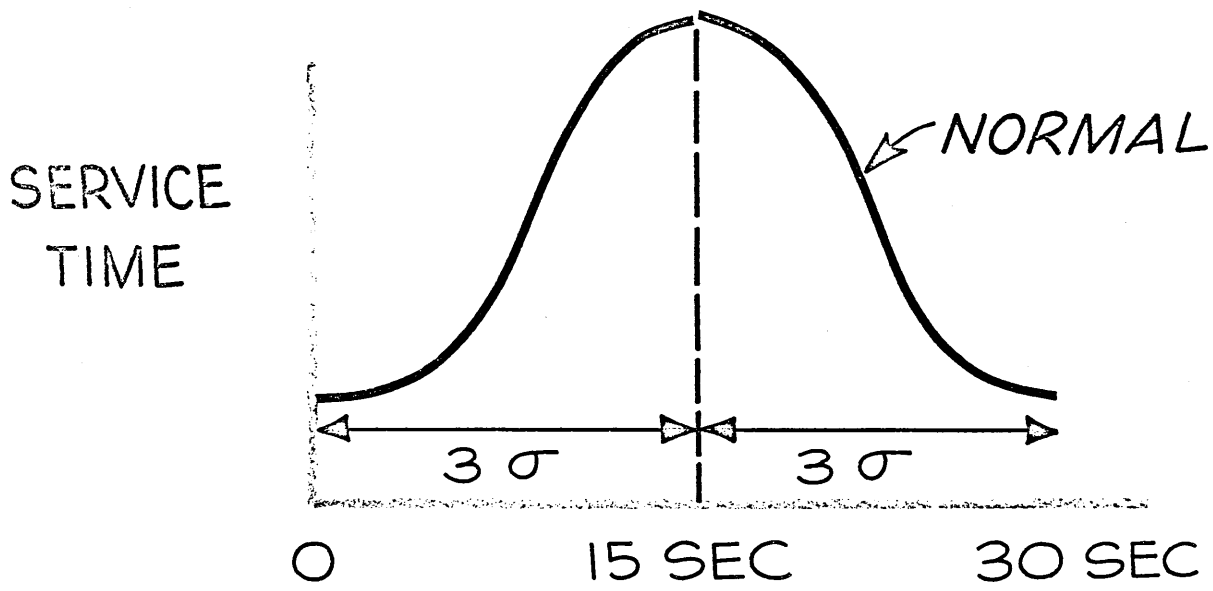


FIGURE 18

BASIC PROBLEM MIX

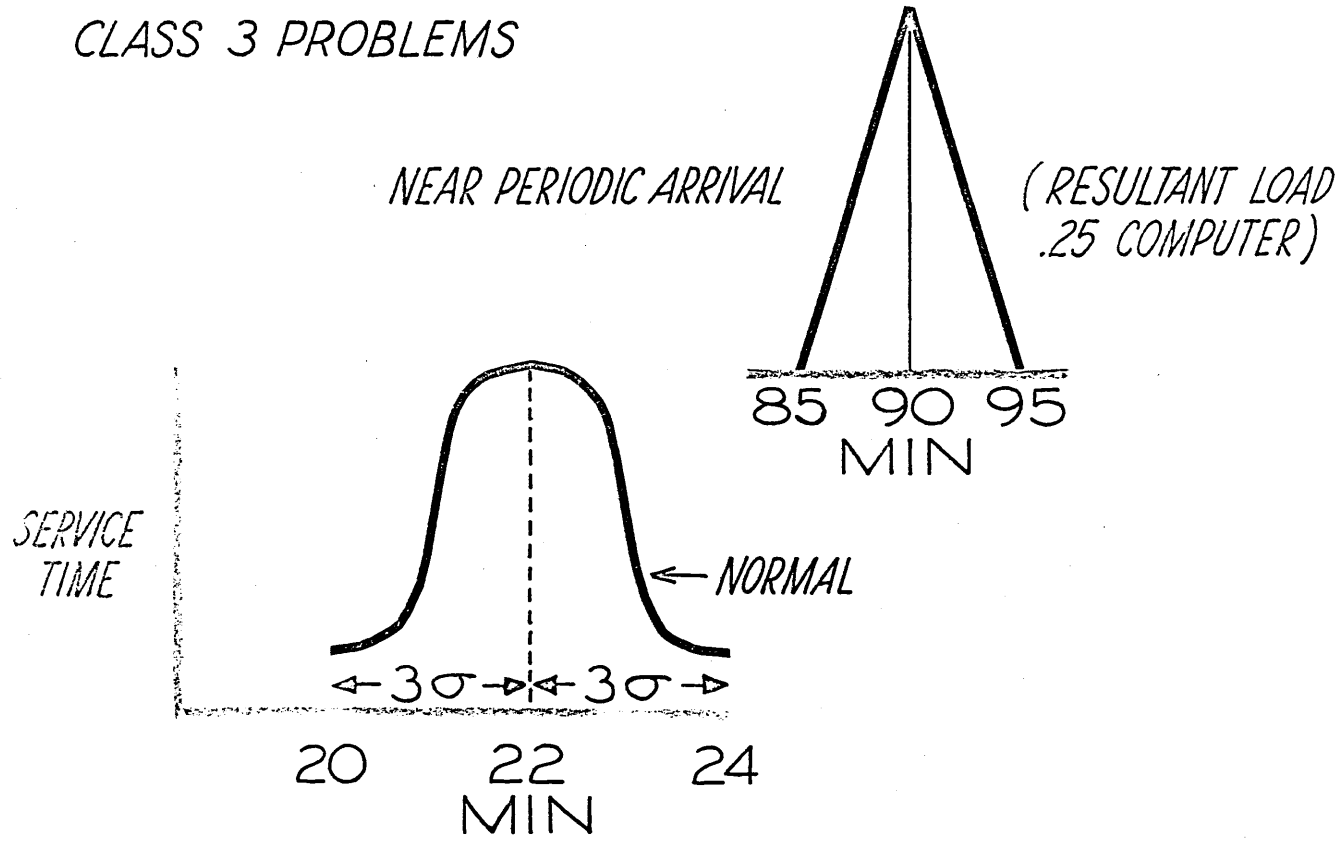


FIGURE 19

BASIC PROBLEM MIX

CLASS 4 PROBLEMS

POISSON ARRIVAL SELECTED TO GIVE LOAD OF 5 COMPUTERS

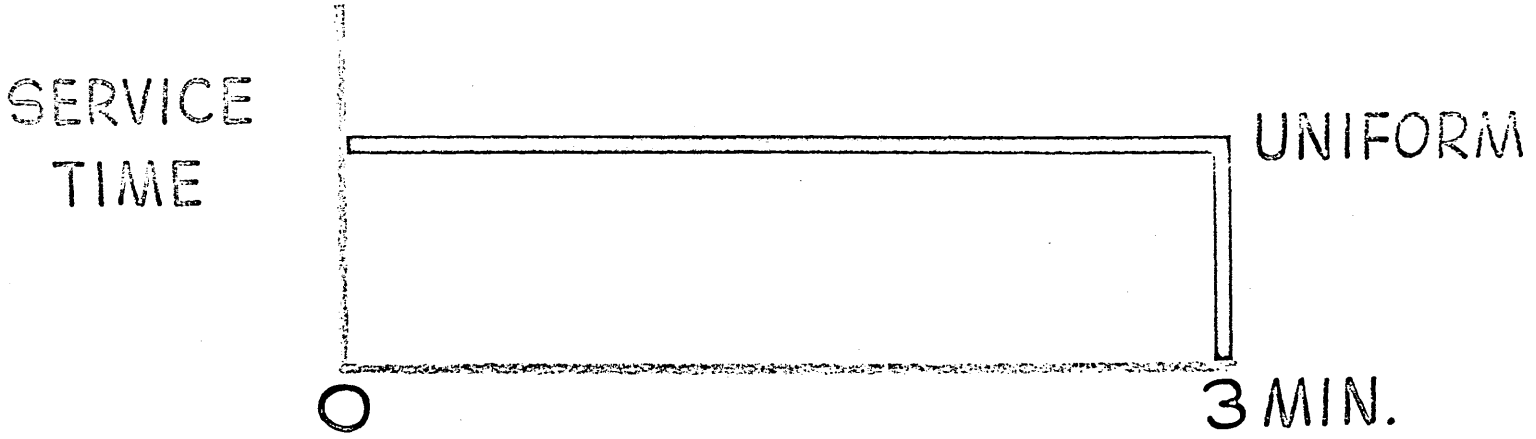


FIGURE 20

BASIC PROBLEM MIX

CLASS 5 PROBLEMS

POISSON ARRIVAL (SELECTED TO GIVE LOAD OF 1 COMPUTER)

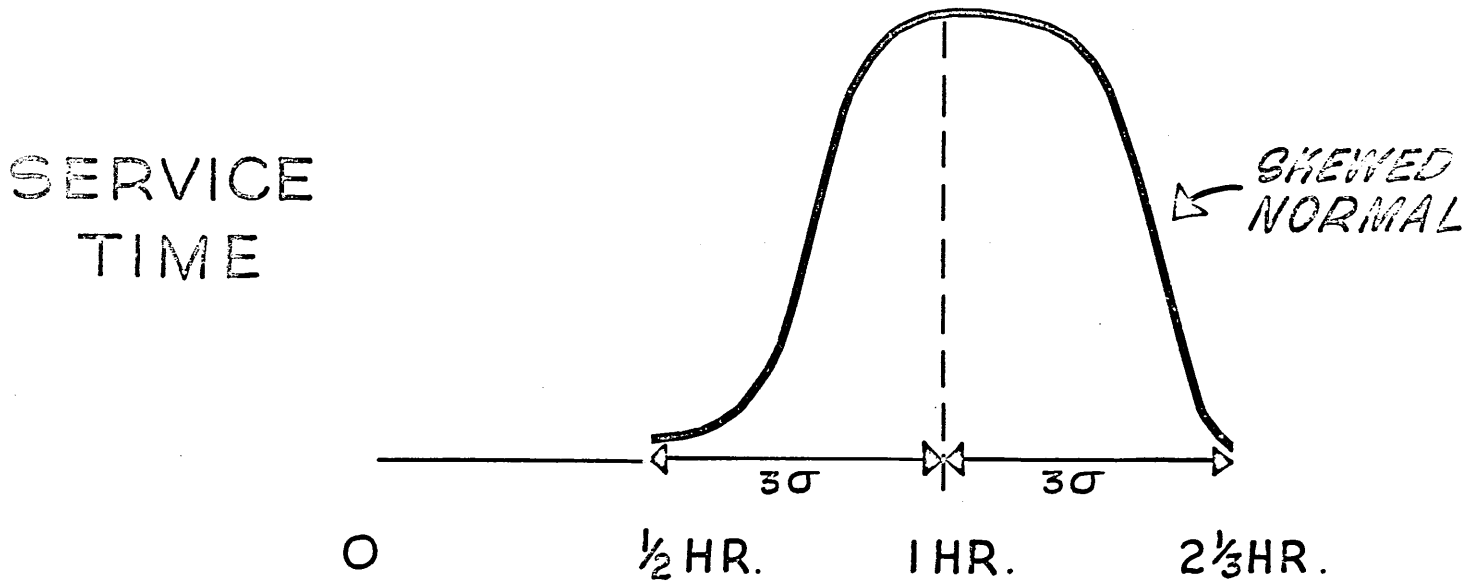


FIGURE 21

TYPICAL RESULTS OF QUEING STUDY

REMARKS	PROBLEM CLASS 2 PROB	2 t	3 PROB	3 t	4 PROB	4 t	5 PROB	5 t	CONCLUSIONS
1. BASIC PROBLEM MIX	90%	26 S	90%	43M	90%	14M	33%	7HR.	SLIGHTLY UNSTABLE CLASS 5 PROBLEM WAITING LIST GROWING
2. BASIC PROBLEM MIX BUT RESTART PENALTY REDUCED FROM 15 SEC. TO 1 SEC.	90%	26 S	90%	30M	90%	6M	90%	7HR.	STABLE SITUATION
3. BASIC PROBLEM MIX BUT CLASS 2 PROBLEM ARRIVAL RATE UP 50% WHILE CLASS 5 PROBLEM ARRIVAL RATE DOWN 50%	90%	26 S	73%	40M	83%	35M	74%	24HR.	UNSTABLE- CLASS 5 WAITING LIST GROWING
4. BASIC PROBLEM MIX BUT CLASS 4 SERVICE TIME UNIFORM 0-1½ MIN INSTEAD OF 0 TO 3 MIN	90%	25 S	90%	35M	90%	8M	36%	5HR	STABLE SITUATION

FIGURE 22

TYPICAL RESULTS OF QUEING STUDY

REMARKS	PROBLEM CLASS 2 PROB	2 t	3 PROB	3 t	4 PROB	4 t	5 PROB	5 t	CONCLUSIONS
5. BASIC PROBLEM WITH REDUCED (1 SEC) RESTART PENALTY AND CLASS 2 PROBLEM ARRIVAL UP 50% - CLASS 5 PROBLEM ARRIVAL DOWN 50%	90%	28 S	90%	32M	90%	8.7M	90%	4.3HR	STABLE SITUATION
6. BASIC PROBLEM WITH REDUCED (1 SEC) RESTART PENALTY ALL ARRIVALS REDUCED 10%	90%	25 S	90%	29M	90%	4.9M	90%	4.9HR	STABLE SITUATION
7. BASIC PROBLEM WITH REDUCED (1 SEC) RESTART PENALTY ALL ARRIVALS REDUCED 20%	90%	24 S	90%	27M	90%	4.6M	90%	4.7HR	STABLE SITUATION
8. BASIC PROBLEM WITH REDUCED (1 SEC) RESTART PENALTY ALL ARRIVALS REDUCED 30%	90%	23.6 S	90%	23M	90%	3.9M	90%	3.4HR	STABLE SITUATION

FIGURE 23

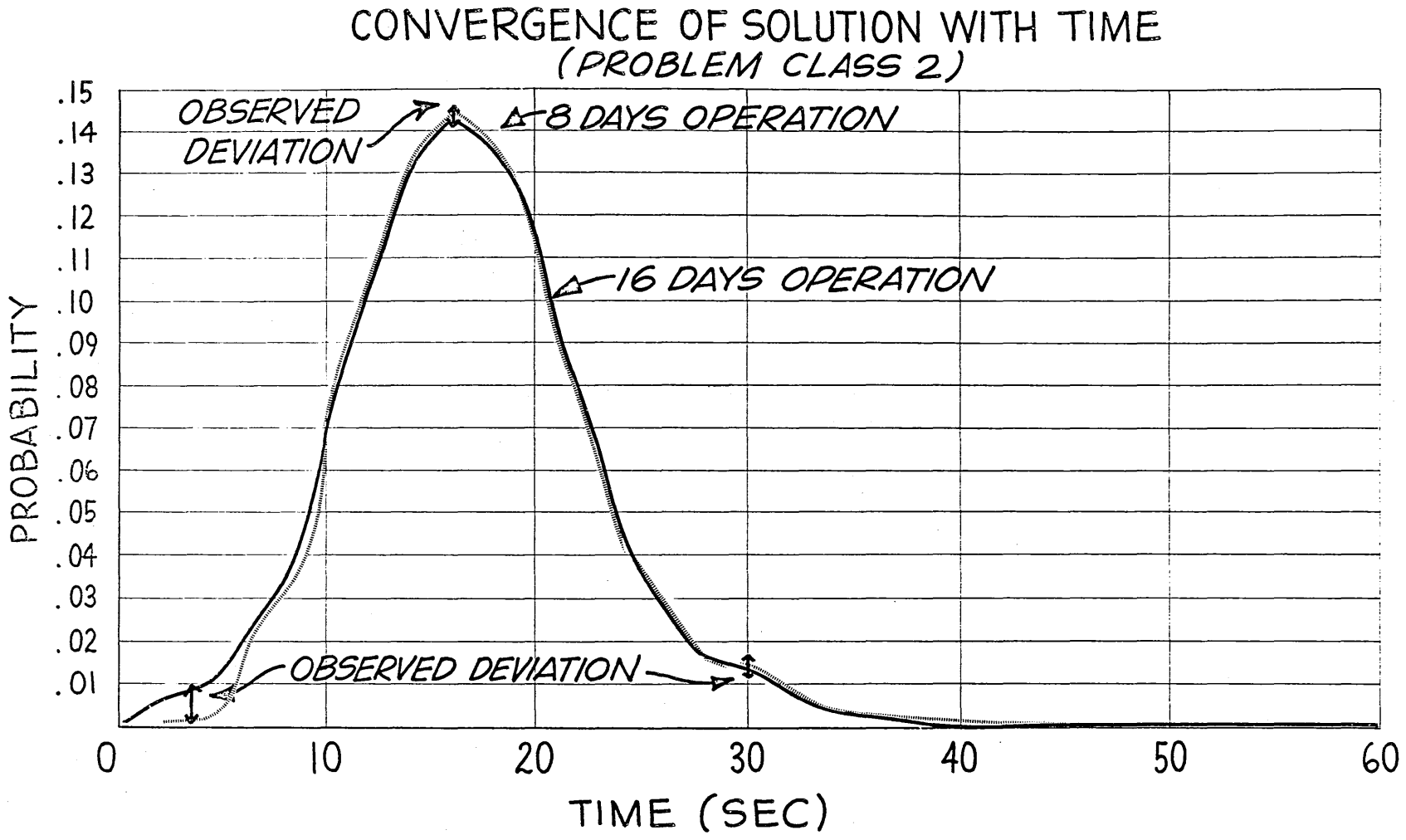

CONVERGENCE OF SOLUTION WITH TIME
(PROBLEM CLASS 2)
PROBABILITY
.15
.14
.13
.12
.11
.10
.09
.08
.07
.06
.05
.04
.03
.02
.01
OBSERVED DEVIATION
8 DAYS OPERATION
16 DAYS OPERATION
OBSERVED DEVIATION
0
10
20
30
40
50
60
TIME (SEC)

FIGURE 24

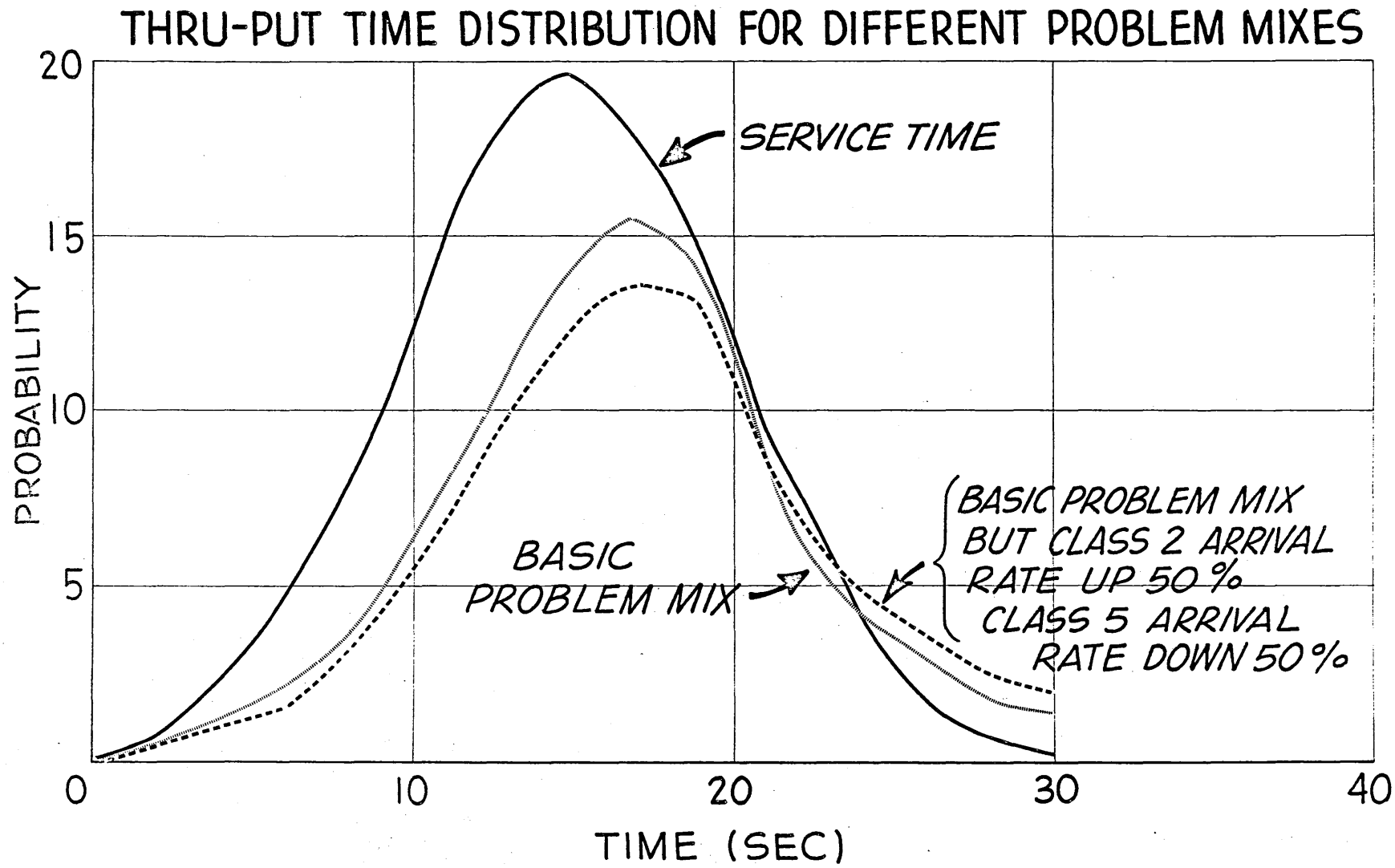

THRU-PUT TIME DISTRIBUTION FOR DIFFERENT PROBLEM MIXES
PROBABILITY
20
15
10
5
0
10
20
30
40
TIME (SEC)
SERVICE TIME
BASIC PROBLEM MIX
BASIC PROBLEM MIX BUT CLASS 2 ARRIVAL RATE UP 50% CLASS 5 ARRIVAL RATE DOWN 50%

FIGURE 25

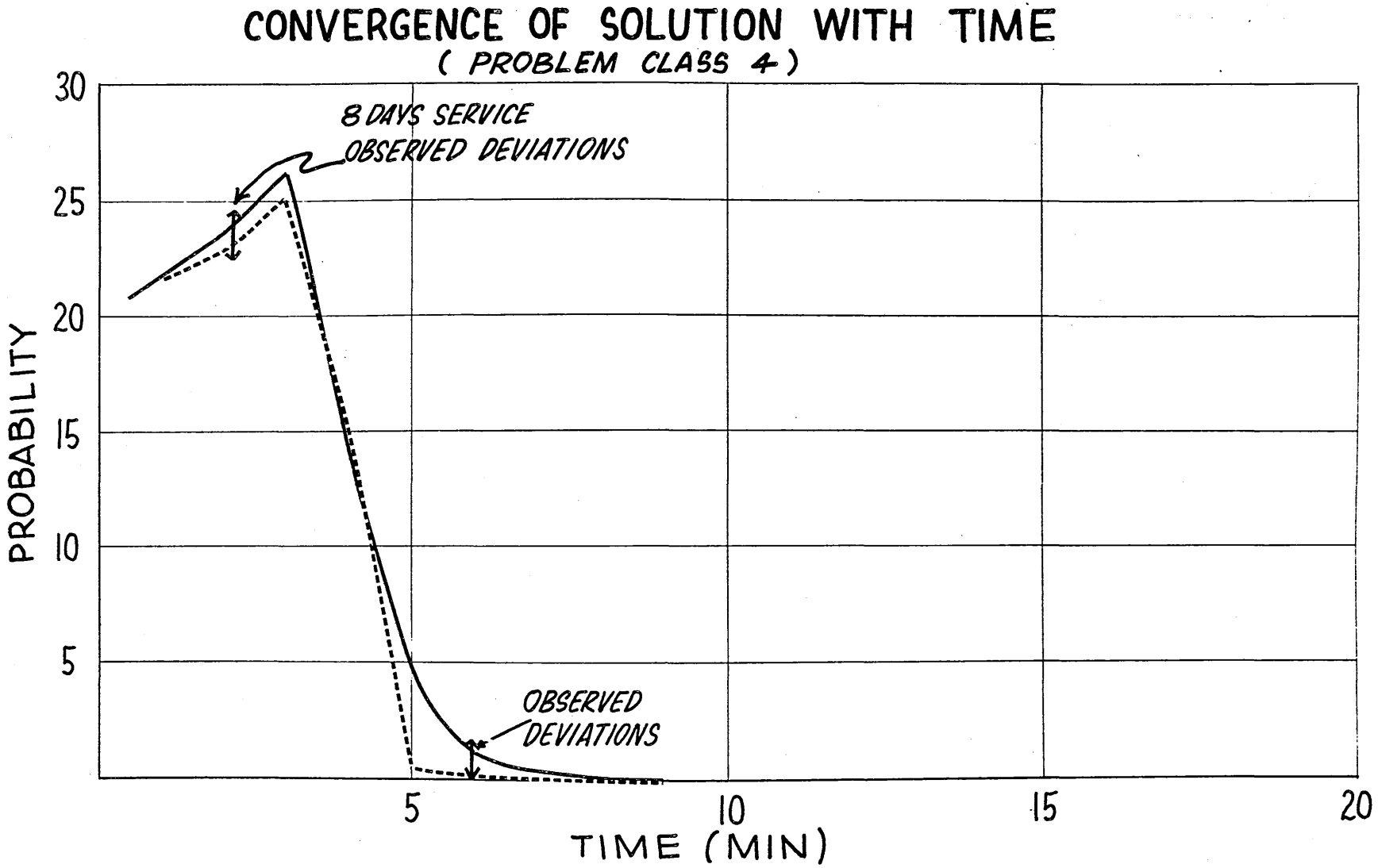

CONVERGENCE OF SOLUTION WITH TIME
(PROBLEM CLASS 4)
8 DAYS SERVICE
OBSERVED DEVIATIONS
OBSERVED
DEVIATIONS
PROBABILITY
30
25
20
15
10
5
5
10
15
20
TIME (MIN)

FIGURE 26

V. APPLICATIONS

A. GENERAL

The growth of the computer from the days of Babbidge and Hollerith to Von Neuman can be characterized rather simply. Computers have grown bigger memories, and faster and more complex arithmetic and control units. In our brief experience with the RW-40 system, it is apparent that small, inexpensive, independent modules can be cascaded to operate simultaneously on problems, producing the same or better results with less hardware. In the following three examples, (search and retrieval, sorting analysis, and event splitting) the diversity of arrangements and their power will be illustrated.

B. SEARCH AND RETRIEVAL

The process of search and retrieval from digital records in this system is limited only by the transfer rate of the storage medium. To illustrate the point, consider searching a magnetic tape with records that vary in length from 100 to 1000 words. The inter-record gap is 1-1/4 inches or 8.2 milliseconds. Assuming 25,000 operations per second, 200 operations can be performed in the inter-record gap. Thus, if only 200 operations per record are required on the average, the tape can be read directly into the computer.

If the amount of processing per record is larger, a Buffer Module is required. Assume the following sequence of operations: the tape unloads at 15 KC into the Buffer Module. At the start of the inter-record gap, the Buffer Module unloads to the Computer Module in, at most, one millisecond. Thus the Computer Module has the remaining time from the record gap plus the next record transfer time from tape to Buffer Module to process the data. A 100 word record is 7 milliseconds of transfer time and the 1000 word record is 70 milliseconds of transfer time for a total of from 380 to 1955 operations per record. If, then, one

assumed a group of n pairs of Buffer Modules and Computer Modules with a tape shunting mechanism for shunting the tape in a cycle of n, the effective number of operations available to any one Computer Module for processing the record (100 words to 1000 words) is from n . 380 to n . 1955.

The mechanism whereby the Tape Module is cycled from Buffer to Buffer is illustrated in Figure 27. When Buffer "i" is loaded, it disconnects from the Tape Module and connects to Computer "i". Computer "i" interrupts Buffer "i+1", and orders it to connect itself to the tape. The n^{th} Computer Module interrupts the first Buffer to complete the cycle. This transfer of the tape from Buffer to Buffer can be accomplished without actually bringing the tape to a complete stop.

The question of general purpose search logic versus special purpose search devices was considered. We soon found that the cost of a special purpose search device so nearly approached that of the general purpose Computer Module that it was necessary to re-assess the speed advantage. It was discovered that the general purpose device was almost as fast. This is because the special purpose search completed the full logic of the search on every record while a serial approach to the logic on a general purpose device often truncated quite rapidly.

C. SORTING ANALYSIS

The process of sorting data has been programmed on the RW-40 using about five different complements of equipment. Of these, two complements were selected for possible use:

1) Four Tape Modules, three Single Buffers, one Computer Module
2) Four Tape Modules, four Single Buffers, one Computer Module

Using a two-tape merge on which the first pass established strings limited only by core size, and on which the remaining phases were strictly merges, the sorting time was computed. With three Single

SEARCH & RETRIEVAL

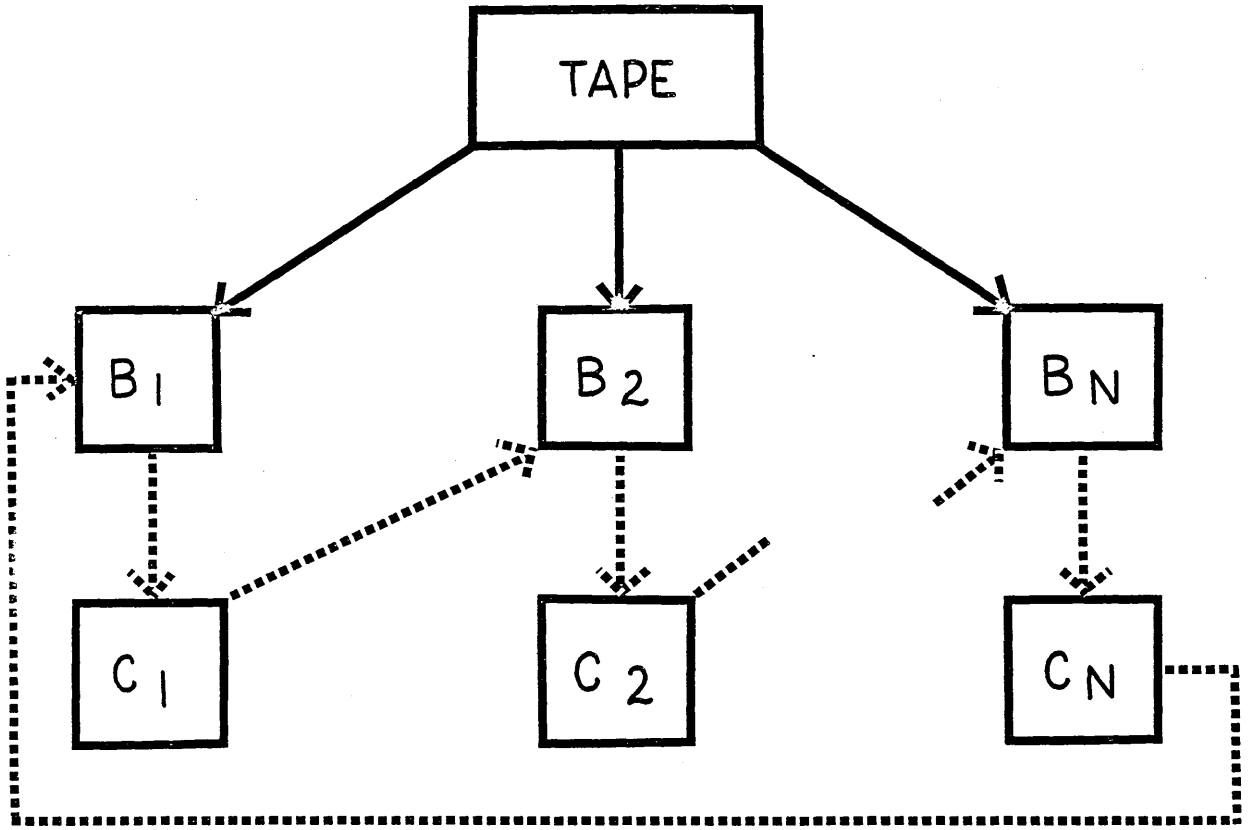


FIGURE 27

Buffers in operation, one is always reading tape, one is always writing tape, while a third is being sorted. Essentially, the sorting performance was calculated as a function of memory size, speed of core cycle, and tape speed. Figure 28 shows the sorting comparison. Items 5 and 6 of the table are the two RW-40 systems. Items 1 to 4 are other systems of generally larger core size and approximately equal cost. It is interesting to note that there is no significant difference between lines 1 and 2.

Aside from a high effective tape rate and the ability to read backward, saving rewind time, the Buffer Module is nicely adapted for these purposes. It is possible for the Computer Module to extract the sorting word and beginning record address from a Buffer Module, sort these and put a program back into the Buffer Module which writes on tape-- all without having to re-order the whole record in the core. Furthermore, an inter-record gap does not have to be suffered when jumping to the next copy address.

D. EVENT SPLITTING

When a single computer is being used to simulate a phenomena with a number of isolatable but interrelated events, it is usual to let each event progress a short distance in time and then compute the interaction between them in some arbitrary or ordered way. Of course the events are continuous and simultaneous in time and as such the simulation is approximate. The speed, or rather the cost, of computing time is now such that the discrete approximation to the continuous event is possible. However, the simultaneity remains untouched. Under these circumstances, it would appear that allocating independent Computer Modules to events, using a flexible interrupt and data communication system to effect the timely inter-relationships, is very natural. Here, of course, a Master Computer is used to synchronize time among the many events.

Air Traffic Control in its full-blown form requires more events to be computationally handled than one single machine can do with an effective approximation to reality. One certain way to handle the full problem is a multiple computer approach.

SORTING TIME (MIN.)

RECORD SIZE			10 WORDS			20 WORDS			100 WORDS		
NUMBER OF RECORDS THOUSANDS			10	20	100	10	20	100	10	20	100
SYSTEM PARAMETERS											
CORE SIZE (WORDS)	CORE CYCLE (μ sec)	TAPE SPEED (THOU. BITS/SEC)									
1. 32,000	12	90	4.5	11	47	8.3	18	87	45	92	518
2. 8,000	12	90	6.1	14	59	13	26	111	54	107	579
3. 10,000	6	375	3.0	6.7	34	5.5	11.8	54	31	57	289
4. 8,000	6	384	1.1	2.5	15	2.2	5.0	30	13	28	165
5. 3,000	10	390	1.3	2.8	17	2.6	5.8	34	15	33	190
6. 4,000	10	390	1.2	2.6	16	2.3	5.2	31	13	29	167

FIGURE 28

VI. SUMMARY

We were asked to solve a problem that exceeded the present state of the computer art. We believe that the concept explored--an automatically managed multiple computer system--provides a new direction for computer development.