B. B. Tasini
S. Winograd

Multiple Input-Output Links in Computer Systems

Abstract: An algorithm is developed for the analysis and design of computing systems having a multiplicity of concurrent and independent information sources and a lesser number of input channels to the processing elements of a digital computer; the input-output links work simultaneously with the processing and computing elements of the system. Utilizing this algorithm, such problems pertaining to the amount of hardware and the interconnection of its components can be resolved. For an optimum system design, its parameters are subject to two major criteria, i.e., the amount of information transmitted to the central processing elements per unit time, and the degree of overlapping of input output operations with computing (or other processing operations) that can be attained. Part I of this paper is devoted to information sources of the sequential access, serial-by-character (or-bit) transmission class. Magnetic tape units are specifically dealt with. With minor modifications the method is applicable to other information sources of the same class. In Part I, three fundamental system configurations are discussed and results of computations are summarized. In Part II, information sources of the quasi-random and random access class are investigated within the framework established in Part I. Disk files were selected as a specific representative of the quasi-random and random access information sources.

Introduction

It is well known that a large disparity exists in rates of information flow between the high-speed electronic portions of a computing system and its peripheral equipment. This is primarily due to the upper bound imposed on the information transmission rates by the electromechanical input-output devices such as magnetic tapes, card and paper readers, magnetic drums, and magnetic disk files.

In order to increase the amount of information transferred to or from main memory of the computing system, it is desirable to provide several information sources or sinks (i.e., inputs or outputs) operating concurrently and independently of each other, and simultaneously with the computing processes that take place within the central processing unit. In more common parlance, we are talking about computing systems having "READ/WRITE-compute" capabilities and a multiplicity of input or output devices operating concurrently and independently. The purpose of this paper is to develop an algorithm for the analysis and design of input-output links in which multiplicity of information sources or sinks will time-share the central processing unit of a digital computer.

In an excellent and comprehensive paper, Boyell[1] has tackled the problem of time-sharing in digital computers in various programming situations and has dispelled some of the illusions on this subject. While Boyell's approach to the problem may be termed "macroscopic", the approach adopted in this paper can be regarded as "microscopic" inasmuch as it goes into details, on the gross system level, in answering the various questions posed by the designers of computing systems. Recently, Boudreau and Kac[2] addressed themselves to the problem on a rather restricted basis, leaving some questions still unresolved, at least from the engineering point of view.

To illustrate the type of systems considered in this paper, Fig. 1 depicts in general outline one configuration of a system. For convenience we shall discuss, here and throughout this paper, input operations only, since they represent the most stringent conditions; output operations can be deduced by minor simplifying modifications and by assumption of a reversed flow of information. In Fig. 1 there are S independent information sources transmitting information concurrently

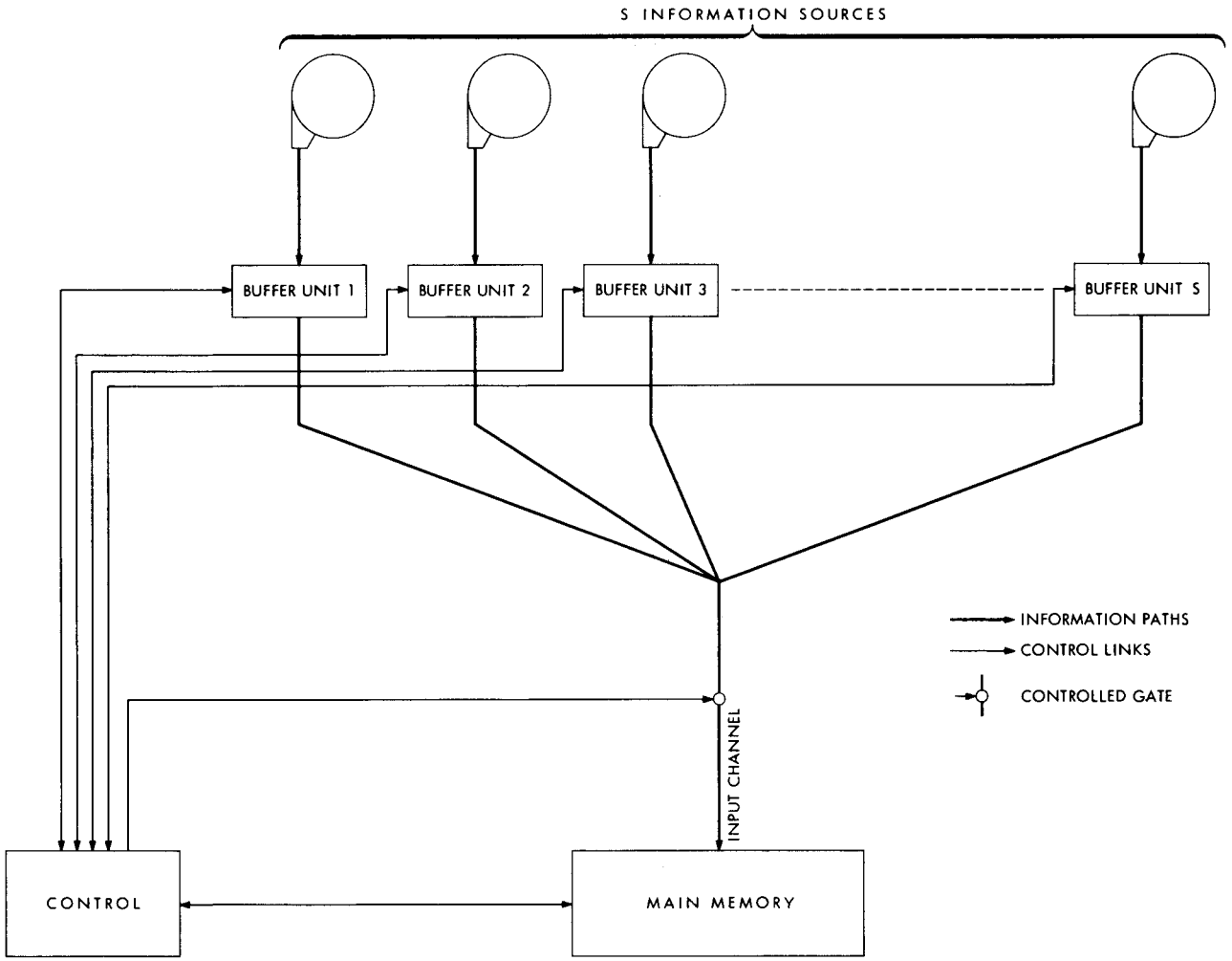


Figure 1 **A single-input-channel system.**

via a single input channel to the main memory of a digital computer or a computing system.

It can be seen that a single input channel[3] is time-shared by S information sources. The main memory, on the other hand, is shared by the input channel and the processing elements of the system. The entire input operation is controlled and coordinated by an organ designated as *Control*. The buffers shown provide a mechanism for assembling information into larger units for transmission to memory.

Another type of system is shown in Fig. 2. Here a group of information sources S_1 time-share input channel A and a group S_2 share input channel B. Each input channel requires a separate access to the main memory. In conventional systems this implies separate main memory units, each with independent controls and addressing. In general, the systems discussed in this paper fall into two broad classes: (a) single-input-channel systems, and (b) multiple-input-channel systems.

There are three classes of information sources that should be considered. One class includes all sources in which access to any item of information is sequential;[4] magnetic tapes, paper tape, and card readers are examples. The information out of such sources is transmitted in a serial-by-character (or -bit) fashion. The second class includes magnetic disk files and magnetic drums, magnetic card files, et cetera. These information sources are characterized by a quasi-random[4] access to large blocks of information within which access is sequential. Information sources of the third class are the random access memories. Magnetic core storage is a representative example. In Part I we shall consider information sources of the sequential-access variety. Magnetic tapes, because of their prominence, will be dealt with specifically. Information sources of the quasi-random and random variety will be the subject matter of Part II, in which disk files will serve as a vehicle of exposition.

To date, the design of multiple input-output links has been carried out by using some rule-of-thumb trial and error and mostly intuition, which sometimes

turned out to be right. The following questions, then, become pertinent to the systems designer:

What is the capacity and type of buffer?

What is the configuration of the optimum system?

How many input-output links can be connected in a system configuration?

What is the rate of information transmission into the system?

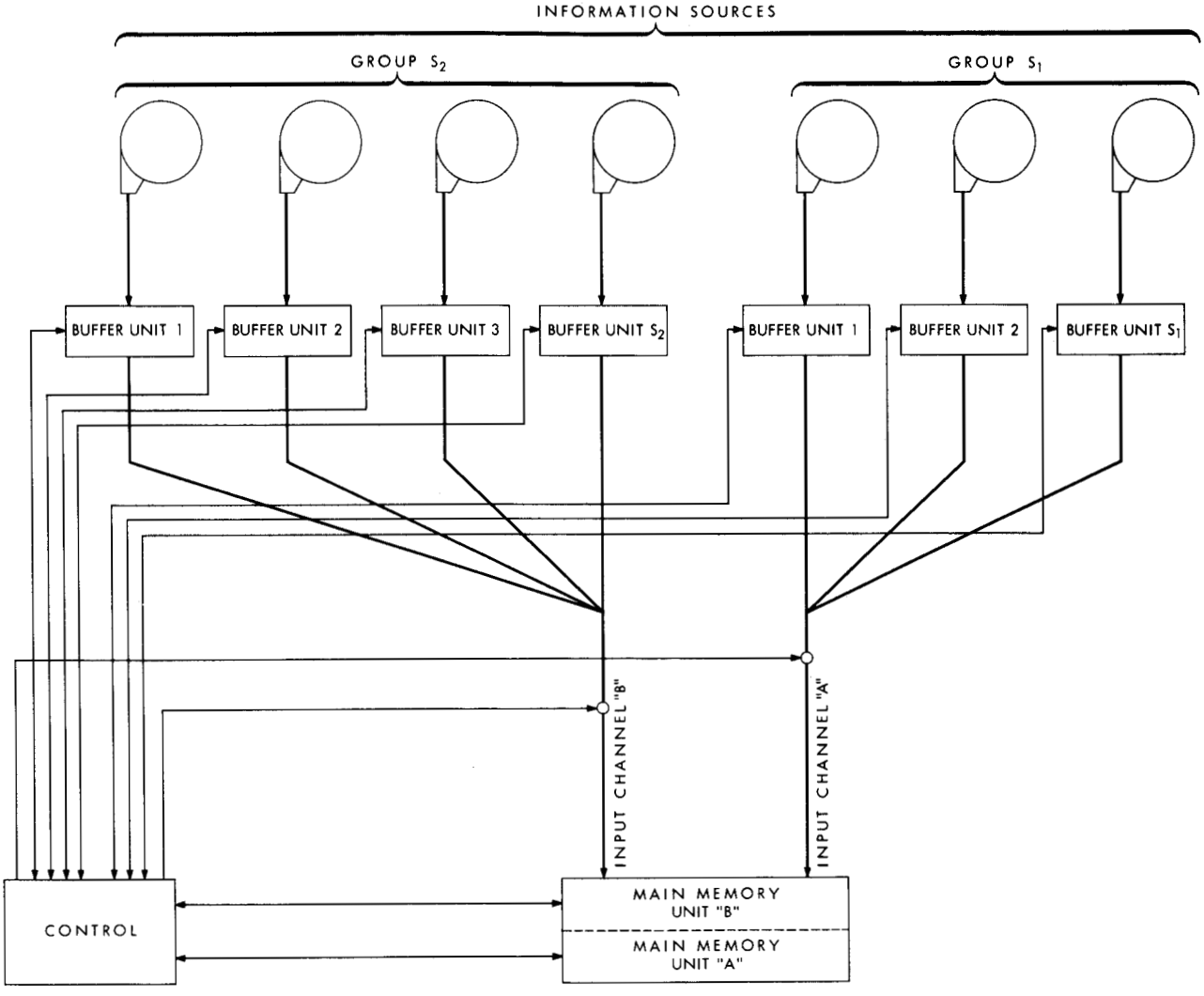
What is the degree of overlapping of input-output operation and processing that can be obtained?

These questions should be answered with the objective of attaining an optimum system, striking a balance amongst the amount of information transmitted per unit time, the degree of overlapping, and last but not least, the amount and complexity of hardware involved. Usually this balance will require a compromise design because of conflict in some of the attributes of the system.

The algorithm developed in this paper provides the answers to the questions we have just raised. The problem of independent and concurrent multiple-information sources with a lesser number of receiving facilities is related to the field of queueing and scheduling. However, the conventional theory and algorithms available could not be utilized since the operation modes of the systems of interest do not fit any of the models stipulated by classical queueing theory. In this paper we have developed the entire theoretical and mathematical formulation, based on the probabilities and statistics of the problem. The parameters which characterize a given system generate probabilities of the occurrence of certain events; these in turn are used to evaluate the various "macroscopic" entities which give a measure of the effectiveness of the system.

As was previously mentioned, this paper is comprised of two parts. In Part I we shall deal with information sources of the sequential-access type. After a general discussion on the method of approach we shall

Figure 2 **A multiple-input-channel system.**



describe three fundamental system configurations[5,6] and summarize some of the results that were obtained from actual computations[5,7] using the algorithm. Most of the material presented in Part I, especially that pertaining to system configuration and mode of operation, will be used as background to the presentation in Part II, which is devoted to information sources of the quasi-random and random-access class.

To summarize, the algorithm provides answers to such mundane problems as how much hardware is necessary and in what way should its various components be interconnected, in order to obtain an optimally effective system using two major criteria: *the amount of information transmitted per unit time* and *the degree of overlapping of input-output operations with information processing*. Unlike the costly and lengthy methods of simulation, the algorithm (when programmed for a digital computer) can provide the desired information quickly and conveniently.

It should be emphasized that the design process by which one arrives at the optimum system is that of analysis rather than synthesis. Furthermore, the process is iterative, i.e., one set of parameters is selected and the algorithm used to obtain a measure of effectiveness of the system. Then one parameter or more is varied at a time to measure again the suitability of the system with respect to the two major criteria or some other criterion deemed essential in a particular application. The process is repeated until the optimum system is obtained.

PART I: SEQUENTIAL ACCESS INFORMATION SOURCES

Method of approach

This section will be devoted to a general discussion of the problem and the parameters which characterize a system. The discussion will explain the assumptions made and the limitations imposed or inherent in the method.

Succinctly defined, the problem at hand is that of coordinating the operation of a multiplicity of information sources which are concurrent and independent of each other, and a lesser number of receiving facilities. In our case the receiving facilities are the input channel to main memory and the associated control. As an illustration, consider the system in Fig. 1. The buffer in each link is of finite capacity. The information is transmitted from the source to the buffer serially-by-character (or -bit). This will be true for the information sources considered in this paper. When this buffer is filled to its capacity a request for service by the input channel is sent to the control organ. The request is granted if the input channel is not busy servicing another buffer; information can then be transmitted from the buffer to main memory. Granting a request for service also involves designation of the locations in main memory allocated to the incoming information. However, if the input channel is busy transmitting the contents of another buffer, the granting of the service is delayed until the input channel is released. In fact, a queue of these delayed requests can form. Therefore, the total time required to complete the requested service comprises the time period spent in queue plus the time required to transfer the entire contents of the buffer to main memory.

A request for service must be granted and completed within a very definite time period. For example, for a serial-by-character information source feeding a single buffer, this period is the reciprocal of the character rate. In the event that a request can not be fulfilled within this time period, an additional delay ensues because of corrective action that must be taken. Since the buffer is of finite capacity, means must be provided to ensure no loss of information. During the corrective action information will not be transmitted and a way must be found to relocate the place at which this occurred in order to resume transmission. For example, for magnetic tape units the corrective action may take the following form: The tape is stopped, rewound to a known position (to the interrecord gap, say), restarted in the forward direction, and finally (by some means that we shall not go into here[6]) the location of the last character read is found. Reading can then be resumed, provided in the meantime the buffer had been cleared of the previous information. From the foregoing description, the queueing nature of the problem is rather apparent.

The systems considered in this paper are listed below. We shall consider two systems of the single-input-channel class and only one of the multiple-input channel:

a) Single-input-channel systems:
   1. The Minimum System
   2. The Parallel Buffers System

b) Multiple-input-channel systems:
   1. *p*-Input Channels System

The names assigned to the various systems succinctly describe the salient features of each; these, of course, will be detailed in our discussion. The system parameters at the designer's disposal are listed in Table 1.

Although we have stipulated in each case a particular mode of system operation (fully described later when we deal with each system) the set of fundamental and derived parameters in Table 1 proved to be quite general in its definition, and from experience in using the algorithm, adaptable to reasonable changes in other situations.

The algorithm developed here can yield a variety of results and information pertaining to a system, e.g., various probabilities of occurrence of certain events. However, when viewing the system as a whole, we are mostly interested in the two major attributes of a system that measures its effectiveness, namely, the performance of the system and the availability of main

Table I **Fundamental and derived parameters**

Symbol	*Definition* and *relation to other quantities*
Fundamental parameters	
X	The capacity of a buffer in main memory words.
S	The number of tape units operating concurrently.
F	The rate of character reading off tape (characters/second).
T_{mc}	Main memory cycle.
T_{ss}	The time duration for either acceleration or deceleration of a tape unit for starting or stopping, respectively.
T_b	The time required to unload one word from a buffer to main memory.
T_a	Time lag from the instant an input channel starts servicing a requesting buffer to the instant information can actually be transmitted to main memory.
r	The number of characters in a main memory word.
C_R	The number of characters in a record.
B	The number of buffers in parallel fed by a single tape unit. This applies to the Parallel Buffers System only.
p	The number of input channels. Applicable only to the multiple-input-channel systems.
Derived parameters	
T_c	The time required to read one character tape. $T_c = 1/F$
Y	The number of memory cycles required to transfer the contents of a buffer to main memory. Y = smallest integer $\geq X$
T_w	The time required to read one word off tape. $T_w = rT_c$
T	The time required to fill a buffer. $T = T_w X$
T_h	The time period during which the input channel is busy rendering service to a buffer unit. $T_h = T_a + YT_b$
T_s	The time lost for active processing and spent on retrieval of information. $T_s = 4T_{ss} + T_c C_R Q$ $Q = 1$ or 2
T_m	The time interval at the end of which unloading of a buffer into main memory (or another buffer) must be completed. $T_m = T_c + (B - 1)T$

memory. The performance of a given system is the amount of information received by the system per unit time, measured in suitable units.

The availability of main memory is defined as the fraction of total time main memory is available to the processing and computing elements of the systems to perform simultaneously tasks other than input-output; it is a measure of the degree of overlapping that can be obtained in a computing system.

Underlining the assumed mode of operation of the systems considered are some assumptions which we shall now enumerate. These hold for all systems discussed.

1. The computer is capable of receiving and processing of the information transmitted and will not interact with the tape units. A tape unit can be stopped only as a result of interaction among the requesting buffers.[8]
2. The discipline in the queue is that of first-come, first-served.
3. The central processing unit gives priority to the requests of the buffers, on main memory, over the requests of the processing elements.
4. The S information sources operating concurrently have the same nominal character rate. However, they do not operate in synchronism and read rate variations from the nominal value, whether transient or steady state, are assumed.[9]

It should be noticed that although we have characterized the information sources as sequential access, serial-by-character, we do not tackle the problem of the effect of information access time on the system performance or other attributes. Rather, it is assumed that the S information sources are in a state ready to transmit the information, after it had been accessed to, concurrently and continuously for a reasonable length of time. What is important here is the serial-by-character (or -bit) transmission. In effect we analyze a given system under the severest conditions of full load.

After discussing this background information we may now tackle the specific systems.

Single-input-channel systems

• *The Minimum System*

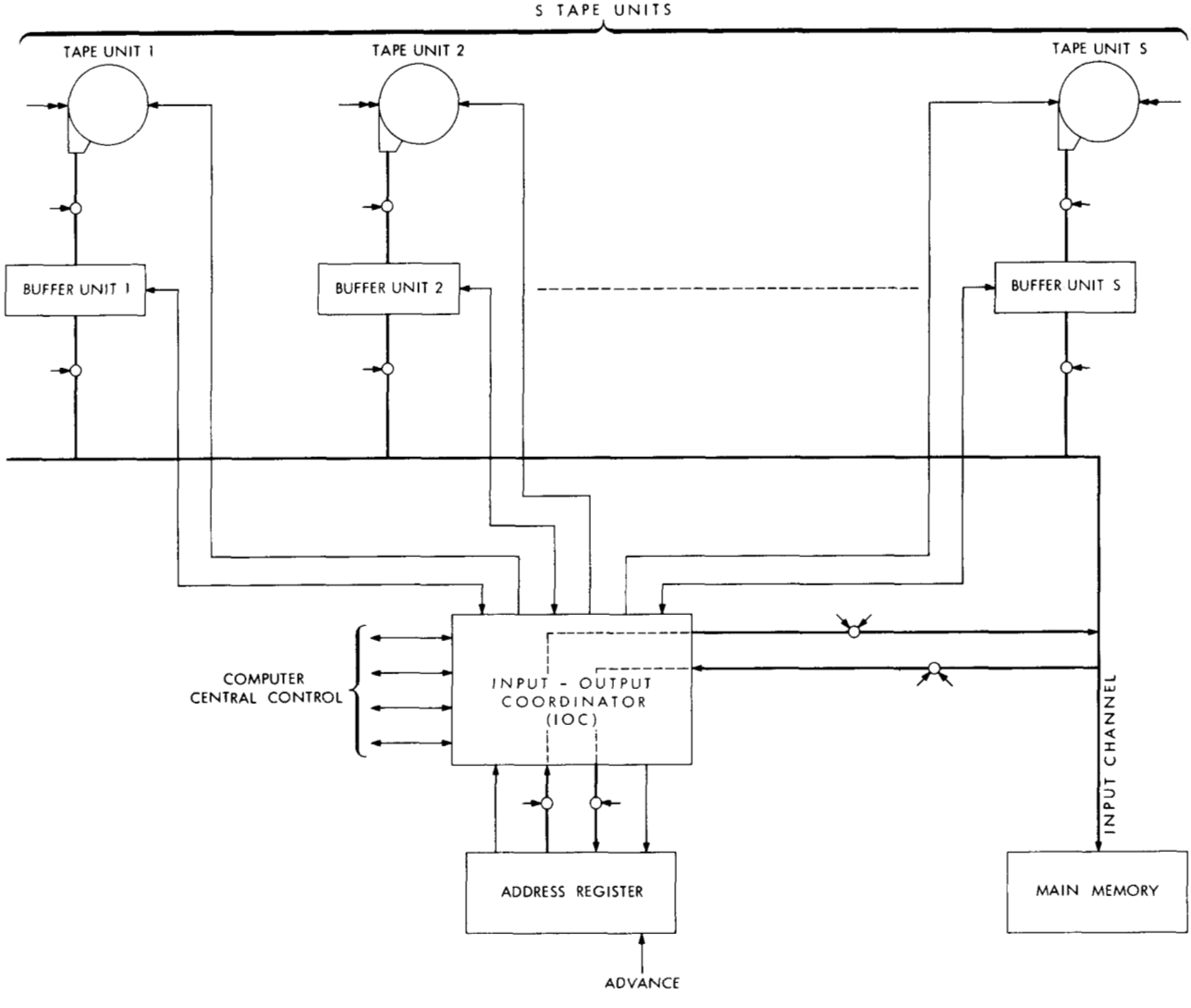
The term "Minimum System" accurately describes System I (Fig. 3). As the name implies, a single channel, one buffer unit per tape unit, and a single address register are the essential minimum requirements for a system handling S tape units operating concurrently. Each tape unit feeds a buffer which, in turn, transmits the information into the main memory of the computer via a single input channel, time-shared by all buffers. Furthermore, all tape units and their associated buffers share a single address register. This address register

contains the address in main memory to which, at any instant, the information will be sent.

Consider an application where S tape units are required to supply data to the Central Processing Unit (CPU). The main memory is considered to be a part of the CPU. S input instructions in sequence are partially decoded in the main program and transmitted for further decoding to an autonomous control unit which we shall call the *Input-Output Coordinator* (IOC). Thenceforth, the entire input operation will be controlled by the IOC. The task of the CPU is reduced to receiving and processing of information. Proper interlocking controls between the CPU and IOC should be provided. The instruction will identify a particular tape unit and provide the base address. The first word transmitted from a particular tape unit will be assigned to the location in main memory specified by the base address. As the input operation proceeds, this address will be continuously modified. It is assumed that a certain number of registers in main memory or in a separate memory unit are specifically assigned and accessible to the IOC. As soon as an input instruction is fully decoded in the IOC, the particular tape unit is selected and placed in operation. The S tape units will start operating almost simultaneously, since the time to decode the input instructions is negligible in comparison to the time it takes to place a tape unit in operation. Once in operation, the tape units are not in synchronism. The IOC renders the entire input operation independent of the CPU as much as possible, subject only to essential interlocking controls.

Let us now consider a tape unit and its buffer. Assume that the instruction to place the tape unit in operation has already been given. The tape unit transmits the information to the buffer in serial-by-

Figure 3 **System I—The Minimum System.**
Single-arrow gate: controlled only by IOC.
Double-arrow gate: controlled by IOC or CPU.



character form. A *character* is defined as a unit of information that may be identified outside the computer. It usually consists of a certain number of bits arranged according to a specified code. These bits are read in parallel. Every tape unit has a single-character storage buffer incorporated in it. This buffer is essential to the operation of the tape unit and consists of the reading amplifiers and associated flip-flops. Unless otherwise specified, this single-character buffer will be implied in the tape units of all systems discussed in this paper.

A second buffer, of a finite size, assembles these characters into larger units called *words*. A word is defined as that unit of information that can be transmitted in parallel from the buffer to main memory in one memory cycle. When the buffer is filled to its capacity, the following sequence of operations is initiated:

1) The buffer requests service from the input channel. This request is transmitted from the control circuits of the buffer unit to the Input-Output Coordinator.
2) If the input channel is free, the address register is cleared and loaded from main memory or a special register.
3) The entire contents of the buffer are transferred a word at a time to the main memory. The contents of the address register are increased by 1 for every word transmitted.
4) The contents of the address register are re-stored in the same location used in Step (2), until the next request for service from the tape unit under consideration.

In Step (1) the input channel may be busy servicing another buffer. The IOC forms a queue of all requests arriving while the input channel is unable to render service. The requests are then handled on a first-come, first-served basis. Steps (2) and (4) require one memory cycle each. A memory cycle will be required for every word transmitted in Step (3). All this activity should be terminated before a new character arrives from the tape unit; otherwise, the tape unit must be stopped in order to retrieve the lost information. The tape unit is then stopped, backspaced to the beginning of the record and restarted in the forward direction. This is the "corrective action" referred to previously.

The information on tape is assumed to be grouped into records, each of a certain fixed number of characters. Each record is terminated by an end-of-record mark and an end-of-record gap to allow space for deceleration. A gap is also provided at the beginning of a record.

Throughout this paper it is assumed that the computer is capable of receiving and processing the information transmitted and will never interact with the tape unit. The tape unit can be stopped only as a result of interaction among tape units. It is also assumed that main memory gives priority to requests of the input channel. The information from a tape unit is stored in sequentially located memory cells. Scattering of information if desired can be effected by programming.

The purpose of this detailed description of a minimum system is twofold. First, by presenting a conceivable hardware configuration, the theoretical formulation becomes closer to actual applications. The second purpose is to indicate the manner in which a system having a multiplicity of tape units operating concurrently can operate independently of the CPU, except for some essential interlocking controls.

The difference between the various systems discussed in this paper is reflected in the modification applied to, or the relationship existing between, some of the fundamental parameters and derived quantities (see Table 1). For the Minimum System, $T_b = T_{mc}$; $T_a = 2T_{mc}$; $T_m = T_c$. It remains only to define T_s, the time lost for active receiving and processing of information and spent on corrective action. It is given by $T_s = 4T_{ss} + T_cC_RQ$.

When a tape unit is required to stop, the tape unit is first decelerated and stopped; then restarted, accelerated and backspaced to the beginning of a record, where it is again decelerated, stopped, and restarted in the forward direction. There are, therefore, four time periods during which the tape unit is either accelerating or decelerating. The term $4T_{ss}$ in the expression for T_s accounts for these four "stops" or "starts". It should be noted, however, that T_s can still be defined for unequal deceleration and acceleration times. An appropriate modification will have to be applied to the term $4T_{ss}$. The magnetic tape traverses the space taken by the record proper at full normal speed. The time taken is given by the term T_cC_RQ. The quantity Q can have two values, depending upon the procedure adopted.

Procedure I: Assume that on the average the tape unit will be required to stop when the middle of the record passes under the reading head. The condition is recognized, but the tape is allowed to advance to the end of record mark and gap. Only upon reaching the end-of-record gap will the tape begin to decelerate. The tape will then be backspaced to the beginning of the record, where it will be decelerated and stopped. The tape unit is then restarted in the forward direction. In addition to the four "stops" and "starts" the tape traverses a distance equivalent to two record lengths. Therefore, for this procedure $Q = 2$.

Procedure II: Again, it is assumed that on the average the tape unit will be required to stop in the middle of a record. The tape is immediately decelerated and stopped. It is backspaced to the beginning of the record and then restarted in the forward direction. For this procedure, in addition to the four "stops" and "starts" the tape traversed a distance equivalent only to a single record. Hence, $Q = 1$. This procedure will be adopted in this paper.

T_s is independent of any system configuration and will apply, as defined by Procedure II above, for all systems.

The entire mathematical formulation and derivation of the algorithm for the Minimum System have been relegated to Appendix A. [In Appendix B of Ref. 6 an example is given that highlights the computations.] A pivotal and fundamental result is the fraction of total time that a tape unit will be expected to stop on the average, given that S tape units operate concurrently. It is defined by Eq. (A.8) of the Appendix and denoted by $P_{ave}(S)$. The tape unit utilization, performance of a system, and availability of main memory are derived using $P_{ave}(S)$. Tape unit utilization is defined as the fraction of total time that a tape unit is expected, on the average, to operate without being stopped. It is denoted by η and is given by $\eta = 1 - P_{ave}(S)$.

In the Appendix an upper and lower bound is established for η [Eq. (A.9)]; the result is $1 \geqq \eta > R_I/S$, where R_I is the integral part of (T_m/T_h) and is defined as the number of tape units that can operate concurrently with a resulting tape unit utilization of unity.

The performance of a system was until now defined as the amount of information transmitted and processed per unit time. To be more specific, let time be measured in seconds and amount of information in records. The performance of a system, expressed as the number of records transmitted and processed per second, is $A = \eta FS/C_R$.

It can be seen that system performance is proportional to tape unit utilization. Indeed, if we define the efficiency of a system as the ratio of the actual performance of a system A, to the performance of an ideal system for which $\eta = 1$, we find that the efficiency of a system is equal to the tape unit utilization η.

The availability of main memory is denoted by ρ and is given by $\rho = 1 - (T_h/T)S\eta$.

Availability of main memory is the fraction of the total time under consideration that main memory may perform tasks other than, and simultaneously with, input-output. It should be noted that availability of main memory and tape utilization are conflicting attributes of a system. An increase in tape unit utilization will cause a decrease in the availability of main memory. In an actual system design, a suitable balance should be obtained between these two entities.

The availability of main memory is derived with the assumption that during the entire time span under scrutiny data is transferred to main memory continuously. As we shall presently see, the availability of main memory is a measure of the potential overlapping of input-output operations with processing that can be obtained.

We now introduce a new entity which we shall call *the demand on main memory*. As the name implies, it is defined as the amount of time that main memory is busy fulfilling the requests for memory references by the CPU during processing. It can also be expressed as a fraction of total time and will be denoted by δ. The demand on main memory of any particular computing system can be derived from statistical studies of sample programs which are representative of the applications in which the particular computer will be used (Cf. Ref. 7, Appendix A, Section 1). The availability of main memory ρ, obtained by the algorithm can be looked upon as the "supply" of main memory time made available to the CPU by the peripheral equipment.

We now define the degree of overlapping of a particular system σ, as the ratio of the availability of main memory ρ, to the demand on main memory δ. When $\rho \geqq \delta$ we can state that processing and input-output operations are 100% overlapped. Thus we have: $\sigma = \rho/\delta$.

The results of actual computations[7] obtained for the minimum system can be summarized as follows:

Generally speaking, in any given system, the tape unit utilization and availability of main memory decrease with an increase of the number of tape units operating concurrently.

An increase in the character rate, other parameters being kept constant, results in deterioration of tape unit utilization.

The effect of increase in main memory cycle is detrimental to the system performance.

For any given system, tape unit utilization decreases with increase of buffer size.

The tape unit utilization η satisfies the inequality $1 \geqq \eta > R_I/S$.

Some of the above conclusions could have been deduced intuitively and expressed qualitatively, especially with the aid of some of the intermediate formulae and relationships existing among system parameters that were developed here. The significance, however, lies in the quantitative results and measures that the algorithm can provide.

A very important result that was obtained from the investigation of a rather extensive set of sample systems was, that for the minimum system a one-word buffer yielded maximum tape unit utilization for a given number of tape units S. While it was true that certain sample systems were found to have maximum tape unit utilization for a buffer of more than one word capacity, the slight improvement in practice would not warrant using more than one-word buffers unless an improvement in availability of main memory can be achieved.

In Figs. 4 and 5 a sample of results for one particular minimum system is provided. In Fig. 4, tape unit utilization is plotted as a function of buffer capacity (in main memory words), for $S = 1, 2, 3, \cdots 10$. The availability of main memory is plotted as function of buffer capacity in Fig. 5.

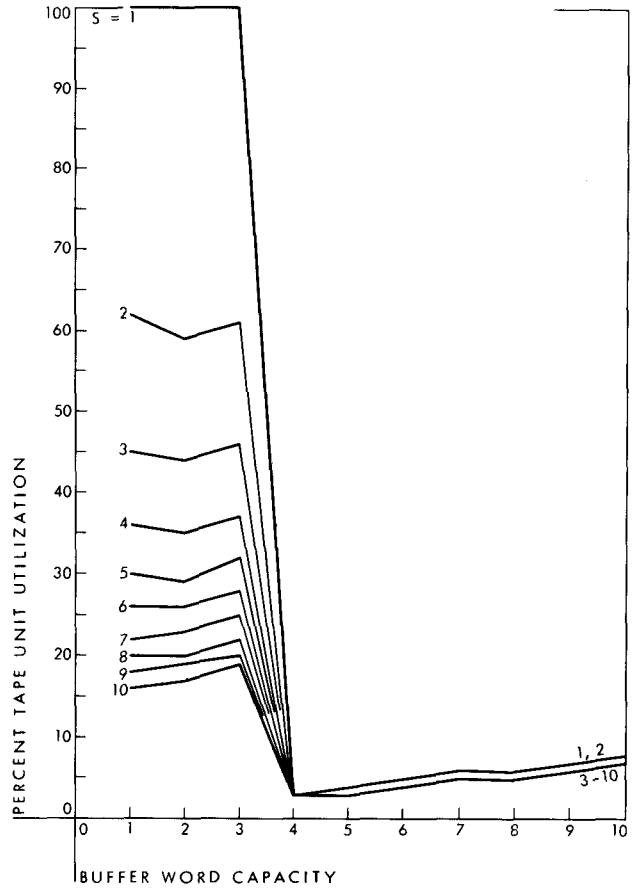
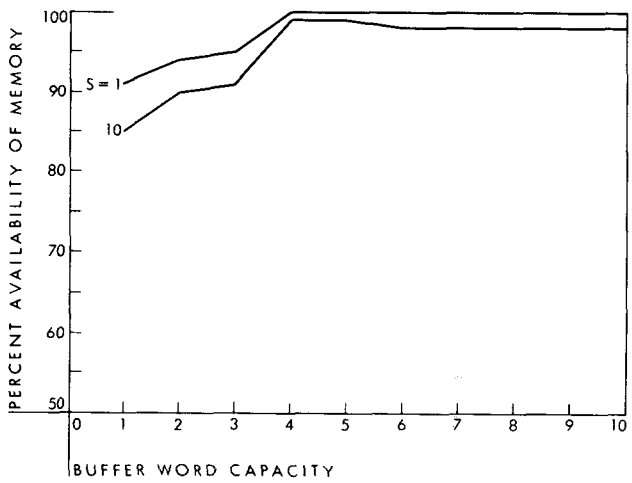


Figure 4 **Tape unit utilization plotted as function of buffer capacity (in main memory words) for a given Minimum System.**

$F = 15{,}000$ *char/sec* $B = 1$
$T_{mc} = 12$ *μsec* $r = 6$ *characters*
$T_b = 12$ *μsec* $C_R = 120$ *characters*
$T_{ss} = 10.0$ *msec*

Figure 5 **Availability of main memory plotted as function of buffer capacity for same Minimum System as for Figure 4.**



In general, for systems of the minimum system configuration, it was found that curves for tape unit utilization have the form as in Fig. 4, with a maximum around $X = 1$ or $X = 2$ and a sharp drop to a very low value at some X for which $T_c \leqq T_{mc}(X + 2)$. The curves for availability of main memory in Fig. 5 were found to be too congested; therefore, only the curves for $S = 1$ and $S = 10$ were plotted, thus defining a region wherein reside the curves for $S = 2, 3, \cdots 9$.

• *The Parallel Buffers System*

The Parallel Buffers System—System II—is shown in Fig. 6. It is identical in configuration and mode of operation to the Minimum System except that the buffer unit associated with a tape unit may have two or more buffers connected in parallel. In Fig. 6, a buffer unit is shown to be comprised of only two parallel buffers—Buffer I and Buffer II. The information is automatically transmitted to Buffer II whenever Buffer I is filled to its capacity and requests service from the input channel. The process is cyclical such that when Buffer II is full, Buffer I will receive the information transmitted from the tape unit. This can be extended to any number of buffers in parallel.

The relations enumerated for the Minimum System are applicable to the Parallel Buffers System except that T_m is now given by $T_m = T_c + (B - 1)T$, where B is the number of buffers in parallel contained in a buffer unit associated with one tape unit. It is assumed that all paralleled buffers are of the same capacity. However, the case of unequal buffer capacity can also be handled. A system such as System II may be desirable where very fast tape units and a comparatively slow main memory are employed. Providing B parallel buffers resulted in an increase of T_m. The term $(B - 1)T$ is the time required to fill the remaining $(B - 1)$ parallel buffers while one full buffer is in queue for service by the input channel. The mathematical formulation and derivation of the Appendix still holds for System II. The expressions for η, A, ρ, and σ stated for the Minimum System still hold here.

The Parallel Buffers System proved to be remarkably effective in improving the tape unit utilization over that obtained from an exactly similar system in the Minimum System configuration. In fact, one is tempted to use the term "overeffective", since the improvement in tape unit utilization is accompanied by a relatively low availability of main memory, because of the large amount of information that can be transferred.

A sample of results for a Parallel Buffers System is given in Figs. 7 and 8. Unlike the Minimum System, as the buffer capacity increases (for a given S) tape unit utilization for the Parallel Buffers System increases. In general, the curves of tape unit utilization were found to be "monotonically" increasing with buffer capacity until eventually $\eta = 1$ (or 100%) is reached and maintained therefrom. No local maximum

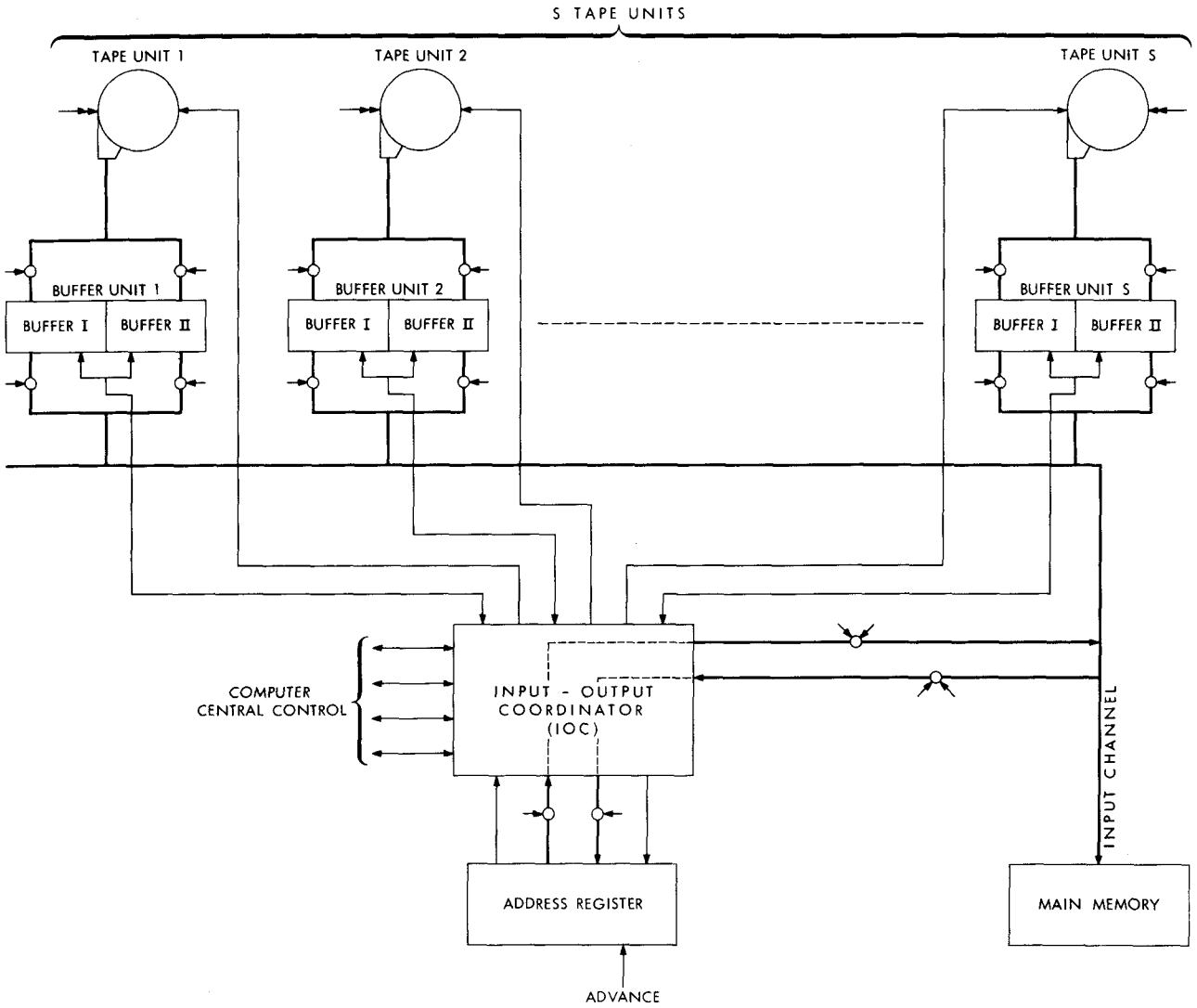


Figure 6 **System II—The Parallel Buffers System.**

of tape unit utilization (and a corresponding buffer capacity) could be located, as was the case with the Minimum System. The curves for the availability of main memory are also monotonically increasing with buffer size; however, they do so rather very gradually. In Ref. 7, the possibilities of improving the availability of main memory were investigated. The conclusion was that the improvement is possible, albeit very costly in terms of hardware.

Multiple-input-channel systems

• *p-Input-Channels System*

One multiple-input-channel system[6] is depicted in Fig. 9. Here, in System III, there are in general p groups of tape units and the buffer units associated with them. Each group is assigned to an input channel and an address register. For example, group 1 has S_1 tape units and S_1 buffer units sharing input channel 1 and address register 1 and working independently of and simultaneously with other groups. In general, for p groups, p input channels and address registers must be provided; p main memory units are assumed. In effect, we have combined several systems such as System I or II into a larger system.

It was stated above that System III is an assembly of systems like System I or II. Indeed, all parameters and definitions applicable to System I or II are valid for System III.

The results for System III can be deduced from the analysis of System I (or II) by a very simple process. Let n be the number of tape units that operate concurrently in the system as a whole. We may write $n = S_1 + S_2 + S_3 + \cdots S_p$.

The fraction of total time that a tape unit may be expected to stop on the average, given that a total of n

tape units operate concurrently, is given by

$$P_{ave}(n) = \left[\sum_{i=1}^{p} S_i P_{ave}(S_i)\right] \Big/ n \, .$$

$P_{ave}(S_i)$ can be derived from a system like System I or System II in which S_i tape units operate concurrently.

For the special case for which $S_1 = S_2 = S_3 = \cdots S_p = S$ we obtain

$$P_{ave}(n) = pSP_{ave}(S)/n = P_{ave}(S) \, .$$

Finally, we obtain for the system as a whole:

$$\eta = 1 - P_{ave}(n)$$

$$A = \eta F n / C_R$$

$$\rho = 1 - (T_h/T)n\eta \, .$$

The above expression for ρ is applicable to the situation where the p-memory modules are considered as a whole and only one module is accessed from the CPU at a time. For the case where the full p-tuple simultaneous access capability is to be utilized, we have $\rho_1 = 1 - (T_h/T)n\eta/p$, and this is the availability of main memory/module.

In general, the multiple-input-channel systems are not very effective in drastically improving tape unit utilization with a small number of input channels. By a brute-force method they provide a very costly solution for a system balanced with respect to tape unit utilization and availability of main memory; the latter only if one considers multiple access to the p modules from the CPU.

To illustrate the above statement, let it be required to improve the tape unit utilization of the Minimum System as represented by Fig. 4, by using a multiple-input-channel configuration. Furthermore, assume that each group of tape units will have the same number of tape units S, such that $Sp = n$ (the total number of tape units in the system); also we assume a one-word buffer. If $n = 6$, for example, we can choose either

Figure 7 **System II—Tape unit utilization.**

$F = 150{,}000$ *char/sec* $\quad B = 2$
$T_{mc} = 2$ *μsec* $\quad r = 6$ *characters*
$T_b = 2$ *μsec* $\quad C_R = 120$ *characters*
$T_{ss} = 10.0$ *msec*

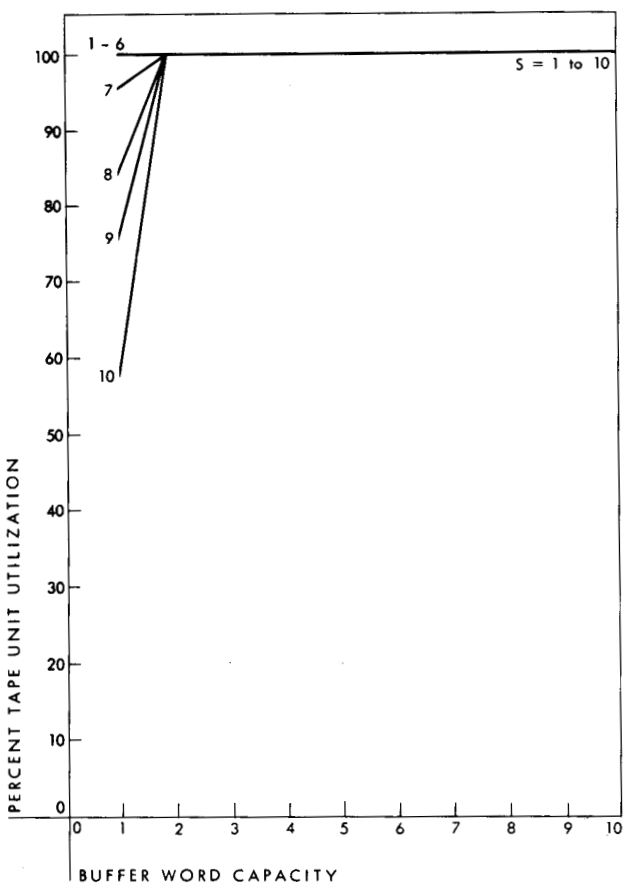
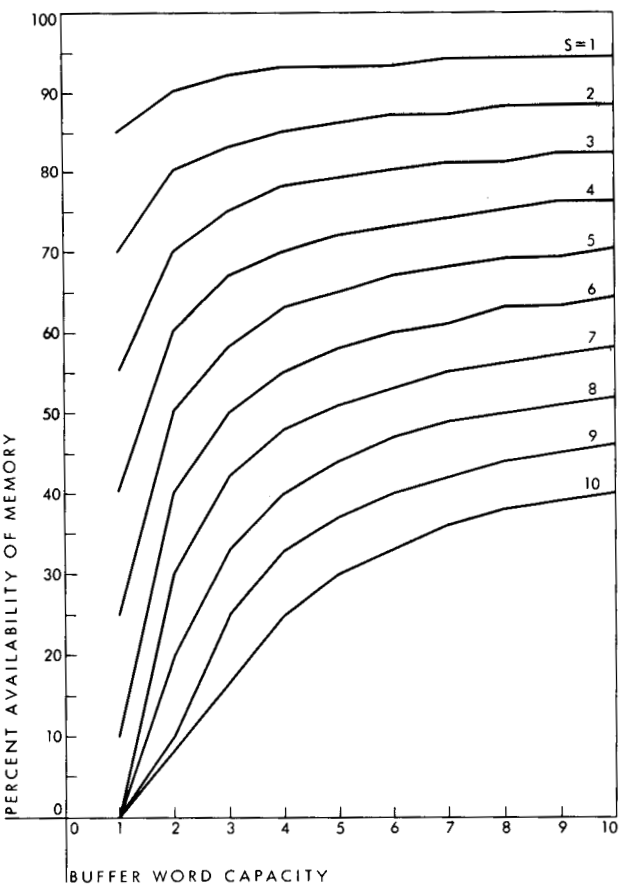


Figure 8 **System II—Availability of main memory for same system as in Figure 7.**



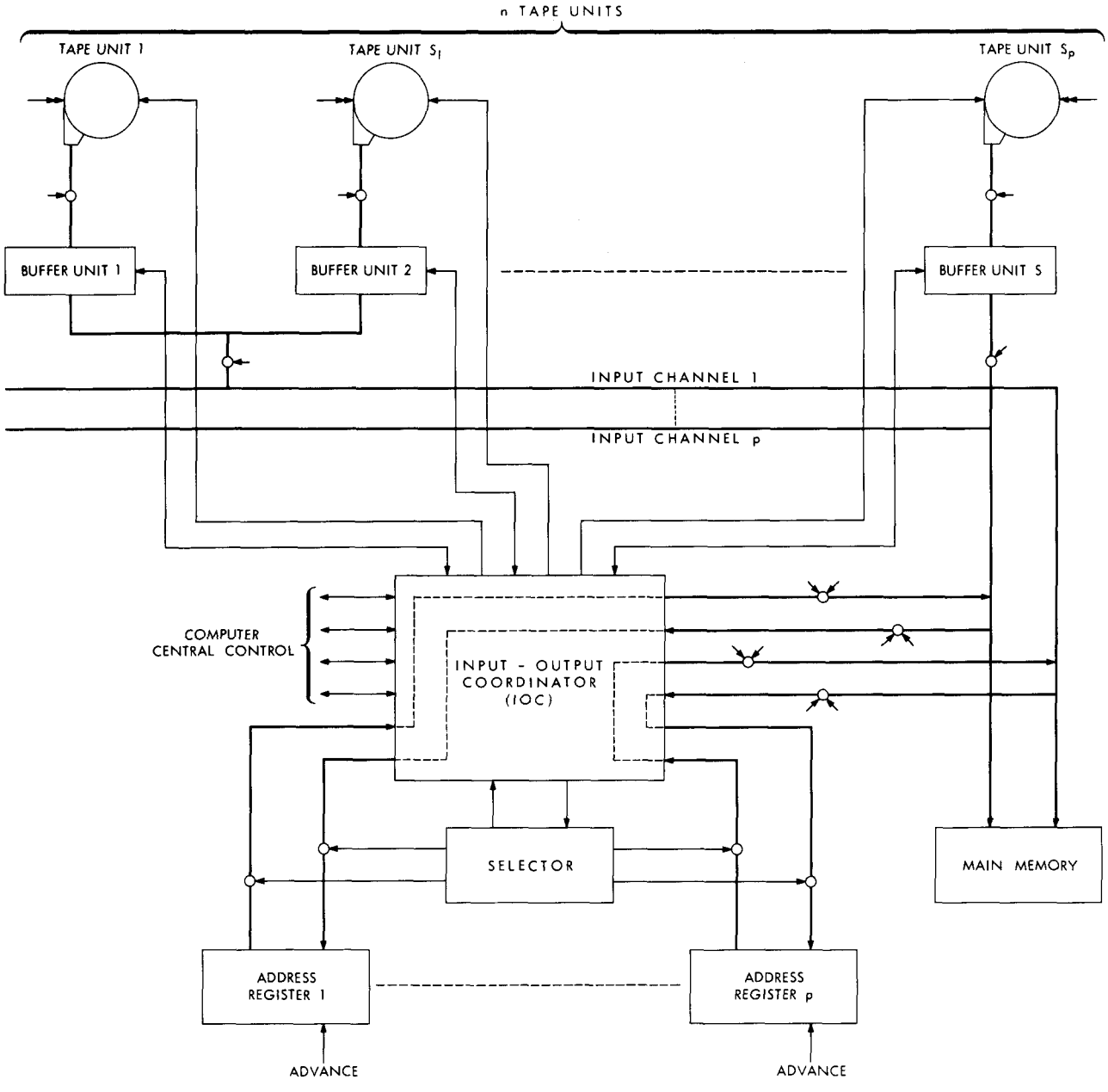


Figure 9 **System III—p channels and p address registers.**

$p = 6$, $S = 1$, or $p = 3$, $S = 2$, or $p = 2$, $S = 3$ with the corresponding $\eta = 1$, $\eta = 0.53$ and $\eta = 0.46$, respectively.

It can be seen that when the tape unit utilization of a Minimum System is low a considerable amount of hardware is required in order to improve it through a multiple-input-channel configuration.

Conclusions

Generally speaking, in systems such as were discussed in Part I, where a multiplicity of information sources operate independently and concurrently and where concurrence with the CPU is also desired, it is very difficult to obtain a high degree of both tape unit utilization and availability of main memory. Both of these requirements can be realized only with increase in amount and complexity of hardware. In actual practice, once an application for a particular computing system has been singled out and specifications laid down, compromises can be made in order to reduce the amount of hardware below the level indicated by theory.

We would like to conclude the discussion by summarizing the salient features of each of the systems considered. This will be done in a general manner.

For System I, the Minimum System, it was found

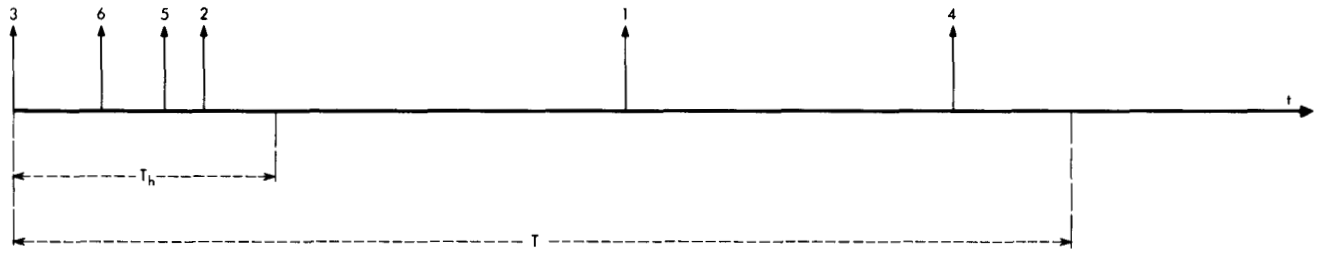


Figure 10 **An example of the random arrival of requests by buffers within the time interval *T*.**

that a one-word buffer will always provide maximum tape unit utilization. As the character rate increases, faster main memories will be required in order to obtain a high degree of tape unit utilization and availability of main memory.

System II, the Parallel Buffers System, was characterized by a high degree of tape unit utilization and a low percentage of availability of main memory. It can be shown that improvement in availability of main memory can be effected.

The multiple-input-channel systems proved to be useful in cases where high-speed tapes and a relatively slow memory are utilized. However, the multiple-input-channel systems do not usually achieve high tape unit utilization and availability of main memory with a small number of input channels.

Appendix A: The minimum system. Mathematical formulation and derivation

A tape unit, or several tape units, will have to be stopped if too many buffers request service from the input channel almost simultaneously. In order to find the probability that this will occur, the problem is broken into two parts:

Problem I: What is the probability that L buffers will request service from the input channel almost simultaneously? This will require a definition of what is meant by "almost simultaneously".

Problem II: Under what conditions will the L buffers be j too many (which means that j tape units will have to be stopped) and what is the probability that this will occur?

Solution to Problem I: As a first approximation, it is assumed that all S buffers will request service in any interval T. Actually, some tape units may be stopping; as a result, the number of buffers requesting service in any interval T is less than S. Let the interval T start at the instant any buffer is granted its request for service. The probability that L additional buffers will request service during the time interval T_h of the beginning of T is sought. We define that $(L + 1)$ buffers request service almost *simultaneously* if L additional buffers request service while the first one is being serviced. Figure 10 is an illustration of one possible way in which the buffers request service within the interval T. An upward arrow indicates the instant a buffer requests service and the number above the arrow indicates a particular buffer. In this example $L = 3$. It is assumed that the tape units are not synchronized and are independent of each other. The probability that a specific buffer (say buffer #1) will request service within the interval T_h is: T_h/T. The probability that it will not is: $(1 - T_h/T)$. The probability $P^L(S)$ that L *additional* buffers (out of the remaining $(S - 1)$ buffers) will request service within the interval T_h is:

$$P^L(S) = \binom{S-1}{L}(T_h/T)^L(1 - T_h/T)^{S-L-1}. \qquad (A.1)^*$$

Equation (A.1) is derived from the probability that $(S - 1)$ Bernoulli trials with probability u for success and $v = 1 - u$ of failure, result in L successes and $(S - L - 1)$ failures $(0 \leqq L \leqq S - 1)$. Here, success will imply that a request for service arrives within T_h, and failure, its negation.†

Solution to Problem II: In an interval T_m, $R = T_m/T_h$ buffers can be serviced. If a buffer will have to wait $(R - 1)T_h$ or less from the instant it requested service to the instant it received it, the tape unit associated with that buffer will not be stopped. If the time it must wait is more than $(R - 1)T_h$, the tape unit will be stopped. In general, R is not an integer and is made of the integral part R_I and the fractional part R_f, i.e., $R = R_I + R_f$. In other words, if *less than* R_I additional buffers request service, none will be stopped. The emphasis on "less than R_I" is required because the first buffer used as a reference to start the interval T must be included. If more than R_I additional buffers request service, at least one tape unit will be stopped. The $R_I{}^{\text{th}}$ additional buffer requesting service will cause its tape unit to stop if the request arrived within $(1 - R_f)T_h$ of the beginning of the T interval under consideration; the tape unit will not be stopped if the request arrived later. Figure 11 illustrates the two cases for $R = 31/4$, $S = 4$. In the first case (Fig. 11a) the third additional buffer requests service within R_f as measured from the end of T_h and therefore its tape unit will not be stopped;

* $\binom{S-1}{L}$ is the binomial coefficient.

† See for example William Feller, *An Introduction to Probability Theory and its Application*, pp. 105-106.

while in the second case (Fig. 11b) the request arrives within $(1 - R_f)T_h$ causing the tape unit to stop.

We define the probability $P(j, S)$ that j tape units will stop when S tape units are operating concurrently as:

$P(j, S)$ = (Probability that $(R_I + j)$ additional buffers request service while the input channel is busy.)
× (Probability that the R_I^{th} tape unit will not be stopped.)
+ (Probability that $(R_I + j - 1)$ additional buffers request service while the input channel is busy.)
× (Probability that the R_I^{th} tape unit will be stopped.)

In order to evaluate $P(j, S)$, we must know the probability that the R_I^{th} tape stops given that $(R_I + N)$ additional buffers requested service. This probability will be denoted by $P_s(R_I + N)$ for $0 \leqq N \leqq S - R_I$. The R_I^{th} tape unit will be stopped if and only if R_I or more of the additional buffers request service within $(1 - R_f)T_h$. Again, the assumption that the $(R_I + N)$ tape units are not synchronized and work independently implies that the $(R_I + N)$ additional requests arrive any time within T_h with the same probability. Therefore, the probability that $(R_I + 1)$ requests out of the $(R_I + N)$ arrive within $(1 - R_f)T_h$ is:

$$\binom{R_I + N}{R_I + i}(1 - R_f)^{R_I + i} R_f^{N - i} \qquad 0 \leqq i \leqq N \qquad \text{(A.2)}$$

and therefore,

$$P_s(R_I + N) = \sum_{i=0}^{N} \binom{R_I + N}{R_I + i}(1 - R_f)^{R_I + i} R_f^{N - i} . \qquad \text{(A.3)}$$

Or, if we let $k = N - i$ we get

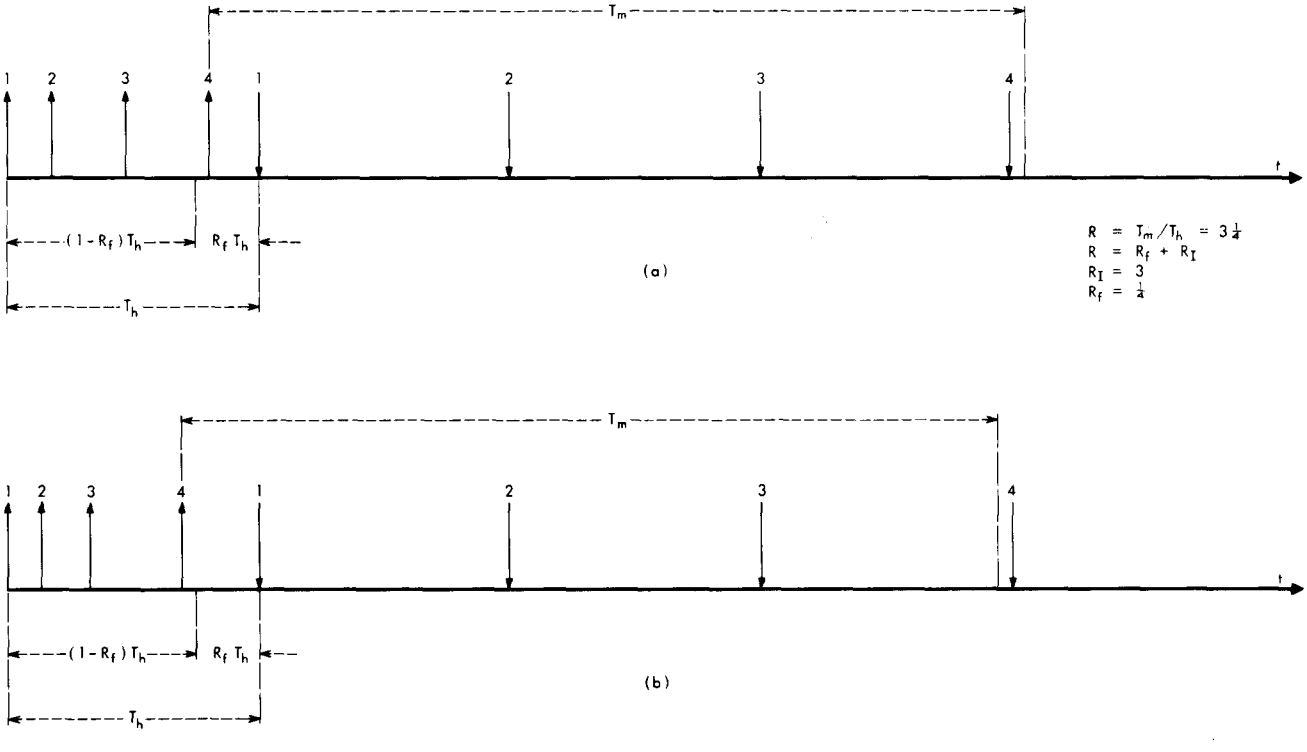
$$P_s(R_I + N) = \sum_{k=0}^{N} \binom{R_I + N}{k}(1 - R_f)^{R_I + N - k} R_f^{k} . \qquad \text{(A.4)}$$

We can now evaluate $P(j, S)$:

$$P(j, S) = P^{R_I + j}(S)[1 - P_s(R_I + j)] + P^{R_I + j - 1}(S) \times P_s(R_I + j - 1) \qquad 1 \leqq j \leqq S - R_I . \qquad \text{(A.5)}$$

As an example, let $P(3, S) = 10^{-3}$; this means that on the average every 1,000 requests for service from the address register three tape units will be stopped more or less at the same time. However, we are not merely interested in the probability of stoppage with respect to the number of buffers requesting service. A measure of system utilization is the percentage of total time j tape units will be stopped. As a first approximation, we select an idealized model in which no additional tape units can be stopped when less than S tape units operate concurrently. In a sufficiently large time span, let q be the number of requests generated when exactly S tape units were operating concurrently. The time taken for these requests is qT/S. From the definition of $P(j, S)$, the number of

Figure 11 **Graphical representation of the relationship existing among T_h, T_m, R_f, and R_I.**



requests that caused exactly j tape units to stop is $qP(j, S)$; and the time period during which only $(S - j)$ tape units were operative is therefore $qT_sP(j, S)$. Let $T_i(i = 0, 1, 2, \cdots, S - R_I)$ be the time period during which only $(S - i)$ tape units operate concurrently and none of these were stopped. We get

$$T_0 = qT/S$$

$$T_1 = qP(1, S)T_s$$

$$T_2 = qP(2, S)T_s$$

$$\vdots$$

$$T_{S-R_I} = qP(S - R_I, S)T_s .$$

The total time we consider is

$$t = qT/S + qT_s \sum_{i=1}^{S-R_I} P(i, S)$$

$$= q\left[T/S + T_s \sum_{i=1}^{S-R_I} P(i, S)\right] .$$

The fraction of total time that j tape units are not utilized is denoted by $P^t(j, S)$ and is given by

$$P^t(j, S) = qT_sP(j, S)/t$$

$$= SP(j, S)T_s \Big/ \left[T + ST_s \sum_{i=1}^{S-R_I} P(i, S)\right] . \quad \text{(A.6)}$$

$P^t(j, S)$ will sometimes be referred to as the probability with respect to time that j tape units will be stopped given that S are operating concurrently. By using Eq. (A.6) we can obtain $P^t(0, S)$, the probability that no tape unit will be stopped. Thus,

$$P^t(0, S) = 1 - \sum_{i=1}^{S-R_I} P^t(i, S)$$

$$= T \Big/ \left[T + ST_s \sum_{i=1}^{S-R_I} P(i, S)\right] .$$

To obtain the correct results for $P^t(j, S)$ we have to consider what happens when some tape units have stopped while $(S - 1)$, $(S - 2), \cdots,$ or $(S - R_I)$ were operating. Let $Q(j, S)$ be the correct values corresponding to $P^t(j, S)$. We proceed as follows:

$$Q(0, S) = P^t(0, S)$$

$$Q(1, S) = P^t(1, S)Q(0, S - 1)$$

$$Q(2, S) = P^t(2, S)Q(0, S - 2) + P^t(1, S)Q(1, S - 1)$$

$$Q(3, S) = P^t(3, S)Q(0, S - 3) + P^t(2, S)Q(1, S - 2) + P^t(1, S)Q(2, S - 1) .$$

Or, in general,

$$Q(j, S) = \sum_{i=1}^{j} P^t(i, S)Q(j - i, S - i) \quad 1 \leqq j \leqq S - R_I . \quad \text{(A.7)}$$

In order to compare the performance of systems, we require the expectation that a tape unit will stop. We define $P_{ave}(S)$ to be

$$P_{ave}(S) = 1/S \sum_{j=1}^{S-R_I} jQ(j, S) . \quad \text{(A.8)}$$

The quantity $P_{ave}(S)$ is the fraction of the total time during which a tape unit is not utilized for transmission of information. Hence, $[1 - P_{ave}(S)]$ will be defined as the tape unit utilization and denoted by η.

A very crude lower bound can be established for η. This is obtained from the following physical reasoning. It has been shown that when S tape units operate concurrently at least R_I of them will never stop. As a "worst case" example, assume that the remaining $(S - R_I)$ stop all the time. Therefore, $P_{ave}(S) = (S - R_I)/S = 1 - R_I/S$.

Actually, on the average, more tape units will continue to operate. Therefore

$$\eta > 1 - (1 - R_I/S) = R_I/S . \quad \text{(A.9)}$$

Equation (A.9) holds only for $S > R_I$. When $R_I \geqq S$, $\eta = 1$.

The performance of a system has already been defined as the amount of information processed per unit time. Let

D = Time unit chosen in seconds.

$C = DF$ = Number of characters transmitted by a tape unit during one time unit.

A = Number of records processed per unit time.

The performance is expressed by

$$A = SC[1 - P_{ave}(S)]/C_R = \eta CS/C_R . \quad \text{(A.10)}$$

It can be seen that the performance is proportional to the tape unit utilization η. It will be recalled that only one avenue of access to main memory is available. The problem to be tackled next is that of availability of main memory, expressed as fraction of total time, to perform other tasks besides input.

In y units of time y/T_w words can be transmitted by one tape unit; y is measured in the same units as those for T_w. But since a tape unit is not operating all the time, only $(y/T_w)[1 - P_{ave}(S)]$ words will be transmitted by one tape unit and $(y/T_w)S[1 - P_{ave}(S)]$ words by S tape units. Every X words fill a buffer and cause a request for service. The number of requests for service in y units of time is: $(y/XT_w)S\eta$. Main memory is busy for the entire duration of servicing a buffer, i.e., T_h. The total time main memory is busy in y units of time is: $yT_h/(XT_w)S\eta$. Let Θ be the probability that the main memory is busy. We obtain

$$\Theta = (1/y)(yT_h/XT_w)S\eta = (T_h/XT_w)S\eta . \quad \text{(A.11)}$$

But $XT_w = T$, so that

$$\Theta = (T_h/T)S\eta . \quad \text{(A.12)}$$

The availability of main memory ρ is given by

$$\rho = (1 - \Theta) = 1 - (T_h/T)S\eta \,. \qquad \text{(A.13)}$$

Symbols and their definitions used in this Appendix are given in Table 2.

PART II: RANDOM ACCESS INFORMATION SOURCES

Introduction

The problem of system balance of input-output links in which a multiplicity of serial-by-character (or -bit) information sources were employed was tackled in Part I.

Here, we address ourselves to the task of developing an algorithm for the analysis of input-output links having a multiplicity of information sources of the quasi-random and random-access class. Magnetic disk files and drums are examples of such information sources. We shall specifically deal with magnetic disk files; the analysis will, however, apply equally well to magnetic drums.

The problem confronting us may be succinctly defined as follows: Given a number N of random-access information sources (the reading arms of a disk file) operating concurrently in a certain stipulated manner, what is the least number of buffers M ($\leqq N$) or arms to yield the maximum utilization of either buffers or arms?

The other parameters of importance are the average access time and the average time to read an item of information.

It should be emphasized that because of assumptions concerning the probability distributions of certain entities, the algorithm can provide the information sought, subject to these assumptions.

In the following section we shall briefly discuss the operation of a magnetic disk file and, in more detail, the particular stipulated system. The results of actual computations will be presented in the Results and Discussion section. The mathematical formulation and derivation is given in Appendix B.

Description of system

Fig. 12 is an outline of the system to be analyzed. As shown, we have divided the system into two parts. One part consists of the disk file proper and the other of the input channel which includes the buffers.

The disk file consists of a number of magnetically coated disks rotating on a common shaft. A number of arms carrying the READ/WRITE heads are provided. The arms are mechanically positioned to read (or write) information on a specified location on any disk. The disk is coated on both faces and thus any arm carries actually two READ/WRITE heads (this is not shown in Fig. 12) to enable access to information on either of two faces of a disk.

To simplify our discussion, we shall henceforth refer to READ operations only; the WRITE operations will be executed in almost the same manner, assuming a reversed flow of information.

The information is stored in concentric tracks on the surface of the disk in either serial-by-bit or serial-by-character (parallel-by-bit) fashion. To retrieve an item of information an instruction to SEEK AND READ is given. The address of this item is sent to one of the

Table 2 **Symbols and nomenclature used in the mathematical formulation**

Symbol	*Definition* and *relation to other quantities*
$P^L(S)$	The probability that L additional buffers request service within T_h given that S tape units operate concurrently.
R	$R = T_m/T_h$
R_I	The number of buffers which can request service almost simultaneously within T_h, causing none of the associated tape units to stop. It is the integral part of R.
R_f	The fractional part of R.
$P(j, S)$	The probability that j tape units will have to be stopped, given that S tape units operate concurrently.
$P_s(R_I + N)$	The probability that the R_I^{th} additional tape unit will be stopped, given that $(R_I + N)$ additional tape units requested service within T_h.
$P^t(j, S)$	The probability with respect to time that j tape units will be stopped, given that S tape units operate concurrently. $P^t(j, S) = ST_sP(j, S) \Big/ \left[T + ST_s \sum_{j=1}^{S-R_I} P(j, S)\right]$.
$Q(j, S)$	The corrected values of $P^t(j, S)$. $Q(j, S) = \sum_{i=1}^{j} [P^t(i, S) \times Q(j - i, S - i)]$.
$P_{\text{ave}}(S)$	The expectation that a tape unit will stop, given that S tape units operate concurrently. $P_{\text{ave}}(S) = \left[\sum_{j=1}^{S-R_I} jQ(j, S)\right] \Big/ S$
η	Tape unit utilization. $\eta = 1 - P_{\text{ave}}(S)$
A	System performance expressed as the number of records transmitted and processed per second. $A = \eta SF/C_R$
ρ	Availability of memory, expressed as the ratio of the time main memory may perform tasks other than input-output to the total time under consideration. $\rho = 1 - (T_h/T)S\eta$

access registers and the arm is positioned to the disk and track specified. Once positioned at the target track, the arm waits until the beginning of the track is reached (as distinguished by a special mark) and then reads identification information which contains the disk number and the track number. These are compared with the information stored in the access register. If a match exists, information can be read as specified by the rest of the address and/or instruction. The absence of a match constitutes an error.

In general, the address of any particular item may represent up to six coordinates, depending on what atom of information is of significance for processing purposes. Thus, the address may specify the following:

Disk number
Track number
Sector number
Record number
Word or field number
Character or bit.

From the reading arm the information is transmitted via the input channel to main memory.

In most disk files to date, the reading arms can seek simultaneously; however, the actual information transmission is carried for one arm at a time. In other words, only one buffer is available. In this paper we would like to investigate systems wherein more than one buffer is provided, so that more than one arm can read simultaneously.

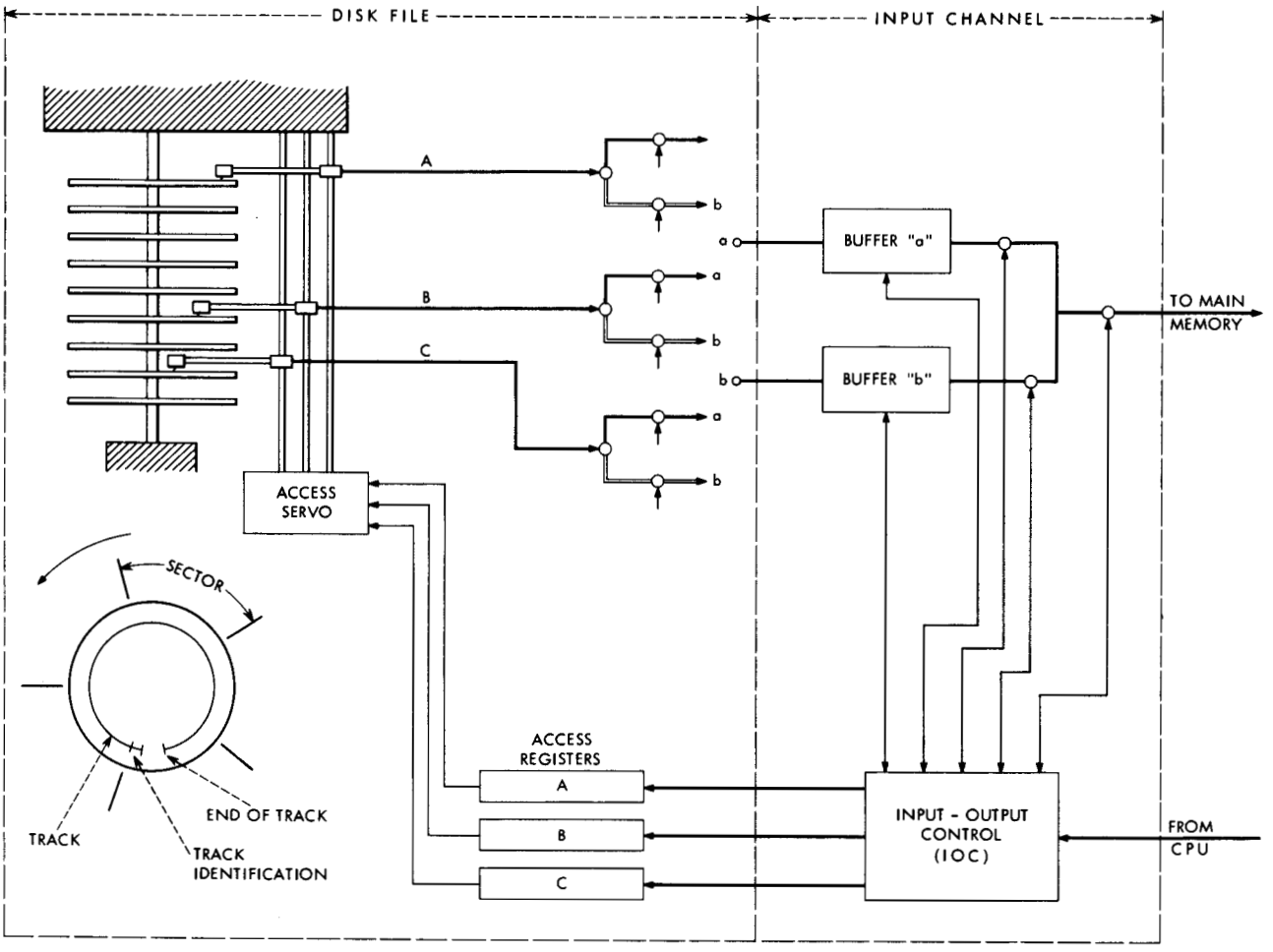
In Fig. 12 the system shown has three reading arms which share two buffers. These buffers, in turn, share the single transmission link to main memory. As shown, the input-output control (IOC) supervises the entire operation. In order to retrieve an item of information the following sequence of events is initiated.

1) An instruction (or several instructions in sequence) to SEEK AND READ is partially decoded in the central processing unit (CPU), and is then transmitted for further decoding and action to the IOC.

2) The address part of the instruction is set up in any access register which is found free. The address is interpreted by the access mechanism and the arm starts its travel to the target track.

3) After reaching the target track the reading arm

Figure 12 **The disk file and its input channel to CPU.**



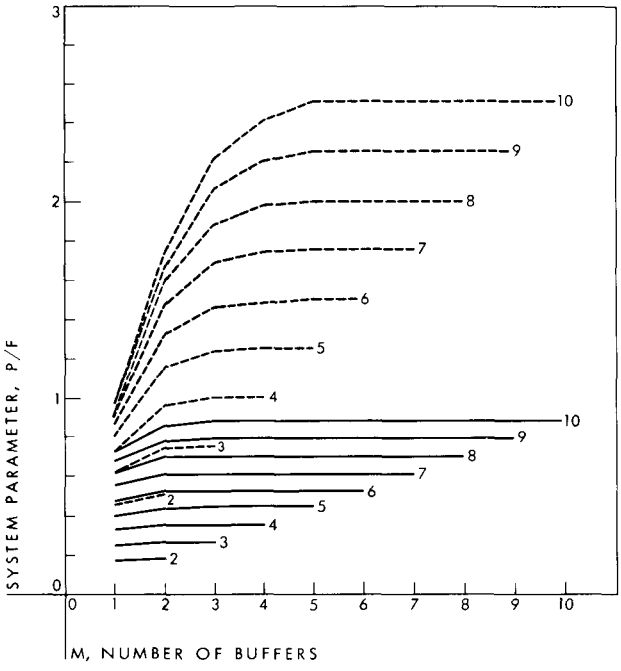


Figure 13 **P/F plotted as function of number of buffers M for $R = 0.5$, $S = 0.05$ and $W = 1.0$ (solid curves), and $R = 0.5$, $S = 0.2$ and $W = 1.0$ (dashed curves).**

waits until the beginning of the track is reached. The identification information (disk number and track number) is read and compared with the corresponding part of the address in the access register. This serves as a check for the functioning of the access servomechanism. An error signal will be transmitted when a mismatch exists. The arm then proceeds to "zero in" on the item of information as specified by the remainder of the information in the access register.

4) If a buffer is available, the information is read and transmitted to the buffer. The buffer is held by an arm for the entire duration of reading T_r. The transmission of information is delayed if a buffer is not available and Step (3) is repeated.

The information is transmitted serially by character (or bit). The buffer assembles the characters into larger units (words). When the buffer is filled to its capacity, the information is transmitted a word at a time to main memory under the control of the IOC. Although we describe here only a single input channel, the system can have any configuration described in Part I. In fact, we assume the mode of operation of the IOC and the buffers to be the same as described in Part I.

For the purpose of our investigation it is assumed that:

1) The only "bottleneck" is caused by the buffers. The transmission to main memory does not cause any delay. In other words, the rate at which the contents of the buffers are transmitted is sufficiently high. In Appendix B we shall discuss and derive a relationship expressing this rate as a function of the character rate, number of buffers, and memory cycle.

2) When an arm terminates the reading of information, it immediately receives another instruction to SEEK AND RETRIEVE another item of information; i.e., we assume that at any instant there is a backlog of demands on the disk file and thus we can test a system at full load.

Results and discussion

This section is devoted to the presentation of results of calculations using the algorithm developed in Appendix B. The computations were executed on an IBM 7090.

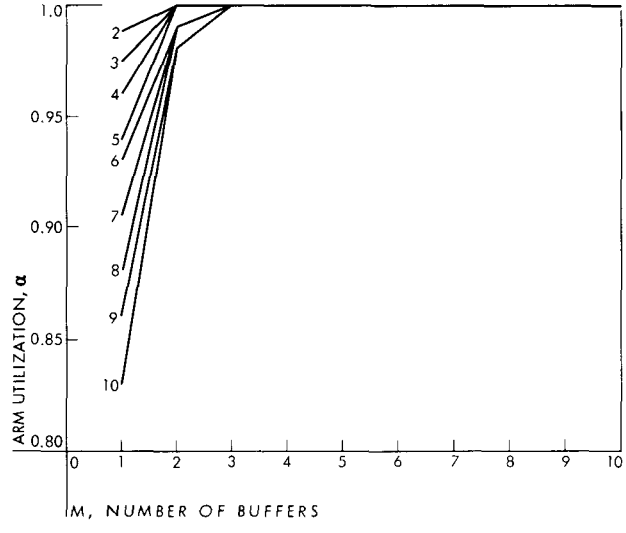
For the purpose of calculations the parameters R, W, and S can be normalized with respect to any of them. We have chosen to normalize with respect to W. The results are summarized graphically in Figs. 13, 14, and 15. Two sets of the parameters R, W, and S were used. In the first $R = 0.5$, $S = 0.05$, and of course $W = 1.0$. This corresponds to $T_r : T_s : T_w = 2 : 20 : 1$, i.e., the average time to read is a full revolution. The other set of parameters corresponds to a file with much shorter seek time, with $R = 0.5$, $S = 0.2$, and $W = 1.0$. ($T_r : T_s : T_w = 2 : 5 : 1$).

In Fig. 13 we have plotted P/F ($= M\eta$), which is proportional to system performance, as a function of the number of buffers. Each numbered curve corresponds to a particular value of the number of arms N. The results of both sets of parameters (R, W, and S) are contained in Fig. 13.

The following observations and conclusions can be made in regard to system performance:

1) For a given number of buffers M and R, W, and

Figure 14 **Utilization of an arm plotted as function of number of buffers M for $R = 0.5$, $S = 0.05$ and $W = 1.0$.**



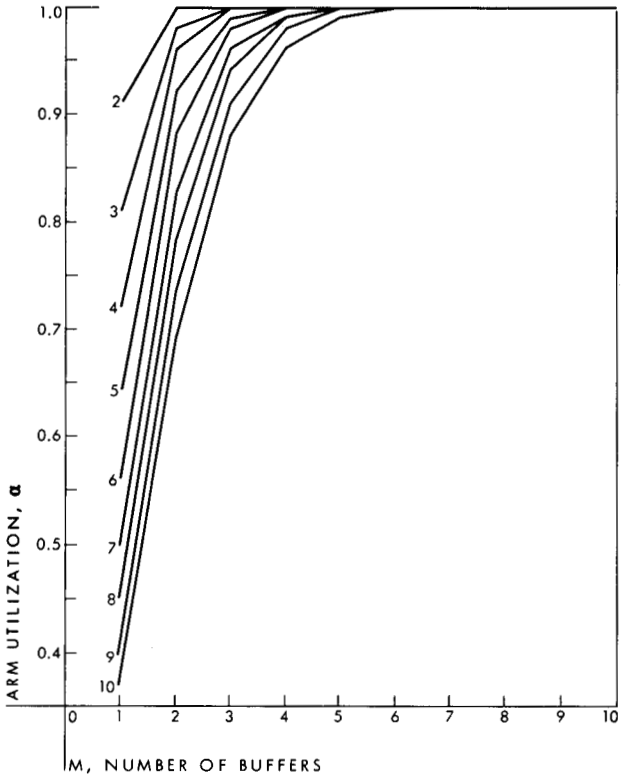


Figure 15 **Utilization of arm plotted as function of number of buffers M for $R = 0.5$, $S = 0.2$ and $W = 1.0$.**

S, the performance increases with increase in the number of arms N. The rate of increase with respect to N of system performance is greater for the system with shorter seek time.

2) Overall, the performance is markedly improved by the system with a shorter seek time. In our particular example a fourfold decrease in T_s produced approximately a threefold increase in performance.

3) For a given N, the rate of change of system performance with respect to M is greatest when M is increased from one to two buffers. The performance generally increases with increase in M; however, quite rapidly a saturation point is reached.

4) For a given number of arms N, the performance can never exceed $(NT_r)/(T_r + T_w + T_s)$.

In Figs. 14 and 15 the utilization of an arm α has been plotted as a function of the number of buffers M for the two sets of the parameters R, W, and S. Again, each numbered curve corresponds to a particular value of N. The most important result that can be gathered from the results for the utilization of an arm, together with the system performance curves, is that probably the minimum optimum system should have two buffers for fast systems, i.e., files with a short access time. For files having a long access time it can be seen that increase in the number of buffers beyond one is not worthwhile. Any balanced system will probably be a compromise among system performance, utilization of an arm, and availability of main memory.

It will be useful to go through an example to illustrate certain points. Suppose that it is desired to link a CPU with a disk file having $R = 0.5$, $S = 0.2$, $W = 1.0$, $N = 4$, and $M = 1$. The objective is to achieve a main memory availability, $\rho = 96\%$ for example.

From Fig. 13 we find $M\eta = 0.72$, and from Eq. (B.11), $\rho = 1 - 0.72\,(T_h/T)$.

The question to be answered now is: given a character rate of 12 Kc/s (kilocharacters/second) and a one-word buffer (six characters, say), what should the main memory cycle be?

$T_h = 3T_{mc}$ (for a one-word buffer),

where T_{mc} is the memory cycle,

$T = 6T_c = 500\ \mu\text{sec}$.

We obtain

$T_{mc} = 9.25 \approx 10\ \mu\text{sec}$.

Obviously, we could have started the design thread from the CPU end and found the disk file characteristics required to match a given memory speed.

The inequality in Eq. (B.10) is satisfied, since $[T_c/T_h] = [83.33/30] = 2 > M\ (= 1)$.

It should be noted that since we have chosen only a single buffer the arm utilization is rather low. In fact, 28% of the time an arm is blocked and not utilized due to the fact that a buffer is not available.

If we choose to employ two buffers instead of the one, and leave all other parameters the same, we shall find that the availability of main memory is reduced only to 94%, with an improvement in system performance ($M\eta = 0.96$) and arm utilization ($\alpha = 0.96$).

Conclusions

The algorithm presented in Part II of this paper should provide the information necessary to system designers in order to optimally link the central processing unit of a computer to information sources of the random-access class, subject to the assumptions and mode of operation of such systems which were stated in the text.

An important result of the investigation which we would like to restate is that in actual practice it will not pay to increase the number of buffers beyond a reasonably small number. Part I and Part II have covered the most important types of information sources.

Appendix B: Quasi-random and random-access information sources.

• *Mathematical formulation and derivation*

We start our discussion by listing and defining the

parameters of the problem. Let

N = the number of arms that can seek simultaneously
M = the number of buffers
t_{sk} = the average time to find a track
T_r = the average time of reading operation
T_t = time for half a revolution.

t_{sk} and T_r are application dependent and will be derived from appropriate statistical data.

It is assumed that the probability distribution of the time duration of any operation (be it seeking, reading, or waiting for the beginning of a track) is exponential. This is the only assumption we make concerning the distribution of time durations.

The probability that the SEEK operation will last a time duration t (or longer) is given by

$$Pr(t_{sk} \geqq t) = e^{-St}, \tag{B.1}$$

where $S = 1/T_{sk}$. Similarly, for the READ and WAIT (for the beginning of the track) operations respectively, we obtain

$$Pr(t_r \geqq t) = e^{-Rt} \tag{B.2}$$

and

$$Pr(t_w \geqq t) = e^{-Wt}, \tag{B.3}$$

where $R = 1/T_r$ and $W = 1/T_t$.

In general, the probability distribution of time duration of any operation O is given by

$$Pr(t_0 \geqq t) = e^{-\lambda t} = P(t)\,. \tag{B.4}$$

The distribution is the exponential distribution with mean $1/\lambda$. The left-hand side of Eq. (B.4) should read: "The probability that an operation O lasts a length of time t or longer."

Let A be defined as the event for which the operation O lasted an additional interval Δt given that the operation has lasted for time t prior to the interval Δt. The last condition may be considered as the event B. We may express the conditional probability as $Pr(A/B) = Pr(AB)/Pr(B)$, where A/B is equivalent to the statement: "event A given event B."

$$Pr(AB) = Pr(t_0 \geqq t + \Delta t) = e^{-\lambda(t+\Delta t)}\,.$$

AB is the event that the operation will last for a time $(t + \Delta t)$ or longer.

$$Pr(B) = e^{-\lambda t}\,.$$

Hence,

$$Pr(A/B) = e^{-\lambda(\Delta t)}$$

The probability that an operation O will terminate during the interval t is

$$1 - e^{-\lambda(\Delta t)} \approx \lambda(\Delta t)$$

for a sufficiently small Δt.

At any instant we may observe i arms seeking, j arms reading (such that $j \leqq M$), and k (where $k = N - i - j$) are waiting for the beginning of the track. It can be seen that any two indices of i, j, and k can be used to characterize the state in which the system is found at any instant.

Let $P_{i,j}(t)$ be defined as the probability that there will be i arms searching and j arms reading (and hence $N - i - j$ arms waiting for the beginning of the track) at time t.

We shall evaluate the probability of finding the system in the state (i, j) at time $(t + \Delta t)$. This may be expressed as the sum of four independent compound probabilities as follows:

I. P_1—the product of the probabilities that
  a) The system is in state (i, j) at time t $[= P_{i,j}(t)]$
  b) There are no completions of seek during Δt $[= 1 - iS(\Delta t)]$
  c) No arm completes reading during Δt $[= 1 - jR(\Delta t)]$
  d) No arm completes waiting for the beginning of a track during the interval Δt $[= 1 - q(N - i - j)W(\Delta t)]$.

$$\begin{aligned} P_1 &= P_{i,j}(t)[1 - iS(\Delta t)][1 - jR(\Delta t)] \\ &\quad \times [1 - q(N - i - j)W(\Delta t)] \\ &= P_{i,j}(t)[1 - iS(\Delta t) - jR(\Delta t) - qkW(\Delta t)] \\ &\quad + O(\Delta t)\,, \end{aligned}$$

where $k = N - i - j$ and $O(\Delta t)$ is a sum of terms involving higher powers of (Δt), and,

$$q = \begin{cases} 1 & \text{for } j < M \\ 0 & \text{for } j = M\,. \end{cases}$$

The reason for the factor q will be explained later.

II. P_2—the product of the probabilities that
  a) The system is in the state $(i + 1, j)$ at time t $[= P_{i+1,j}(t)]$
  b) One arm completes seeking during Δt $[= (i + 1)S(\Delta t)]$
  c) None of the arms complete reading during Δt $[= 1 - jR(\Delta t)]$
  d) None of the arms terminate waiting for the beginning of a track during Δt $[= 1 - (k - 1)W(\Delta t)]$.

$$\begin{aligned} P_2 &= P_{i+1,j}(t)[(i + 1)S(\Delta t)][1 - jR(\Delta t)] \\ &\quad \times [1 - (k - 1)W(\Delta t)] \\ &= P_{i+1,j}(t)(i + 1)S(\Delta t) + O(\Delta t)\,. \end{aligned}$$

III. P_3—the product of the probabilities that
  a) The system is in the state $(i, j - 1)$ at time t $[= P_{i,j-1}(t)]$
  b) None of the arms completed seeking during Δt $[= 1 - iS(\Delta t)]$

c) None of the arms completed reading during Δt [$= 1 - jR(\Delta t)$]
d) One arm terminated waiting for the beginning of a track during Δt [$= (k+1)W(\Delta t)$].

$$P_3 = P_{i,j-1}(t)(k+1)W(\Delta t) + O(\Delta t)\,.$$

IV. P_4—the product of the probabilities that
a) The system is in state $(i-1, j+1)$ at time t [$= P_{i-1,j+1}(t)$]
b) None of the arms completed seeking during Δt [$= 1 - (i-1)S(\Delta t)$]
c) One arm completed reading during Δt [$= (j+1)R(\Delta t)$]
d) None of the arms terminated waiting during Δt [$= 1 - kW(\Delta t)$].

$$P_4 = P_{i-1,j+1}(t)(j+1)R(\Delta t) + O(\Delta t)\,.$$

Hence,

$$\begin{aligned} P_{i,j}(t+\Delta t) &= P_1 + P_2 + P_3 + P_4 \\ &= P_{i,j}(t)[1 - iS(\Delta t) - jR(\Delta t) - qkW(\Delta t)] \\ &\quad + P_{i+1,j}(t)(i+1)S(\Delta t) \\ &\quad + P_{i,j-1}(t)(k+1)W(\Delta t) \\ &\quad + P_{i-1,j+1}(t)(j+1)R(\Delta t) + O(\Delta t)\,, \end{aligned}$$

or,

$$\begin{aligned} [P_{i,j}(t+\Delta t) - P_{i,j}(t)]/\Delta t = &-[iS + jR + qkW]P_{i,j}(t) \\ &+ (i+1)SP_{i+1,j}(t) \\ &+ (k+1)WP_{i,j-1}(t) \\ &+ (j+1)RP_{i-1,j+1}(t) \\ &+ O(\Delta t)\,. \end{aligned}$$

Passing to the limit as $(\Delta t) \to 0$ we obtain:

$$\begin{aligned} d[P_{i,j}(t)]/dt = &-(iS + jR + qkW)P_{i,j}(t) \\ &+ (i+1)SP_{i+1,j}(t) + (k+1)WP_{i,j-1}(t) \\ &+ (j+1)RP_{i-1,j+1}(t)\,. \end{aligned}$$

In the steady state $d[P_{i,j}(t)]/dt = 0$ and we can dispense with the t dependence in our notation. We obtain the following set of simultaneous linear equations:

$$\begin{aligned} &-(iS + jR + qkW)P_{i,j} + (i+1)SP_{i+1,j} \\ &\qquad + (k+1)WP_{i,j-1} \\ &\qquad + (j+1)RP_{i-1,j+1} = 0\,, \end{aligned} \tag{B.5}$$

where $k = N - 1 - j$, $0 \leqq j \leqq M$, and $M \leqq N$. We also have the conditions that:

$$\sum_{i,j} (P_{i,j}) = 1$$

and,

$$q = \begin{cases} 1 & \text{for } j < M \\ 0 & \text{for } j = M\,. \end{cases}$$

In general, we shall have $(M+1)[2(N+1) - M]/2$ probabilities and hence equations to consider. For example, for $N = 3$, $M = 2$ the state probabilities are:

$$\begin{array}{lll} P_{0,0} & P_{0,1} & P_{0,2} \\ P_{1,0} & P_{1,1} & P_{1,2} \\ P_{2,0} & P_{2,1} & \\ P_{3,0} & & \end{array}$$

In the foregoing steps to derive Eqs. (B.5) we have assumed that transitions occur via one and only one event. No two events can occupy the same interval (Δt). When constructing Eqs. (B.5) each term should actually be multiplied by a factor w such that for any $P_{m,n}$

$$w = \begin{cases} 1 & \text{for } m \text{ or } n \geqq 0 \\ 0 & \text{for } m \text{ or } n < 0 \\ 0 & \text{for } m + n > N \\ 1 & \text{for } m + n \leqq N\,. \end{cases}$$

Equations (B.5) have the general form of

$$-\alpha P_{i,j} + \beta P_{i+1,j} + \gamma P_{i,j-1} + \delta P_{i-1,j+1} = 0$$

where α, β, γ, and δ are the rates of transition out of or into the corresponding states. A negative sign denotes a transition out of a given state. Thus, Eqs. (B.5) state the fact that in the steady state the number of transitions per unit time out of state (i, j) equals the sum of the number of transitions per unit time into state (i, j) from the allowable neighboring states.

In Eqs. (B.5) the factor q was introduced to take care of the case $j = M$. In this case it is known that no arm can terminate its waiting for the beginning of the track within (Δt) since all buffers are busy reading. For the contrary to occur two events must occur simultaneously: first, a buffer has to terminate its reading and become free, and second, the arm has to be exactly at the beginning of the track to begin transmitting to the buffer. This is contrary to our assumptions that no two events can occur simultaneously in the same interval (Δt).

We shall derive now some of the entities required to evaluate the performance of the system. First, let us define the utilization of a buffer as the fraction of total time that we may expect a buffer, on the average, to be busy receiving information. This will be denoted by η. Thus,

$$\eta = \left[\sum_{i=0}^{N} \sum_{j=1}^{M} jP_{i,j}\right] \Big/ M\,. \tag{B.6}$$

Similarly, we define the utilization of an arm as the fraction of total time that it will be expected on the average to be utilized. An arm is gainfully utilized when it is either searching or reading; it is not so utilized when it waits for the beginning of the track.

It should be recalled that in the development leading to Eqs. (B.5) there were two kinds of "waiting for the beginning of a track." An arm has to wait for the beginning of the track either because no buffer is available, or as part of the access cycle. These will be referred to as a waiting of the first or second kind, respectively. When calculating the utilization of an arm we should not penalize the system because of the waiting of the second kind. To this end we proceed as below.

Let μ be the fraction of total time that an arm is *not* utilized due to waiting of the first and second kind.

$$\mu = \left[\sum_{i=0}^{N}\sum_{j=0}^{M} kP_{i,j}\right] \Big/ N \,. \tag{B.7}$$

When $M = N$ we can derive a particular solution for μ by inspection:

$$\mu_{M=N} = T_t/(T_{sk} + T_r + T_t) \,.$$

$\mu_{M=N}$ takes in account only waiting of the second kind, since when $M = N$ no other waiting can occur.

In a span of time t ($t \gg T_r$), $tM\eta C$ (C is a proportionality constant) readings (of duration T_r) have taken place, resulting in $t(\mu - \varepsilon)N$ units of time spent in waiting of the second kind. Thus, ε is the fraction of total time that an arm is *not* utilized due to waiting of the first kind.

Applying direct proportion we obtain

$$\mu - \varepsilon = (M\eta\mu_{M=N})/(N\eta_{M=N}) \,.$$

But again by inspection,

$$\eta_{M=N} = T_r/(T_{sk} + T_r + T_t) \,.$$

Hence,

$$\varepsilon = \mu - (M/N)(T_t/T_r)\eta \,.$$

Let the utilization of an arm be denoted by Θ. We get

$$\Theta = 1 - \varepsilon = 1 - \mu + (M/N)(T_t/T_r)\eta \,. \tag{B.8}$$

The performance of the system can be defined as the average total effective rate of information transmission. We shall denote the performance by P measured in characters/second.

$$P = M\eta F \text{ characters/second} \,, \tag{B.9}$$

where F is the rate at which characters are transmitted from arm to buffer. It can be seen that for a given F the performance is directly proportional to $M\eta$.

One of the assumptions we made in deriving the algorithm was that the single input channel shared by the buffers does not cause any delays. In order for this to be realized the following inequality should be satisfied.

$$M \leqq [T_m/T_h] \,, \tag{B.10}$$

where the square brackets indicate "integral part of" and T_m and T_h are as defined in Part I.

The above inequality is rather conservative in estimating the value for M since it takes into account the worst case when all buffers receive simultaneously information for a period T_r. The equality in Eq. (B.10) should provide a value for M with a reasonable margin of safety.

Another criterion of performance is the availability of main memory. This is defined as the fraction of total time that main memory is available to perform tasks other than input-output. The availability of main memory will be denoted by ρ and is given by

$$\rho = 1 - M\eta(T_h/T) \,, \tag{B.11}$$

where T is the time to fill one buffer (cf. Part I).

References and footnotes

1. Roger L. Boyell, *J. Soc. Indust. Appl. Math.* **8**, No. 1, 102, (March 1960).
2. P. E. Boudreau and M. Kac, *IBM Journal* **5**, No. 2, 132-140 (1961).
3. The reader should not confuse the term *input-output link*, with *input (or output) channel* respectively. The term *link* will apply here to the path traversed by the information from or to the device via the buffer to the beginning of the *input (or output) channel*, which is the final and common avenue of access to main memory.
4. The terms *random* and *sequential* are often abused, especially the former. A redefinition, therefore, is in order. By *sequential* (non-random) *access* we shall mean a mechanism such that the retrieval time of any item of information out of a storage medium is a function of the location of the last item retrieved. A *random access memory* is such a memory in which any item of information (picked at random) can be retrieved in the same interval of time, which is usually a constant of such a memory. In the *quasi-random memories*, such as magnetic disk files, there is a range of values of access times,

bounded, to be sure, to the blocks of data; the access is sequential within any selected block. The block of information is a collection of contiguous items of information.

5. The paper presented here is based on three previously published papers. These are listed in Refs. 6, 7, and 10 below.
6. B. Tasini and S. Winograd, IBM Research Report RC-268, IBM Research Publications, Yorktown Heights, N.Y., May 31, 1960. "System Balance, Part 1: Input-Output Links—Theoretical Analysis." (This report is the basis of the present paper).
7. B. Tasini, IBM Research Report RC-370, IBM Research Publications, Yorktown Heights, N.Y., December, 1960. "System Balance, Part 2: Input-Output Links—Design Procedure." (A program for carrying out computations on an IBM 704 computer included).
8. In Ref. 7 above we derive the necessary corrections to be applied when the CPU is required (for some reason or other) to interfere in the operation of the tape units.
9. The method does not allow intermixing of two types of information sources having different nominal rates of transmission, in the same system——say, paper tape readers and magnetic tape units. This is not a very severe restriction, since it is possible to calculate the effectiveness of the system using an average character rate, taking in account proportions of each type of information source to be used in the system. Alternatively, the effectiveness of the system can be derived using first one rate and then the other, thus defining a bounded region.
10. B. Tasini and S. Winograd, IBM Research Report RC-475, IBM Research Publications, Yorktown Heights, N.Y., May 9, 1961. "System Balance, Part 3: Input-Output Links—Analysis of the Performance of Magnetic Disk Files." (This report is essentially the same as Part II of the present paper.)
11. Philip M. Morse, *Queues, Inventories and Maintenance*, John Wiley & Sons, Inc., 1958.
12. W. Feller, *An Introduction to Probability Theory and its Applications*, John Wiley and Sons, Inc., 1950.

Received March 22, 1961