The Multi-Sequence Computer as a Communications Tool

J. N. ACKLEY†

THIS is a report on the merging of two fields: communication switching and computers. Recent advances in the computer art make it possible to satisfy the ever increasing communication switching requirements brought on, in part, by computers themselves employed in centralized data processing systems. Present record communication systems have significant delays which are not primarily caused by the transmission times but by the time required for the operations in the communication message switching centers.

In the past, most communication has been from human to human. Communication systems have become more complex and automated to meet the ever increasing needs of commercial and military activities. We are faced with a revolution. Increasingly, communication will be between humans and machines and between machines and machines. This transition will place more stringent requirements on accuracy, reliability, and speed.

Present day electromechanical switching centers are limited in their speed of operation due to their electromechanical nature and due to limitations of the transmission means, namely, teletype. Little error detection and correction capability is presently found in these systems.

Centralized data processing requires the error free transmission of large volumes of data to a central point. Usually, the communication with machines or between machines involves little redundancy such as that found in plain English. The computer, although it can make validity checks on the data it receives, cannot fill in missing letters or words as a human can. Since the present electromechanical systems are special purpose devices, all messages routed through the system must adhere to a very rigid format. This leads to difficulty when trying to integrate data gathering devices and different types of computers with different codes and formats. Military command control systems, especially, require data inputs from varied sources.

Stored program techniques could solve many of these communications problems. A programmed switching center could provide the error checking and error correction procedures as required. It could be programmed to translate from one code to another, indeed, perhaps from one language to another. Various speeds and code structures could easily be accommodated.

† International Electric Corporation, Paramus, New Jersey.

In addition to improving the features presently found in some switching centers, stored program techniques could be used to implement features which are not practicable with electromechanical or special purpose switching systems. A programmed switching center could generate, receive and interpret service messages to and from other machines or human operators. It could test and monitor all of the communication links and reroute messages if necessary to avoid inoperative links. One of the biggest advantages of a programmed switching center is its ability to be reprogrammed to account for changes in operational procedures, routes and equipment.

One of the most important considerations in employing computer techniques to the communications switching problem is the large number of input and output channels required. Also, all channels must operate simultaneously and independently. Military practice requires that each message be forwarded as far as possible whenever the communication link is available. Thus, the communication switching center must accept a message on each communication line whenever the subscriber wishes to transmit.

For many years the bottleneck on efficient use of computers and on the application of computers to real-time systems has been the problem of integrating input-output devices. Most input-output schemes have involved a large amount of equipment external to the central computer to provide for buffering and control. Now that the versatility of the high speed random access core memory has been fully appreciated, an almost limitless number of schemes is possible. Indeed, a single computer may use several schemes to integrate various input-output devices.

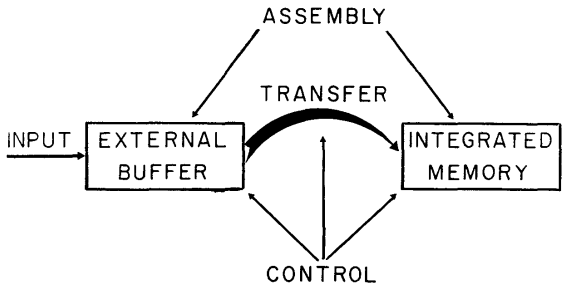

Fig. 1—Input-output classification.

In order to discuss the various schemes and compare their advantages and disadvantages, it is necessary to establish a classification system. There are

four parameters, as shown in Fig. 1, which characterize an input-output scheme:

Assembly refers to the process of packaging or unpackaging information into definite size units. In order to characterize the assembly (or the disassembly) process, one must specify how and where the transformation takes place among bits, characters, words, records, and files.

Buffer refers to the unit external to the central computer which holds information until it can be transferred. The buffer may also be involved in the assembly process. It is characterized by its capacity, access time, and assembly features.

Transfer refers to the process of exchanging information with the integrated (addressable) memory of the central computer. The transfer is characterized by the number of bits transferred in parallel.

Control refers to the process which determines the sequence of operations of an input-output channel. In order to carry out the control function, control words must be supplied to a control device. Typical words that are usually involved in input-output are:

Selection code
Number of units of information to be transferred
Address in integrated memory at which the transfer is to begin
Address in external device at which the transfer is to begin
Location of next control word

Some or all of the control words are required for any input-output transfer. The device which uses these control words can be the central computer or a separate input-output control device which provides for the proper sequence of operations.

In designing a system, engineering compromises must be made. Many combinations and permutations of the above factors can be made. Each permutation will have certain advantages and disadvantages which must be weighed against the system application. For example, to take the extremes, a system can be designed like the IBM 709 system which has an external control device (the Data Synchronizer). This device has a register for buffering and additional registers for storing the control words. This system requires only a core memory cycle for a transfer and thus takes little time away from the central computer during input-output transfers. However, this scheme requires extensive hardware for the external control device.

On the other hand, a system may be designed to store all of the control words in integrated memory and the central computer could supply the control. Such a system would require a minimum of equipment external to the central computer for control of the input-output transfers. However, in a system application which requires a large volume of input-output, a large percentage of the capacity of the machine would be tied up in the input-output transfers.

A typical communications switching center may have 50 to 100 two-way communication lines. In order to have a computer perform the functions of a store and forward switching center, it must have corresponding numbers of input and output channels. In addition, it must have a sufficient processing capacity to determine distribution and optimum routing, priority, and to perform validity and parity checks if required. Not only must it provide for these large numbers of input and output channels, but all channels must be capable of operating simultaneously. Neither of the extreme systems described above is satisfactory for this type of operation. However, both of these schemes can be made practical for this type of application by multiplexing as shown in Table I.

TABLE I

IN-OUT SYSTEMS FOR COMMUNICATION SWITCHING

Assembly	Buffer	Transfer	Control
1. Bits to character external	Character	Character break-in	External-multiplexed interrupt or multisequence for special cases.
2. Bits to character external	Character	Character programmed	Central computer-multisequence

For example, the external control device can be multiplexed so that it may service a number of input and output channels simultaneously. This also requires that the central computer have a number of independent memory banks so that the transfer of the control words from storage to the external control device and back to integrated memory on each input or output transfer will not saturate the system.

By multiplexing the central computer, it can be used to handle a large number of communication lines. A central computer can be made to time share over a large number of channels by utilizing the multisequence configuration.[1] This configuration provides a number of program counters and a system for switching from one program counter to another, in response to external stimuli.

In order to select the proper scheme, a detailed study must be made of the particular communication switching center requirements. The external control scheme gives a higher capacity under certain conditions, but it requires more equipment. The study of

[1] W. A. Clark, "The Lincoln TX-2 Computer Development"; J. W. Forgie, "The Lincoln TX-2 Input-Output System"; *Proceedings of the W.J.C.C., 1957.*

the communication requirements must determine the ratio of the number of core memory cycles required for internal processing to the number of core memory cycles required for input and output transfers. If this ratio is sufficiently high, then the central computer control scheme is to be preferred, since relieving the central computer of the burden of input-output transfers would free only a relatively small percentage of the system capacity. On the other hand, if this ratio is low, then the external control scheme is to be preferred since a low ratio implies that the main function is that of input-out transfers.

In a system which requires utmost accuracy achieved by the use of redundancy checks, and which must automatically route and reroute messages to account for communications outages and supply other functions such as code translation; this ratio may be in the order of 2 or 3 to 1. Thus, only a 33 to 50 percent increase of capacity is the maximum that could be expected by utilizing an external control device and multiple access integrated memory — neglecting memory reference conflicts.

Another significant factor which affects total capacity of the system is the assembly process and the capacity of the external buffer. Since most communication is based upon characters of five, six, seven or eight bits, it is desirable to handle the assembly from bit to character externally.

Systems have been proposed utilizing only a single bit external buffer, with the assembly from bit to character to word to message in integrated memory. However, the capacity is severely reduced since now a transfer must take place on each and every bit of a message. Increasing the size of the external buffer on the other hand increases the capacity, but at the same time increases the amount of equipment. The maximum buffer size to be considered is that of the word length. Therefore a compromise must be reached between the amount of equipment in the external buffers and the capacity of the system. This compromise is influenced by the number of lines to be terminated, by the speed of the circuitry available to the programmed element, and by the nature of the transmission system.

Those who have been through a computer development program will immediately ask, "Is it necessary to design a special computer for the communication switching system?" or, more generally phrased, "Can a general purpose computer be utilized?" This question is not easily answered. The answer must be based on a thorough study of the communication system application. Surely, if the only job of the programmed element is that of communication switching, then there is no need for floating point or even multiply or divide instructions in the instruction vocabulary. Thus, simplifications can be made in the design of a programmed element which is to be used solely for communication. The specialized programmed element can be optimized for communications.

On the other hand, if the computer is to be used for both communications and computations, it may be desirable to design an external control device which can be adapted to a general purpose computer. However, the range of applicability of the general purpose computer seems rather limited. If the computational requirements are very great, it may be profitable to employ the special purpose programmed element for the communications switching center and also to act as an input-output processor for a much larger data processor. This philosophy is illustrated in the STRETCH and LARC computers where a simple input-output processor is used to relieve the complicated, high speed data processor of the simple routine tasks associated with input and output editing. The in-out processor can also be used to operate or schedule the operation of the larger data processing system to achieve a much higher utilization of the data processor than would be possible with a human operator.

In selecting a computer or designing a computer for communication system switching center application, consideration must be given to the peak and average traffic rates. Generally, the utilization factor of communication lines is approximately 0.1 to 0.2. This surprisingly low figure is justified because of the queueing problems which would ensue with a very high utilization of any communication link. Assuming a Poisson distribution of message arrivals, a utilization factor approaching unity would result in a queue approaching infinity. Therefore, to minimize the queueing problems and to assure rapid transmission of the message, the utilization factor is desirably kept approximately at 0.2. The message processing capacity of the switching center must also be designed to exceed the average traffic load in order to eliminate excessive queues of messages awaiting processing.

The use of a multisequence computer for a communication message switching center permits the termination of a number of lines which would saturate the computer with input transfers if all lines were operating at full capacity all of the time. The internal and output processes can be assigned a lower priority in the multi-sequence scheme and these operations suspended until the input peak passes. The probability of such an occurence is extremely small. Since it also is extremely rare that many of the communication lines would be simultaneously busy with input traffic, additional lines may be terminated with the probability that the switching center would lose an input character. In a completely automated system, this would be caught by the error detection and correction system and cause only a slight delay in transmission. The amount of excess message processing capacity required to reduce the processing queue to satisfactory proportions depends upon the delays permitted and upon the priority structure of the messages.

A programmed Traffic Control Center is currently being fabricated at ITT Laboratories. This Traffic Control Center, which utilizes the multisequence technique, was undertaken as a study project in 1957 at the Laboratories and was then proposed by this author as the solution to the communication message switching problem of the SAC Control System, Project 465L, in July 1958. The system design resulted in a computer with several distinguishing design characteristics which make it peculiarly efficient in handling communications. The central computer is multiplexed by use of the multisequence technique. Separate memory units are provided to store 256 program counters and 256 index registers. One index register is associated with each sequence; thus, in essence, 256 separate sequences may time-share the central computer. Each communication line is terminated by a simple character buffer and each buffer has associated with it a service request flip-flop. As data becomes available, the service request flip-flop is set and competes for time on the central processing unit. Each instruction has provision for a break-bit or a dismiss-bit, as shown in Fig. 2, which when set indicates that the present sequence may be interrupted in favor of a higher priority sequence or that the present sequence may be dismissed, in which case the service request flip-flop is cleared and the remaining sequence with the highest priority is activated.

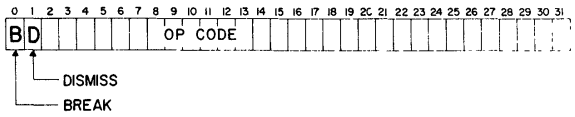

Fig. 2—Instruction word.

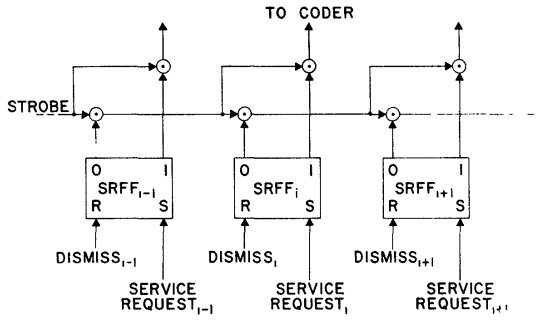

Fig. 3—Service request scanner.

To take advantage of statistical averaging of the inputs and outputs, an on-demand scanner (shown in Fig. 3) does not operate on a fixed cycle, but service is requested immediately if no other channels are busy at the same time. In order to incorporate devices of different speeds, and to provide an orderly procedure for servicing requests if more than one should arrive at a time, the service request scanner operates on a priority basis. A strobe signal proceeds from one service request flip-flop to the next until the first one which is set is encountered, at which point the strobe signal is routed to the coder.

Fig. 4 shows a typical input channel. When the strobe signal arrives at the coder it is converted into binary code. If the instruction that the central computer is executing contains a break bit, the output of the coder will be compared with the number in the sequence register. If they are not equal, a sequence of higher priority number has requested service. Notice that the number in the sequence register operates a switch which selects the particular external device to be used at any given time and routes control signals to the external device. The dismiss signal is issued whenever the present sequence has completed all operations necessary to answer the service request. This signal clears the service request flip-flop and prepares it for receiving the next request for service from the external device.

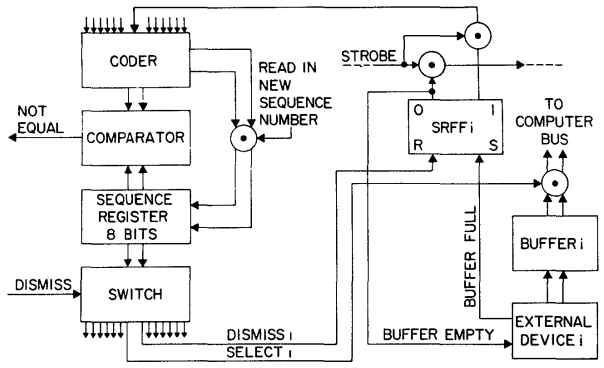

Fig. 4—Typical input channel.

Fig. 5 shows the registers involved in changing sequences. If the number from the coder is found not equal to the number in the sequence register and the current instruction has a break or dismiss bit, a change of sequence is called for. Since the control must return eventually to the old sequence whenever that particular channel requests service again, the program counter must be stored for future reference.

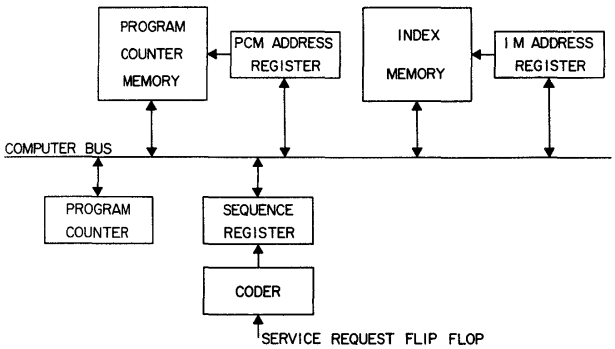

Fig. 5—Registers involved in changing sequences.

Therefore, the first operation is a transfer of the sequence number from the sequence register to the program counter memory address register. The program counter is then stored at the location so speci-

fied. After the old program counter has been stored, the new sequence number is read from the coder into the sequence register and then to the program counter memory address register. The stored program counter stored at this location is now placed in the program counter and the computer continues to take instructions specified by the program counter.

In order that each sequence may have its own index register, the contents of the sequence register are transferred into the index memory address register and the contents of the location so specified are used for the indexing operation. Since the program counter memory and the index memory are independent units and can operate concurrently with the main integrated memory of the computer, changes of sequence can take place without any loss of time, that is, an instruction with a break or dismiss in one sequence may be immediately followed by an instruction in some other sequence.

Note that if the contents of the accumulator were stored along with the contents of the program counter, we would truly have a multiplexed computer. However, this feature was not necessary for the communication switching center, but care must be taken during programming in placing break and dismiss bits only at those points in a sequence where the contents of the arithmetic unit are immaterial.

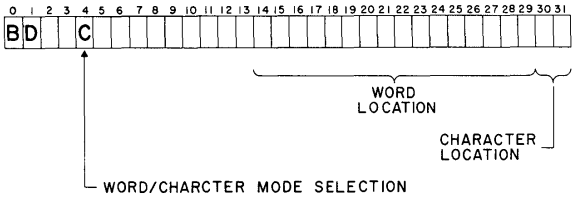

Fig. 6—Instruction word.

Since most communication and data processing systems have five, six, seven or eight bit characters, most instructions are capable of operating in two modes. In the word mode, the operation applies to the entire computer word of 32 bits. In a character mode a single 8-bit character is referenced. Since there are four eight-bit characters in a 32-bit word, an addressing scheme as shown in Fig. 6 was devised which permits convenient addressing of consecutive characters in memory. The two least significant bits of the address part of an instruction refer to a character position within a word. In the word mode, these bits are ignored. The instruction vocabulary contains a liberal quantity of logical instructions, but does not contain multiply or divide or any other numerical operations other than add and subtract.

Most instructions have a bit to indicate the repeat mode, as shown in Fig. 7. This permits most input-output transfers to be accomplished with a single instruction requiring two core memory cycles. As the characters are transferred in from the external buffer,

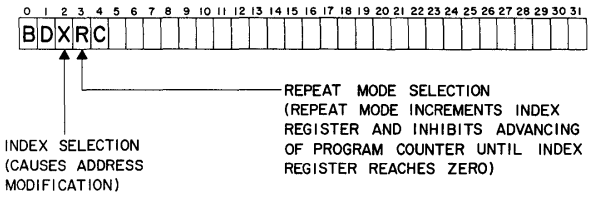

Fig. 7—Instruction word.

a check on parity and a check for a control character are performed by common circuitry. If either condition prevails, it is indicated to the program by means of a skip. Otherwise the input-output transfers are handled by a single instruction in the repeat mode with the dismiss-bit set. This causes the character to be transferred, the index register to be incremented, and the present sequence dismissed in favor of some other sequence. The message processing sequence has the lowest priority and processes messages between transfers.

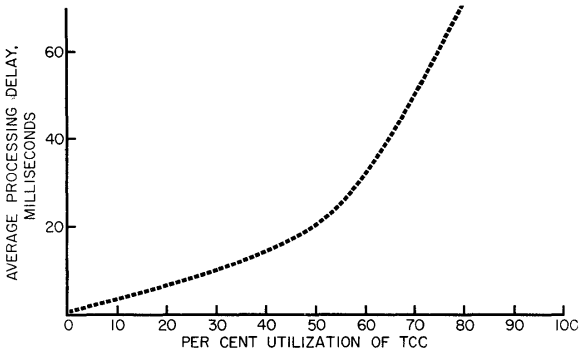

Fig. 8—Comparison of per cent utilization (of TCC) *vs.* average processing delay.

Fig. 8 shows the comparison of the percentage of utilization of the Traffic Control Center versus the average processing delays. This truly represents an advance in the communication message switching art. Present switching systems delay each message by many minutes. Now the delay is measured in milliseconds.

Most comprehensive communication systems require both circuit and message switching. The programmed switching center can effect digital circuit switching by associating, by program means, a particular input channel with a particular output channel. The delay under these conditions would be only a few microseconds and would be insignificant compared to the delays in long transmission lines. However, since the traffic is just coming in and going out without any processing, the capacity of the programmed element is used unnecessarily. In a communications system which requires a large amount of circuit switching, it will be preferable to terminate all communication lines in an electronic cross-bar switch, as shown in Fig. 9, which has many trunks to the programmed element and is under its control.

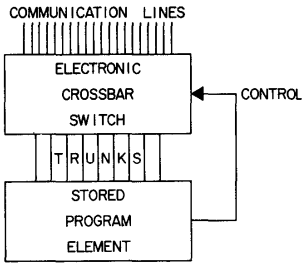

Fig. 9—The ultimate communications switching center.

This configuration would permit both message and circuit switching on an intermixed basis and permit changing from one type of service to another by simple indications to the stored program.

Discussion

W. G. Stevens (*IBM*): What is the maximum character rate on a single channel?

Mr. Ackley: The maximum character rate on a single channel is determined basically by the number of core cycles per second. In the present ITT design, which has an 8-microsecond memory cycle, the maximum speed is 62,500 characters per second. This is available over to the totality of inputs.

Mr. Stevens: With all channels operating and the computer transferring characters from input to output channels, but performing no editing, what is the maximum character rate per channel?

Mr. Ackley: The maximum average input rate is fifteen 250-character messages per second. That runs it up to 3,750 characters per second.

G. W. Bleisch (*Bell Tel. Labs.*): What provisions are made for multiple address messages?

Mr. Ackley: I hate to say it but this is programming detail. This has frustrated quite a few of the communicators using this system. The entire message processing operation is determined by stored program.

K. Enslein (*Brooks Research*): Have you given any thought to what an electronic crossbar would be composed of?

Mr. Ackley: That is outside my field of interest although there are several developments under way. I understand that Stromberg Carlson is working on one and ITTL has been also working on an electronic switching system known as Digicom.

D. A. Bourne (*IBM*): What error reduction scheme is used?

Mr. Ackley: In our system we have both parity per character and also parity characters at the end of each message providing a cross check. If an error is detected by these parity checks, a request for automatic retransmission is given by the Traffic Control Center which causes the transmitter to retransmit the message. As an additional feature, again this is a program detail, every message must be acknowledged. If a transmitter sends a message and it is not acknowledged in three seconds it automatically retransmits the message.

R. G. Turner (*Philco*): How is message priority established? How do you determine relative priority of input signals?

Mr. Ackley: The relative priority of input channels is not related to the priority of the messages. The priority of the input channels is to permit the integration of different speed devices and provide orderly procedure for handling a multiplicity of requests. As to the priority of the message, the message format has in it a provision for indicating the priority, and again the priority processing is a program detail. First thing after it receives a message, the program checks the priority, and establishes proper program connections to process it on a priority basis.

E. B. Cohen (*Auerbach Electronics*): How much message storage capacity does the system have?

Mr. Ackley: This is determined by the system applications. This ITTL system has 16,000 words of core, 130,000 words of drum and approximately 12 tapes for bulk storage. Usually messages come in and go out so fast there is little working storage required in the core memory and no additional storage is required for messages except for the journal which is on magnetic tape.

D. Dittberner (*IBM*): You mentioned a study program in 1957. Did this give rise to a piece of hardware and who sponsored the study?

Mr. Ackley: This was a study sponsored with ITT General Development Funds and it was just that, a study, not a development.

H. P. Peterson (*Lincoln Lab.*): Can any sequence get at the program counters and index registers of other sequences?

Mr. Ackley: Yes, all of the program counter core memory and the index core memory is addressable.

Mr. Peterson: What is the number of instructions in your machine?

Mr. Ackley: Instructions list runs, I think, 40 some instructions.