Design of Univac®- LARC System: I

J. P. ECKERT,† J. C. CHU,† A. B. TONIK,† AND W. F. SCHMITT†

THIS TALK is a progress report and in many respects a final report on the Univac® — LARC system which has been developed by Remington Rand Univac. It is a companion not only to another talk on LARC design being given at this time but with an earlier talk given three years ago to the EJCC in New York.[1]

The talk three years ago described the objectives of a system still in the design stage. This talk describes progress since then. In order to prepare for this talk we have, of course, reviewed the one delivered three years ago. What we read was quite interesting. We found that in one sense of the word we have no progress to report. Fortunately in the larger sense of the word we have great progress to report. The area in which no progress, or more precisely no change, has been made deals with the design objectives of the LARC system. All of the goals set up for the LARC system three years ago have been very safely achieved. None of the speeds of any of the operations in the LARC system have been changed. Basic decimal-checked addition takes exactly one microsecond as described three years ago. Memory cycle times of both one microsecond and four microseconds have been adhered to.

By a conservative design approach and by choosing existing components, we expected LARC to be completed in 1958. Although we did use then existing components, modifications required to increase reliability on some of the components, along with the logical complexity of the system, delayed completion until 1959. If we had compromised our speed or reliability objectives, the delay would have been much shorter.

Now we come to the point where great and for computer engineering unusual progress can be reported. We have precisely met and proven the validity of our original goals on the world's fastest, most versatile computing and data processing system as described in our earlier published talk. Fig. 1 shows a typical LARC system as it would appear in an operating installation.

The original objective of the LARC system was to build a good system which pressed the limits of the art while still maintaining balance between its various elements. We feel a good system is one in which everything involved is accomplished in a manner which is not only dependable but consistent in the degree to which it presses the art at the time the design is frozen. The talk of three years ago established the goals. We have now accomplished them in the area of logic circuits, high-speed memories, input-output equipment, and most important of all in overall system organization. Further, we have established an automatic procedure for the prodigious record keeping required to carry out such a design. The companion paper will discuss this point further.

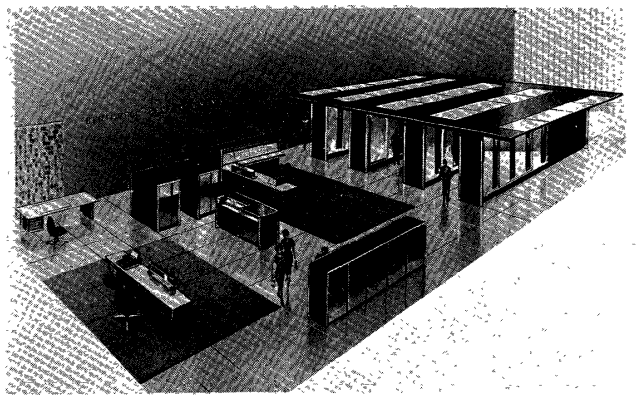

Fig. 1—A typical LARC system as it would appear in an installation.

By a balanced system, we mean a system designed to obtain the greatest work output for the costs involved. Since the problems faced by different users vary, considerable flexibility of the input-output equipment and the memory equipment of the computer system is required. In addition to providing more flexibility in these areas than any other existing system, the LARC system has the added flexibility of enabling either one or two computing units to be included as part of the system to allow increased speed.

Computers and Processor

A basic LARC system contains a Computer and a Processor, each of which has most of the attributes of a general purpose Computer but perform different functions in the system. The primary function of the Processor is the flexible, parallel, and coordinated control of all input-output equipment and transfers between this equipment and the memory system. The Computer is designed to perform rapid arithmetic computation with a minimum of interference.

The two Computers in an expanded system (see Fig. 2) can be programmed and controlled to solve jointly a single problem; or each can solve inde-

† Remington-Rand Univac Div., Sperry-Rand Corp., Phila., Pa.

[1] J. P. Eckert, "Univac-Larc, The Next Step in Computer Design," *Proc., 1956 EJCC,* (A I E E Special Publication T-107) pp. 16–19.

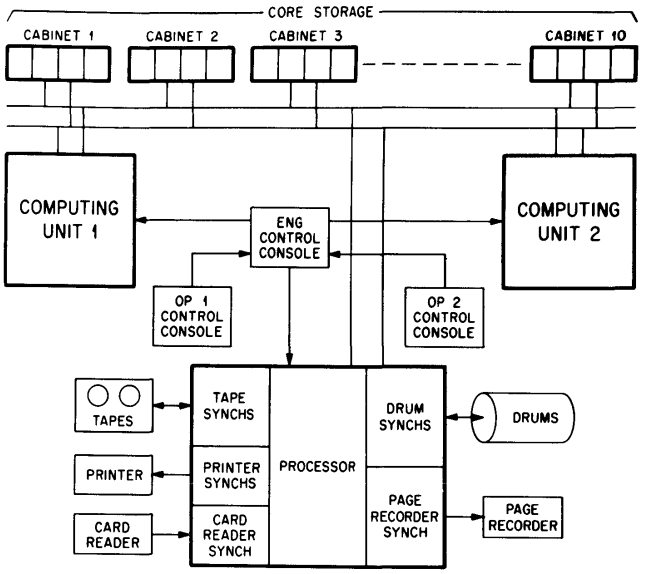

Fig. 2—Block diagram of a LARC system.

pendent problems. The Processor is designed to take care of the input-output and other on-line equipment needs of both Computers and to do any necessary editing of the input-output data. If input-output demands are not excessive, it may also be used to run various side routines such as sorting and merging. It receives compact instructions from the Computer and expands them by program to provide the control, interlocking, and timing required to operate simultaneously a large group of on-line equipment. This function requires only limited arithmetic ability.

To allow for simultaneous communication between the various units (a necessity with the high degree of parallel operation achieved) with a minimum of switching circuitry, a time-slot system of inter-unit communication is used. Time-slotting of the elements of a computer system is quite new and is one of the features which has given LARC such a great increase in speed and flexibility without a corresponding increase in cost.

In LARC we sought to minimize the total amount of electronic equipment by considering the logical design as a whole, rather than seeking to optimize the equipment required for each individual instruction. For example, there are many methods of performing multiplication which reduce the number of individual addition operations required without expanding equipment requirements. We examined a number of these multiplication methods and picked the arrangement which required the minimum amount of equipment, not just for multiplication but for all of the machine instructions, frequently making compromises on some specific instruction. A modified short-cut multiplication procedure was finally employed that requires only one addition per multiplying digit and allows 11-by-11 digit products to be formed in 8 microseconds. Floating-point multiplication of 9-by-9 digits also takes 8 microseconds, the extra time being used for normalizing. The division process is of fixed length, requiring 5 cycles or $2\frac{1}{2}$ microseconds per quotient digit. It requires 28 microseconds for a floating-point division or 32 microseconds for a fixed-point division.

The pulse code in LARC was carefully chosen to yield a ninety percent chance of a single digit superposition error being detected. Of course, since most such errors occur on several digits, the chance of not detecting such an error is remote. However, where a single digit must be transferred special checking circuits are employed to reduce this possibility; or some special circumstance is noted which makes the chance of detection far greater.

The names Processor and Computer assigned to the central units of the LARC are somewhat misleading. The entire LARC system may be considered a data processor. What we call the Processor in the LARC system contains a computer of its own in addition to all of the circuits necessary to synchronize the on-line equipment into the system. In designing the LARC system, we carefully avoided incorporating any feature that would increase costs unless it realized a much greater proportional increase in over-all speed and performance. On the basis of this criterion alone, the incorporation of a computer within the Processor is justified, although it does provide other important benefits by way of increasing flexibility and simplifying programming and communication within the system. If the LARC Computer were designed to perform the duties of the Processor, it would require additional instructions and other facilities. In such a system, all of the circuits devoted to synchronizing the on-line equipment would still be required. In a typical LARC system, about two-thirds of the Processor consist of circuits exclusively involved in synchronizing on-line equipment. The Processor computer represents the remaining one-third of the Processor or about one-sixth of the equipment and cost of the Computer and Processor combined. Considering also the cost of the on-line equipment used in the system, the incorporation of a computer in the Processor has contributed between one-seventh and one-eighth to the total cost and complexity of the LARC system. The LARC Processor, however, may relieve the Computer of half of the work load it would otherwise have to bear. We thus have achieved as much as eight percent increase in speed for every one percent increase in cost resulting from incorporating the computer in the processor.

Since the programs for the Computer and Processor are stored in separate memories, debugging can be done independently, so that the programming is

usually made easier. Thus we have speeded up LARC without overstepping the state of the art either in circuits or in the demands put upon present day programming technology.

Storage

There are four levels of storage in the LARC system which differ in speed, capacity, and cost per character.

The first level of storage operates on a one-microsecond cycle. It consists of a number of registers that may be used interchangeably as accumulator registers for storing operands and results or as index (B) registers for storing constants used in addressing operations. Up to 99 12-digit one-microsecond registers may be included in each Computer unit.

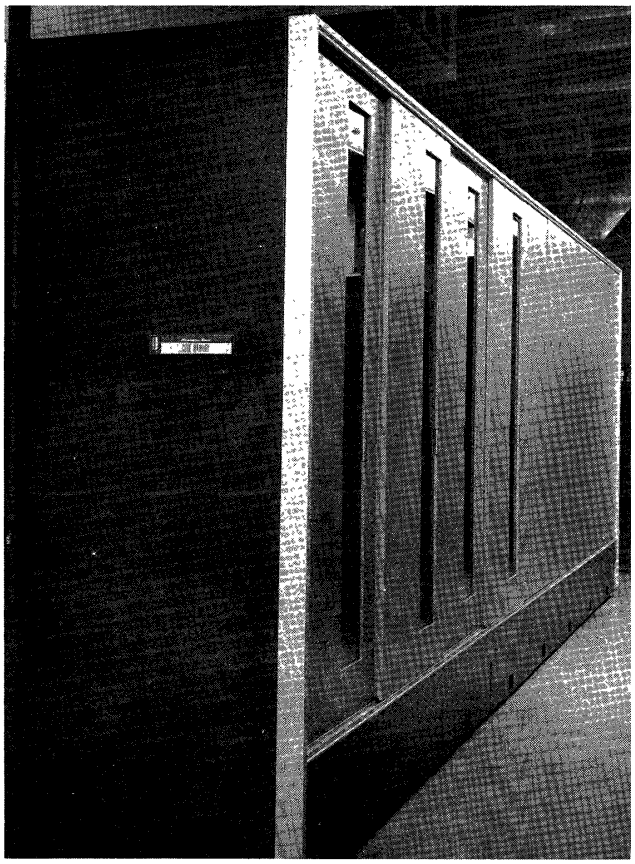

Fig. 3—A 10,000 word (600,000 bit) core memory unit.

The second level of storage is a ferrite-core storage which has a read-write cycle of four microseconds. Fig. 3 shows a 10,000-word (600,000 bit) core memory cabinet. It is accessible to both Computers and the Processor. It serves as both the main storage and buffer storage of the system and as a common communication link between the Computers and the Processor. To increase the rate at which reference may be made to the main storage, it is divided into 2500-word modules. Since the Computers and the Processor have access to the same storage, they can control each others' instruction sequences as desired. Nearly 100,000 12-digit words of ferrite-core storage may be included in the system (6,000,000 bits of core storage).

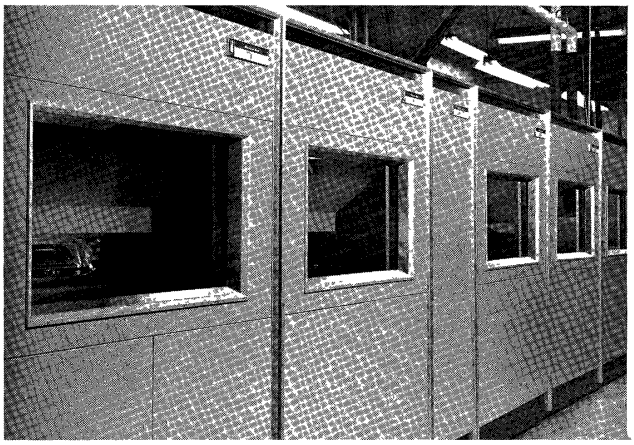

Fig. 4.

The third level of storage consists of movable-head magnetic drums. Fig. 4 shows a group of such units. Data can be transferred between the main storage and the drums over one of several simultaneous drum circuits at a data transfer rate of 500,000 decimal digits per second (2,500,000 bits per second). A maximum of 72,000,000 digits of drum storage (360,000,000 bits of drum storage) may be included in a system. The drum units are capable of transferring information twenty times as fast as the tape units, and match the high computing rates of the LARC system. In addition they provide for random access in a fraction of a second. Several drum synchronizers are employed so that computation and printing can proceed in parallel with the loading and unloading of drums to the tape units.

The moving-head drums themselves are designed very conservatively. They rotate at 884 RPM. They are recorded at 448 pulses to an inch and 29 channels to the inch. Several times this pulse rate and channel density have been achieved in the laboratory on these drums. The magnetic heads which move on a carriage along the top of the drums are moved with moderate accelerations. This is possible since the drums are used in pairs, first moving one head while another head is reading and then alternating so that the synchronizer is never idle. This arrangement allows a very conservative approach to the mechanical problems and a fool-proof head-moving mechanism of simple and rugged design driven by a servo motor. The drum assembly is mounted in an air-tight enclosure. Inside air is circulated through special filters to collect any dirt particles. The design allows the units to be opened

for repair in a non-air-conditioned room. The magnetic heads fly on the hydrodynamically generated air film generated by the rotating surface of the drum. Fig. 5 shows a head flying on the surface of a drum. No filtered high pressure air supply which might direct contamination under the head is needed. A simple mechanism lowers the head on the drum without causing even momentary contact with the drum. An electrical circuit detects any contact between head and drum and instantly retracts the head if any contact occurs.

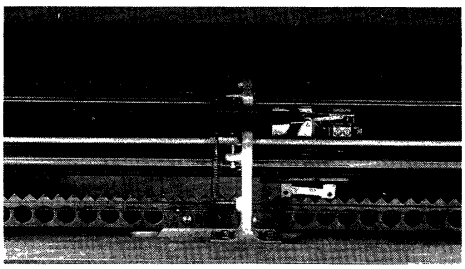

Fig. 5—A head flying on the surface of a drum.

Flying-head drum units have been life-tested for well over a year's continuous operation to ensure that their reliability is consistent with the reliability of the solid-state circuit equipment. The order of reliability achieved by these drums is believed to exceed considerably that obtainable on tape or disc equipment at this time.

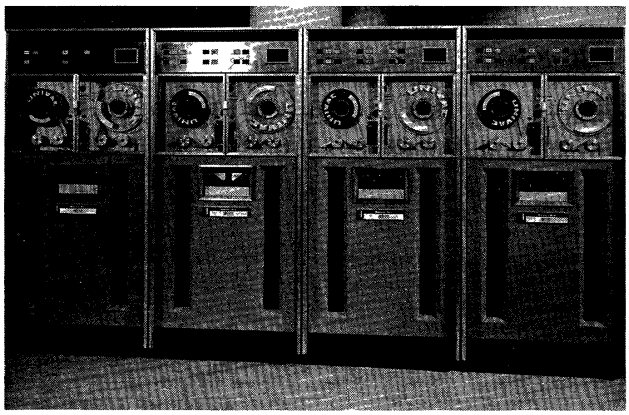

Fig. 6 A group of UNIVAC tape units.

The fourth level of storage consists of magnetic tape units which transfer data at 25,000 alphanumeric characters per second. Fig. 6 shows a group of Univac tape units. The tape units provide for input-output and long-term storage of data. LARC presently uses medium-speed tape units which not only have the advantage of being compatible with other Univac systems and off-line input-output equipment but have the advantage, due to moderate pulse densities and moderate tape velocities, of being quite reliable. Faster tape units as they become available may be used with the LARC. Tape speed is not usually a limitation, however, since the bulk of the input-output load has been taken off of the tape units in the present LARC system by the drum units.

On-Line Equipment

The on-line equipment includes, in addition to the tapes and drums used for input-output and intermediate storage:

1. An electronic page recorder, which employs a cathode-ray tube and microfilm, as shown in Fig. 7. It provides high-speed recording of output data. Sixty-four symbols are available for tabulating and curve plotting. Plots are complete with titles, scales, and grid patterns. While originally specified at 25,000 characters per second, this has been changed to 20,000 characters per second to allow more accuracy in the curve-plotting mode. Fig. 8 is a chart of the units for a typical and for an expanded LARC system.
2. Electro-mechanical line printers for multiple-copy printing of numerical characters at 720 lines per minute or printing of combined alphabetic and numeric characters at 600 lines per minute. The synchronizers will provide signals for printers operating up to 1200 lines per minute.
3. A card reader for introducing data into the system directly from punched cards.

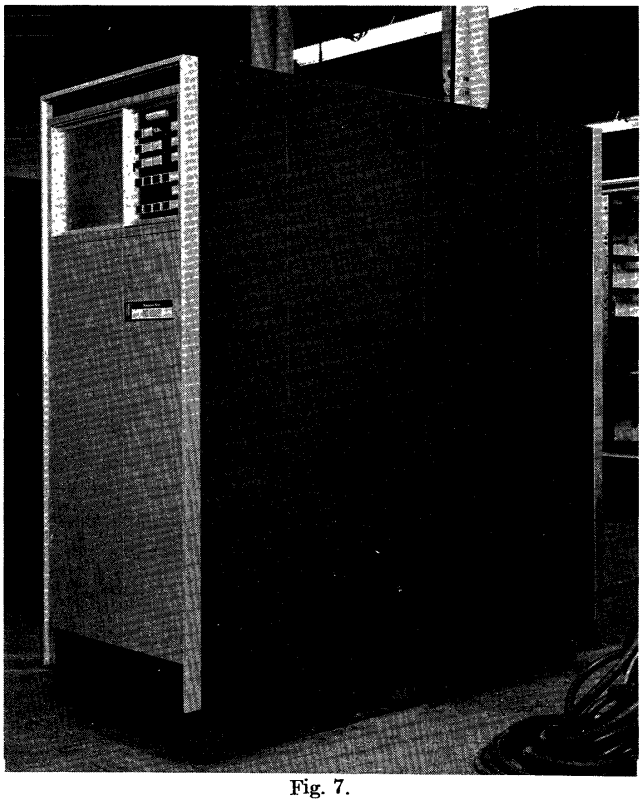

Fig. 7.

EQUIPMENT NAME	TYPICAL	EXPANDED
Magnetic Core Storage Units (2500 words each)	8	39
Computers	1	2
Multipurpose Fast Registers (per Computer)	26	99
Processor	1	1
Drum-read Synchronizers	2	3
Drum-write Synchronizers	1	2
Tape Read-Write Synchronizers	2	4
Electronic Page Recorder Synchronizer	0	1
High-Speed Printer Synchronizer	1	2
Card Reader Synchronizer	0	1
Console Printer Synchronizer	1	1
Tape Positioning Checker	1	1
Magnetic Drum Storage Units (250,000 words each)	6	24
Uniservo II Magnetic Tape Units	12	40
Electronic Page Recorders	0	2
High-Speed Printers	1	2
High-Speed Card Readers	0	1
Control Consoles	1	2
Numeric Keyboards (one per Console)	1	2
Alphanumeric Console Printers (one per Console)	1	2

Fig. 8—Modular units of a typical and completely expanded UNIVAC LARC system.

4. Console typewriter printers with an attached paper-tape reader and punch.

The original design called for a Processor to contain a minimum of five "synchronizers" and a maximum of eight synchronizers as required for simultaneous operation of the various pieces of on-line equipment. The capabilities of the LARC system turned out to be such that in order to provide more flexibility for customer requirements the minimum was raised to seven and the maximum to fourteen. The synchronizers in themselves have flexibility in the pulse rate they can accept and this, along with the concept of the Processor and the optional number of synchronizers that may be used, is one of the more important features in obtaining the high speed and flexibility of the LARC system.

LARC Consoles

Supervisory control of the LARC system is achieved through the use of the LARC Operator's Console and the LARC Engineer's Control Console. Fig. 9 shows an operating console to the left and an an engineering console to the right. The LARC Operator's Console includes a decimal display unit, an automatic typewriter, and an input keyboard. This console is designed for the utmost simplicity of operation and includes only the manual-intervention and start and stop buttons. Certain signals from the on-line apparatus are also displayed where operator attention is required.

The LARC Engineer's Control Console includes a replica of the Operator's Console and in addition, all necessary equipment for the monitoring and control of all of the LARC units. A special feature of

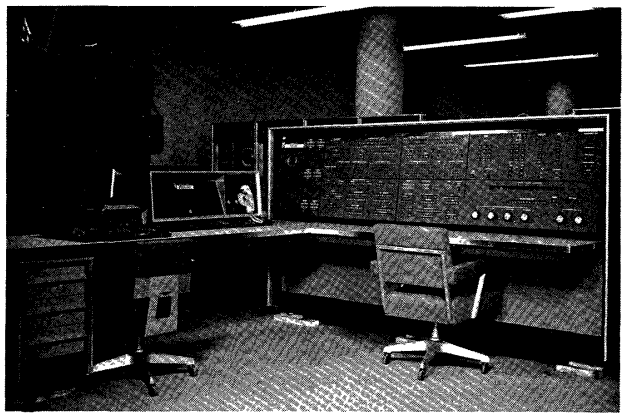

Fig. 9—An operating console to the left and an engineering console to the right.

the consoles is the ability to monitor the operations of the computing unit in a unique way. The digital display units and corresponding binary display units may be connected to any one of a number of key points of the computing unit. Synchronizing equipment allows these registers to display at a particular pulse time and program step selected by the engineer. Various modes of operation, including an error-display mode for trapping both permanent and intermittent faults, greatly reduce trouble shooting time. The console also includes all voltage-monitoring and alarm indicators for the various units, power control for the system, marginal-checking controls, and all necessary error and contingency indicators.

Complete Parallel Operation

The designers of LARC realized that even with automatic programming some programs must first be written by hand; and further, that programming must not be so difficult to understand that the people maintaining these machines cannot interpret the situation when trying to locate a fault. For these reasons, all of the parallel time-saving features built into LARC were done in a way that would require the minimum of planning on the part of the programmer and would be as similar as possible to his present programming techniques. Therefore, in our system design we have been consistent with the straightforward electronic and mechnical designs employed. Table I shows a list of the equipment in simultaneous operation in a maximum LARC system.

TABLE I

Complete Parallel Operation in LARC

(A list of equipment in simultaneous operation in an expanded LARC system)

1. *Input-output equipment*
   Read from 3 drum units
   Write on 2 drum units

Position drum head assemblies
Read from 2 tape units
Write on 2 tape units
Position a tape unit or check read a tape unit
Rewind all tape units as required
Print on 2 line printers
Record on electronic page recorder
Print on console printer
Read fast card reader
Operate decimal display unit

2. *Computers*
   Compute on data by 2 computing units
   Edit or compute on input-output data and control on-line equipment by input-output Processor
3. *Operation of instructions in computing unit*
   Different parts of five consecutive instructions are performed at once.
4. *Decimal arithmetic*
   60-bit parallel self-checking arithmetic circuits
5. *Automatic checking*
   During operation of instruction examine for possibility of tracing, programmer making a mistake, and computer making a mistake.

Without exception, we have obtained overall system speed without resorting to difficult programming tricks such as might require a knowledge of the detailed timing of the machine and the setting up of complicated interlaced patterns of data and instructions. There are many examples of the point we are trying to make here. The memory organization is one of the best examples. At the time the LARC was conceived we decided to have a very fast core memory (one-microsecond cycle time) in combination with a much larger somewhat slower core memory (four-microsecond cycle time).

Our first thinking on how to use these two memories was to take all information out of the slower memory and put it in the faster memory — instructions, operands, index numbers, etc. Computing would then be carried out almost entirely from this fast memory. When we were finished, a group of operation results from the fast memory would be returned to the slower memory. This idea led to all kinds of difficulties. First, information must be taken out of and put back into the smaller fast memory both rapidly and frequently in most of the problems studied. The large memory, being four times slower, would have to be separated into several parts so that by a time-interlacing system it could keep pace with the small memory. In turn the smaller, fast memory would also have to be broken into at least two or three parts so that parts could be loaded and unloaded while another part was in use. In addition, the fast memory could no longer be a hundred words or less and still be effective. It would usually have to be at least one thousand words. Even then, this complicated process would not work on many problems where the nature of the information coming in is such that one does not know for a few hundred words ahead where his next instructions and data are coming from. Since LARC is a very fast new machine and will be used in problem areas which have not been well investigated, we did not feel that the introduction of such restrictions on the programming would be practical. Less is gained by this arrangement than one might expect.

Instruction routines are frequently long enough between transfer instructions that they might just as well be taken from the slower memory, interlaced in the same way as would be required to transfer to the fast memory. Some small loss of time using only the slower memory can occur when a transfer order is encountered due to interruption of the interlace pattern. A similar argument will often hold for strings of data. Thus the faster memory, except for accumulator and index registers, does not help the speed situation much but would be a considerable program complication. In LARC, however, to facilitate matters still further rather than purely time-slotting all of the memories together to get a high speed flow of information, we have done this in such a way that the interrelationship is automatic and does not have to be programmed. Up to eight one-half-microsecond subdivisions of the four-microsecond memory cycle are provided for time-slot operation. The programmer does not have to plan how to intermix information in order to achieve the time-slotting necessary to match memory speeds to the Computer. Because of the time-slot system, the four-microsecond memory system looks almost like a one-half-microsecond memory in an overall system sense.

In this system it is possible for the Computer to ask for information which is not yet available. This is because the Computer receives orders ahead of their execution time in order that circuits have time to set up without delaying the operation. Since LARC is to achieve its speed without the programmer's being required to give attention to timing problems occasioned by parallel operation, special circuits accommodate the situation when inconsistent logical demands occur. These circuits very rarely delay computation in actual operation.

When two computing units are used in a LARC system, the common memory system allows either separate operation of the units or, alternately, any degree of interplay desired. A common memory system and certain common instructions provide the means of interrelating operation. To handle the priority problem involved in the parallel operation of many different units, the memory units give pref-

erence first to the input-output synchronizers, then the Processor, and finally the computing unit.

One kind of parallelism not commonly thought of as parallelism occurs with LARC's one-microsecond decimal self-checking adder. Clearly, we could convert input data into binary form and checking could be done by program. These extra operations take additional time as well as additional programming effort. In LARC's decimal-checking adder these operations are effectively combined in parallel with the arithmetic operation. A binary unchecked adder would have to be considerably faster to equal the speed of LARC's decimal-checking adder. The adder is normally used as part of a sequence in which an instruction is obtained and corrected as necessary by an index register, operands are obtained, exponents are sensed, numbers are shifted as necessary, the sum is obtained, and finally the sum and new exponents are put in an arithmetic register. All of these operations take place in a four-microsecond interval.

LARC does not expect the programmer to take care of such timing and interlocking problems. Parallelism has been obtained in a system that does not reduce flexibility or require advanced programming technology.

We feel that LARC is a very important step forward in system design in that all of its units are in balance both in regard to speed, reliability, and cost for the present state of the art. Sufficient flexibility has been designed so that as new auxiliary equipment is developed, it can be effectively added to the system. Univac I was a better-balanced system than any other computer of its era. This statement has never been challenged. The LARC system is similarly well balanced.

Discussion

See "Design of Univac-LARC System: II."