Computers of the Future

REX RICE†

Introduction

THIS PAPER considers the advances required in many related technologies to revolutionize the construction and use of digital data processing systems. In the following discussion we are particularly concerned with the radical change in fabrication technology and wish to analyze the effect that this change will have on our methods of computer design and specification.

Present Methods

The manufacturing techniques used in the electronic portion of today's digital data processing systems are illustrated in Fig. 1. The active devices are standardized in these systems. Circuit standardization is established at what may be defined as the Boolean function level. Circuits for AND, OR, Invert, Latch, Trigger, etc., are standardized individually. The pluggable packaging usually combines several circuits, either of the same type or in selected groups. A major system function such as a complete working storage register and all its controls, an arithmetic processing unit and its controls, etc., is obtained by assembling a group of circuit packages on a panel and interconnecting the circuit packages with individual wires. At the time the individual circuits and packages are designed and optimized, very little information is available regarding their specific employment in systems functions.

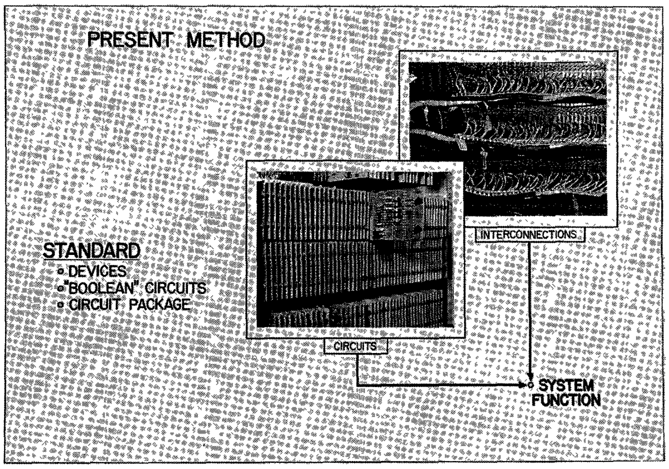


Fig. 1—Present method.

A digital "system function" may be defined as a combination of logical elements interconnected and timed to perform major operational sequences in a data processor. One of our future objectives is to create major digital system functions in one continuous, automated manufacturing sequence.

Future Methods

A possible future method for producing major system functions such as complete working storage registers, process units, memory arrays, etc., is illustrated in Fig. 2. We envision this manufacturing line as a set of printing presses through which a conveyor system passes. Substrate material is placed on the conveyor and proceeds through the line. At each stage one pattern of interconnections, insulation, or active material is printed on the substrate. As required, bake ovens, etc., may be strategically placed. Here, devices are standard by virtue of the materials used. These materials are applied by a standardized method to produce active elements, interconnections, insulation, etc., in batches. The plates, inserted in each press, are made in an automatic machine which develops the appropriate layout under equation control for major system segments.

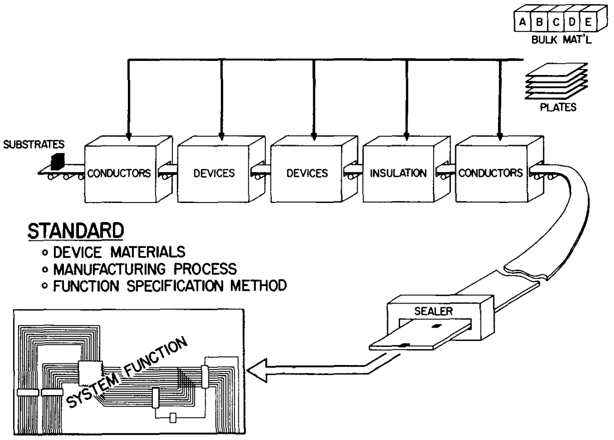


Fig. 2—A future method.

The figure illustrating future methods is only diagrammatic. The manufacturing method chosen will probably depend on the basic component technology and may be different for each type of component. Before complete automation is realized it will be necessary to manufacture active elements separately and to rely on automatic testing and insertion. The field will be dynamic and the illustration indicates a trend, not a specific technique.

---

†IBM Research Laboratory, Poughkeepsie, N. Y.

Illustrative Example of a System Function

A serial-by-digit, decimal adder is used to illustrate a system function as shown in Fig. 3. This represents a portion of an arithmetic processing unit. The digital code assumed is a decimal "one out of ten" representation, chosen because decimal matrix addition is well understood. Other examples or codes would have served equally well.

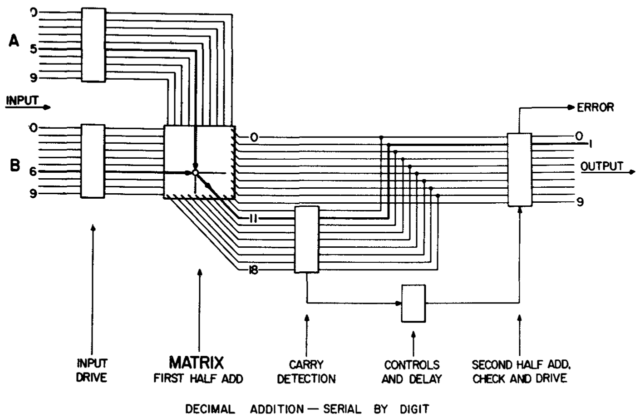


Fig. 3—Illustrative example of a system function.

In this function a pair of decimal digits enters a process unit at A and B and the added result is obtained at the output. A matrix, to be described in detail, performs the first half-addition. Other elements provide input drive, output carry detection, recombination, and the second half addition. It is also necessary to store the presence or absence of a carry, so that as succeeding pairs of digits are processed, the second half-addition circuit may be activated. Let it be assumed by way of example that A equals 5 and B equals 6, as emphasized with heavy marked lines. In the matrix the 5 on the vertical axis together with a 6 on the horizontal axis activates an AND circuit which places an output on the eleventh diagonal. After passing through the carry detection element, the eleventh diagonal is recombined with the output line 1. The carry condition is remembered for later use. Let us now consider circuits for the matrix in more detail.

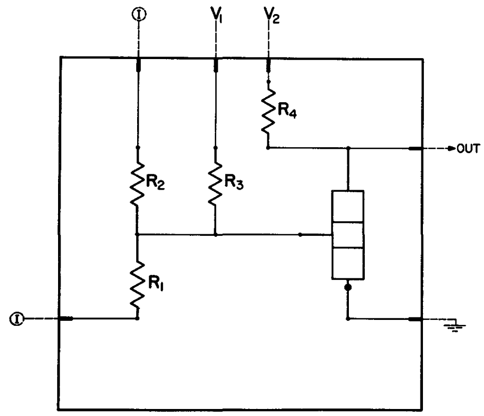


Fig. 4—Standard "and-inverter" circuit (TRL).

Matrix Utilizing Individual, Standardized Boolean Circuits

The circuit in Fig. 4 is a Boolean standardized two-way AND circuit with one transistor, four resistors, and various internal interconnections. Several outputs may be wired together to form an appropriate OR circuit. A two-way circuit is chosen, since for our purposes in the addition matrix a three- or four-way AND circuit has no advantage.

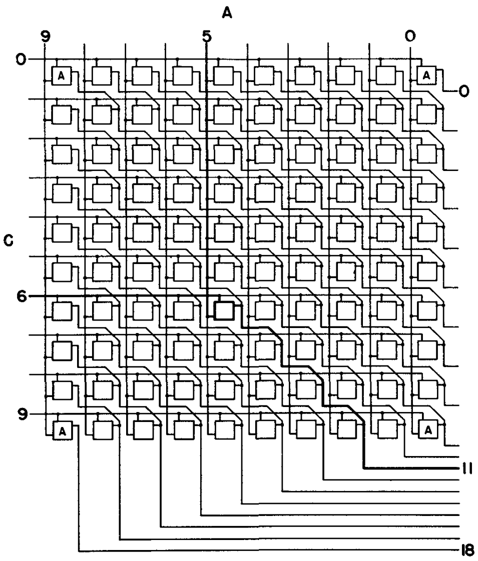


Fig. 5—Matrix utilizing standardized "Boolean" circuits.

A ten by ten matrix of these AND circuits is illustrated in Fig. 5. For clarity, the internal circuit connections and devices have been omitted. In the matrix, addition is accomplished by the coincidence of current on any pair of lines such as $A = 5$ and $B = 6$. When the AND circuit at this intersection is active, its output is placed on the eleventh diagonal. For packaging purposes the designer has the choice of packaging several AND circuits on a single pluggable unit. When the circuits were optimized, only the two-way AND logic together with the output loading conditions were known.

Let us now reexamine this same matrix from a system rather than a circuit viewpoint (Fig. 6). In this specific matrix element only one AND circuit in the $A = 5$ column and the $B = 6$ row is "on." This is a system consideration and was not known at the time the Boolean AND circuit was optimized. The vertical column $A = 5$ will now be considered as a single element.

System-Tailored Circuits

A circuit which is tailored to this system function

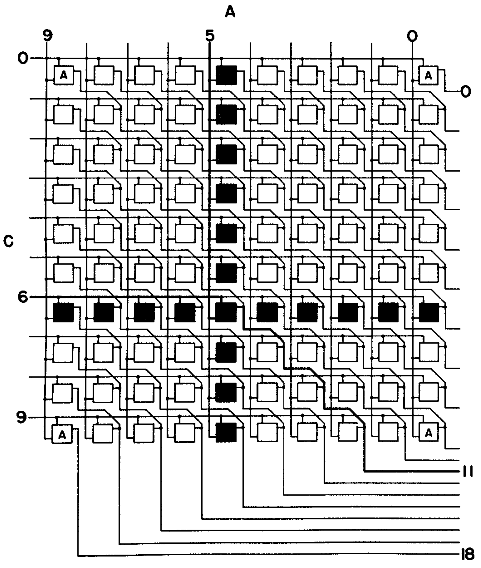


Fig. 6—Matrix utilizing standardized "Boolean" circuits.

is illustrated in Fig. 7. For convenience, transistors have been shown, although other devices such as relays, tubes, cryogenic devices, etc., could have been used. The input *A* supplies current to a common control which goes to all the bases of the ten transistors. Since only one line on the *B* input to the emitters is active at any instant, only one transistor will be conducting. Let us now examine the addition matrix utilizing this "system tailored" circuit.

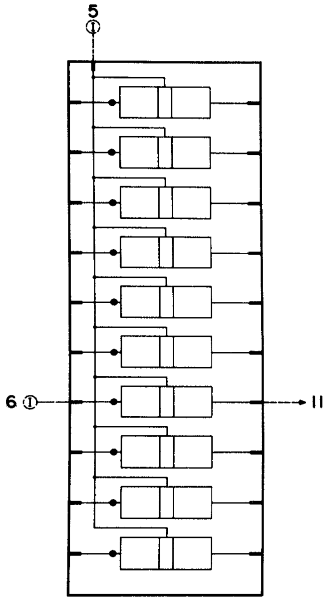


Fig. 7—System tailored circuit (CS).

Matrix Utilizing System-Function Circuits

The complete matrix is again shown in Fig. 8, this time utilizing ten of the system-function circuits. The "*A*" entries on the vertical axis go directly to the common control connections of the ten AND circuits. The "*B*" entries are connected to the emitters of the ten transistors in each of the ten circuits. The collectors are connected to the output lines, which are functionally equivalent to diagonals in the previous matrix. Note the identical configuration of the wiring to the inputs of all ten matrix columns. The outputs of each "system AND" circuit are connected in a pattern which drops down to the next output line for each successive group. Thus, to add 5 to the number entering *B* the sixth AND circuit is activated. The number 6 on the *B* entry is moved down five units on the output, giving a sum of 11. Although the number of transistors required in both matrix examples remains the same, the passive elements are eliminated and the packaging pattern for both interconnections and devices is drastically improved.

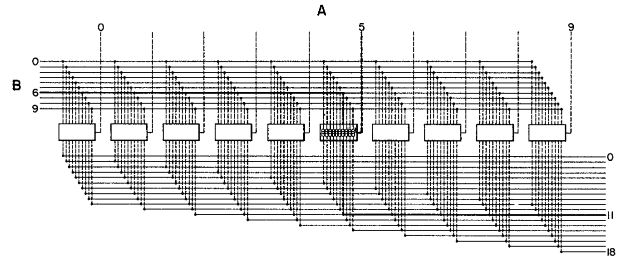


Fig. 8—Matrix utilizing "system function" circuits.

In the illustration the solid lines represent a layer of interconnections on the front of a printed substrate and the broken lines, a second layer on the rear. Connections through the substrate are indicated by dots. Inasmuch as ten system-function circuits are used, ten component packages consisting of active elements only may be mounted on a single substrate that contains the complete interconnection wiring.

A computer may be described as "a bunch of wires connected by active elements." This second method of matrix design underscores that definition. Three important features become apparent in this example. First, careful attention to system-function circuits will lead to logical layouts that are much easier to express algebraically for equation-controlled manufacturing. Second, the amount of packaging and interconnections, and the number of elements involved can be reduced over present methods. Third, new system-function device specifications will emerge.

System-Tailored Devices

The previous discussion presented an example in which circuits and system-function logic were combined using standard transistors. Present active devices are individual elements packaged separately, as shown in Fig. 9. The connections between the active and passive elements are generally made by individual wires, although more recent systems use printed wiring for circuit packages.

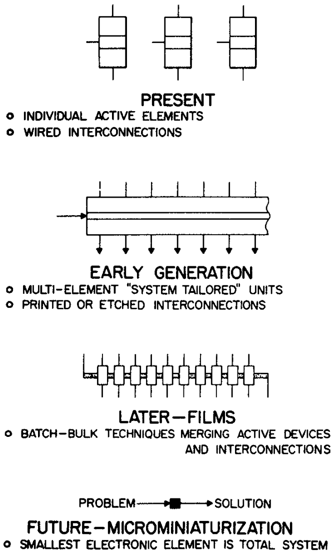


Fig. 9—Devices "system tailored."

In an early generation, multi-element system-tailored devices will be available. In addition, a much greater proportion of the interconnections will be etched and printed. Multi-element miniaturized components have been made available in small quantities by American Bosch Arma, the Diamond Ordnance Fuze Laboratory, Hughes Aircraft, RCA, Texas Instruments, and others. Programs in molecular electronics to permit the use of plating and vacuum-deposition processes are also receiving attention. Much of this work is for military applications but will probably be available for commercial use in the near future.

The production of interconnections and active elements in one continuous manufacturing process will occur with the introduction of films, either thick or thin, into systems. At this time, semiautomatic methods of manufacture will be mandatory. Here it is obvious that separate considerations of system functions, circuits, and devices may no longer exist. Magnetic coupling is used to accomplish switching in thin film cryogenic systems and speeds are very high. One suspects that nature also provides a medium speed and cost arrangement if we are clever enough to detect it.

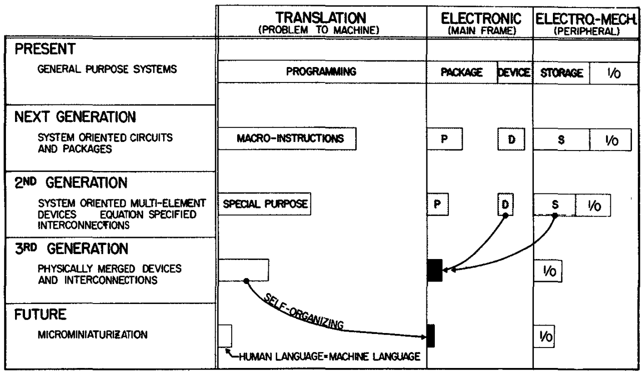
Further in the future we may anticipate true microminiaturized systems constructed from automatic, computer-controlled processes utilizing bulk materials. The late Professor Dudley Buck has defined a microminiature computer as: "A computer on a scale which could never be looked at in an optical microscope." In this technology, the cost of active elements will approximate the cost of interconnections. Logical designers may enjoy the luxury of utilizing thousands of active elements to perform logical functions of a complex nature.

One of our major objectives is to reach the future system illustrated here. Let us now consider some of the more important work to be done to make this possible.

DIGITAL DATA PROCESSING

APPROXIMATE RELATIVE COSTS

The bar graph (Fig. 10, Line 1) shows the approximate relative costs of processing data in presently available commercial general-purpose digital systems. Problem preparation and programming costs are generally accepted as being approximately half of the total. The remaining costs may be divided into two major items: electronic main-frame costs and electromechanical peripheral-equipment costs. The percentages vary from system to system, but are essentially as follows: The cost of main-frame electronics varies between 15 and 25 percent of the total, and includes the main random access storage, the arithmetic and logic unit, and controls. In the main-frame, the switching devices cost approximately one-third and the packaging (which includes circuit cards, panels, interconnections, frames, display, covers, etc.), approximately two thirds. The cost of the electromechanical portion of a system may vary between 25 and 35 percent of the total and may be divided into two parts. The first is bulk storage involving mechanical motion. This part includes tapes, discs, drums, etc., and their attendant electronic equipment. The second part is the input-output equipment, including communication devices.



Present Generation

General-purpose systems predominate at the present time. This is probably due to the relatively high cost of research and development coupled with long design and manufacturing lead-times for initial production. Instructions usually include an operation, one or two addresses, and a few special control bits. The instruction code at the machine language level is relatively "micro" due to the general-purpose requirement and for other reasons not covered here.

System specification normally starts with a market analysis so that a potential product may be defined. Performance, storage volume, input-output equipment, etc., are established at this time. Available standard circuits and packages are considered during the specification of system logic. Outputs from the system design are block diagrams, or equations, or both. At this stage we do not know where each device or circuit will be placed, nor the length of interconnections.

In programming, present generation machines use autocoders to translate from problem language into machine language. The autocoders, in many instances, involve execution time and occupy storage space. This combination of autocoders and machine language is the result of the programmer's desire to have a machine language different from the one technology is able to economically provide.

Devices used in present systems, both active and passive, are individually manufactured by semiautomated methods. This allows individual testing, selection, and replacement in the event of malfunction.

The circuits are Boolean optimized and the minor packaging assemblages usually include several elementary functions. Recent trends as evidenced in machines like the Philco TRANSAC, are toward the inclusion of more Boolean-type circuits on each pluggable element. Interconnections are a mixture of printed cards and hand inserted wires and cables.

The major mechanical design of a system starts when logical specification and Boolean standardized circuits are available. With this information, the active and inactive elements may be located and packaged. For the first time, lead lengths become accurately known. The output from mechanical design is generally a complete set of blueprints which go to the manufacturing engineering groups.

In the peripheral equipment area the bulk storage usually involves magnetics and includes much mechanical equipment. Access to data in this type of storage is either serial-by-bit or serial-by-character. The input-output equipment is essentially mechanical, taking data from a keyboard to a buffer storage and, later, taking data from a buffer to a printer to produce hard copy.

Servicing is usually done by a combination of electrical tests and diagnostic programs. It involves locating the defective active or passive elements and substituting new pluggable cards.

The specification and design of present systems is thus essentially a serial process in which most major elements are individually standardized and then assembled to make a system. The design feedback loops, while many, have rather high impedance.

Next Generation

The next generation as illustrated by the bar in Fig. 10, Line 2, may be *characterized mainly by system-oriented design and manufacturing techniques.* Commercial machines will probably remain general-purpose in nature.

The bars illustrating approximate relative cost on this and succeeding generations does not necessarily indicate that the cost of an equivalent advanced machine will be reduced. The length of the bars represents the relative proportionate cost for each of the major elements in a system for a particular generation. Past experience has shown that as more powerful techniques become available we solve larger problems; therefore, we have an option of obtaining more computing for our millions or reduced costs for the same amount of processing. This is obviously a designer's choice and will be adjusted to suit requirements as he specifies a particular system.

A major change will occur in the specification of systems. Logic and circuits will be merged to produce new system function circuits utilizing standard devices. The physical location of components, the interconnection lengths and paths, and layout of the package will be specified as an integral part of logic. To attain these objectives a new "system-function algebra" is necessary. This algebra, which will begin with the logical Boolean expressions, must be enriched to include the active and passive device characteristics, the physical location of all components, the interconnection paths and lengths, and timing.

Programming in this generation will be done with more powerful macro-type instructions. Machine language instructions will approximate the level typified by coding systems such as FORTRAN. Relatively speaking, more hardware will be in the instruction controls with the objective of making programming easy and fast.

Improved single-function devices and some use of multifunction devices may be anticipated.

A major change in packaging as well as in logic-circuit specification will occur in this generation. Complete system functions will be packaged on one replaceable element. Interconnections will be etched, printed, evaporated, or batch produced by other automated techniques. Manufacturing equipment, methods, and mechanical design techniques must undergo the appropriate changes.

Service will be accomplished by locating and replacing malfunctioning major system functions. If the individual devices are expensive, they may be replaced at a testing and service center so that the system function may be returned to stock. If not, the whole unit may be discarded. Extensive built-in checking and automatic program diagnosis will be included. The logic of the machine will require more redundancy for checking and diagnostic purposes.

The next generation of systems thus involves major improvements in logical design and packaging. New devices or other research items are not necessarily required.

Second Generation

Two major changes characterize the second generation systems. (Fig. 10, Line 3). *First, system-tailored multi-element devices* will be used extensively. This will influence mechanical design, packaging, and manufacturing equipment. *Secondly, special-purpose machine systems to solve classes of problems* will be made on the same manufacturing line. The logical specification of these machines will be generated by computers utilizing system-function algebra. Extensions of the algebra will control the manufacturing setup. This combination will drastically reduce design and production lead times and cost of the product.

The availability of special-purpose systems will ease programming difficulties through the use of application-tailored languages to solve related classes of problems.

System-function design techniques and devices will be applied to bulk storage. For input-output, electronics will replace mechanical equipment wherever possible.

No on-line service will be performed since the machine will be able to select alternate logical paths in the event of a malfunction. At inspection periods, previously-flagged defective system elements will be removed and replaced.

Third Generation

The true revolution begins in the third generation. (Fig. 10, Line 4). Here, *device, package, and interconnections are inseparably merged.* Major system functions will be produced from bulk materials in computer-controlled continuous manufacturing processes. Techniques such as vacuum deposition, electron-beam writing, spraying, printing, etc., will be utilized, depending on device technology chosen relative to the speed and cost range desired. The use of three-dimensional connections will alter packaging concepts. Miniaturization for complete systems may now be realized. This miniaturization will allow dramatic increases in the number of active elements available for both logic and storage.

The availability of vast amounts of homogeneous storage with internal logical capabilities will drastically alter programming methods. In particular, built-in symbolic addressing will eliminate the inefficient and tedious housekeeping associated with present-day machines. Coupled with special-purpose instruction sets, this will allow machine language to approximate problem language.

The input-output equipment will now be reduced to that which is used to communicate with humans or from machine to machine, since bulk storage is now merged with the main frame.

Service will be simple because automatic error detection and correction by the machine will allow continuous operation. Defective elements will be replaced at the next service period.

Future Generation

We may envision a few aspects of future generations now (Fig. 10, Line 5). True *microminiaturization* meeting Professor Buck's definition will be realized. Self-organizing systems will become possible due to microminaturization and better understanding of the logic involved. The use of self-organizing systems to find optimum solutions to problems will allow us to synthesize more economical, special-purpose systems for on-line use.

For programming, we may anticipate that machine language will approximate or equal human language if we have progressed properly to this point and if we use self-organizing systems appropriately. A major change in input-output techniques is required. Voice and pattern recognition, and vastly improved display and printing systems are needed.

In this generation, service will be accomplished by throwing the whole computer away.

In summary, to progress from the present day data processing capabilities to more desirable future systems, we require greatly increased logical capabilities, vast amounts of storage, improved input-output methods and more speed. All these elements tend to require microminiaturization, batch-bulk processing, automated logical synthesis, and equation-controlled manufacturing. Consequently, both speed and system cost require and benefit from this revolution.

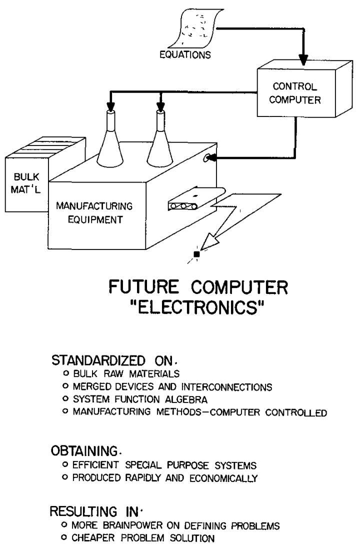


Fig. 11—Future computer "electronics."

CONCLUSION

Future computers (Fig. 11) will be standardized as follows:

1. Interconnections and active devices will be made in a continuous process from bulk raw materials to finished product.
2. The device, circuit, and interconnection technology will merge.

3. System-function algebra will be used to specify all aspects of design.
4. Completely automated, computer-controlled manufacturing methods will be used.

From these techniques we will obtain efficient special-purpose digital data processing systems. They will be produced economically with short design and construction lead-times through complete automation. This will result in more brain power being devoted to discovering and defining new problems, and in their cheap, efficient solution.

DISCUSSION

J. H. Felker: (*AT&T*) I would like to hear you complete the job of prophecy, Mr. Rice, and give us some idea of the timetable you envision for these first, second, third (and) fourth generation machines.

Mr. Rice: I have given that question considerable thought. We are working on the next generation right now in many research laboratories. The universities are probably ahead in some respects in their thinking on research in this area. It is not necessarily true that each generation requires the specific items at the same time as shown in the paper. If we develop microminiature computer devices ahead of new programming techniques, they may be utilized early. I suspect, and this is a personal observation, that the first models of microminiature computers are ten years off and the other items for the next generation are scattered from three to five years away in production. This is a guess on my part.

Mr. Felker: Thank you. With the three year period it takes to design and get production of the conventional computer, how can you anticipate anything as drastically different from what we do today as your microminiature computer in only ten years? Where are the people and the knowledge that will permit this in ten years?

Mr. Rice: I agree that the ten years is probably on the optimistic side. However, you will note that the methods of specification for what I call the "algebra" of these systems includes the device characteristics and the physical layout. This implies that much of our early work is in development of a new "system function algebra." Once this algebra is automated, the design of new systems will be done rapidly, and we will be less dependent on present day design techniques.

H. Richmond (*System Development Corp.*): What is meant by "machine language is approximately problem language"? What is done in this case if a new variable is needed and your hardware is built?

Mr. Rice: We have to recognize in our future designs that problem language is not static. In other words, FORTRAN, if I may use that example, has already proven that we need extensions. Therefore, I think the computer designers — and I happen to be one who believes this — must design control sections which admit that programming language is dynamic. We should be able to incorporate new instructions without going back and completely rewiring. There is much research work to be done on the type of control situation implied. I, for one, am very anxious and excited about working in this area.

J. Feitler (IBM): What about analog-computer logic with digital-computer hardware with many arithmetic elements (100 to 1000-plus arithmetic elements) using microminiature components at "3rd generation level"?

Mr. Rice: I am not certain that I fully understand the implication of analog-digital computer hardware. If you mean we are working on separate portions of the problem in parallel, using the accuracy obtained by digital techniques, I think there are existing machines showing this tendency. Assuming that we can assemble these systems to solve the classes of problems we have to solve, this is an interesting area for development. As to the 3rd generation I don't think I would hazard a guess.

L. B. Harris (*GE*): How do you propose to implement self-checking of system functions, that is, to pin-point the trouble?

Mr. Rice: Much work is being done on this subject in various research and development laboratories. I think we have to reanalyze where we want to spot errors. For example, in the talk a complete arithmetic process unit is shown as a single system function. I purposely chose the one-out-of-ten code in this example, because it is possible to put a single check device at the far end of the system. If more than one pulse arrives, there is trouble. If less than one pulse arrives, there is trouble. If only one pulse arrives, I would assume it is correct, because the logical paths do not cross. Much research remains to be done in this area, so I don't have a complete answer. I believe we should analyze how small or how large an element should be when we look for trouble. We should probably diagnose trouble in major elements rather than at the Boolean circuit level. We should also examine our need for a single code throughout a complete system. That is to say, do we need the same bit code in the processing element that we need in bulk storage. There are many ways of tackling the problem, and I think we will have to look to future generations for the complete answer.

C. H. Propster (*GE*): What reason do we have to think a self-organizing computer will ever be produced?

Mr. Rice: Perhaps you are in a better position to answer this question than I am. I believe that two things are necessary before self organizing systems are more than (if I may use the expression loosely) ideas: First, we have to really understand what we want to do in the system to make it self-organizing. This is the logical consideration. Secondly, it is fairly obvious it will take lots of components, so we have to develop the manufacturing techniques to produce large numbers of components economically. Whether or not we will get to the most blue-sky systems is hard to predict, and I will shy away from that. I think that manifestations of self-organizing systems are possible, and that they will be developed.

P. J. Scola (*GE*): On the throw-away computer, what will the input-output wiring look like? Will there *be* any input-output?

Mr. Rice: This is a very difficult question to answer, even in an hour and a half. At all stages in the future, we will need communications from humans to the machine. We hope that voice recognition will allow us to get from a human to the machine language. In the throw-away portion, I am specifically referring to the electronic elements of the computer: that which we now know as the main frame. In particular, the capacity of the bulk storage associated with the main frame is drastically increased. This will reduce the peripheral equipment such as tape, discs and so forth. So in effect we will be throwing that section of I/O away. The concept of throwing away is also hard for *me* to accept. However, I ask myself how are we going to repair microminiature devices; and I come up with the answer that we had better make them cheap enough so we can throw them away.

F. Panch: Would you care to speculate on what kind of computers might be in use twenty to forty years from now?

Mr. Rice: Frankly, I have trouble envisioning what I call future computers. I think that the major changes beyond these generations will be in new uses for computing systems. If we can make computer language approximate human language, or at least equal problem language, the challenge will be in what we do with the system and in making the systems cheaper so we can use them more frequently.

R. J. Brousseau (*U of C*): In saying that computer language should approach problem language in future computers, are you suggesting that the computer hardware should accomplish the functions now being borne by present automatic programs, such as mnemonic instructions, symbolic memory names?

Mr. Rice: The answer to that question, in terms of generalities, is yes. First we need better devices to go into memory so we may perform logic in the memory itself. At a time when this technique is sufficiently advanced, we may expect that instead of "addressing," we can tell the memory to find a particular field of data by specifying the "tag" inherent in the data. There are several other logical techniques which may be used for symbolic look-up. The extent to which designers can do this is dependent on the person specifying the problem. He must establish a set of rules that is fixed.