NASA TM X-1544

SOME ARGUMENTS FAVORING NON-CONVENTIONAL TYPES OF COMPUTERS

By Wayne A. Lea

Electronics Research Center
Cambridge, Mass.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – CFSTI price $3.00

SOME ARGUMENTS FAVORING NON-CONVENTIONAL TYPES OF COMPUTERS

By Wayne A. Lea
Electronics Research Center

SUMMARY

This paper considers the basic question of the adequacy of conventional, general-purpose digital computers, and whether new, non-conventional machine structures would be useful for expanding the capabilities of machine performance, to permit automatic solution of problems previously handled only by humans. Included in the class of "non-conventional" machines discussed herein are hybrid machines, multiprocessors, highly parallel computers, Holland and SOLOMON machines, bionics models, diffused-function anastomotic networks, pattern recognition machines, "linguistically intelligent" machines, and other more exotic machines.

Several arguments favoring non-conventional machine structures are presented. Considered separately are arguments based on: (a) the opinions of certain experts; (b) ultimate theoretical capabilities of various machine types; (c) structures of biological counterparts in brains and nervous systems; (d) the potential for natural-language programming; (e) relative computational economies and efficiencies; and (f) the potential for successful development of new types of computer structures. Such arguments demonstrate the problems one can get into by confining study to present-day digital computer organizations, and the potential advantages of new, user-oriented designs.

I. INTRODUCTION

Digital computing machines are being applied to an ever-expanding set of problems, extending from simple arithmetic operations to such exotic tasks as mechanical language translation, information retrieval, visual pattern recognition, automatic speech recognition and synthesis, theorem proving and playing checkers and chess. It has often been pointed out that digital computing machines can do anything which can be described in detail with a finite set of instructions.* A general-purpose computer is considered a universal machine in the sense of Turing; it can imitate any other machine. In

* As stated, the quoted assertion is completely false. There are certain problems which cannot be handled by Turing machines, and thus cannot be expected to be handled by digital computers. See section III of this paper.

particular, if the human brain is a machine, a computer with access to sufficient memory can then, in principle, imitate the human brain exactly. Hence it can, according to this argument, perform the same operations, such as language translation, pattern recognition, theorem proving, and the like.

There are, however, certain strings attached to this prediction [ref. 1, p. 308]. No mention is generally made of time or economy relations, and such imitation arguments usually neglect the fact that one machine imitating another is greatly slowed up by the mechanics of describing the one machine in terms of the second. Even more important, however, is the fact that, under any known techniques one machine can imitate another or carry out its operations only if one can describe exactly, in precise detail, the first machine or the desired operation to be performed. Of course, at least for the present, this cannot be done for the human brain and many of its functions [ref. 1, p. 308].* The general question remains open as to whether there are operations (performable by humans or otherwise) which cannot be adequately handled by digital computers. In fact, it appears that there are several distinct questions which are involved in establishing the capabilities of digital computers organized and operating in accordance with today's technology. There is, on the one hand, the question of <u>ultimate capabilities</u>: Are there theoretical limitations which disallow performance of certain operations, regardless of the size of accessible memory space, with present day digital computers? Then, there is the question of <u>practical capabilities</u>: Are there practical limitations of economy, efficiency, or utility which restrict the ange of operations performable with present-day digital computers? Related to these questions is one concerning <u>whether new machine structures would be useful for expanding the capabilities of machine performance</u>, particularly for such human-like functions as pattern recognition, language translation, theorem-proving, and natural-language communication. This memorandum is concerned with these questions.

The author's investigation of non-conventional machine structures was begun with the vague thought that machines built around logical, natural-to-human primitive structures, combined in natural (and, thereby, non-conventional) organizations, might simplify a number of man-computer interaction problems, including the problems of translating from natural language to programming language, and compiling from pro-

---

* Dreyfus [ref. 2] argues that it never can be done because of certain inherent ambiguity tolerances in human actions, etc. His views are still in the heat of controversy with Minsky, Papert, and others.

gramming language to machine languages. To indicate the expected changes in the logic of machine structure, the term "non-Boolean machines" was adopted, and will be used in the following discussion.*

The Question of Non-Boolean Machines

The introductory comments suggest that the basic question to be considered in this document is whether conventional digital computers are adequate for satisfactory performance of certain natural "human" operations, or whether "non-Boolean" machine structures would be necessary or more adequate. No attempt will be made to <u>prove</u> the necessity or value of non-Boolean machines. Rather, several types of arguments favoring their application and development will simply be considered. Such arguments will hopefully demonstrate the need for <u>careful consideration</u> of the problems one might get into by confining study to present-day digital computer organizations, and the potential advantages of new, user-oriented designs.

There are various forms of arguments which may be presented justifying the investigation of non-Boolean machines. Preliminary consideration of arguments based on the following will be made:

1. The opinions of certain experts (section II)

2. Ultimate capabilities of machines (section III)

3. Structures of biological counterparts in brains, nervous systems, and the like (section IV)

4. The value of natural-language programming (section V)

5. Relative computational economies or efficiencies (section VI)

6. The potential for successful development of new types of computer structures (section VII).

The arguments of experts in computation and related fields will be considered first.

---

*However, it should be noted that the term "non-Boolean machines" does not define or delineate any clear class of machines. The term is only suggestive of the non-conventional logic or structure of machines; that is, machines with some degree of significant deviation from standard digital or analog computers. The reader is cautioned against reading too much into the term. Essentially, it may be replaced by the longer and less suggestive term "non-conventional machines" in all cases.

II. THE ARGUMENT FROM OPINIONS OF EXPERTS

Many experts (including Shannon, McCulloch, Turing, Minsky, MacKay, Dreyfus, Taube, Davis, Kleene, McNaughton, Wang, et al.) have discussed the universality of digital computers and have debated whether the digital computer can solve any problem which can be precisely stated. Also, Shannon and Pierce (ref. 1) have pointed out that just because a computer can do something does not mean that it should do it. The desires of the users and the needs for optimal use of the computer and human capabilities ought to be considered in determining what tasks to automate. As Pierce says:

> "The fact that a general-purpose computer can do almost anything does not mean that computers do all things equally well. Some things they do much better than human beings; some things they do worse. Machines are not people. While it is highly desirable to strengthen their weakness, the greatest immediate gains, and some long-range ones as well, will come from exploiting their strengths.
>
> "Partly, the strengths of computers are associated with certain types of problems necessarily involving a great deal of straightforward computing on a great deal of input data, as opposed to game playing or recognition problems [ref. 1, p. 295]."

He also noted that the strengths of computers are associated with their organization and programming.

Agreeing with these arguments, Shannon expanded on them to note that such arguments (as well as those to be discussed in sections III through VII) suggest that

> "...efficient machines for such problems as pattern recognition, language translation, and so on, may require a different type of computer than we have today. It is my feeling that this computer will be so organized that single components do not carry out simple, easily described functions. One cannot say that this transistor is used for this purpose, but rather that this group of components together performs such and such a function. ...In a machine of the type I am suggesting, it would be impractial to describe the purpose or action of any single component. I know of very few devices in existence which exhibit this property of diffusion of function over many components [ref. 1, pp. 309-310]." (As will be shown in section IV, there are certain brain functions which appear to be achieved in this gross, associative-structure way.)

Shannon acknowledges that "such a computer may lead us in something very difficult for humans to invent and something that requires very penetrating insights [ref. 1, p. 310]." But, he continues [ref. 1, p. 310], "If this sort of theoretical problem could be solved within the next few years, it appears likely that we shall have the hardware to implement it." Considering the advances in computer and device technology, he concludes that it is reasonable to ask:

> "Can we design with these a computer whose natural operation is in terms of patterns, concepts, and vague similarities rather than sequential operations on ten-digit numbers? Can our next generation of computer experts give us a real mutation for the next generation of computers [ref. 1, p. 310] ?"

In other words, can we build computers whose natural operations, or primitives, are readily expressible as patterns, concepts, similarities, or logical propositions which are natural for the human to use in natural-language communications?

Vannevar Bush echoed Shannon's remarks [ref. 1, p. 311] , observing:

> "To me some of the most interesting machines built by man are neither digital nor analog. I think that this field is somewhat neglected today; yet in it lies much of the promise for the future."

Continuing this train of thought concerning the capabilities of machines of various structures, Walter Rosenblith [ref. 1] stated:

> "We are hopeful that we shall gain added insight into both brains and computers by analyzing their structural principles in relation to their programmability. Some of Claude Shannon's remarks on serial versus parallel operation, on diffused-function operation which hark back to a theme of von Neumann's, make this hope more explicit. Present-day neuroanatomy and neurophysiology do not furnish us with recipes of what mix of hierarchical organization, specificity, randomness, and redundancy, will yield a particular performance in the sensory domain, for instance. These sciences are likewise incapable of telling us today how evolution came to make such extended use of analog-to-digital (and vice versa) recoding and interaction schemes. Perhaps those who preside over the evolution of computers will contribute to the study of brain function by developing a series of calculi of relations that will in some sense transcend the digital versus analog dichotomy."

These comments raise the issues as to the potential of digital versus analog, and parallel versus serial, machine structures, <u>and</u> the possibilities of other machine structures not fitting into these opposing categories. The digital-versus-analog and serial-versus-parallel dimensions of machine distinction are only two of many dimensions which might be possible in distinguishing machine structures. Shannon [ref. 1, pp. 319-320] has noted this, in saying that:

> "Computing machines as we have them today, the analog and the digital, are only two of a vast population which we have not yet explored. What I am saying in my challenge is that we should look to other possible machines in the same general area of information processing."

Thus reiterating his position and clarifying the fact that he was not specifically suggesting that parallel machines would be better than the present serial forms, "but only that they would be different and might lead to the possiblity of doing more easily some of the things which seem difficult in serial machines [ref. 1, p. 319]," Shannon suggested we ask, "Are there other types of computing machines that will do certain things better than the types we now have?" He closed with this comment: "I am suggesting that there well may be such machines."

With regard to the potential for different types of machine structure, Minsky has noted [ref. 1, p. 322] that:

> "Every designer of programming languages now-a-day is thinking about more complex associative memories for serial computers. Parallelism is in the air. The next generation of machines, or the one after the next generation, will certainly have large parallel aspects."

However, he noted that, despite the comments of Shannon and Pierce, "we are not sure that it [a conventional computer, like the IBM 7090] cannot solve many of the difficult intellectual problems that face us." Thus, in deference to some of the arguments given herein favoring new machine structures, Minsky shows more faith in the capabilities and utilities of present conventional machine structures.

One controversial figure who takes issue with Minsky and other researchers involved with "artificial intelligence" is H. L. Dreyfus [ref. 2]. Although Dreyfus could hardly be termed a computer expert, he has raised a number of general questions about the ability of present computers to exhibit "artificial intelligence"; i.e., to do things which would be considered to be "intelligent" if humans had done them. In a paper which has caused a stir of frustration at M.I.T., and in the general academic community concerned with artificial intelligence, Dreyfus suggested [ref. 2, p. 222] that "there are four distinct types of intelligent activity, only two of which do not presuppose ... [certain] human forms of information processing and can therefore be programmed." He concluded that:

> "Significant developments in artificial intelligence in the remaining two areas must await computers of an entirely different sort, of which the only existing prototype is the little-understood human brain."

Dreyfus has raised some interesting points, which certainly favor consideration of non-conventional machine structures.

It appears that regardless of what viewpoint is taken about the possibility of intelligent activity by machines, there is general agreement about the possibility of future machines taking on distinctively different structural forms, particularly if they are to perform such human-like tasks as theorem-proving, pattern recognition, and the like.

This is illustrated by contrasting Drefus' reasons for non-conventional machines with those of pro-artificial-intelligence researcher Paul Armer [ref. 3, p. 393] :

> "While I do believe that today's digital computers can exhibit intelligent behavior, I do not hold that the intelligent machines of the 1970's will necessarily resemble today's machines, either functionally or physically. In particular, in my desire to see machines pushed further out in the continuum of intelligence, my interests in the dimension of speed are very minor; the organizational aspects (sophistication of the information processes) are obviously much more important. Likewise, I hold no brief for the strictly digital approach; a combination of analog and digital equipment may prove to be better. I do not mean to disown the digital computer for it will be a most important tool in the endeavor to advance in our continuum."

All these opinions of experts in computation and related fields might be summarized by suggesting that non-Boolean machines may be significantly more useful than conventional digital machines, particularly for such problems as language translation, pattern recognition, theorem proving, game playing, and other problems previously handled only by humans. Dimensions of analog-versus-digital and parallel-versus-sequential operation may be considered in guiding one to the development of non-Boolean machine structures, but it is best to keep an open attitude toward other dimensions of distinction if one is to make best use of the potential of the whole "population" of possible machine structures.

III. THE ARGUMENT FROM ULTIMATE CAPABILITIES

The question of ultimate capabilities as raised in the introductory comments concerns whether or not there are theoretical limitations which disallow performance of certain operations, regardless of computation time and the size of accessible memory space, with present-day computers. This theoretical question leads one to consider the predictions of Turing machine theory, or general automata theory, concerning capabilities of various machines. This is particularly true in the light of how a general-purpose computer with unbounded memory is known to be equivalent to a universal Turing machine [refs. 4, 5, 6]. The assertion of their equivalence dictates that there are no problems which are solvable by any Turing machine that cannot be solved by such open-ended digital computers, and vice versa. Hence, considering the capabilities of Turing machines would indicate the maximum problem-solving potential for present computers (maximum conditions occuring when a potentially infinite tape or other storage and input-output system is available).

The equivalence of universal Turing machines and computers demonstrates that if one is to justify a new machine type on the basis of its having more ultimate capability than present computers, he needs to show that his new machine handles problems which a universal Turing machine cannot handle. This, in turn, raises the obvious question of whether there are any problems which a Turing machine cannot solve. Goedel's theorem [ref. 7], Turing's results [ref. 6], and other more recent results by Davis [ref. 6], Chomsky [ref. 8], and others [ref. 9], demonstrate that there are some such problems. Perhaps, then, non-Boolean machines might be built which solve such problems, and hence would have more ultimate capabilities than conventional computers. This is a natural and tempting supposition. *

However, one must be somewhat cautious about misunderstanding the results obtained in the "pure science" of general automata theory (with its associated predictions about ultimate capabilities of mathematical machines) and their implications on the "applied science" of computation theory (with its associated questions about what practical computers can effectively do). This is particularly well brought out by Davis' comments in discussing the practical implications of the abstract theory of computability:

> "The existence of universal Turing machines, another result of the theory, confirms the belief of those working with digital computers that it is possible to construct a single "all-purpose" digital computer on which can be programmed (subject of course to limitations of time and memory capacity) any problem that could be programmed for any conceivable deterministic digital computer. This assertion is sometimes heard in the strengthened form: anything that can be made completely precise can be programmed for an all-purpose digital

* However, its acceptance would imply the denial of Church's thesis [ref. 10] and the generally accepted universality of Turing machines. It would contradict the commonly accepted viewpoint that Turing machines can compute anything which is "effectively" (i.e., "mechanically") calculable.

computer. However, in this form, the assertion is false. In fact, one of the basic results of the theory of computability (namely, the existence of nonrecursive, recursively enumerable sets) may be interpreted as asserting the possibility of programming a given computer in such a way that it is impossible to program a computer (either a copy of the given computer or another machine) so as to determine whether or not a given item will be part of the output of the given computer [ref. 4, p. vii]."

The error involved in suggesting (e.g., ref. 1, p. 308) that "anything that can be stated completely precise can be programmed on a general-purpose digital computer" illustrates the errors one can get into by careless restatement of the formal results of mathematical machine theory. Davis [ref. 6], Goedel [ref. 7], Turing [ref. 6], and others [ref. 5] give numerous other examples of precisely specifiable, but unsolvable problems. In addition, a number of practical, significant problems unsolvable by universal Turing machines have been found in mathematical linguistics and other fields (e.g., ref. 8 and other references therein).

Thus, there exists a set of problems (not all of which are merely pedagogical) which, though amenable to precise statement, are unsolvable by universal Turing machines, and thus unsolvable by general-purpose computers, regardless of time and memory capacities.

It is tempting to infer from this that computers are inadequate for handling certain human-like activities or problems and, thereby, that computers cannot be artificially intelligent. This might encourage one to look for new (i.e., "non-Boolean") machine types which would succeed where the present computers are expected to fail. This, for example, appears to be part of the motivation behind Dreyfus' conclusion [ref. 2, p. iii] about the ultimate ineptitude of present computers and the need for "computers of an entirely different sort." (See also section IV of this paper.) Certainly, a confusion concerning the practical significance of Goedel's theorem and associated limitations of abstract machines is involved in the conclusion drawn by Nagel and Newman [ref. 11, p. 1695] and reiterated by Taube [ref. 12, p. 4] that:

"Goedel's conclusions also have a bearing on the question whether calculating machines can be constructed which would be substitutes for a living mathematical intelligence. Such machines, as currently constructed and planned, operate in obedience to a fixed set of directives built in, and they involve mechanisms which proceed in a step-by-step manner. But in the light of Goedel's incompleteness theorem, there is an endless set of problems in elementary number theory for which such machines are inherently incapable of supplying answers, however complex their built-in mechanisms may be and however rapid their operations. It may very well be the case that the human brain is itself a "machine" with built-in limitations of its own, and that there are mathematical problems which it is incapable of solving. Even so the human brain appears to embody a structure of rules of operation which is far more powerful than the structure of currently conceived artificial machines. There is no immediate prospect of replacing the human mind by robots."

Regardless of how much one may agree with Nagel and Newman's conclusion concerning the relative powers of human minds and "calculating machines," one should be able to see the error in their application of Goedel's results. Goedel's incompleteness theorems (and several other subsequent results in mathematical logic, recursive function theory, and the theory of computability) demonstrate the existence of problems unsolvable by Turing machines and, hence, by digital computers. But, no one has ever demonstrated (and it is unlikely that they ever could demonstrate) that any of these unsolvable problems are solvable by humans or any particular new types of machines. That is, no machines have been found that exhibit more power than universal Turing machines. Moreover, it has been argued rather convincingly by Putnam (ref. 13) and Scriven (ref. 14, and also ref. 15, pp. 139-140) that Goedel's theorem is no more an obstacle to a computer than to humans. One may then conclude with Arbib [ref. 15, p. 140] that "Goedel's theorem is not to be taken as a proof that no machine can be intelligent," and, more generally, that Goedel's theorem and other associated results are not to be taken as proof of inadequacies exhibited solely by present digital computers.

The argument up to this point might be summarized by saying that, since general-purpose digital computers, when endowed with potentially infinite memory, are behaviorally equivalent to universal Turing machines, ultimate-capability arguments favoring non-Boolean machines will depend on exhibiting problems solvable by some new machine type but unsolvable by a universal Turing machine. On the assumption that such exhibited problems could be considered to involve only "effectively calculable" functions, such results would refute Turing's hypothesis [ref. 6] and Church's thesis [ref. 10]. Frankly, the results to date, and the viewpoints taken by most researchers interested in this problem, would suggest that the search for such problems would be fruitless. Nevertheless, it is an open question as to whether such problems and associated non-Boolean machines do exist. The burden of proof, however, rests on him who would assert that such problems are to be found.

The discussion of ultimate capabilities as has been considered here does not involve the amount of time required for computations. Such time considerations might be considered to be most closely related to the questions of efficiency and economy to be discussed in section VI. Yet, there is a sense in which it is reasonable to consider computation times in relation to ultimate machine capabilities. Since many of the human-like operations which one may wish to automate (e.g., visual pattern recognition, speech recognition and synthesis, game playing, and the like), and which are not being handled completely adequately by present digital computers, are most properly handled as real-time problems of man-computer interaction, it is appropriate to consider ultimate capabilities in comparison to what have been called "real-time automata" [ref. 5, pp. 402-411, and refs. 16, 17]. Thus, it may be most appropriate in the long run to compare the ultimate capabilities of conventional and various non-Boolean machines in terms of operations they can perform within certain time limits following input times, rather than in terms of unlimited-time Turing machine operations.

In considering the significance of real-time operations, it is important to note several ways in which real-time operation may affect ultimate capabilities. McNaughton noted in 1961 [ref. 5, pp. 403-6] that Yamada's work [refs. 17, 18] suggests that there are computable transformations (i.e., operations performable by a universal Turing machine) which cannot be performed in real-time (at least by any known real-time device, and, probably, by no real-time operating device). Thus, the set of real-time computable functions is apparently a proper subset of the set of all computable functions.

On the basis of this prediction one might expect that two general-purpose machines which differ in structure may (although each is equivalent to a universal Turing machine in the ultimate, time-independent sense) differ in the set of real-time operations they may perform. This is well brought out by several open questions which McNaughton noted [ref. 5, pp. 403-6]:

> "For example, for real-time operations a Turing machine with one tape is (possibly) not the most general type of growing automaton. It is an open question whether Turing machines with several tapes can do real-time operations that Turing machines with a single tape cannot do. Insight gained from Yamada [ref. 18] (the first work to my knowledge to discuss real-time operations in the theory of automata) seems to favor an affirmative conjecture on this question...
>
> "...there may be real-time operations performable by [Holland's] iterative circuit computers but not performable by multitape Turing machines. As far as I know, this is an open question...
>
> "An open problem is whether the multiplication of arbitrarily large numbers is a real-time operation.
>
> "...The problem of characterizing (by some mathematically interesting necessary and sufficient condition) [symbol] sequences that result from real-time operations is an unsolved one."

These questions have been the subject of much subsequent research [refs. 16, 17, 19].

There is thus a significant difference between the ultimate capabilities (i.e., with potentially infinite memory) of a machine operating in real-time and one operating in general Turing machine fashion without time restrictions. It may well be that real-time capabilities are of most interest to the question of whether non-Boolean machines are required. Yamada's results and the possibility of real-time operations performable by non-conventional machine types (such as Holland's iterative circuit computers), but not performable by present sequential processors, apparently favor the consideration of non-Boolean machine structures.

IV. THE ARGUMENT FROM BIOLOGICAL COUNTERPARTS

Interesting aspects of the above question of ultimate capabilities are concerned with the functional or "behavioral" study of machines, and (with the exception of the effects of real-time constraints) have little or no association with the duplication of machine structure or techniques for solution. In contrast, the argument from biological counterparts as it will be presented in this section is very much concerned with specific structures and techniques for problem solution.

The point of the argument from biological counterparts is just this: the synthesis of machines to perform certain cognitive processes may be advanced and simplified by studying and duplicating the techniques and structures used in the biological systems which successfully perform those functions. Thus, if one wishes to have adequate pattern recognizers, theorem provers, language translators, and so forth, following the biological counterparts as working, structured models may ease the design process. This is essentially the viewpoint of the field of bionics, which, as McCulloch has said [ref. 20, p. 393] "is concerned...primarily with an attempt to understand sufficiently well the tricks that nature actually uses to solve her problems, thus enabling us to turn them into hardware."

It is true that duplication of human brain structure is no more needed to achieve corresponding behavior than one needs to build airplanes like birds or ships like ducks or fish. But, just as in those cases, there may be some laws and principles of operation which are common to the natural and artificial systems, and which it may prove profitable to discover. Thus the implication that duplication of structure is necessary to the duplication of behavior is not being made. On the contrary, it should be clear that duplication of terminal behavior will not necessarily require duplication of techniques of solution or of "machine" structure. It is helpful to keep this distinction in mind when considering the implications of biological counterparts of machine structures. For further discussion of this distinction, see MacKay's excellent studies [refs. 21-23] . However, turning this argument around may make it equally clear that the distinction between duplication of behavior and duplication of structure does not negate the value of using the structural features of available biological systems to aid in the structural design of machines which duplicate behavior of those biological systems. It still is true that following the biological counterpart as a working, structured example may ease the machine synthesis process. (An example of the design of mechanisms by use of biological prototypes is given in reference 24, especially page 292.) This apparently was the viewpoint of vonNeumann, who is said to have believed that more is to be learned about machines from the study of neurology [and other biological sciences] than is to be learned about biological systems from the study of machines [ref. 12, p. 127] .

The potential for simulating cognitive behavior of biological systems has been the subject of great debate since the advent of the digital computer. (For representative examples of resulting "artificial intelligence" studies, see reference 25). One recent critic of certain attempts to simulate cognitive processes has been Dreyfus [ref. 2], who, as noted in section II, has concluded that certain cognitive behavior cannot be simulated:

> "An examination of... [difficulties encountered in attempting to simulate cognitive processes on digital computers] reveals that the attempt to analyze intelligent behavior in digital computer language systematically excludes three fundamental human forms of information processing (fringe consciousness, essence/accident discrimination, and ambiguity tolerence). Moreover, there are four distinct types of intelligent activity, only two of which do not presuppose these human forms of information processing and can therefore be programmed.
>
> Significant developments in artificial intelligence in the remaining two areas must await computers of an entirely different sort, of which the only existing prototype is the little-understood human brain [ref. 2, p. iii] ."

Dreyfus attempts to formulate and criticize the common assumption, shared by many workers in artificial intelligence, that humans face the same difficulties in cognitive processing as machines do, and that, therefore, the difficulties encountered by machines in simulating cognitive behavior obviously can be overcome, since humans overcome them. It is beyond the interest and scope of this paper to discuss or criticize Dreyfus' arguments. It may be noted in passing that, in one sense, they seem to constitute another in the series of arguments to the effect that "you'll never get digital computers to do such-and-such". Such arguments are believed by MacKay and others [refs. 22, 23, 26] to be "foredoomed as soon as the speaker has been induced to say precisely what behavior he would regard as satisfactory [ref. 22] ," but Dreyfus' objection to this would seem to be that these very processes cannot be precisely described in determinate step-by-step fashion (ref. 2, especially p. 49). It is their intractability to such standard algorithmic solution, their inherent vagueness or "global" character, that led Dreyfus to look toward "computers of an entirely different sort [ref. 2, p. iii] " to handle these cognitive processes.

In general, if any such behavioral inadequacies for cognitive processing by digital computers could be demonstrated, they would certainly favor the development of non-Boolean machines. This is directly analogous to the observation in the previous section that problems ultimately unsolvable by universal digital machines might favor development of new machine types.

From these remarks one can see that any inability of present digital computers to duplicate certain human cognitive behavior would favor the development of non-Boolean machines. But, in addition, even if present computers could be shown to duplicate all such behavior successfully, it is not clear that that alone would be considered satisfactory. As Shannon and McCarthy [ref. 27, p. vi] have pointed out,

> "...it is possible, in principle, to design a machine with a complete set of arbitrarily chosen responses to all possible input stimuli.... Such a machine,

in a sense, merely looks up in a "dictionary" the appropriate response... Such a machine...does not reflect our usual intuitive concept of thinking. This suggests that a more fundamental definition [of 'thinking' and of 'artificial cognitive processors'] must involve something relating to the manner in which the machine arrives at its responses... something which corresponds to differentiating between a person who solves a problem by thinking it out and one who has previously memorized the answer." [Emphasis added.]

Similar reasoning has led Kochen, MacKay, and their colleagues [ref. 26, p. 2] to conclude that:

"...although output behavior is crucial, it is entirely legitimate to examine the way in which it is produced, including the structure and its information flow-map, as a way of checking that it is "genuine" and not "rigged" or "accidental."

MacKay has thus been led to consider a more challenging problem than merely duplicating input-output behavior, namely, "How far is it possible to envisage an artificial mechanism that would not only imitate human behavior, but work internally on the same principles as the brain [ref. 22, p. 263] ?"

When one considers this more demanding question, one quickly notes marked differences between the manner in which the brain functions and is structured and the manner in which conventional digital computers operate. These differences of technique and structure may be important to the justification and synthesis of non-Boolean machines, as was pointed out by the comments of Shannon, Pierce, Rosenblith, and others quoted in section II. Consider again, for example, the comments of Shannon [ref. 1, p. 309] :

"If there are these important differences at the psychological level between computers as we have them today and brains, one may raise the question as to whether this is a reflection of a different internal organization, and if so, what are the chief differences? I believe that, in fact, there is very little similarity between the methods of operation of the computers and the brain. Some of the apparent differences are the following. In the first place, the wiring and circuitry of the computers are extremely precise and methodical. A single incorrect connection will generally cause errors and malfunctioning. The connections in the brain appear, at least locally, to be rather random, and even large numbers of malfunctioning parts do not cause complete breakdown of the system. In the second place, computers work on a generally serial basis, doing one small operation at a time. The nervous system, on the other hand, appears to be more of a parallel-type computer with a large fraction of the neurons active at any given time. In the third place, it may be pointed out that most computers are either digital or analog. The nervous system seems to have a complex mixture of both representations of data."

It was these distinctions (and other arguments to be considered in this memorandum) that led Shannon to conclude that "efficient machines for such problems as pattern recognition, language translation, and so on, may require a different type of computer than any we have today [ref. 1, p. 309]."

One of the most frequently noted distinctions between computers and their biological counterparts is the use of analog, as well as digital, processing in brains (for example, ref. 28, Ch. 1; ref 29, p. 27; ref 27, p. 5; ref. 30, p. 91; ref. 1, pp. 309, 311, 314 319 ; ref. 31, p. 68). Although crude approximations to the action of individual neurons in brains and nervous systems permit somewhat of a digital "on-off" characterization of nerve pulses, there are distinctively analog processes involved in individual neurons, in the operation of nervous "subsystems", and in general operation of nervous systems. Von Neumann noted this in his excellent introduction to the similarities and differences between computers and brains [ref. 31]. For example, he noted that:

> "...processes which go through the nervous system may...change their character from digital to analog, and back to digital, etc., repeatedly. Nerve pulses, i.e., the digital part of the mechanism, may control a particular stage of such a process, e.g., the contraction of a specific muscle or the secretion of a specific chemical. This phenomenon is one belonging to the analog class...the nerve-pulse part of the system, which is digital, and the one involving chemical changes or mechanical dislocations due to muscular contractions, which is of the analog type, may, by alternating with each other, give any particular process a mixed character [ref. 31, pp. 68-9]."

Add to these factors the mixed nature of the summation and threshold operation of a neuron (e.g., ref. 27, p. 5; refs, 32, 33; ref. 31, pp. 52-60), the frequency encoding of stimulus intensities (e.g., ref. 34, p. 123; ref. 31, p. 79), and the presence of some analog properties in the neuron pulse [ref. 35, p. 23], and one can readily see sharp distinctions from present computers of either strictly digital or strictly analog form.

These mixed aspects of the nervous system are not trivial features insignificant to the mode of operation and general terminal behavior of these biological systems. The logic of threshold operation and the adjustment of peripheral threshold levels by the reticular formation (to effect changes in logical function performed by the neural network), for example, are directly involved in what Warren McCulloch calls "the great problem of the nervous system", the reticular formation, or core of the brain structure [ref. 33]. The reticular formation controls what signals the brain will receive from peripheral nerves, and decides to what basic mode of behavior the whole organism will be committed. It does so by analog feedback controlling the thresholds of peripheral neurons, and by a triadic and intentional decision logic which defies representation by conventional logic systems [ref. 33].

A basic characteristic of the reticular formation is its "redundancy of potential command" resulting from broad parallel and associative processing in its iterative networks [ref. 33]. This is an example of the general use of parallel processing in biological systems. Numerous other examples could be given. In general, von Neumann has summarized this important distinction between biological mechanisms and conventional artificial ones by remarking that [ref. 31, pp. 51-2]:

> "...an efficiently organized large natural automaton (like the human nervous system) will tend to pick up as many logical (or informational) items as possible simultaneously, and process them simultaneously, while an efficiently organized artificial automaton (like a large modern computing machine) will be more likely to do things successively...That is, large and efficient natural automata are likely to be highly parallel, while large and efficient artificial automata will tend to be less so, and rather to be serial.
>
> "...it should be noted, however, that parallel and serial operation are not unrestrictedly substitutable for each other."... [Translation from a serial scheme to a parallel one may be impossible or difficult, requiring a change in logic and procedural organization, and, conversely] "the desire to serialize a parallel procedure may impose new requirements on the automaton...Hence the logical approach and structure in natural automata may be expected to differ widely from those in [conventional] artificial automata."

Von Neumann [ref. 30] and others [refs. 35, 36] have also developed models for how biological systems achieve reliable computation with unreliable components, by the use of parallel processing or "multiplexing" [ref. 30, p. 63ff], logical "coding" [ref. 34, pp. 123], and other mechnisms not prevalent in conventional computers.

The several advantages of parallel processing have led to the development of new machine structures such as "multiprocessors" and "iterative circuit computers." Yet, it seems that, in spite of its definite importance to the justification of non-conventional machine structures, strict parallelism alone is not enough. It appears that many brain functions (such as certain peripheral data processing operations in a nervous system [ref. 36, p. 123], some inter-neural interactions in visual sensory systems [ref. 37], and aspects of the reticular formation operations [refs. 38, 39] are best represented not by many distinct, simultaneous parallel computations, but rather by some involved, "integrative", "associative" processes in which many components are involved in a single computation and single components do not carry out simple, easily separated functions.* The apparent presence and importance of such

* This form of associative or integrative processing is, in particular, achieved by what McCulloch calls anastomotic networks [ref. 36] in which many simple components or neurons are replaced by a rank of complex components each member of which is computing, in general, a different function of its inputs.

associative or "diffused-function" processing in human information-handling systems seems to be a particularly strong point favoring the development of non-Boolean machines.

It is this form of diffused-function computer which Shannon looked to for handling such problems as pattern recognition, language translation, and the like. The value of diffused-function operation, particularly in the light of its apparent use in biological systems, encourages one to look toward the development of non-Boolean machines with this form of modus operandi.

Such associative or "diffused-function" operation violates the standard principle of "one organ for each basic operation [ref. 31, p. 13]." Although this principle has been helpful in the design of system structures and programming techniques which permit easy understanding of the functions of all components and subsystems at all times, it is by no means necessary to the successful performance of any given terminal behavior. Its extensive use in digital machine design may in the long run turn out to be an unfortunate result of historical accident (or poor design choice) rather than an intrinsic requirement of machine operation. The fact that analog computers are not built with only one organ for each basic operation illustrates the potential dispensability of the principle [ref. 31, pp. 13-14].

The principle of "one organ for each basic operation" has resulted in another major discrepancy between digital machine organization and apparent structures of biological counterparts; that is in terms of the embodiment of memory in the two systems. Von Neumann has observed, "The 'only one organ for each basic operation' principle necessitates...the providing for a larger number of organs that can be used to store numbers...The totality of these organs is called a 'memory'...[ref. 31, p. 14]." The result of the single-function principle is thus the separation of memory from other operations, yielding the familiar "memory unit" of digital computers. There is no indication that memory exists as such a separate organ in biological systems. In fact, there are strong reasons to suspect that it may be distributed throughout the nervous system. Witness, for example, this observation by neurophysiologist O. H. Schmitt:

> "In all probability the central nervous system, including its memory and computing functions, is a widely distributive statistical time-place-state system where memory of a particular event is smeared out over some millions of cells and these same cells simultaneously hold many millions of other memory traces [ref. 40, p. 251]."

This distribution of memory throughout the system is more like that of Holland and SOLOMON machines than conventional computers [refs. 41, 42]. Von Neumann and others [ref. 31, pp. 60-68; refs. 43, 44, 45] have discussed some possible ways in which memory may be physically embodied in the nervous system.

In general, these studies of the nervous system suggest that the "functional blocks" (e.g., memory unit, arithmetic unit, control unit, and the like) of conventional computers are not in direct correspondence with any "blocks" in brains, and that there are thus general differences in structure of computers and biological information-processing systems. These differences do not necessarily imply differences in ultimate or practical capabilities, but they do favor development of non-Boolean machines if one accepts the premise that duplication of performance is faciliated by duplication of structure.

The argument from biological counterparts, as presented in this section, may be summarized by noting the many ways in which techniques and structure of biological systems differ from conventional computers. Computation in nervous systems is not strictly digital; it involves analog processes of many types. Synaptic digital-to-analog and analog-to-digital transformations, input summation processes, threshold operation and corresponding threshold logic, frequency and spatial codings of stimulus intensities, and even the fact that the neural pulse is not strictly on-off in nature. illustrate the analog and mixed analog-digital features of the biological systems. Parallel processing and diffused-function operation prevail in anastomotic networks, with attendant redundancies of operation and improved system reliability with unreliable components. No separate "memory units," and the like, are apparent in these biological systems.

Thus, despite frequently noted similarities between digital computers and biological systems, there are marked differences between them. These differences do suggest definite value in developing non-Boolean machines, if one accepts the premise that the desired duplication of human cognitive behavior is facilitated by the duplication of structures of biological counterparts. However, this does not suggest that one must confine structures of new machines strictly to the forms of human brains or other biological counterparts. Such a restriction might prove no less of an error than confining structures to conventional digital computer forms. The objective is not necessarily to duplicate the brain, but rather to make use of its helpful working "hints" about how to structure machines for human-like tasks such as pattern recognition, theorem proving, speech recognition and synthesis, language translation, and the like.

V. THE ARGUMENT FROM NATURAL-LANGUAGE PROGRAMMING

One basic way in which the operation of conventional computers differs from that of biological prototypes is in terms of the languages used. As von Neumann observed, "the language of the brain" is not "the language of mathematics" or the "machine language" of today [ref. 31, pp. 80-82]. This distinction is also exhibited at a higher level, in the sense that the languages presently used in communication within and among machines, and between machines and man, are distinctively different from natural languages used in human communication.

An argument favoring non-Boolean machines can be associated with this distinction between machine and human languages. It is perhaps the most interesting, but at the same time perhaps the most vague and nebulous, argument favoring non-Boolean machines. It is concerned specifically with the development of natural, flexible means of communication between man and machine. Crudely said, this argument suggests that machines built around logical, natural-to-human primitive structures, combined in natural (and, thereby, probably nonconventional) organizations, might simplify a number of man-computer interaction problems, including the problems of translating from natural language to programming language, and compiling from programming language to machine languages.

Consider in detail the nature and importance of the distinction between human and machine languages, the relationships of such to machine structures, and the implications on the effectiveness of man-machine communication.

As a result of extensive training and practice, man speaks and understands a natural language. He thus has a language for communicating. Machines, on the other hand, are designed and structured in such a way as to have their own languages for communicating. If there is a language barrier or difference between the man and the machine, one or both may have to be adjusted in what language to use. Thus, either the machine must learn the man's language, or the man must learn the machine's language, or each must compromise and learn some intermediate language.

Although there may not be universal agreement about which (man or machine) should adjust the more to meet the language of the other communicant, many computer researchers would suggest that man has done more than his share of accommodating to the machine. Witness the assertion of Green, Wolf, Chomsky, and Laughery [ref. 46, p. 207] that:

> "Men typically communicate with computers in a variety of artificial, stylized, unambiguous languages that are better adapted to the machine than to the man."

An extreme case of man's accommodating to the machine has occurred with binary or numerical machine codes used extensively since the advent of the digital computer. Such binary coding may be appropriate to the internal structure of the present two-valued Boolean-logic machines, and may in the ultimate sense be capable of representing any given message one might wish to communicate. But it would not, in general, be the most economical and effective language of communication. Economy of expression, for example, as exemplified by lengths of statements or symbol-sequences, may be improved by increasing the size of the "vocabulary" of possible symbols in the expressions. As the vocabulary size (V) increases from two (binary) symbols to larger numbers, the length (k) of sequences needed to represent any set D of distinguishable messages will decrease in accordance with the relation $D = V^k$, or, equivalently, $k = \log D/\log V$.

The progress of computer language design has demonstrated the awkwardness and severe constraints which are invoked upon the computer user when he attempts to communicate in a language of binary form. Although the digital computer is well adapted to rapid handling of binary signals, humans are not. Real-time, man-machine communication in such a language is next to impossible. Added to such difficulties are the consequent needs for the user to understand the gory details of machine structures, instructions, and operational manipulation of programs.

Because of the many inconveniences of such detailed machine coding, and basically as a result of the shorthand used by computer designers and users, a variety of assembly languages evolved, so as to allow naming and reference to general procedures involving several specific machine operations. Also, as the next step toward making languages for communication with machines appears more like the languages of mathematics and other usual notations used by humans, programming languages like FORTRAN were developed. The development of such high-level programming languages resulted in such advantages as drastically reduced costs of programming, easy, precise statement of problems without the need to specify all minute details of machine operation, and a wider spread use of computers by scientists and businesses. The brevity of statement, flexibility, and increased naturalness of these more powerful languages has contributed much to the effectiveness of man-computer communication. They also make better use of the extensive training and experience humans already have with "natural" languages.

The trend has thus been toward languages with larger vocabularies, more flexible structures, and in general, more natural form. The utility of user-oriented programming languages which were closer to natural languages than to machine codes was thus empirically demonstrated. It is easy to understand why such empirical results should obtain. The inefficiency and awkwardness of an extremely small (binary) vocabulary had to be avoided. On the other hand, at the other extreme of very large vocabulary size, very short statements would result, but severe requirements on memory capacity would be required to distinguish the members of the very large vocabulary.

This trade-off between length of expression and vocabulary size appears to have been involved in the evolution of natural languages. The large, but clearly restricted, vocabulary used by English speakers makes it possible to express complex thoughts in reasonably short statements.

In general, both the empirical demonstration of the utility of natural-like programming languages (as displayed by the cited history of computer language forms) and the above argument about the virtue of a compromise between vocabulary size and brevity of statement illustrate the value of programming languages which are close to natural languages in form. Further justifications for natural-language-like programming abound in the literature [refs. 47, 48, 26]. For example, witness Green et al in their assertion that:

> "For convenience and speed, many future computer-centered systems will require men to communicate with computers in natural language [ref. 46, p. 207]."

The importance of natural language programming has been recognized as a major objective of the extensive DEACON project [ref. 47, p. 1]:

> "The first objective is to allow natural English as a language for man-machine communication.... English, as a natural language, is an ideal language for man-machine communication. It is a language that society has evolved for communication of an extremely wide range of data, relationships, and concepts, and it is the language most familiar to military and civilian managers, to intelligence analysts, and, in general, to any senior user who has achieved his position of responsibility at least partially through facility in the use of spoken or written English.
>
> "In many circumstances it is important that responsible senior men be able to work directly with the computer system, rather than through an intervening screen of programmers. This will be practical only when a natural language such as English is the vehicle of communication.
>
> "A second and perhaps more important reason for using English is the great flexibility the language possesses. Expressions of infinite variety, subtlety, novelty, and complexity can be constructed from English words and phrases through the use of a relatively small number of rules of grammar. The language has evolved so that it is easy for new concepts and relationships to be described in a sentence that can be formulated and understood by any native speaker."

In summary, there are ample reasons to consider having man-machine communication be achieved with languages similar or identical to natural human languages. The inefficiencies and awkwardness of machine coding are avoided, the flexibility and expressive power of natural languages are available, the enhanceability of natural languages permits novel expressions, the gory details of machine operation do not have to be understood or programmed by the user, and the training and experience which humans have with natural languages are effectively utilized, thus avoiding costly and time-consuming retraining of computer users. These _a priori_ reasons are substantiated by empirical results and trends in the computer industry, as higher level and more natural programming languages evolve.

Thus, in a sense, the extensive accommodation of the man to the machine has been recognized and is steadily diminishing. The machine is being called upon to accommodate more and more to the man's language in an attempt to bridge the gap or language barrier between man and machine. This is being accomplished not by basic changes in the internal structure of the computer, but rather by the use of translation equipment (compilers, and the like) intermediate between the computer and the user.

The present language barrier problem and solution techniques might by illustrated as in Figure 1. The computer user has a problem expressed in a language natural to the problem (English, mathematical expressions, and so forth). He must somehow translate this problem statement into a form interpretable by the computer's central processing unit. With the present conventional programming language compromise, he does so by writing a corresponding program in the programming language, i.e., he translates from natural to programming language. The machine's programming language processor (compiler, interpreter, or the like) translates this program into a corresponding sequence of basic assembly-language statements. These assembly language statements are in turn translated into basic machine-language operations.* The compilers and assemblers involved are costly and time-consuming to build. Thus, communication with the machine under the present compiling technique is a multiple step process, with associated inefficiencies and complications.

Natural Language → Programming Language → Assembly Language → Machine Language

Figure 1. - Multiple step translation between natural and machine languages

---

This present technique for man-machine communication thus represents an (admittedly rather useful) compromise solution to the language barrier between man and machine. The man must translate his natural-language problems into a high-level programming language, and the machine is responsible for translating from the programming language to its internal machine language [ref. 1, p. 299].

The trend toward more natural programming languages is easing man's translation process while complicating the machine's. More and more demands are being placed on the compilers and other language processors to achieve effective man-machine communication with natural-like languages. In this sense, the machines are being asked to learn and act more intelligently in their communication with men.

* In some cases, the operations of the compiler and assembler are combined, reducing the programming-to-machine-language translation to essentially one step.

The various forms of programming languages developed within this conventional compilation method of translating between man and machine languages are to a considerable extent a function of machine structures. Davis has, for example, pointed out that:

> "The emphasis on and use of procedure-oriented languages has probably been a direct result of the von Neumann concept of digital computers adopted universally by the computer design industry. This concept is, of course, characterized by the sequential procedure of producing a problem solution, incorporated bodily by designers of procedure-oriented languages [ref. 49, p. 125].
>
> "...procedure-oriented programming languages based upon the von Neumann sequential computer concept comprise almost the total set of existing programming languages [ref. 49, p. 128]."

This dependence of present programming languages upon the standard von Neumann concepts of "one organ for each basic operation" and sequential operation may prove a handicap to communication with machines. As the argument in section IV suggests, parallel and diffused-function operation may be most appropriate in the design of future machines, and for those new modes of operation new languages may be needed.

Newell, Shaw, and Simon [ref. 50, p. 41] have appropriately observed that "communication is limited by the intelligence of the least participant" and "the rise of effective communication between man and computer will coincide with the rise in the intelligence of the computer." Then if, in fact, Dreyfus and others are correct in asserting (cf. section IV) the need for new types of machines to obtain more machine intelligence, better communication between man and machine will result from the development of such new ("non-Boolean") machines. The present compiling methods are, in a sense, an attempt to achieve the same results with no such basic changes in machine structure.

An alternative solution to the standard multiple-step translation technique is to bridge the gap between natural and machine languages by eliminating (or drastically reducing) that gap. If one makes the machine language equivalent to, or at least close to, human natural languages, the problem of translation is simplified.

In other words, it appears that much of the need for compromise, retraining, and so forth, on the part of the man or machine could be eliminated if the machines were so designed that languages natural to them were similar to human natural languages. The simplification resulting from such new machines may be illustrated by comparing the simple one-step translation of Figure 2 with the multiple-step process of Figure 1. As the "machine's natural language" becomes progressively more like human natural languages, this translation process becomes progressively simpler.

Natural Language $\longrightarrow$ "Machine's Natural Language"

Figure 2. Proposed translation for non-conventional machines

This alternative solution thus involves building machines which are in some sense "linguistically equivalent" to humans. It would permit a human to communicate with the machine with little or no special training, thus making maximal use of the extensive training the human already has in a language, and opening up the communication channel with the machine to a wider user population. The computer may thus become a potential tool to people who are not interested in learning its mode of operation, its mechanical idiosyncrasies, and its particular details of structure.

However, the big question remains: How can such machines be designed and built? No answer is immediately forthcoming, but it appears that the successful development of such "natural-language computers" would be helped by extensive linguistic analysis of natural languages, formal linguistics studies, studies in artificially intelligent and "comprehending" machines [ref. 26], and general communication studies. Thus, it may be that construction of machines built around the primitives and structures of natural human languages will have to wait until more complete models of natural languages and the communication process are developed (e.g., ref. 49, p. 118). Perhaps. Perhaps not. This would be a question of <u>when</u> to have non-Boolean machines with such structures, not a question of <u>why</u> -- unless, of course, we consider the possibility that an adequate model of natural language is not forthcoming in <u>any</u> future, foreseeable or not. (This relates to the question of the likelihood of success in building needed non-Boolean machines, which will be considered in section VII.)

The point of the present argument is <u>why</u> have such non-Boolean machines. The arguments given in this section may be summed up as follows. Efficient, effective communication is facilitated by "higher level" languages in contrast to numerical machine coding. Also, humans use natural languages machines do not, and it is probably better for flexible man-machine communication and work-load sharing if the machine adjusts more than the man. The use of non-Boolean machines with natural-language-like structures would also eliminate the need for the present multiple-step, man-machine translation processes. These arguments, coupled with other justifications for natural language programming, strongly favor the development of such non-Boolean machines.

Thus, the generally acknowledged need for machine structures and languages which are suitable for natural-language man-computer communication favors the development of non-Boolean machines.

VI. THE ARGUMENT FROM COMPUTATIONAL ECONOMIES

In at least one sense all arguments favoring non-conventional machine structures could be stated as being based on either ultimate capabilities of machines or else some form of economies. Either a machine is capable (in the ultimate sense) of solving a problem or it is not. If it is not, then any need to solve such problems requires the development and use of other machines. However, if it is capable, there still may be justification for rejecting its use if it is uneconomical or inefficient, or if it does not provide solutions when needed and in appropriate form, and so forth. If machine A is more economical, efficient, "cost-effective," or easy to use, than machine B is, practical motivation exists for selecting machine A over machine B. This may, in fact, be the case with certain "non-Boolean" machines in preference to their conventional machine counterparts.

Thus, just because a digital computer <u>can</u> do almost anything doesn't mean it <u>should</u>. Universality of capability does not imply universality of utility or applicability. Pierce [ref. 1, pp. 300, 305] and Shannon [ref. 1, p. 308] have recognized and stressed this, as is evidenced by their arguments quoted in section II of this paper. Von Neumann [ref. 31, pp. 5-6] also recognized that certain machine structures (e.g., conventional analog and digital analyzers) were appropriate for some problems while not for others.

Studies of the efficient applicability of various machine types to certain problems continue to play a dominant role in the development and justification of new machines and in the optimization of problem solutions [refs. 51, 52, 53, 54]. For example, the editors of a general workshop on computer organizations have concluded that speed of operation (as a measure of computational economy) is <u>the basic justification</u> for new machine types:

> "Admittedly it is a rare job that cannot be performed on the common serial type computer but the price to be paid in terms of equipment and programming to perform operations in a reasonable time is in many cases tremendous. This leads us to the conclusion that the basic justification for new computer organizations is the same as the original motive for the von Neumann computer--speed of operation [ref. 55, p. iv]."

It is beyond the scope of this paper to discuss the many detailed arguments which could be (and, in many cases, <u>have been</u>) presented justifying the use of specific machine structures for selected classes of problems. The number of computer operations required to solve a specific problem, the computation time, the required memory capacity, the cost of computer time, the cost of equipment, utilization of mass production economies, the time and cost of programming, difficulty of understanding machine operation, enhancement of computational capabilities, and a myriad of other criteria may be considered in such studies. Recently, in particular, some such economic criteria have been discussed as justification for various parallel-processing computers [refs. 42, 51, 52, 55]. In general, such economic arguments frequently are shown to justify new, non-conventional machine structures.

It is interesting to note that this adjustment of machine structure to best match the problems being tackled generally results in special-purpose computers, in contrast to general-purpose "universal" machines. There is nothing inherently bad about such specialization, particularly in the light of the potential economic savings, but it is expected that many computer researchers will consider such a trend (toward special-purpose computers) as a retrogression in computer design. This need not be considered to be the case. It may turn out that having a number of special-purpose computers to solve several types of problems may, with modern technology, be more economical and advantageous than having a single general-purpose machine.

More importantly, it appears possible that universal machines can be built with entirely new structures (e.g., like Holland's iterative circuit computers) which could be more economical, in the overall sense, for handling most or all of the problems presently handled by digital computers. Thus, given two universal machines, such as the conventional general-purpose digital computer and a new "non-Boolean" machine of universal capability, one may, in the general sense, be more economical than the other.

Many researchers, including this author, believe that, considering the vast population of machine types possible, and the fact that present machines represent only a small portion of that population (and a portion which has resulted from the initial groping in the relatively new field of automatic computation), it is quite likely that new machine types, yet to be discovered and built, may turn out to be far more economical, efficient, and effective than present machines. This justifies the systematic search for new machine types, i.e., for non-Boolean machines.

VII. THE ARGUMENT FROM LIKELIHOOD OF SUCCESS

The previous arguments suggest ample justification for the development and use of non-conventional machine organizations. But to the practical-minded investigator such arguments favoring new structures would not be enough without some hope for being able to successfully design and implement these ideals into working systems. The argument from the likelihood of success, as it will be presented in this section, favors non-Boolean machines because of the definite potential for successful development of new types of computer structures.

In a weak sense, "non-Boolean" machines of some types have already been proposed, designed, and, in some cases, implemented. Machines which do not fit the pattern of the conventional digital computer include such types as analog analyzers, hybrid computers, parallel computers and iterative circuit computers (such as Holland's machines [ref. 52], SOLOMON machines [ref. 42], and ATHENE computers [ref. 56]), McCormick's pattern recognition machines [ref. 54] PERCEPTRONS, variable structure machines [ref. 51], and so forth. To varying degrees, these different machines exhibit definite differences from the structure and mode of operation of conventional digital computers. The conception, design, implementation, and even the programming and practical use of these machines have proven to be not so formidable problems as was originally expected. A vast number of devices utilizing advanced techniques of microminaturization, batch fabricating techniques, and the like, are now available. Indeed, it appears that Shannon's prediction [ref. 1, p. 310] that we will have the necessary hardware (and, to some extent, software) to implement theoretical conceptions of new machine types is being fulfilled.

However, to solve the intriguing problems of language translation, information retrieval, pattern recognition, game playing, theorem proving, and so forth, and to design "a computer whose natural operation is in terms of patterns, concepts, and vague similarities [ref. 1, p. 310]," drastic mutations in machine structure will apparently be needed. Such drastic changes will involve such concepts as: the "diffused function" operation mentioned by Shannon [ref. 1, pp. 309-310] and called for in section IV of this paper; the very general, real-time computation capability discussed in section III; the structural correspondence with biological counterparts discussed in section IV; and the natural-language, linguistically intelligent machines proposed in section V. Admittedly, these concepts require new insights, a better understanding of human information processing, and better models of cognitive processing, real-time computation, and natural languages. The question then is whether reasonably adequate models and insights are forthcoming, in either the near future, the distant future, or ever. There are no known ultimate theoretical barriers to achieving such models, and the rapid advances within cybernetics, linguistics, and computation theory in the past few decades show great promise for the successful development of adequate models and consequent machines in the foreseeable future.

As only one example of recent developments which may provide the needed technology for the implementation of new machine types, consider the potential of "domain tip propagation logic (DTPL) [ref. 57]." DTPL is a new thin-film technique for obtaining all-magnetic logic and memory. Information is stored in domains of reversed magnetization within controlled regions of low coercive force channels imbedded in a thin-film element of generally high coercive force. Propagation of such information-storing domains occurs at the domain tips, and is controlled by applied magnetic fields, resulting in microminiature high-bit-density storage and logic. It has been observed that:

> "The speed and directionality of propagation and the stray field strength of domain tips suggested that this mode of domain growth, properly controlled, is adaptable to a great variety of logical operations of a new and yet unexploited nature [ref. 57, p. 348]."

and that certain characteristics of DTPL "can lead to an enormous variety of possible device operations [ref. 57, p. 350]." Thus this new technique not only provides high storage and logic densities, good use of batch fabricating, and all-magnetic logic, it also shows a wide potential for new logic devices.

In particular, the effects involved in DTPL permit digital and some analog operations, some threshold effects, an apparent similarity between the propagation of domain tips and the propagation of nerve pulses in neural nets, interconnections between DTPL logic elements which are of the same material as the elements themselves, relatively slow domain tip propagation and potentially long-duration, minute-size delay units and shift registers. All these features are directly pertinent to bionic models which would use principles of biological counterparts to aid development of successful new machines.

The net result of the comments in this section is to display significant potential for achieving successful design and implementation of non-Boolean machines. When coupled with the previous arguments from experts in computation, ultimate capabilities, biological counterparts, natural-language programming, and computational economies, this potential for success encourages one to carefully consider the development of non-Boolean machines.

REFERENCES

1. Pierce, J. R., with Shannon, C. E., and Rosenblith, W. A .: What Computers Should Be Doing--Computers and the World of the Future, M. I. T. Press, Cambridge, Mass., 1962, pp. 290-325.

2. Dreyfus, H. L.: Alchemy and Artificial Intelligence. RAND Memorandum P-3244, Santa Monica, California, December, 1965.

3. Armer, P.: Attitudes toward Intelligent Machines -- Computers and Thought (E. A. Feigenbaum and J. Feldman, eds.). New York, McGraw Hill, 1963, pp. 389-405.

4. Davis, M.: Computability and Unsolvability. McGraw Hill, New York, 1958.

5. McNaughton, R.: The Theory of Automata: A Survey--Advances in Computers. vol. 2, pp. 379-421, 1961.

6. Turing, A. M.: On Computable Numbers with an Application to the Entscheidungsproblem. Proc. London Math. Soc., ser. 2, 42, pp. 230-265, 1963; see also correction in 43, pp. 544-546, 1937.

7. Goedel, K.: On Formally Undecidable Propositions of Principia Mathematica and Related Systems (Translated by B. Meltzer). Basic Books, New York, 1964.

8. Chomsky, N.: Formal Properties of Grammars--Handbook of Mathematical Psychology (Bush, Galanter, and Luce, eds.). vol. 2. Wiley, New York, 1963, pp. 323-417.

9. Lea, W. A.: The 'Spectrum' of Weak Generative Powers of Grammars. Mechanical Translation, vol. 9, No. 1, pp. 10-14, March 1966.

10. Church, A.: Introduction to Mathematical Logic. Princeton University Press, Princeton, N. J., vol. 1, 1956.

11. Nagel, E., and Newman, J. R.: Goedel's Proof--The World of Mathematics (J. R. Newman, ed.). vol. 3. Simon and Schuster, New York, 1956, p. 1695.

12. Taube, M.: Computers and Common Sense. McGraw-Hill, New York, 1961.

13. Putnam, H.: Minds and Machines -- Dimensions of Mind (Sydney Hook, ed.). Collier Books Division of Crowell-Collier Publishing Co., New York, 1961.

14. Scriven, M.: The Compleat Robot: A Guide to Andriodology -- Dimensions of Mind (Sydney Hood, ed.). Collier Books Division of Crowell-Collier Publishing Co., New York, 1961.

15. Arbib, M.: Brains, Machines, and Mathematics. New York, McGraw-Hill, 1964.

16. Rabin, M. O.: Real-Time Computation. Israel Journal of Math., vol. 1, pp. 203-211, 1963.

17. Yamada, H.: Real-Time Computation and Recursive Functions Not Real-Time Computable. IRE Transactions on Electronic Computers, vol. EC-11, pp. 753-760, 1962.

18. Yamada, H.: Counting by a Class of Growing Automata. Doctoral dissertation, University of Pennsylvania, Philadelphia, Pennsylvania, 1960.

19. Hennie, F. C.: On-line Turing Machine Computations. IEEE Transactions on Electronic Computers, vol. EC-15, pp. 35-44, February, 1966.

20. McCulloch, W. S.: The Imitation of One Form of Life by Another-Biomesis -- Biological Prototypes and Synthetic Systems. vol. 1, Plenum Press, New York, 1962, pp. 393-7.

21. MacKay, D. M.: Mindlike Behavior in Artifacts. Brit. J. for Phil. of Science, vol. II, pp. 105-121, 1951.

22. MacKay, D. M.: On Comparing the Brain with Machines. American Scientist, vol. 42, pp. 261-268, 1954.

23. MacKay, D. M.: The Use of Behavioral Language to Refer to Mechanical Processes. Brit J. for the Philos of Science, vol. XIII, No. 50, pp. 89-103, 1962

24. Babcock, M.: Some Physiology of Automata. Proc. Western Joint Computer Conference, pp. 291-298, 1961.

25. Feigenbaum, E. A., and Feldman, J.: Computers and Thought. McGraw-Hill Book Co., New York, 1963.

26. Kochen, M., MacKay, D. M., Maron, M. E., Scriven, M., and Uhr, L.: Computers and Comprehension. RAND Memorandum RM-4065-PR, April 1964.

27. Shannon, C. E., and McCarthy, J.: Automata Studies. Princeton University Press, Princeton, New Jersey, 1956.

28. Wooldridge, D. E.: The Machinery of the Brain. McGraw-Hill, New York, 1963.

29. Turing, A. M.: Computing Machinery and Intelligence--Computers and Thought (Feigenbaum and Feldman, ed.). McGraw-Hill Book Co., New York, pp. 11-35, 1963.

30. von Neumann, J.: Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components (C. E. Shannon and J. McCarthy, eds.). Automata Studies, Princeton University Press, Princeton, N. J., pp. 43-98, 1956.

31. von Neumann, J.: The Computer and the Brain. Yale University Press, New Haven, Conn., 1958.

32. McCulloch, W. S.: Biological Computers. IRE Trans. Electronic Computers, vol. EC-6, No. 3, pp. 190-192, September 1957.

33. McCulloch, W. S.: Commentary -- Communication: Theory and Research (L. Thayer, ed.). Charles C. Thomas, Publisher, Springfield, Illinois, 1964, Ch. XVI.

34. Davis, H.: Peripheral Coding of Auditory Information -- Sensory Communication (W. A. Rosenblith, editor). M.I.T. Press, Cambridge, Mass., 1961, pp. 119-141.

35. McCulloch, W. S.: Reliable Systems Using Unreliable Units--Disorders of Communication. vol. XLII, Research Publications, Association for Research in Nervous and Mental Disease, 1964, pp. 19-28. (A good list of references on reliable computation with unreliable components is included.)

36. McCulloch, W. S., Arbib, M. A., and Cowan, J. D.: Neurological Models and Integrative Processes--Self-Organizing Systems-1962 (M. Yovitts, G. T. Jacobi, and G. P. Goldstein, eds.). New York, Spartan, 1962.

37. MacKay, D. M.: Interactive Processes in Visual Perception -- Sensory Communication (W. Rosenblith, ed.). M.I.T. Press, Cambridge, Mass., pp. 339-355, 1961.

38. Kilmer, W. L., and McCulloch, W. S.: Toward a Theory of the Reticular Formation. Fifth National Symposium on Human Factors in Electronics, pp. 7-25, 1964.

39. McCulloch, W. S., and Kilmer, W. L.: Introduction to the Problem of the Reticular Formation. Automata Theory, Academic Press, Inc., New York, 1963, pp. 269-277.

40. Schmitt, O. H.: The Design of Machines to Simulate the Behavior of the Human Brain. IRE National Convention, 1955 Symposium, pp. 240-255, 1955.

41. Holland, J.: A Universal Computer Capable of Executing an Arbitrary Number of Sub-programs Simultaneously. Proceedings of the Eastern Joint Computer Conference, 1959, pp. 108-113.

42. Slotnick, D. L., Borck, W. C., and McReynolds, R. C.: The SOLOMON Computer -- A Preliminary Report. Proc. of 1962 Workshop on Computer Organization. Spartan Books, Inc., Washington, D. C., pp. 66-92, 1963.

43. Gibson, J. J.: The Problem of Temporal Order in Stimulation and Perception. J. Psychol., vol. 62, pp. 141-149, 1966.

44. vonFoerster, H.: Memory without Record -- The Anatomy of Memory, D. P. Kimble (ed.). Science and Behavior Books, Palo Alto, pp. 388-433, 1965.

45. Young, J. Z.: The Organization of a Memory System. Proc. Royal Soc., vol. 163, pp. 285-320, 1965.

46. Green, B. F., Wolf, A. K., Chomsky, C., and Laughery, K.: BASEBALL-An Automatic Question Answerer -- Computers and Thought (E. A. Feigenbaum and J. Feldman, eds.). McGraw-Hill, New York, 1963, pp. 207-216.

47. DEACON Project: DEACON Breadboard Processing. Report AD-612 165, TEMPO Division, General Electric Co., Santa Barbara, California, 1964. (Available through Clearinghouse, U. S. Dept. of Commerce, Springfield, Virginia.)

48. DEACON Project: Phrase-Structure Oriented Targeting Query Language. Final Report, vol. 1, Contract AF30(602)-3568, TEMPO Division, General Electric Co., Santa Barbara, California, September 1965.

49. Davis, R. M.: Programming Language Processors -- Advances in Computers. vol. 7, 1967, pp. 117-180.

50. Newell, A., Shaw, J. C., and Simon, H. A.: Chess-playing Programs and the Problem of Complexity--Computers and Thought (E.A. Feigenbaum and J. Feldman, eds.), McGraw-Hill, New York, 1963, pp. 39-70.

51. Borck, W. C., Gregory, J. C., Hudson, J. R., and Murthy, J. C.: The Design of a Variable Structure Parallel Network Computer. The Proceedings of the Symposium on Prospects for Simulation and Simulators of Dynamic Systems, Books, Inc., New York, August, 1967, Ch. 14.

52. Holland, J.: Iterative Circuit Computers. Proc. Western Joint Computer Conference, San Francisco, 1960, pp. 259-265.

53. McCormick, B. H., and Narasimhan, R.: Design of a Pattern Recognition Computer with Application to the Automatic Scanning of Bubble Chamber Negatives. Report file No. 463, Univ. of Illinois Digital Computer Laboratory, Urbana, Illinois, July 1962.

54. McCormick. B. H.: The Illinois Pattern Recognition Computer. Report No. 148, Univ. of Illinois Digital Computer Laboratory, Urbana, Illinois, August 1963.

55. Barnum, A. A., and Morris, A. K.: Workshop on Computer Organization. Spartan Books, Inc., Washington, D. C., 1963.

56. Schwartz, J.: Large Parallel Computers, Journal of ACM, vol. 13, pp. 25-32, January 1966.

57. Spain, R. J.: Domain Tip Propagation Logic. IEEE Transactions on Magnetics, vol. MAG-2, no. 3, pp. 347-351, September 1966.

Electronics Research Center
National Aeronautics and Space Administration
Cambridge, Massachusetts, December 1967
125-23-02-31