U. S. AIR FORCE

PROJECT RAND

RESEARCH MEMORANDUM

PARALLEL CONTROL

John Nash

RM-1361

27 August 1954

Assigned to _____

The RAND Corporation
1700 MAIN ST. • SANTA MONICA • CALIFORNIA

SUMMARY

Some ideas are presented for new designs of the control system in high-speed digital computers. The basic idea is to decentralize control with several different control units capable of directing various simultaneous operations and intermediating them when appropriate.

PARALLEL CONTROL

John Nash

This memorandum concerns some ideas for new designs of the control system in high-speed digital computers. The ideas are yet in an immature and rather unspecific form, but this is a subject that deserves some attention and thought for the future.

Indeed, the idea is more or less futuristic, and is more appropriate for the "electronic brains" of the future than for the computers now used, or under construction, or even planned. The basic idea is simple. Instead of having a single control unit sequencing the operations of the machine in series (except for certain subsidiary operations as certain input and output functions) as is now done, the idea is to decentralize control with several different control units capable of directing various simultaneous operations and interrelating them when appropriate.

Let us consider some of the advantages of this sort of development of computing (and data processing) machines.

1. Speed of computation

This would apply mainly to problems admitting computations in parallel. One cannot expect the speed with which, say, a multiplication or a memory access, can be performed to increase indefinitely. So greater speed of computing will come most strongly from being able to do many of these operations at once.

Some argue that having a big machine that can do a problem fast is not economically advantageous over having several slower small machines. But isn't it much better to have one machine that takes a day for a problem than 100 which take 100 days for a problem?

Of course, this isn't the most important point in the speed question. The more important "intelligence" quality of a fancy machine with a good internal programming library will be mentioned later.

2. Expansibility

A machine based on parallel control can be designed so that it can be readily expanded, more or less indefinitely, by the addition of more arithmetic, memory, control, etc. units. This could be a very desirable feature especially when a valuable library of sub-routines, etc., had been built up.

3. Self-Maintenance

A parallel control machine could locate points in itself needing repair. It would tend not to be completely incapacitated by any single material failure.

4. Ease of programming, the "intelligence" quality, ability to handle complex problems efficiently.

We could define the intelligence of a machine in terms of the time needed to do a typical problem and the time needed

for the programmer to instruct the machine to do it. Suppose a human can do a problem in 3 hours and the machine in 1 minute but programming takes 5 hours. Then the machine has no advantage for the isolated problem, and would be worth while only when many similar problems are to be done. Simplifying the programming requirements so that the machines can take less explicit instructions is extremely important for the development of the art. This does not require special design of the machine to take instructions more readily but rather the development of interpretative programs by which translation from more abstract into more explicit instructions can be effected by the machine itself. This process is capable of indefinite extension. Its cost is in machine time, either spent in constructing an explicit program before computing or in translating during computation from abstract to explicit instructions. Here parallel control offers a distinct advantage since it permits the translation to occur simultaneously with the computation, indeed several different stages of translation might be occurring simultaneously.

Before saying any more about the advantages of parallel control we should give a more concrete idea of a design for a parallel control computer. Consider a machine having

a) logical units (control units)

b) arithmetical units

c) fast memory units

d) slow memory units

e) input units

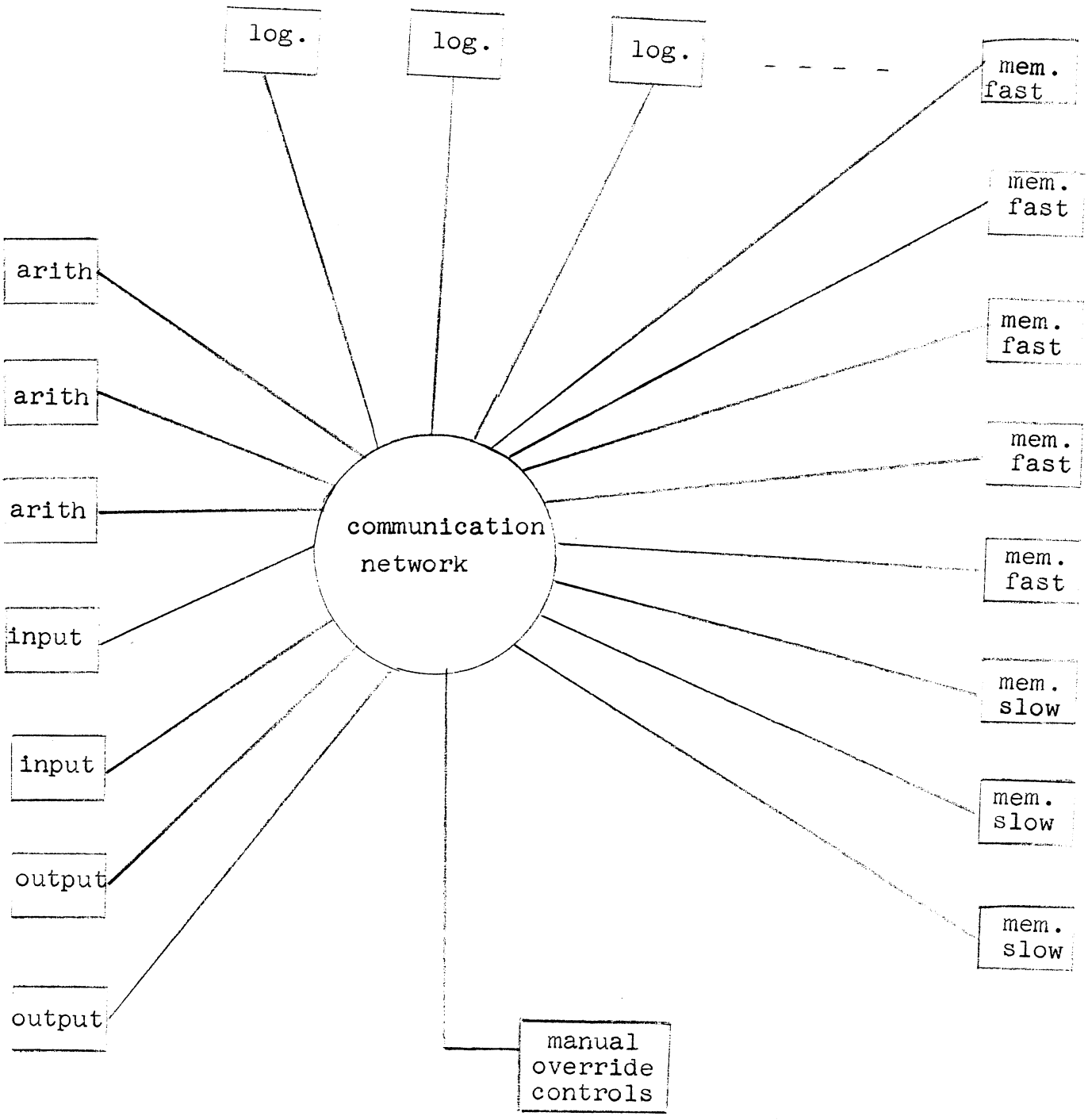
f) output units

g) the communication network.

The more routine aspects of control are separated from the rest. Each arithmetic unit is, for example, to have the capacity to follow a sequence of orders stored in the fast memory so long as no decisions are required. A decision is made when the sequence of orders followed depends on the result of following them to some point where two or more possible continuations would exist, the continuation followed depending on that result. Division should not be a decision requiring process; the arithmetic unit should divide in response to a single order.

When a decision must be made the relevant information is referred to a logical unit. Also the logical units should handle all information processing that is not properly an arithmetical operation or an input or output operation. In this category would fall program translation, etc. The design of the machine should make it unnecessary for a logical unit to refer to an arithmetic unit in performing any purely logical task, such as manipulating orders, modifying programs, searching the memory.

A block diagram of such a machine would be like this:


log.
log.
log.
mem. fast
mem. fast
mem. fast
mem. fast
mem. fast
mem. slow
mem. slow
mem. slow
arith
arith
arith
input
input
output
output
communication network
manual override controls

For different applications, the balance between the numbers of units of various types would be different. It would generally be desirable to have fairly many fast memory units to minimize waiting in line for access.

It is quite likely that for certain purposes, one would want specialization within the classification "logical unit" or the other types of units. For example, for a translating machine an arithmetical unit is pointless but specialized input readers would be valuable.

Now let us consider what functions the various types of units should perform.

1. Arithmetic Unit. Should have the capacity to obey commands to:

a) perform an arithmetic operation

b) request a specified number from a specified memory unit and place it for operation

c) send out the result of an arithmetic operation to a specified memory unit to be stored in a specified location

d) refer to a specified memory locus for a new command, and then continue in canonical sequence obeying the commands stored at following memory loci:

e) be responsive (or unresponsive) to an overriding command transmitted from a logical unit,

f) stop,

g) "report" to specified logical unit.

2. Fast Memory Unit. The fast memory units are to be passive and respond to direct requests for specified stored numbers from other units or to direct instructions to store a transmitted number at a specified locus. So their functions are described as functions of the other units.

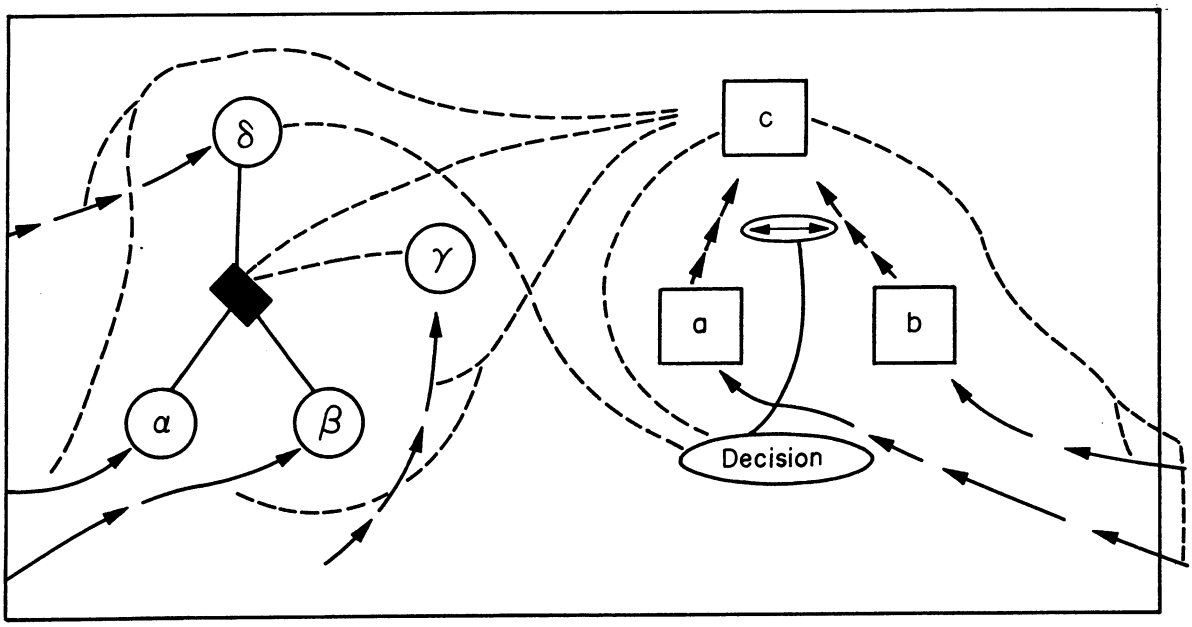
3. Slow Memory Unit. Obeys commands in the same manner as an arithmetic unit but has different functions, these are to read from a fast memory a specified number or a specified sequence of numbers in canonical order or to do the reverse, sending numbers into a fast memory.

4. Input and Output Units. Follow commands in the same manner as an arithmetic unit and act between fast memory units and forms of communication with the exterior of the machine.

5. Communicating Network. This apparatus is analogous to an automatic telephone system. It must transmit orders and data from one unit to another as directed by the unit sending the order or direction. It must have a provision for dealing with two or more messages directed simultaneously to the same unit so that only one is permitted at a time. If a unit is delayed in getting its message through, it waits until the message goes through unless it is acting under a command that instructs it to some alternative in such a case. This is a "decision," so this would apply only to a logical unit. All other types would wait.

6. Logical Unit. In this system all powers not delegated to the other types of unit are reserved to the logical units. They may be considerably more richly equipped with powers to perform various specific logical operations than are the control units of current machines. A logical unit should have enough basic operations so as not to require the help of an arithmetic unit in building up (efficiently) more complex or special logical functions or decisions. The writer cannot give anything close to an optimal or efficient set of basic operations now, but can illustrate a set that gives an idea of how the logical units might function.

Assume a logical unit can contain at one time seven numbers (which term will include data in the form of a set of binary digits regardless of whether or not it is to be interpreted as a number) stored at places a, b, c, α, β, γ, δ. The numbers α, β and δ are to be thought of as data. a, b, and c are commands. γ is partly like data and partly like a command, as we shall see. It can be thought of as a subsidiary command sometimes given meaning by c. The command c is the operating command determining the current operation of the unit.



Schematic of Logical Unit Function

Dotted lines indicate influence or control. c and δ control the decision on which command a or b is to succeed c, if either. This is the basic decision operation. c and γ control the production of a number δ as a logical function of α and β. Think of c as commanding that a logical function of a certain type be computed from α and β and of γ as determining the specific function of that type.

We must assume here (really this is an assumption for simplicity) that numbers are longer than addresses so that a number can contain both instructions and a memory locus. Consider the following set of commands (c is the command):

A) Replace c by the number in memory locus _____.
(When no rule for replacing c is given explicitly in a command c is replaced by the next command in canonical sequence.)

B) Replace c by a if δ has property_____, otherwise by b. The property of having first digit 1 would be enough, but it might be more efficient to have some other decisions built in as well.

C) Compute the logical function of type _____ described by γ of the numbers α and β and place the result as δ.

D) Send δ to memory locus _____.

E) Send δ to arithmetic unit _____ as an overriding command.

F) Send δ to logical unit ____ etc. E.g., to other unit types.

G) Send δ to logical unit _____ as a datum. (As distinguished from a command)

H) Be responsive to a command received from another logical unit.

I) Accept a command received from another logical unit for storage as a.

J) Reject commands received.

H, I, J are commands that remain in effect until countermanded (by H,I, or J). When a number δ is sent to another unit and rejected by that unit the sending unit must follow command b, otherwise the next command in sequence as usual.

K) Be receptive to a datum sent from another logical unit and store it as β.

L) Be unreceptive.

M) Place the number in memory locus _____ in position as α

M') .....as β

M") .....as γ

M"') .....as a

M"") .....as b

N) Stop being responsive to commands or reports

O) Be responsive to report from an arithmetic or other lower unit. Such a report contains the address of a command and the logical unit obtains that command from memory and places it as c.

P) Be unresponsive to reports.

Now we can consider what logical functions should be included under (c).

c1) Produce δ as the number which has the same digits as α in the places where γ has 1's and the same as β where γ has zeros.

c2) Produce δ as the number which has 1's where α and β have the same digit and zeros where they differ.

c3) Produce δ by permuting the digits of α where γ has 1's one step cyclically

c4) Produce δ by having a zero where γ has a zero and where γ has a 1 by having a 1 unless the corresponding digits of both α and β are 1's. (Non-Conjunction)

The attempt at a specific illustration above is, of course, nothing more than that. It's just to give a picture of the functions contemplated for logical and arithmetic and other units in a parallel control machine.

Of course, decision making does not have to be set up as a specific function in a logical unit. Decisions can be made by taking data numbers, transforming them into order numbers, and using them as orders. The programming could be set up to use this device, which would be slower.

The computation of logical functions could be separated from the other functions of a logical unit. It is not clear whether or not this would be advantageous.

Ultimate Advantages

Perhaps this type of machine organization would have the most value for very large machines intended to have a wide range of applicability. Consider a large machine which is to be built at a time (in the future) when the costs of computation have decreased, when very large fast memories are feasible, when printed circuits etc., can be mass produced automatically. When such a development exists one will want the human labor by mathematicians and programmers needed in solving a problem reduced to minimum. The mathematician should have to do little more than state the problem and general method of computation to the machine in almost ordinary mathematical style. The machine should have considerable interpretative capacity and should itself

work out the program of computation, descending level by level of abstraction to the level of specific orders and storage patterns for the computation.

To do this sort of interpretative work the machine must have a quite large general interpretation program stored within it in readiness for new problems. In principle current machines could do this, though not necessarily very efficiently or rapidly, if they had a sufficiently developed program for the purpose stored in their memories. But an interpretative program adequate to essentially eliminate programming would be too much for their memories. Now we can see how the large parallel control machine capable of handling several problems simultaneously would have an advantage. The interpretation program would only need be represented once. So a big advantage of one big machine is that it can afford to store much more basic information, and especially store more with rapid accessibility, than could any one of a collection of smaller machines.

If logical units could be made cheap enough, and the same for memory, one could afford to have trial and error and search processes used in the machine. Trial and error processes and learning processes (which would require a lot of memory) would be helpful in developing high order interpretative capacity. Ultimately trial and error processes, combined with search or association processes, abstraction processes, and learning or conditioning processes should lead to the learning machine or even to the genuine thinking machine. For this development parallel logical operation will certainly be important.

It is interesting to consider what a thinking machine will be like. It seems clear that as soon as the machines become able to solve intellectual problems of the highest difficulty which can be solved by humans they will be able to solve most of the problems enormously faster than a human.

In closing, the human brain is a highly parallel setup. It has to be.