

*Prof. van Neumann,  
with the compliments of  
the authors.*

*g.s. booth*

GENERAL CONSIDERATIONS IN THE DESIGN OF AN ALL PURPOSE ELECTRONIC DIGITAL  
COMPUTER.

by

ANDREW D. BOOTH

and

KATHLEEN H. V. BRITTEN

CONTENTS

SECTION

PAGE NUMBER

0.00 Introduction.

0.10	Projected machines.	1.
0.20	General principles.	2.
0.21	Organisation.	2.
0.30	The scale of notation.	3.
0.40	The input and output.	3.

1.00 The memory.

1.01	Desirata.	5.
1.10	Macroscopic elements.	5.
1.20	Scan v.s. switching methods.	5.
1.21	Combinatorial memory.	6.
1.30	Thermal memories.	7.
1.40	Optical memory.	7.
1.41	Kerr effect as selectron mechanism.	7.
1.50	Acoustic memory.(serial)	8.
1.51	Acoustic memory.(parallel)	9.
1.60	Electrical memory.	9.
1.61	The medium.	10.
1.70	Magnetic memory.	10.
1.71	Magnetic memory for parallel operation.	10.
1.72	Permanence.	10.
1.80	Electronic memory.	10.
1.90	General conclusions.	11.
1.91	Input output.	11.
1.92	The intermediate memory.	12.
1.93	The high speed memory.	12.
1.94	The memory for orders.	12.

2.00 The arithmetic unit.

2.10	The binary point.	13.
2.20	Negative numbers.	13.
2.30	Addition.	14.
2.31	Dynamic and static accumulators.	14.
2.40	Subtraction.	14.
2.41	Subtraction by complements.	14.
2.42	The inverted carry over.	15.
2.50	Multiplication.	15.
2.51	Negative numbers.	15.
2.52	Extended accuracy.	16.
2.60	Division.	16.
2.61	The iterative method.	16.
2.62	The direct method.	16.
2.70	Extraction of the square root.	17.
2.71	The iterative method.	17.
2.72	The direct method.	17.

Gifford - 3 20 57 Von Neumann

3.00 The control.

3.10	Synchronous versus asynchronous operation.	18.
3.20	Specification of orders.	18.
3.30	The control register.	19.
3.40	The control counter.	19.
3.41	Non singular operation.	19.
3.42	Singular operation.	19.
3.50	The decoder-coder.	19.
3.60	The orders.	20.
3.61	Basic order set.	20.
3.62	Contracted notation.	21.
3.70	Extended order set.	21.

## GENERAL CONSIDERATIONS IN THE DESIGN OF AN ALL PURPOSE DIGITAL COMPUTER

### 0.0 Introduction.

The first conception of a calculating machine, capable of executing in a completely automatic manner, a sequence of orders previously given to it, appears to have been due to Charles Babbage. This machine was not completely realised on account of its essentially mechanical nature and the technological imperfections of the engineering processes available in 1830. However, despite this non-realisation, Babbage's plans contain all the logical elements required for the construction of such a computer.

Little progress was made towards the actual construction of a general purpose calculator until the late 1950's, when two projects were initiated in the U.S.A. These were:

- 1) The A.S.C.C. at Harvard.
- 2) The E.N.I.A.C. at Pennsylvania.

The A.S.C.C. is essentially a large Hollerith machine with provision for multiplication and division, and for logical programming. As a Hollerith type machine it is satisfactory, but the programming is complex and the execution of orders slow. For example, multiplication takes a time of the order of ~~five~~ seconds. The memory is indefinitely large but very slow as it consists of punched cards.

The E.N.I.A.C. is the first working electronic machine, but is limited in application since its programming is very complicated, and its internal fast memory is only 20 numbers. Practically, the speed of operation of the E.N.I.A.C. is equal to that of its punched card input ( 400 cards per minute ), since all the electronic processes of the machine are considerably faster.

### 0.1 Projected machines.

It will be seen from the above brief survey, that existing machines are limited in application, either by their slow overall speed ( due to electro-mechanical elements ), or to their limited high speed memory capacity. As a general principle, it can be stated that it is useless to increase the speed of functional units beyond one order of magnitude faster than that of the data input. Thus, for a relay type machine, a multiplication time of much less than 1/100th. sec. is unjustified since punched tape or card data feed is slow compared with this.

Machines at present under discussion or construction, attempt to meet this logical desideratum, and are of two main types:

- 1) Serial operation machines.
- 2) Parallel operation machines.

Of the first type may be mentioned the E.D.V.A.C. and its British copy A.C.E., and of the second, the Princeton Computer project and our own machine. The distinction between the two types is that, in 1) the digits of a number are available one at a time in strict sequence, whilst in 2), all the digits of any number become available at the same time.

It follows that if the minimum reaction time of electronic elements is utilised in both cases, 1) will be slower than 2) by a factor equal to the number of digits in the number considered.

Although, at first sight, it might appear that the design of type 2) machines would be the most complicated, this is, in fact, not the case, since the control needed to execute, for example, multiplication, is much simpler than that required in type 1).

## 0.2 General principles.

The foregoing considerations make it apparent that there are two general methods of designing a fast computing machine.

- 1) A large high speed memory and a few distinct and simple arithmetic units.
- 2) A slower large memory and a number of identical arithmetic units.

In the first case use is made of the fact that any calculation can be broken up into individual parts each of which can be performed separately and in sequence; whilst in the second case many parts of a calculation are performed simultaneously. As an example, consider the calculation of:

$$(ay^2 + by + c)/(dy + e).$$

A type 1) machine would follow this sequence:

- 1) e to product register.
- 2) Form  $d.y$  and add to e in product register.
- 3) Store  $dy + e$  in memory.
- 4) Form  $y^2$  and transfer to multiplier.
- 5) Form  $ay^2$  and leave in product register.
- 6) Form  $by$  and add into product register.
- 7) Add c into product register.
- 8) Divide contents of product register by  $dy + e$ .

whereas a type 2) machine would perform the operations (1-3) and (4-7) in two simultaneous groups and then combine the results in 8), or possibly generate  $ay^2$  and  $by$  simultaneously. Thus the second procedure might effect a saving of time by a factor of 2-3, which could then be set against a memory of proportionate slowness.

It is evident that multiple operations lead to increased complexity, and that the ideal machine would be of type 1).

## 0.21 Organisation.

A study of the principles underlying the basic computing sequences of a few typical problems, makes it evident that the machine must have the following internal constitution:

- 1) A memory.
- 2) A control.
- 3) An arithmetic unit.

Practical considerations, to be discussed in 1.01, show that the memory must consist of two parts:

- 1)a. The internal, high speed memory.
- 1)b. The external, and indefinitely large, storage memory either on tape or on some faster medium.

It is essential that the secondary memory should be available to the machine without the intervention of the human operator.

The control is essentially a means of directing the execution of orders inserted into the machine and must be capable of exercising a certain amount of independent judgment of the type:

"If  $X \geq a$  follow sequence  $D(X)$ , but if  $X < a$  follow sequence  $E(X)$ ."

Here  $X$  is a number resulting from the operations of the machine, and is not known, a priori, to the human operator.

The arithmetic unit must consist of functional units adequate for the performance of at least the operations  $+$ ,  $-$ ,  $\times$ . The inclusion of a unit for division is desirable but not essential, since iterative processes of rapid convergence are available for its programming in terms of the other operations; a similar remark applies with even greater force to the operation  $\sqrt{\quad}$ .

### 0.3 The scale of notation.

Up to this point, no mention has been made of the scale of notation in which the numbers, used by the machine, will be represented. Machines already constructed (0.1) use the standard decimal system both internally and externally and it would appear, in view of the universal use of this scale, that any new machine should conform.

In fact, the scale of notation used inside the machine is almost completely dictated by the available types of memory organ, and, as will appear from later discussions, all present techniques are suited only to the binary, or scale of two, notation.

The attractiveness of binary scale, from the electronic viewpoint, lies in the possibility of using amplitude insensitive elements. Thus the binary counter, or 'flip-flop', is reliably operable with pulses of almost arbitrary shape ( up to direct current ), and having a wide variation in amplitude.

Apart from the high speed memory, binary notation makes possible the use of magnetic tape or wire as the external slow memory and, although slow compared with electronic speeds, this is considerably faster than punched tape or card storage.

### 0.4 The input and output.

The use of binary notation inside the machine raises the question of the means of communication to and from the outside world. It would, of course, be possible to make all conversions to and from binary scale before and after the presentation of a problem to the machine, so that no special provision would have to be built into the latter. This would be quite justifiable if the total output and input was very small compared with the data generated by the machine in the course of its operations, and never communicated to the outside world. In a large class of problems, however, a very considerable amount of initial data has to be inserted ( e.g. matrix elements, and boundary values in the solution of partial differential equations ), and this suggests the desirability of providing special conversion facilities.

Two methods of conversion are immediately suggested:

- 1) A special ad hoc machine or unit.
- 2) The programming, by the machine, of its own conversions.

The first alternative can be ruled out on the grounds of cost and complexity, but an examination of the second shows it to be perfectly feasible and satisfactory in a high speed machine. This is because, at some stage in the preparation of a problem, the human operator must write (i.e. type) the data on to a physical medium, this operation is the limiting factor of the primary input speed and, so long as the machine can make its own conversions in a less time than that taken by the typist, no additional speeding up is necessary. It will be shown, in a later section of this report, that this speed criterion is realisable.

## 1.0 The memory.

It was indicated in the introduction, and may now be stated as a general principle, that once the form of the high speed memory has been decided, most of the other components of an electronic computer become semi-invariant. We shall review, in this section, the various possible memories and indicate our final choice. Only binary memories will be discussed, since no other type is in a sufficiently advanced state of development to justify its consideration.

### 1.01 Desirata.

In order of importance, a high speed memory must be capable of:

- 1) Receiving digital information at speeds of the order of 100 K.C./sec.
- 2) Emitting data at the above speed.
- 3) Retaining its information for electronically long periods ( i.e. of the order of 1 second.)
- 4) Having any particular number erased and replaced at similarly high speed.

Theoretically it should have infinite capacity in both directions, i.e. be capable of holding an infinite number of numbers each having an infinity of digits. In practice, neither of these desirata is realisable, and a compromise has to be made. Examination shows that, to solve many interesting problems, a capacity of from  $10^3$  to  $10^4$  numbers is required. The number of digits in each number is a more speculative matter. In problems which involve the inversion of a matrix the governing factor of the final accuracy is the total number of multiplications and divisions involved, and in even moderately large procedures the rounding off errors may amount to  $10^6-10^8$ . Thus, in order to ensure an accuracy of 1/100 % in the final answer, numbers of total precision  $10^{12}$  must be used; in binary notation this means  $2^{39.8}$  or approximately 40 binary digits.

### 1.10 Macroscopic elements.

The considerations of 1.01 enable an immediate disposal to be made of any idea of using macroscopic binary elements for the memory. Thus, to enable  $10^3-10^4$  binary numbers of 40 digits to be stored,  $2.40.10^3-10^4$  or  $10^5 - 10^6$  half elements would be required, which is quite impracticable.

### 1.20 Scan v.s. switching methods.

Having rejected the idea of using macroscopic elements, there still remains the discrimination between two possible methods of using microscopic memory elements. Suppose for convenience, that we consider a single digit  $n_1$  of any number  $n$ , and suppose the memory elements for this digit of each of  $N^2$  numbers to be arranged on a square lattice :

$$\begin{array}{cccc} l_1 & z_1 & \text{-----} & N_1 \\ - & - & n_1 & - \\ - & & & \\ - & - & - & N_1^2 \end{array}$$



The physical nature of each picture must be such that:

- 1) It emits an impulse when all its impressions are stimulated.
- 2) It emits no impulse if only part of its impressions are stimulated.
- 3) It emits no impulse if any part other than an impression is stimulated.

Although such a scheme would be difficult to realise with present facilities, it should be borne in mind for future development.

#### 1.30 Thermal memories.

No extensive development of thermal type memories has been made. It is evident, however, that the thermal retentivity of various materials, such as chalk, could form the basis of a memory. Most mechanisms of this type are rather slow as they depend on conduction processes so that, unless some indicator of low thermal capacity were used, the memory would be too slow for electronic use. An additional disadvantage lies in the fact that no rapid, and at the same time local, cooling process is available for erasing data.

#### 1.40 Optical memory.

The most obvious application of optical methods to the development of a memory organ would be the use of sound track from cinematograph film. This would give a serial type memory of comparatively low speed and, in view of the fact that a development process has to be interposed between recording and reading off data, and also that once exposed film is no longer usable for the recording of new data, no great investigation has been carried out.

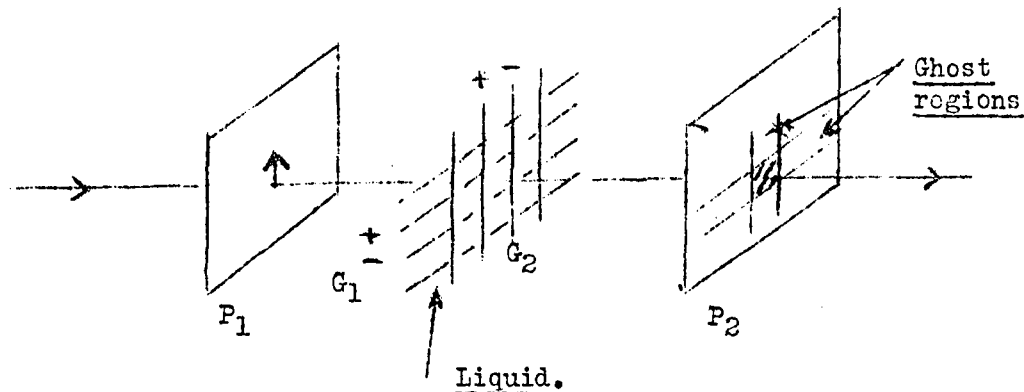
One point is worthy of mention in connection with photographic memories for binary numbers. The normal photographic process is capable of giving half tones, i.e. of giving more information than that required for a purely binary memory. For scale of two black and white would suffice, and it is possible that some photographic process might be found, in which exposure to light of a given wavelength produced a blackening which could be read in some region of the spectrum to which the film was insensitive. This would obviate the necessity of a separate development process.

To carry the process to its logical conclusion, a filmic medium might be found such that exposure to light of frequency  $n_1$  changed it from a transparent state  $s_1$  to a blackened state  $s_2$ , whereas exposure to light of a different frequency  $n_2$  produced the reverse transformation.

Alternatively, the medium might have the property that, under the action of light, the reversible reaction  $s_1 \rightleftharpoons s_2$  took place, such that  $s_1$  was the equilibrium for zero intensity and  $s_2$  that for high intensity. If the state  $s_2$  could be retained at zero intensity by some quenching process, the conditions for use as a binary memory would exist.

#### 1.41 Kerr effect as selector mechanism.

In order for a two dimensional memory of photographic type to have adequate speed, and at the same time not have the complication of electronic scanning with its attendant vacuum, it is necessary to have some kind of switching selection of the type envisaged in 1.20.



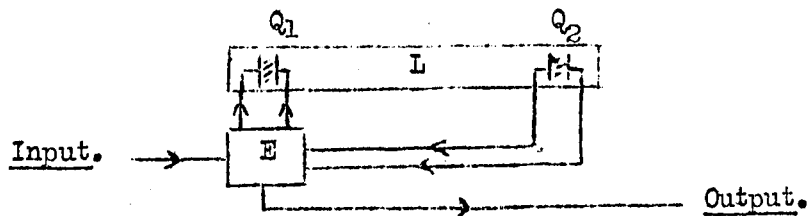
Suppose two sets of parallel wires  $G_1$  and  $G_2$ , mutually perpendicular in direction, to be immersed in a bath of liquid having the Kerr electro-optical property that, in an electrostatic field, the plane of polarisation of transmitted light is rotated. If, with no field applied, two polaroid sheets are arranged for extinction, then, on two adjacent wires in each grid being electrified, the system will transmit light in two bands whose intersection will be of roughly double the intensity of either. This system could be used to sensitise the binary medium and, by placing a photocell behind the medium, could also be used to read off the data. The response of Kerr cells is of the order of  $10^9$  cycles/sec.

Some use might also be made of the fact that, in solid media, the Kerr effect persists for some time (up to 30 secs.) after the removal of the field. This could be made the basis of a memory in its own right.

#### 1.50 Acoustic memory (Serial).

From the essentially dynamic nature of sound phenomena, it follows that the basic acoustic memory would be of the serial type. Considerable development has been carried out on suitable systems for use in conjunction with the E.D.V.A.C. and A.C.E.

The principle of operation is the following, a set of electrical impulses, representing the digit pattern of the number to be remembered, is applied, via suitable electronic apparatus  $E$ , to a quartz crystal  $Q_1$ .



The resulting vibrations are transmitted through a column of liquid  $L$ , (usually mercury), and are picked up by another crystal  $Q_2$ .  $Q_2$  feeds back into  $E$  so that the pulse pattern is preserved.

The number can be read out serially, and either erased from the memory or retained as desired.

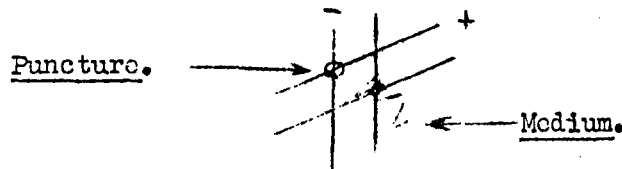
Types of delay at present working have a capacity of the order of 1000 binary digits, and a cycling time of about 1/1000 sec.

### 1.51 Acoustic memory ( Parallel ).

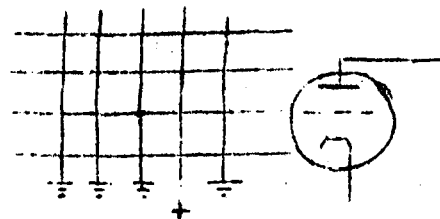
By arranging, say 40, delays in parallel, and placing one of the digits of each number in each line, it would be possible to use the system described in 1.50 for a parallel operation machine. The disadvantage is, however, that the average time required for the location of any number is 1/2000 sec., which is slow compared with other devices to be described later. In addition the control, required to produce synchronism between the various lines, would be complicated.

### 1.60 Electrical memory.

A purely electrical binary memory is the following.



Two parallel grids of parallel wires, in directions at right angles, are separated by a "medium" which is normally an insulator. When a wire of the upper grid is attached to a source of high potential, and a member of the lower grid is earthed, the medium breaks down at the point of closest approach. At any subsequent time the fact of breakdown can be read off, for a particular pair of wires, by earthing all wires in the upper grid save that which is to be sensed and to which a positive potential is applied, and applying the output, via the lower wire, to the grid of a valve.



All the digits of any number can be recorded along a row line of the grid and sensed simultaneously by a number of valves.

### 1.61 The medium.

One suggestion for a suitable medium would be ordinary or waxed paper. Once a hole was produced the carbonaceous products of combustion in the track would have a materially lower resistance than the ordinary paper. The disadvantage of this system lies in the fact that data cannot be erased, however, since the cost would be low, and very large memories could be set up, this fault might not be prohibitive.

Another method would be to use some medium, as in electrolytic condensers, which breaks down semi-permanently on the passage of current in one direction, but is a good insulator in the reverse situation. If such a medium could be repaired rapidly by conduction processes all the requirements would be met.

A gaseous medium, of neon type, is another possibility. Thus if the two grids were held at a potential, just over that required for steady discharge, but well below that required for striking, an impulse on any pair of wires would produce a conducting track and initiate a continuous discharge. Such an arrangement would be fast and easily cleared.

### 1.70 Magnetic memory.

In the field of magnetic media for binary memory, the first place is due to wire and tape. Originally designed for sound recording, the method is particularly appropriate for the input and output of a calculating machine.

Tests have shown that data can be placed on tape or wire with a packing of between 50 and 100 digits per inch, and can be read off at 50 to 100 K.c/sec. Extensive development has been carried out at Princeton, and the method of integrating this type of memory with a high speed inner electronic memory, and with punched tape data input, has been worked out.

### 1.71 Magnetic memory for parallel operation.

The magnetic tape or wire is fundamentally a serial memory of medium speed. There are several methods, however, of making a high speed, parallel operation memory on a magnetic basis. One of the best ideas is to record the data ( in the form of magnetic pulses normal to the surface ) on a cylinder capable of rotation at high speed ( > 1000 revs./sec. ). By having a number of pick up heads in each of the digital channels, and suitable switching arrangements, data could be recorded and read off at better than  $10^{-4}$  sec. per complete number. This rate compares favourably with current ideas on electronic memory ( see 1.80 ), and completely outclasses delay line memories of equivalent complexity.

### 1.72 Permanence.

It is worthy of remark that the magnetic tape or wire provides, not only a high speed memory, but also the only memory ( possibly excepting photographic ) which is permanent if desired. Tapes are in existence which, although made in the early 1900's, are still in good condition.

Another feature is volumetric efficiency. Magnetic wire affords the least space consuming, known, way of storing data.

### 1.80 Electronic memory.

It is not proposed to give a detailed description of the various types of electronic memory which have been proposed, because the development of such devices is possible only in the type of laboratory possessed by valve manufacturing organisations.

In addition the Selectron, which appears to be the most promising memory tube, is still in course of development, and several important features, such as the means of internal amplification, are as yet undecided.

Several memory tubes based on the iconoscope have been considered, these are of the scan type and depend on the possibility of detecting the presence of electric charge on a dielectric by means of the influence of its field on an electron beam. A disadvantage of this type of memory seems to lie in the creep of the charge over the face of the dielectric, this is unimportant in television applications but is a complete contraindication for a calculating machine.

As explained in earlier sections of this report, an adequate binary memory should have two states of equilibrium. In the Selectron these are the states of charge of small dielectric elements. An electron beam can be directed to any point of an array of such elements by means of a switching device. According to the electrical state of the particular element at the point of direction, an impulse is emitted by the valve. The switching of the electron beam to the particular lattice point is achieved by means of two grids of wires, similar to those previously described but so wired that any particular binary digit makes two, and only two, adjacent wires in each grid positive. The arrangements are such that only in the square so defined can the electron beam penetrate the lattice. On such penetration occurring, an external signal is emitted according to the state of the dielectric behind the conducting portion of the grid. It is hoped that the speed of this type of memory will be such that any digit can be located in 10-30 microseconds. The capacity of each tube will be  $2^{12}$  or 4096 digits, and 40 tubes switched in parallel, will give the full 40 digit binary number.

#### 1.90 General conclusions.

It is now possible to make a statement as to the desirable features of the memory, and of the kind of organisation envisaged. The general concept of memory should embrace the input-output organs, since these will, of necessity, involve the making of a permanent record.

#### 1.91 Input-output.

The natural medium for the insertion of data is some form of typewriter, and the decision has to be made as to the medium on which it is to record. It is not impossible that the instrument might inscribe directly on the high speed memory, but this has the disadvantage that the input data may be greater in volume than the capacity of this organ. The organisation envisaged for the input is therefore the following:

- 1) Type data on teletype tape.
- 2) Transcribe from teletype tape to magnetic wire.
- 3) Read off from magnetic wire into high speed memory as required.

This scheme has the advantage that the data on magnetic wire will be available, at moderately high speed, to the high speed memory, and can be taken up by the latter upon order from the control, whereas the teletype tape is, by comparison, very slow. Attractive features of the teletype as a means of primary input are, however,

- 1) That it is universal, and in consequence the problem setter need not be in the immediate neighbourhood of the machine.
- 2) That it affords opportunities of checking input data by comparing independently typed tapes.

### 1.92 The intermediate memory.

Since the high speed memories available in the near future will be of limited capacity, it is evidently necessary to have some form of intermediately fast memory which is capable of storing very large quantities of data in a reasonable volume.

It seems clear from the above discussions, that the magnetic wire forms the solution to this problem.

Facilities will be provided in the control for the high speed memory to refill itself from the wire as required, and servo mechanisms will effect such transfers without the intervention of the operator. In addition, the high speed memory will transfer intermediate results not at once required, and final results of a computation to the wire, ready for transcription into notation comprehensible to the human mind.

### 1.93 The high speed memory.

The decision as to the form of high speed memory best suited to the machine seems limited to a choice between the two types:

- 1) The Seloctron.
- 2) Some modification of the parallel grid switching device.

If it is available at a reasonable price, the Seloctron is the most satisfactory solution, but since the design of the rest of the machine is the same for either alternative one of the other alternatives could be substituted.

### 1.94 The memory for orders.

Up to this point the memory has been regarded essentially as a device for storing and emitting numbers. To make use of the high speeds attainable with the machine it is necessary to have a control capable of directing operations at the same rate. Thus orders have to be stored in a device operable at the same speeds as the high speed numerical memory.

It is an obvious development to express each possible unit order in the form of a binary numerical code, and then to use the memory, already described, to store it. Since the number of possibly useful orders is limited ( $< 2^6$ ) a simple code will suffice, and its exact formulation will be discussed in 3. A less obvious choice lies between separate memories for orders and numerical data, or a single memory. The solution becomes clear, however, when it is considered that a general purpose machine may be called upon to solve problems of two distinct types:

- 1) In which a small number of orders suffices for the solution, and a large quantity of numerical data is used.
- 2) In which a very large and complicated set of orders operates on a small set of numerical data.

It follows that the same memory should be used for numerical data and for orders, and that the control should select its order from this array.

## 2.00 The arithmetic unit.

As stated in ( 0.21 ) the arithmetic unit must be capable of performing the operations of addition, subtraction and multiplication. Some justification is needed for the inclusion of multiplication as a fundamental operation as it could be programmed as a series of additions. Most problems, however, will involve many multiplications and, since the process of coding problems will be simplified and the time of computation reduced if multiplication can be regarded as a basic operation, it will be worth while making the arithmetic unit a little more complex to achieve this.

The case for including a special dividing unit is less clear as a good iterative process exists. It will be shown later, however, that slight modifications to the multiplying unit will enable division to be performed in the same time as multiplication and if the circuits are sufficiently simple the addition may be justified.

Square roots are less easily obtained and, as the operation is of rather infrequent occurrence in normal work, it will probably be better to use an iterative process and programme it each time. ( See however 2.72 ).

## 2.10 The binary point.

It is necessary here to decide on the means of locating the binary point. Two methods are possible:

- 1) Using the floating binary point.
- 2) Using scale factors.

In the first method all numbers are expressed as "binals" ( i.e. binary numbers less than unity ), and the position of the binary point is indicated by an associated characteristic. Thus 10110.111 would appear as .10110111 , 101.

Numbers inserted as binals in the second method have no characteristic, and the use of suitable scale factors is relied upon to keep the results within bounds. This will, of course, increase the time needed for coding problems, but even so this method is preferable for the following reasons. Firstly, the introduction of the floating binary point requires complication of the arithmetic unit to deal with the characteristics in addition etc., and secondly a considerable portion of the memory capacity would be tied up in recording these numbers. Thus if  $n$  significant figures are required,  $\log_2 n$  places may be needed to accommodate the characteristic in the worst possible case, and this space must be left free though its full use may be rare.

For the above reasons it is proposed to use the scale factor method, and in the following discussion only binals will be considered.

## 2.20 Negative numbers.

These will be represented by their complements and prefixed by 1 to indicate their negative nature. Thus -.0110101 will appear as 1.1001011. Although this will reduce the memory capacity to 39 digits, it will be seen to produce considerable simplification in the mechanism required for arithmetical operations.

## 2.30 Addition.

The chief component of the arithmetic unit is a device which will store a binal, add to it a second binal and store the result, this will be called the accumulator. In addition to this, two banks of flip-flops will be required which are capable of storing binals indefinitely, these will be called the arithmetic registers. At least one of the arithmetic registers, and also the accumulator, must have the property of being able to shift their contents to the left and right by as many places as may be desired.

## 2.31 Dynamic and static accumulators.

At this point a distinction must be drawn between the two possible types of accumulator, static and dynamic. When a binal is added to another binal contained in the accumulator provision must be made for carry over from one place to the next higher and, as each carry may itself cause a carry over, these must be done in sequence. In the worst possible case a carry may be propagated along the whole 39 places and, if  $t$  is the time for one carry,  $39t$  must be allowed between each successive addition. This assumes that all carries are being completed at each stage, as is the case with the static accumulator.

In the dynamic accumulator, however, as an addition is performed any carries are stored in a set of flip-flops. When the first addition is completed these flip-flops are cleared into the next upper digit place. The result may be to produce further carries and these are again stored in the carry flip-flops. The next number is now added, and it can be shown that the carries which it will produce can never coincide with those already stored in the carry flip-flops. There is thus no chance of a carry being lost in the process. This process is repeated for the whole series of additions and, at the conclusion of the series, the clearing process is repeated 39 times.

At first sight it might appear that the dynamic accumulator was preferable, especially when a large number of additions are involved as in multiplication. In practice the speed of operation is limited by the possible frequency of the carry pulses; moreover, as will appear when the procedure for division is considered, the rapid automatic method for the latter can only be used with a static accumulator.

Because of the above considerations, it has been decided to have a static type accumulator, a decision which has been strengthened by the progress which has been made towards the realisation of such a unit having an addition speed of better than 1 microsecond.

## 2.40 Subtraction.

Two possible methods will be described for subtraction, the first using complements, and the second using inverted carries-over.

### 2.41 Subtraction by complements.

In this method subtraction is performed by the addition of the complement. For example suppose that we wish to subtract .01101 from .11010. The complement of the first number, .10011, is added to the second number giving 1.01101. The required answer is now  $1.01101 - 1 = .01101$ . The mechanism required is therefore some device which will take the complement of the number to be subtracted by changing all 0's to 1 and all 1's to 0 and adding 1 into the last digit. The complement is now fed into the accumulator, addition takes place and the final result is obtained after erasing the carry-over beyond the binary point.

## 2.42 The inverted carry over.

The second method depends on the fact that binary subtraction can be performed in exactly the same way as addition except that carries-over now take place when the accumulator flip-flops change from 0 to 1 instead of from 1 to 0. No carry over beyond the binary point occurs in this method and the result is therefore obtained directly.

The deciding factor in this matter will probably be the technical difficulties encountered in the circuits.

## 2.50 Multiplication.

Multiplication is essentially a series of repeated additions with suitable shift of position. Thus, suppose that we wish to multiply 10110 by 101. The multiplicand is fed into the accumulator. It is then fed in a second time but now shifted two places to the left, and the result recorded is the required product. Some additional units will be needed here beyond those considered already.

First, a mechanism is needed to produce the shift. This could be incorporated into the accumulator so that, when multiplying, it could be set to shift once automatically at each stage. Alternatively the arithmetic register containing the multiplicand could perform the shift.

It is also necessary to sense the digits of the multiplier in turn and to instruct the arithmetic register to feed the multiplicand into the accumulator if the digit is 1. This could be done by an arithmetic register with shifting facilities, so that the multiplicand is shifted one place to the right at each stage, and the last digit sensed each time.

Thus, as well as the accumulator, two registers are needed, at least one of which must have shifting facilities. Another refinement could be introduced by making the shift mechanisms move an extra place without pausing when a zero occurs in the multiplicand. This would reduce the time required for multiplication on the average by one half.

## 2.51 Negative numbers.

When it is desired to perform multiplications with either or both the multiplier and multiplicand negative, it is necessary to adopt a somewhat more complicated procedure which is detailed in the following table.

- 1) Multiplier negative. The numbers seen by the arithmetic unit are thus  $(1-x)$ , and  $y$ ,  $(1-x)$  being in a shifting register. The product  $(1-x)y$  i.e.  $y-xy$  is formed and added into the accumulator; at the last stage the complement of  $y$ , i.e.  $1-y$  is added in. The accumulator thus contains  $1-xy$ .
- 2) Multiplicand negative. The numbers are now  $x$  and  $(1-y)$ , and, in the normal way the accumulator would receive  $x-xy$ . As the digits of  $x$  become available at the extreme right of the shifting register their complements are taken and fed into the extreme left position of the accumulator. At the end of the operation the accumulator thus contains  $1-xy-1/2^{39}$  and the addition of a unit to the extreme right position of the accumulator produces the correct result  $1-xy$ .
- 3) Multiplier and multiplicand negative. Here the accumulator receives  $1-x-y+xy$ . It is thus necessary to perform both the above processes 1) and 2), giving  $1+xy$ , and then to add in unity to correct the sign.

## 2.52 Extended accuracy.

It is proposed to have means whereby when the partial products are shifted in the accumulator, the terminal digit, which would otherwise be lost, is fed into the left-hand side of the arithmetic register. Since the latter clears itself during the calculation, the final state of the unit is one in which the product is stored with its first 39 digits in the accumulator and its last 39 digits in the register.

## 2.60 Division.

As mentioned before there are two possible methods for division, the iterative and the direct.

### 2.61 The iterative method.

The reciprocal  $x$  of a number  $b$  can be calculated by repeated application of the formula  $x_n = 2x_{n-1} - bx_{n-1}^2$ . This process requires two multiplications and one subtraction at each stage and can therefore be performed with the units already postulated. It can be shown that, if  $e$  is the error in  $x_n$  then the error in  $x_{n+1}$  is  $be^2$ ; as we shall be dealing only with binal numbers, this process will be quite rapidly convergent. Even so it will be necessary to make at least 3 iterations and the time required will thus be about 6 multiplication times.

### 2.62 The direct method.

This method is essentially that of long division; that is, the denominator is successively subtracted from the numerator which is shifted one place to the right at each stage. 1 or 0 is recorded as quotient according as the result of subtraction was positive or negative. The numerator will therefore be placed in the accumulator and the denominator in one arithmetic register. The quotient will appear in a second arithmetic register. The accumulator must be modified, however, so that when a subtraction produces a negative result this result is rejected and the original numbers are restored to the accumulator. This is exactly the process of long division except that the human operator would probably not actually perform the subtraction in order to see whether it were possible.

Provision must therefore be made in the accumulator for storing the numbers of the remainder which are being operated on until the subtraction has taken place. Then, if the result is positive, these will be rejected and the result of the subtraction placed in the accumulator. If it is negative, however, it will be rejected, and the original digits retained in the accumulator.

The machine must thus be able to sense whether the subtraction has produced a positive or a negative result. This can be done very simply by considering the carry-over beyond the binary point. Thus, if the complement method of subtraction is used, the carry over will be 1 or 0 according as the result of the subtraction was positive or negative. If 1 was carried over the machine must therefore retain the result of subtraction in the accumulator, record 1 in the quotient register, and shift the accumulator one place. If no carry over occurs, the result of the subtraction must be rejected, the original number restored to the accumulator with one shift, and the quotient register shifted but no digit recorded.

If the reversed carry over method of subtraction is used, this process will be inverted as 1 will be carried over for a negative result and 0 for a positive.

In practice, no great elaboration of the original accumulator is required to do this, and the process takes exactly the same time as multiplication. Moreover, the coding of a problem is much simplified if it is not necessary to programme division. It seems, therefore, that the direct method is to be preferred.

It should be observed that the static type accumulator is necessary for this process as all carries-over must be completed after each subtraction.

## 2.70 Extraction of the square root.

As with division, there exists a good iterative process for the extraction of the square root. For completeness both this and the direct method will be considered briefly, though, as pointed out in 2.0, the operation of square-rooting occurs comparatively infrequently and it may not be economical to provide a special unit for the purpose.

### 2.71 The iterative method.

The general formula -

$$x_n = x_{n-1} (p+1 - bx_{n-1}^p) / p$$

gives an iterative process for determining the reciprocal of  $b^{1/p}$ . Taking  $p = -2$  this gives the formula-

$$x_n = (x_{n-1} + b/x_{n-1}) / 2 \text{ for } \sqrt{b}.$$

Four iterations would probably be needed to obtain the desired accuracy, and the process would therefore take 4 times the time required for one division. Multiplication by  $1/2$  is trivial, since this merely consists in shifting one place to the right.

If an automatic divider were not available it would be better to use the formula-

$$x_n = x_{n-1} (3 - bx_{n-1}^2) / 2$$

for  $1/b^{1/2}$  and then to multiply the result by  $b$  as no divisions are involved in this process.

### 2.72 The direct method.

This is exactly the elementary method used for extracting square roots, but carried out in binary scale.

It is similar to long division but the divisor is now variable, and the accumulator must be made to shift two places at each stage instead of one.

It is necessary at each stage to take the number from the quotient register, add in 1 two places to the right of its last digit and subtract the number so formed from the contents of the accumulator.

This process requires the same time as division or multiplication, but the deciding factor will, again, be the technical difficulties introduced.

In certain problems of X-ray crystallography with Fourier syntheses, however, the inclusion of such a unit would merit serious consideration, as frequent use would be made of it.

### 3.00 The control.

Up to this point mechanisms, by means of which calculations can be performed, have been described without any reference to the mode of their direction, and it has been tacitly assumed that such a control can be conceived. In this section the precise organisation of this control will be laid down, and the types of order which it must be capable of executing discussed.

#### 3.10 Synchronous versus asynchronous operation.

Whereas delay line type machines must, of necessity, work on the basis of strict timing of operations, this restriction does not apply to a parallel operation machine of the kind considered in this report. It is therefore worth discussing the merits, or otherwise, of an internal timing cycle.

The method of operation of such a timing mechanism is as follows. Suppose that a multiplication requires a time of  $(m)$  micro-seconds, then if a pulse from some central "clock" initiates the multiplication at time  $(t_0)$  a pulse from the same clock at time  $(t_0 + m)$  can initiate the next operation in the computing sequence with the assurance that the previous operation will be complete.

The alternative method is to have the multiplier signal the completion of its operation by emitting a pulse which advances the control to the next operation. Both methods have certain advantages. The method of clock synchronisation permits a calculation to be followed through step-by-step if the operation of the machine is suspect, by the simple expedient of having the operator insert the clock pulses manually. On the other hand, if any unit is malfunctioning so that its cycle of operation is not completed in the standard time, the synchronising pulse may order the commencement of the next operation too early.

On the whole, it seems best to combine both methods and arrange that the clock pulse, and the operation complete pulse, are both required to initiate the next operation. This affords an internal check on the functioning of the machine, so that in the event of the two pulses not occurring within a certain short time of each other, the operation could be automatically stopped and a visible indication given.

#### 3.20 Specification of orders.

It was implied in 1.94 that orders would be stored in the memory in binary coded form, and the means of doing this will now be considered.

In the first place it is evident that, in general, an order will consist of two parts -

- 1) Take a number from the memory.
- 2) Operate on it.

The first part involves the specification of the position of the number in the high speed memory. If, as is projected, this memory has a capacity of about  $2^{12}$  numbers, it follows that 12 binary digits of the order must be used to locate any particular number.

The number of possible operations which the machine can execute is more speculative; it appears certain, however, that it will be less than  $64 (2^6)$ , and probably less than  $32 (2^5)$ . Thus 6 more binary digits will suffice to specify the operation. The total number of digits required for a complete order is thus 18.

Now, since 40 digit binary numbers are the normal unit contents of the memory, it will be possible to store two orders side by side in the memory.

### 3.30 The control register.

The parallel storage of orders in the memory, described above, makes it necessary to have two storage registers in the control; these will be designated Cr<sub>1</sub> and Cr<sub>2</sub> respectively and the orders stored in them will be executed in sequence.

### 3.40 The control counter.

Before performing a calculation in the machine, the operator will draw up an ordered table of the unit operations in the computing sequence and will set these into the memory. The machine will thus follow, as far as the control is concerned, an ordered sequence of events.

### 3.41 Non-singular operations.

By this is meant a programme of operations which are completely predictable by the operator before any machine operations have taken place, and which, in consequence, require the exercise of no independent judgement by the machine.

For such sequences it is sufficient to have a control counter to which unity is added at the conclusion of each machine operation, and which thereupon initiates the next order of the sequence.

### 3.42 Singular operation.

This type of operation is typified by the functioning of the machine during an iterative cycle. Thus, at some stage in a calculation, the machine may be required to follow a sequence of the type -

$$x_{n+1} = f(x_n, x_{n-1} \text{ etc.})$$

until the criterion -

$$x_{n+1} - x_n < \epsilon$$

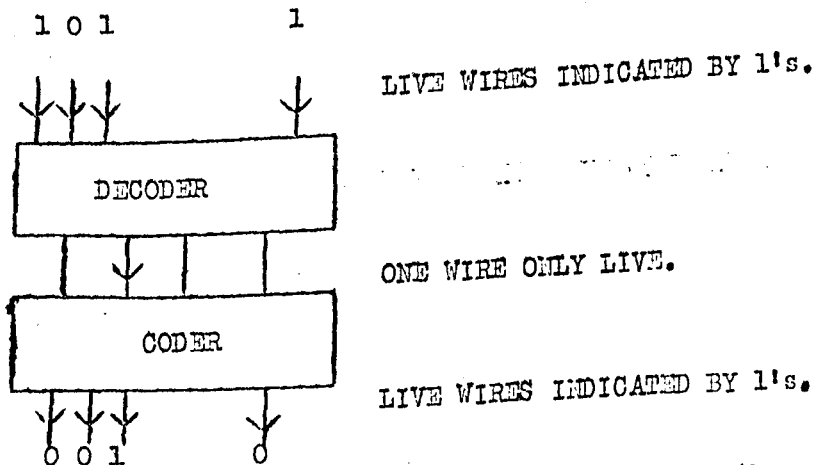
is satisfied and then to proceed on the main computing sequence.

This necessitates the existence of an order which returns the control counter to a previous setting.

### 3.50 The decoder - coder.

Since the number location part of any order gives a representation which is immediately intelligible to the memory, no further treatment of it is required. This, however, is not true of the operational part; here a binary number is to actuate several component parts of the machine. It follows that a decoding table is required which establishes a unique output for any binary input. In addition, since the number of functional units is small, several orders will often have certain functions in common, so that a coding table is required, which gives single inputs to each functional unit, each capable of energisation by the several orders having the particular function in common.

Such decoding and coding tables are already in existence; they have been used in the relay machine of the present authors, and, in modified form, in the E.N.I.A.C.



The general lay-out is shown in the diagram, and it can be seen that the arrangement really corresponds to a table whose output is any preassigned function of its input considered as argument.

### 3.60 The orders.

It is worth while detailing a possible type of basic order set from which all operations could be programmed, and then examining desirable modifications of it.

### 3.61 Basic order set.

Although addition and subtraction might be programmed from logically simpler operations, it seems justified to regard them as basic operations. Thus, basic operations will be -

- 1) Clear accumulator.
  - 2) Set memory to emit absolute values.
  - 3) Add number from position (n) in memory into accumulator.
  - 4) Subtract number from position (n) in memory into accumulator.
  - 5) Clear arithmetic and transfer number from (n) into it.
  - 6) Shift number in arithmetic register one place to right.
  - 7) Shift number in arithmetic register one place to left.
  - 8) Transfer right hand digit of number in accumulator to left hand digital position in arithmetic register and shift number in accumulator one place to right.
- This set of basic arithmetical orders has to be supplemented by discrimination orders to make possible multiplication and division; these are of the type -
- 9) If R.H. digit in arithmetic register is 1 add number from (n) into accumulator.
  - 10) If number in accumulator < 0 add number from position (n) into it and shift numbers in accumulator and arithmetic register one place to left.
  - 11) If number in accumulator > 0 make R.H. digit in arithmetic register 1 and shift contents of accumulator and arithmetic register one place to left.

Now since multiplication and division will take, on the basis of these orders, times of approximately 40tm and 80tm respectively, (tm time of selection and execution of order stored in memory) i.e. 1200 - 2400 micro-seconds, and it appears perfectly feasible to construct a multiplier and divider whose cycle of operation will be about 100 micro-seconds, it seems justified to include these operations among the basic set. This decision produces a considerable modification of the order sequence given above.

In addition, it seems justified to combine order 1) with orders 2), 3), and 4), to give new orders, despite the increase in the total number of orders thereby produced.

### 3.62 Contracted notation.

At this point it is useful to introduce the following contracted notation:-

- M = Memory.
- A = Accumulator.
- R = Arithmetic register.
- C = Control.
- T = Magnetic tape.

For clearing any component, its symbol will be prefixed by c. In the case of partial transfers from accumulator to memory (required in use of tabulated functions etc.) the symbols  $A_L$  and  $A_R$  will be used for the left and right halves respectively. Similarly, to indicate the left and right halves of an order in the memory the symbols  $M_L$  and  $M_R$  will be used. For conditional transfers of the control to orders not in sequence the symbol Cc will be used.

Apart from this the notation is self explanatory. Thus -

+ M → cA

means - "Add the number in the memory (location m understood) to the cleared accumulator."

### 3.70 Extended order set.

The set of orders suggested for the machine may now be formulated.

- 1) +M → cA.
- 2) -M → cA.
- 3) |M| → cA.
- 4) -|M| → cA.
- 5) +M → A.
- 6) -M → A.
- 7) |M| → A.
- 8) -|M| → A.
- 9) M → cR.
- 10) R → cA.
- 11) M × R → cA. = Clear accumulator, multiply M by R and place L.H. 39 digits of answer in A and R.H. 39 digits in R.
- 12) A ÷ M → cR. = Clear register, divide A by M, leave quotient in R and remainder in A.
- 13) C → M<sub>1</sub>.
- 14) C → M<sub>2</sub>.
- 15) Cc → M<sub>1</sub>. = If number in A > 0 shift control to M<sub>1</sub>.

- 16)  $C_c \rightarrow M_r$ .  
 17)  $A \rightarrow M$ .  
 18)  $A_l \rightarrow M_l$ .  
 19)  $A_r \rightarrow M_r$ .  
 20)  $S_r$  = Shift contents of  $A$  one place to right but leave  
 L.H. digits unaltered.  
 21)  $S_l$  = If contents of  $A$  are  $A(0), A(1), \dots, A(39)$  and of  $R$  are  
 $R(0), R(1), \dots, R(39)$ , replace these by  $A(0), A(2), \dots, A(39), 0$ ,  
 and  $R(1), \dots, R(39), A(1)$ .  
 22)  $I$  = Initiate operation of machine.  
 23)  $T_i \rightarrow M$  = Transfer contents of input tape to  $M$ .  
 24)  $M \rightarrow T_o$  = Transfer contents of  $M$  to output tape.  
 25)  $E$  = Signal completion of operation.