Ph.D. Thesis

A COMPUTER FOR SOLVING FIELD PROBLEMS IN ELECTRON BEAM DEVICES

by

Alan Russel Dinnis, B.Sc.

Edinburgh

July, 1962

ABSTRACT

The need is explained for a new type of computer for solving partial differential equations, the Digital Field Computer. The operation of such a machine for solving Laplace's and Poisson's equations is explained and circuits for its realisation, using incremental switching of magnetic ferrite cores, are given. Its operation is predicted by simulation on a Pegasus digital computer, which shows that it solves Laplace's equation correctly.

CONTENTS

1 THE CONCEPT OF A DIGITAL FIELD COMPUTER

1.1 The Need for a Rapid and Accurate Method for Solving Laplace's Equation. In many engineering problems it is necessary to solve Laplace's Equation at some stage. This is true in fields other than electrical engineering, such as in problems involving conduction of heat, but the present work was stimulated mainly by problems encountered in devising methods of design for electron guns and in studies of the behaviour of dense electron beams.

It may seem that the solution of Laplace's Equation is mathematically straightforward and that therefore no problem really exists. But this is not so, as almost invariably, in engineering problems, the boundary values of potential and/or electric field strength cannot be given in a convenient analytic form but are numerical values. The shape of the boundaries themselves may also only be expressible numerically.

One type of problem involving solution of Laplace's Equation under such conditions is that of designing a suitable electrode structure for an electron gun which is to produce a dense beam of electrons. The electrodes must produce at the surface of the beam the correct normal field and potential at all points as determined by the conditions in the particular beam. In certain particular cases these beam boundary conditions can be expressed analytically, and also Laplace's

Equation outside the beam can be solved using the beam edge boundary values in a Cauchy type problem. These are quite simple cases, such as a 2-dimensional straight or curved beam, where complex variable methods may be used, or the cylindrically symmetrical straight beam as solved by Radley[9]. These cases are, however, exceptional and in more complex electron flow patterns, as discovered by Kirstein[1], numerical methods are needed to solve the electron flow problem, with the result that beam surface potentials are given numerically. Numerical solution of Laplace's Equation under Cauchy boundary conditions is notoriously unstable. This instability may not be insuperable, but in most cases it is likely that Pierce's[2] method of design, or some variation of it, will be used. This method has the advantage that it can take account of the effect of modifications necessary to the electrodes for engineering reasons, such as avoidance of sharp corners because of high voltage operation. It turns the Cauchy type problem into one with closed boundaries having boundary values of Dirichlet (V specified at the surface) and Neumann type (normal field specified at the surface). The solution is then determined either by trial and error or by a systematic method, but in either case Laplace's Equation will have to be solved a large number of times to obtain a satisfactory solution.

Another class of problem requiring a large number of solutions to Laplace's Equation is that of electron flow in which electron trajectories depend on the potential field of the electrons themselves and the flow is non- steady state. Such a problem is that of investigating the type of transverse electron beam instability described by Kyhl and Webster[3]. This comes about when a thin-walled hollow beam or a ribbon beam is focused by a longitudinal magnetic field. The electric field produced by the charge in the beam is perpendicular to the magnetic field. Hence transverse motion of the electrons occurs, with consequent beam break-up into a number of vortices. Analysis for small perturbations is possible assuming a sinusoidal disturbance of the beam[3,4], but for large perturbations as observed in practice there seems no alternative to numerical analysis. As each step in such a process would require the solution of Laplace's Equation throughout the region of interest, the computation is likely to be slow unless a rapid means of solving Laplace's Equation is available.

In both of these examples it would be a very considerable advantage if the field solving system could be coupled directly to a general purpose digital computer. In the focusing electrode problem the computer would decide on the electrode shapes to be used, either by a system analogous to the human trial-and-error method or, better, by a systematic method. In the second case the computer would at each step find the motion of all electrons in a short period, feed the new positions to the field solving system and use the new values of the field found by it to compute the next step of the electron trajectories.

1.2 Methods at Present Available. Perhaps the most obvious way of solving such problems, particularly in view of the requirement that any system should be capable of direct coupling to a digital computer, is to use a digital computer actually for solving Laplace's Equation. Of course, this is perfectly feasible provided the problem is small enough, but for many interesting problems, particularly three-dimensional ones, the time taken[10] and storage space needed can be prohibitive, especially in cases where Laplace's Equation must be completely solved at each step in a problem's solution.

It is true that certain problems do lend themselves to solution on a general purpose digital computer. The method of solution in Appendix 2 for Pierce electrodes for a 2-dimensional electron gun system in which electrons reach relativistic speeds is one such case. Although the potentials at the beam edge cannot be expressed in a closed analytic form, it was possible to use numerical integration of a differential equation over the complex plane to determine the potential field outside the beam.

Hence instability of the solution appears to have been avoided. Each point is treated only once and so the method avoids the usual disadvantage of solving numerically a field problem, which is that all points have to be scanned a great number of times before the final solution is obtained. This is, however, a relatively simple problem, the corresponding non-relativistic one is the simplest possible case and so is not typical of what is met in practice. This method is, in addition, inapplicable to the case of a cylindrically symmetrical relativistic beam.

Probably the most commonly used method is the use of some sort of analogue, the electrolytic tank or its solid equivalent, or the resistance network analogue. These devices can be refined to give very accurate results but their accuracy cannot be increased indefinitely. While they have been used in direct conjunction with digital computers for electron trajectory tracing, there is bound to be some difficulty in the analogue-digital link. ~~Like~~ As with most analogue computers, setting up boundary values tends to be a long process as opposed to feeding data into a digital computer. For this reason, control of boundary shapes and potentials in the analogue system by the digital computer is likely to be difficult. These analogue methods have the great advantage, however, that once the boundary values have been set up the solution takes place almost instantaneously. The problem then arises, of course, of reading out the solutions, so that the majority of the time taken for solving a problem is in setting up and reading out.

1.3 The Digital Field Computer.[6,11] This ~~proposed~~ computer, proposed by Dr. Meltzer,[5] is an attempt to combine the desirable features of analogue and digital computers. It will have the speed of the analogue combined with the accuracy and ease of problem changing of the digital computer. As it is a digital device its accuracy can be increased almost indefinitely. It must rely on the use of finite difference methods, as the potential can only be found at a finite number of points. To achieve the speed of an analogue it must work in very much the same way as one, computation proceeding simultaneously at all points in the network. At each mesh point there must therefore be some sort of arithmetic unit in addition to a means of storing the potential at that point.

For maximum simplicity of the machine, the first-order difference equivalent of Laplace's Equation in 2 dimensions will be used. Referring to the square mesh of fig. 1.3.1, this is :

$$4V_0 = V_1 + V_2 + V_3 + V_4 \qquad \ldots\ldots(1.3.1)$$

So that every mesh-point must continuously adjust its potential to the mean of the potentials of the four adjacent points. Thus, if point 4 happens to rise by v volts, point 0 will rise by v/4 volts. This rise in V_0 will then affect V_1,V_2,V_3 and also V4 and

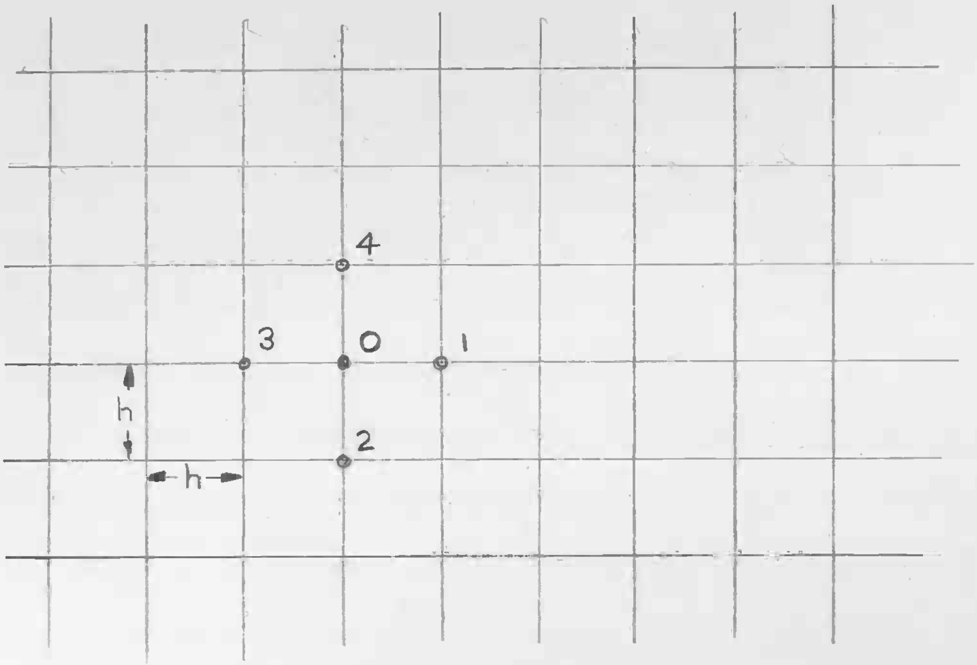


Figure 1.3.1

and so the process will continue until an equilibrium is reached.

1.4 The Mesh-point Units. The unit at each mesh-point must must receive information on potential changes of the neighbouring units, adjust its own potential, and send information on its change of potential to the adjacent points. As it is only the change in potential that need be transmitted between units, and as the interconnexions between units should be as simple as possible, it is advantageous to use a system in which a potential is represented by a series of pulses, the number of which is directly proportional to the potential. For instance, if 1 pulse represents 1 volt, then 100 pulses is 100 volts and so on. This needs only one line between units and it also makes the design of the mesh-point units easier, as will be shown later.

The requirements for the mesh-point unit are, then, that for every four input pulses the unit should emit one pulse to each of the four neighbouring points. Ithdoes not matter from which of the adjoining points the four pulses come from or when they came, they could for instance all be from point 4, or one from each of the neighbouring points. Of course, if point '0' receives a number of pulses during the whole of a computation which is not exactly divisible by four, a certain error is bound to occur. This will not be serious as the total number of pulses will be large, and there is a technique which is described later for minimizing this effect.

There must also be a store at each point, to which one pulse is added every time the unit emits a pulse, in order to have a record of the total potential at the point. This could be a series of cascaded counter stages capable of storing the number of pulses which are needed for the degree of accuracy required in the solution. The storage units must be capable of reasonably rapid readout, as setting up and readout are likely to be much more time-consuming than the actual computing process. It is worth noting that, in view of this fact, it appears to be unlikely that the basic units be capable of the maximum speeds that are at present feasible unless the accuracy required is much greater than that at present envisaged (max. number of pulses at any point about 10^5 , computing time about 5 secs.).

1.5 Boundary Values. Considering only the solution of Laplace's Equation under Dirichlet boundary conditions, the insertion of boundary values is quite straightforward. The potential is specified at every point on the boundary, and so the appropriate number of pulses is fed in at each point. An interesting and useful point is that these pulses need not necessarily be fed in simultaneously. By the principle of superposition, as long as each point is brought finally to the correct potential, it does not matter in what manner it got there. This means that one pulse generator can be used to feed a number of boundary points in turn until all are at the desired value. Also it is easy to change boundary potentials and see what effect this has on the solution.

The boundary points of the mesh emit pulses only when their potential is being set. Pulses which arrive at the boundary from the inside of the mesh system are ignored. If these incoming pulses were acted onin the normal way and the boundary unit emitted one pulse for every four incoming pulses, then its potential would rise in an unpredictable way and it would be difficult to bring the unit to its required potential by feeding in an appropriate number of pulses from outside the mesh.

1.6 Extension to Poisson's Equation. The finite-difference oform of Poisson's Equation is:

$$V_0 = \tfrac{1}{4}(V_1 + V_2 + V_3 + V_4 + h^2\rho/4) \quad ...(1.6.1)$$

where h is the mesh dimension and ρ the charge density in the region of point 'O'. The term $h^2\rho/4$ means that an extra number of pulses corresponding to this value must be fed in from outside the mesh. However, the unit must still take account of the potentials of the adjacent points and these will be affected by the charge density pulses which are fed in. These charge density pulses can, of course, be fed in at any time in the computation.

2 LOGICAL DESIGN OF BASIC SUB-UNIT

2.1 First Type of Basic Unit. Brown[6] suggested that the computer could be built up from a basic unit having two input channels on which pulses may arrive simultaneously on the two input lines or sequentially. The unit must give one output pulse for every two input pulses. This unit by itself would comprise, together with a potential register of some sort, a complete mesh-point unit for one node of a one-dimensional system. Three could be combined to make a two-dimensional unit and six for a three-dimensional one. Such a unit, using thermionic valves, is described by Brown but he decided that the computer was not at that time feasible, due mainly to the high power consumption and cost and the lack of reliability of the circuits.

It was decided that, as a first step, Brown's system should be re-examined to see if by the use of different components, such as transistors and ferrite cores, the computer could be realisable. The first thing done was to make a logical analysis of the requirments for the basic unit so that minimization of the logical elements needed could be achieved.

2.2 Logical Design and Minimization. The 'truth-table' for the unit must first be constructed. The two inputs are labelled x and y. The presence of a pulse on a channel is denoted by the letter,thus: y , the absence of a pulse by the letter primed : y'.A third factor is the state of a memory device, z , which is necessary because there must be an output pulse for every two input pulses even when they arrive at different times. The truth table is:

Present state of z	Next state of z for input of:				Output for input of:			
	00	01	11	10	00	01	11	10
0	0	1	0	1	0	0	1	0
1	1	0	1	0	0	1	1	1

'00' indicates $x'y'$, '01' indicates $x'y$, etc.

From this table can be deduced the condition for the two-state memory device z tobe set to zero:

$$f_0(z) = x'y'z' + x'yz + xy'z + xyz' \qquad \ldots(2.2.1)$$

where + indicates 'or' (union) and multiplication indicates 'and' (intersection). Also, for z to be set to one :

$$f_1(z) = x'y'z + x'yz' + xyz + xy'z' \qquad \ldots(2.2.2)$$

From these two equations it is evident that z changes state (from 0 to 1 , or vice-versa) only when $x'y + xy'$ occurs, i.e. an input on one line only, which fact could be deduced readily without the aid of logic theory.

The unit produces an output for the conditions:

$$f_1(o/p) = x'yz + xy + xy'z \qquad \ldots\ldots(2.2.3)$$

This can be reduced, by inspecting the output column of the truth table and treating it as a Karnaugh Map, to:

$$f_1(o/p) = xy + xz + yz \qquad \ldots\ldots\ldots(2.2.4)$$

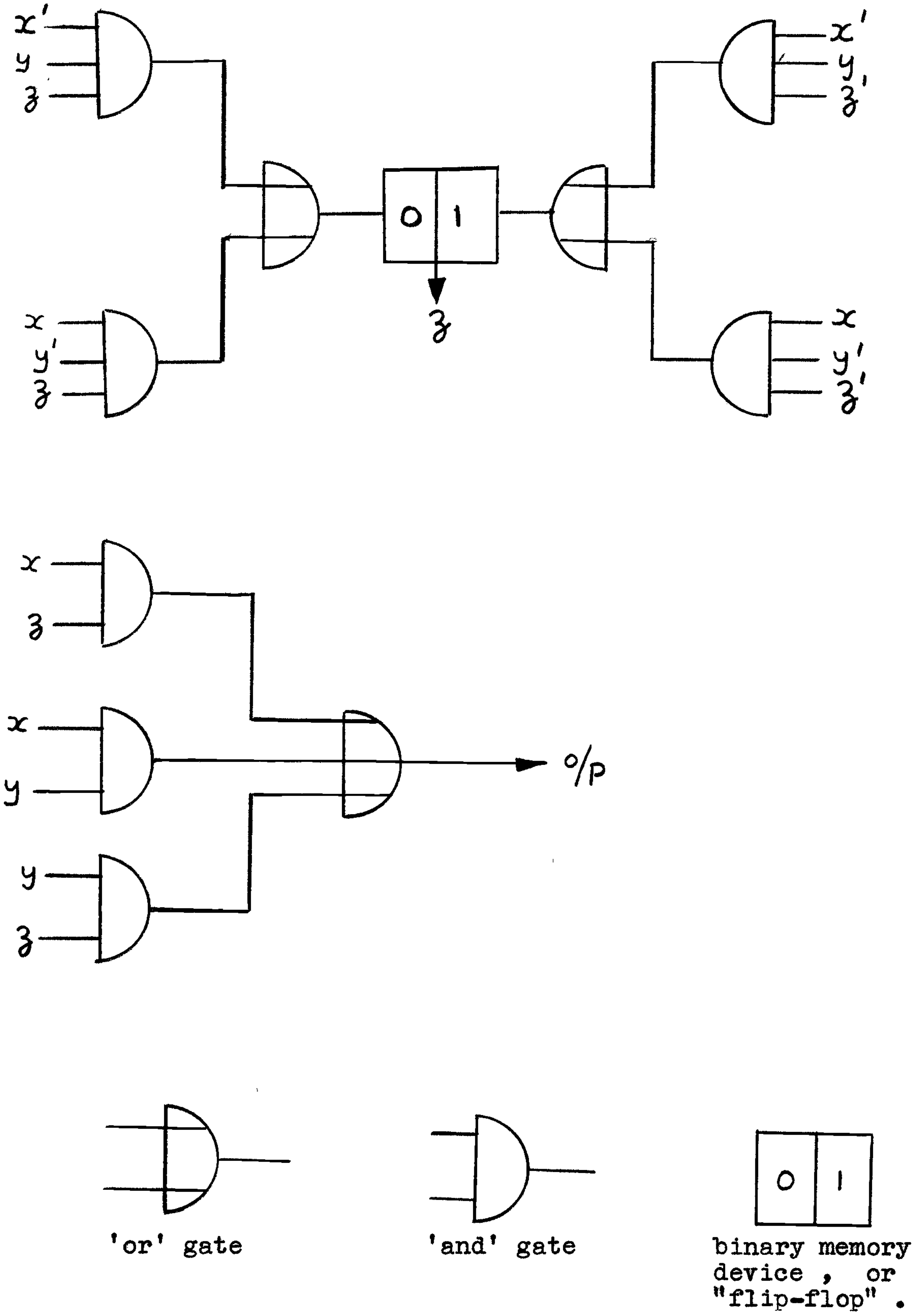

x'
y
z
x
y'
z
0
1
z
x'
y'
z'
x
y'
z'
x
z
x
y
y
z
o/p
'or' gate
'and' gate
0
1
binary memory device , or "flip-flop" .

Figure 2.2.1

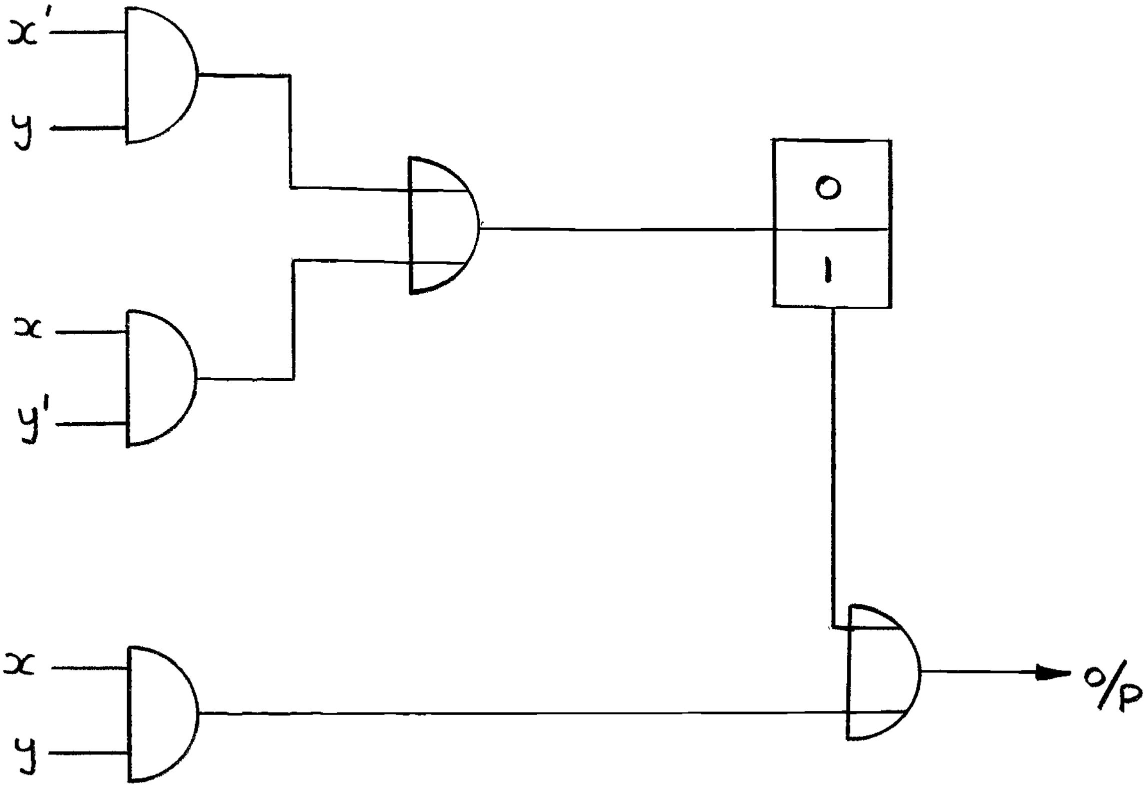


Figure 2.2.2

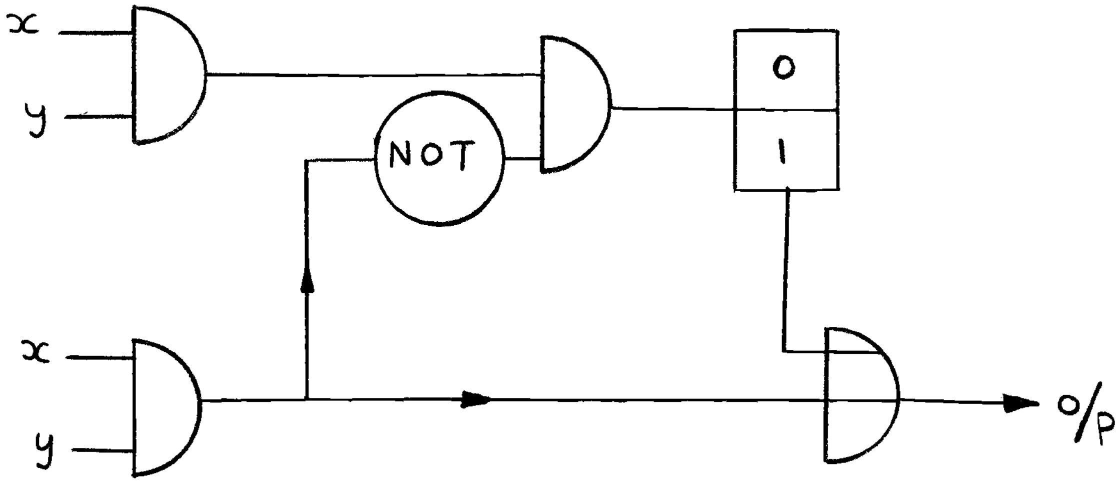


Figure 2.2.3

A diagram of the logical devices needed to achieve the desired result is shown in fig. 2.2.1 .

If it is assumed that the state of z changes for every input to it, and that a pulse is available whenever z changes from 0 to 1, so that $o/p(z) = zxy'+zx'y$, then the simplified circuit of fig. 2.2.2 can be used.

An alternative circuit, incorporating a 'not' element, is shown in fig. 2.2.3 and uses the identity

$$xy'+x'y = (x+y)(x'+y') = (x+y)(xy)' \qquad \ldots(2.2.5)$$

2.3 Results of Minimization. None of these circuits is, in fact, any simpler than Brown's one, and so no progress has been made. However, before giving up this system it was decided to try other circuit components in the same logical system.

It was considered that ferrite core logic offered some advantages over valve and transistor circuits. These are mainly that they are not expensive, have a memory as well as logic capability, and are highly reliable. They are certainly rather slow, but this should not be a great disadvantage, provided the above advantages are realised in practice.

2.4 Core Logic Unit. In ferrite core logic systems, information is continuously moved from one core to the next by a system of clock pulses. The system considered uses two clock phases. On phase 'A', information is driven from cores in phase 'A' to cores in phase 'B'.

At clock phase 'B', information is driven forward again to cores in phase 'A' and so on. The system is therefore based on the core shift register.

Quite complex logic functions can be realised by applying inputs to windings having appropriate numbers of turns wound on a toroidal core. Owing to the 'square-loop' characteristics of the ferrites used in making the cores, the state of magnetization changes, or the core 'switches', only when a certain m.m.f. is exceeded. Therefore an 'and' gate can be realised by applying m.m.f.'s which separately would not switch the core but when applied simultaneously will switch it. In an 'or' gate, more turns are used on each input winding so that the core switches even if only one input is energised. In the figures, the weight given to each input is denoted by a number beside it, +1, +2 etc. A total of $>1\frac{1}{2}$ will switch the core and so produce an output from it at the next clock phase.

The schematic circuit for the basic unit is shown in fig. 2.4.1 . The 'carry-digit', z, is driven, under no-input conditions, alternately between B_3 and A_3. The output from A_3 to B_2 has no effect as there is no other input to B_2 to bring the m.m.f. above the threshold value. If z is present and x or y singly occurs, then there is an output and z becomes zero. This is because the 'or' gate A_2, and also A_3 pass

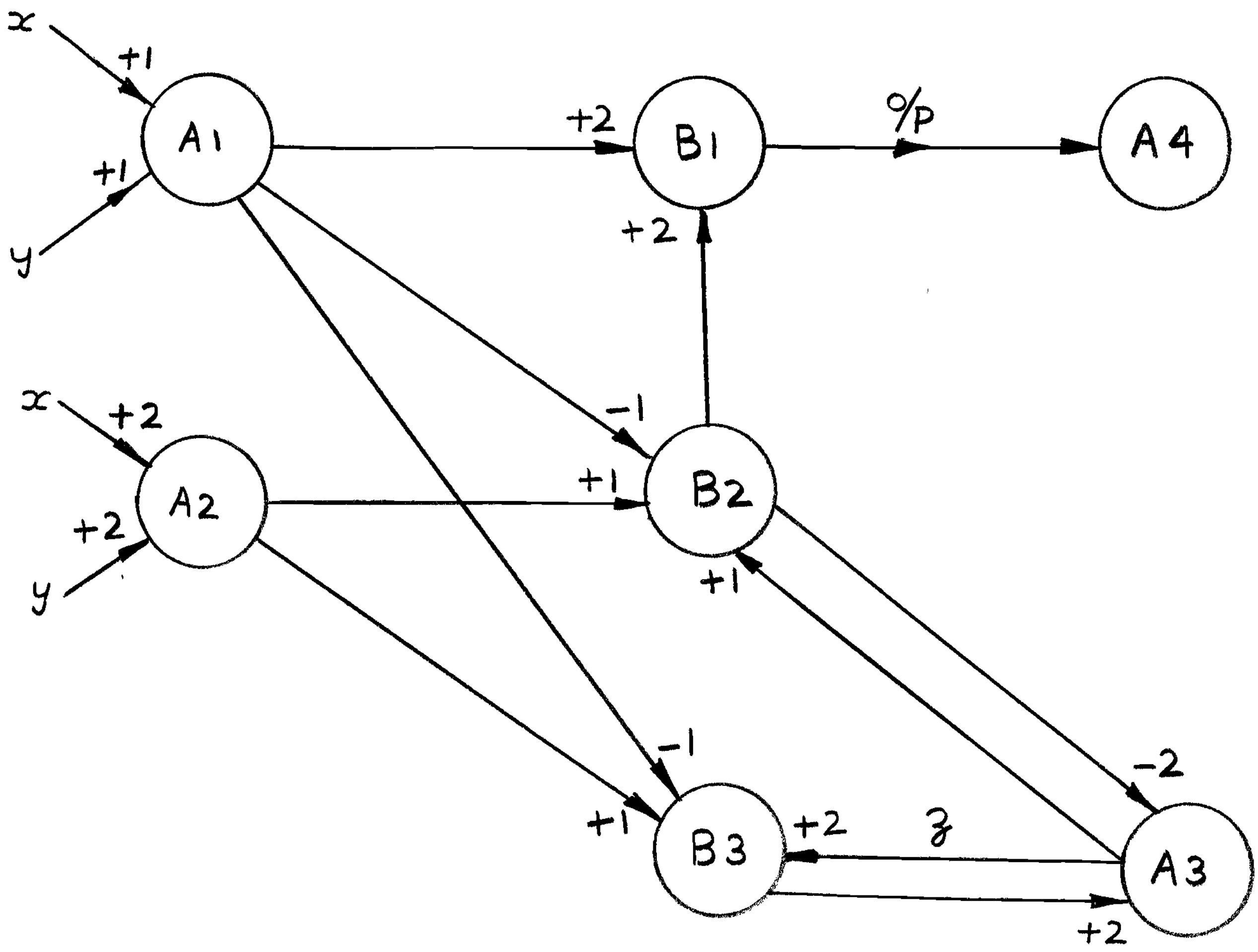


Figure 2.4.1

pulses to the 'and' gate B_2 and hence cause an o/p. z is erased by the -2 pulse from B_2 to A_3 . If z is zero to start with, then 1 is written into B_3 and hence z, and there is no output. z is not erased since there is no output.

If x and y occur simultaneously, an output is produced via A_1, but the 'or' gate A_2 is also actuated. This is not wanted and so the effect is nullified by the (-1)'s to B_2 and B_3. B_1 is provided to give an output at the correct phase.

The unit therefore obeys the logic requirmen ts for a basic unit. It uses 6 cores and 6 diodes besides winding and connecting wires.

2.5 Potential Register Using Cores. A possible arrangement for a binary counter and store, to record the total number of pulses emitted by a unit, is shown in fig. 2.5.1

A_2 and B_1 form a dynamic store, a pulse passing continuously from one to the other if there is a '1' in this stage of the register. The output from the mesh-point unit is passed to A_1 and A_2 simultaneously. If there is '1' in this stage already, then A_1 switches and passes a -2 pulse to B_1 hence erasing the '1' in the dynamic store of this stage. If the store is zero to start with and a pulse arrives, then there is no output from A_1, but a pulse is put into A_2 and so the store becomes '1'.

This register uses 3 cores and 3 diodes per stage.

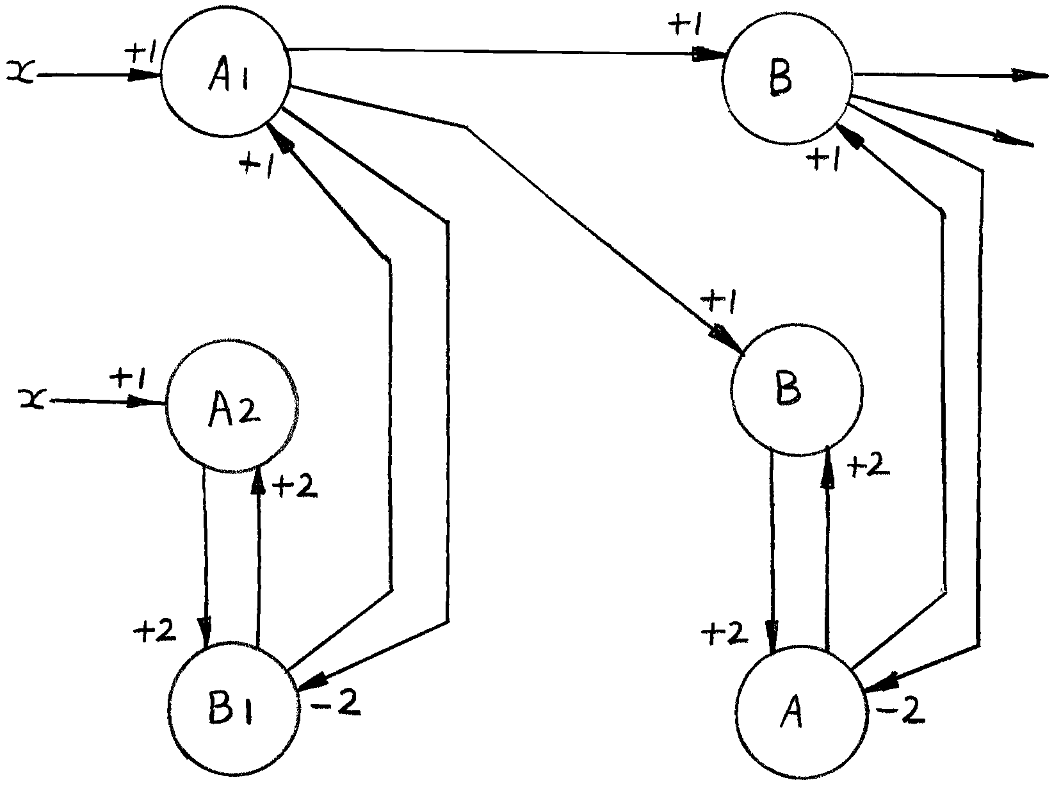


Figure 2.5.1

2.6 Simplification of the Computer System. The system described above is still quite complicated, so it is worth considering fundamental changes in the mode of operation.

First of all, it is evident that provision for the units to deal with simultaneous arrival of pulses adds components which would not be needed if it were known that pulses would always arrive sequentially and never simultaneously. This implies that the units adjacent to any particular unit must always emit pulses to that unit at different phases in the clock cycle. It will be essential to have a clocked system in this case, of course. If the above condition can be satisfied then the basic unit simply becomes a scaler which counts to 2,4 or 6 depending on whether the problem has one, two or three dimensions. A slight complication is added by the fact that the output must be stored till the appropriate clock phase for that unit to emit a pulse to its neighbours. However, the final unit is considerably simplified.

An important advantage of using a scaler as the basic unit is that similar scalers can be used for the potential register of the unit. The circuitry proposed would need the same number of components for a count of ten scaler as for a count of two one, so that fewer stages need be used in the potential register than if binary stages had to be used.

2.7 Arrangement and Phasing of Units. The phasing of the system must be so chosen that a unit receives a pulse from only one of its neighbours at a time. This involves the use of several phases per clock cycle, 3 for a one-dimensional system, 5 for two dimensions and 7 for three dimensions. This can be seen by examination of fig. 2.7.1 .

Referring to the two-dimensional case of fig. 2.7.1(b), the operation is as follows. A unit emits a pulse, provided it has received enough pulses in the last clock cycle, to all asjacent points at its clock phasw. The numbers in the mesh indicate the appropriate phases for each point. All input pulses to a unit go into a scaler which, when it has received 4 pulses, resets itself to zero and sets the output store so that this unit is ready to emit a pulse at its clock phase. If the scaler does not reach 4, it simply atays in the same state until it receives more pulses, however many clock cycles later, and there is no output from the unit so long as 4 is not exceeded. There is always enough storage space in the unit, even if it enters a cycle with 3 in the scaler and receives a pulse from each neighbouring unit. At the end of the cycle there will be 3 in the scaler again and the output store store will be set.

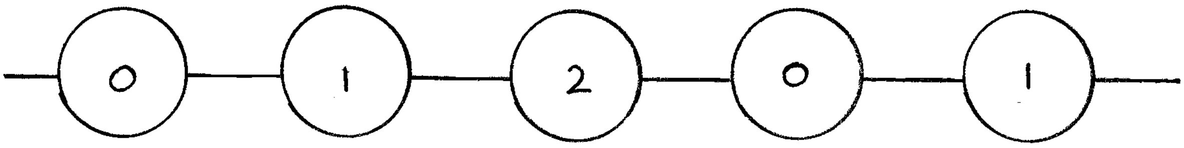
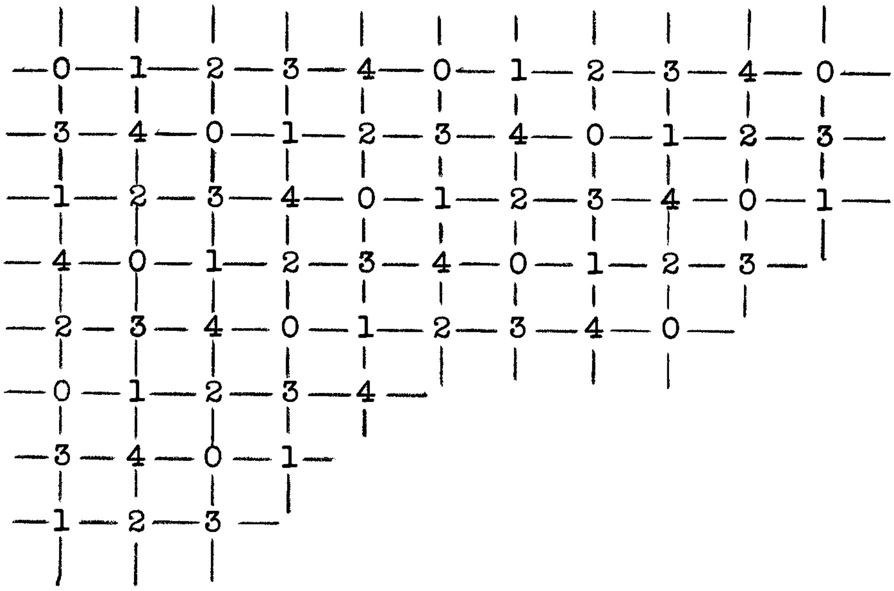
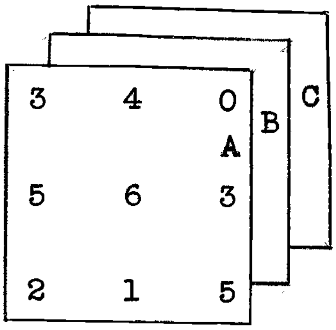


Figure 2.7.1 (a)



```
0 1 2 3 4 0 1 2 3 4 0
3 4 0 1 2 3 4 0 1 2 3
1 2 3 4 0 1 2 3 4 0 1
4 0 1 2 3 4 0 1 2 3
2 3 4 0 1 2 3 4 0
0 1 2 3 4
3 4 0 1
1 2 3
```

Figure 2.7.1 (b)



Plane 'A':-

4	0	2	1	5		
6	3	4	0	2		
1	5	6	3	4	0	2
0	2	1	5	6	3	
3	4	0	2	1		

Plane 'B':-

1	5	6	3	4	
0	2	1	5	6	
3	4	0	2	1	5
5	6	3	4	0	
2	1	5	6	3	

Plane 'C':-

3	4	0	2	1
5	6	3	4	
2	1	5	6	
	0	2	1	5
6	3			

Figure 2.7.1 (c)

3 PRINCIPLES OF USE AND OPERATION OF FERRITE CORES

3.1 Magnetic Properties of Ferrite Cores. The applications of toroidal ferrite cores only will he dealt with. They will, in general, be wound with more than one winding, usually all having a multiple number of turns. The important properties of such components depend on theit ability to store iformation indefinitely, without donsuming any power, and on their rectangular B-H loops.

The latter property gives the core a 'threshold' property. This means that when the sum of the m.m.f!s caused by the currents flowing in the core windings exceeds a certain threshold value, then the core starts 'switching' or, in other words, the magnetic flux in the core starts to change. If the m.m.f. remains large enough for long enough, then the core will end up sarurated in the opposite sense from that in which it started.(The applied m.m.f. is presumed to be applied in the opposite direction to a previous 'setting' m.m.f.). This is the way in which such cores are most commonly used, in memory matrices, shift registers and logical 'and' and 'or' elements. The core always ends up saturated one way or the other, these states being designated '0' and '1' usually.

3.2 Simplifying Assumptions.[7] It is usual to make some simplifying assumptions about core behaviour, in order to facilitate calculations. These are generally, that the rate of change of flux,dØ/dt, is constant

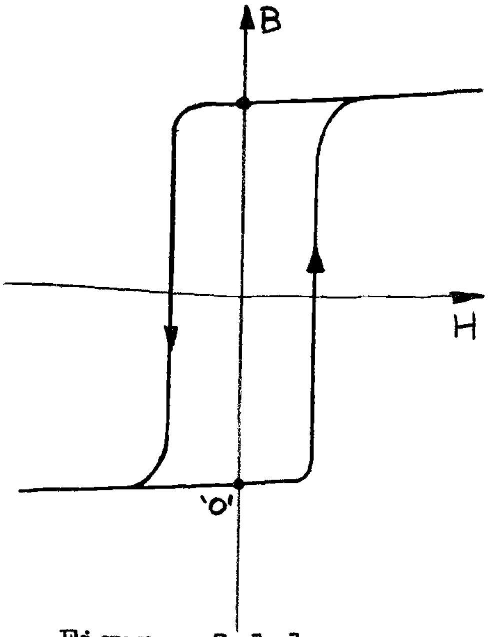


Figure 3.1.1

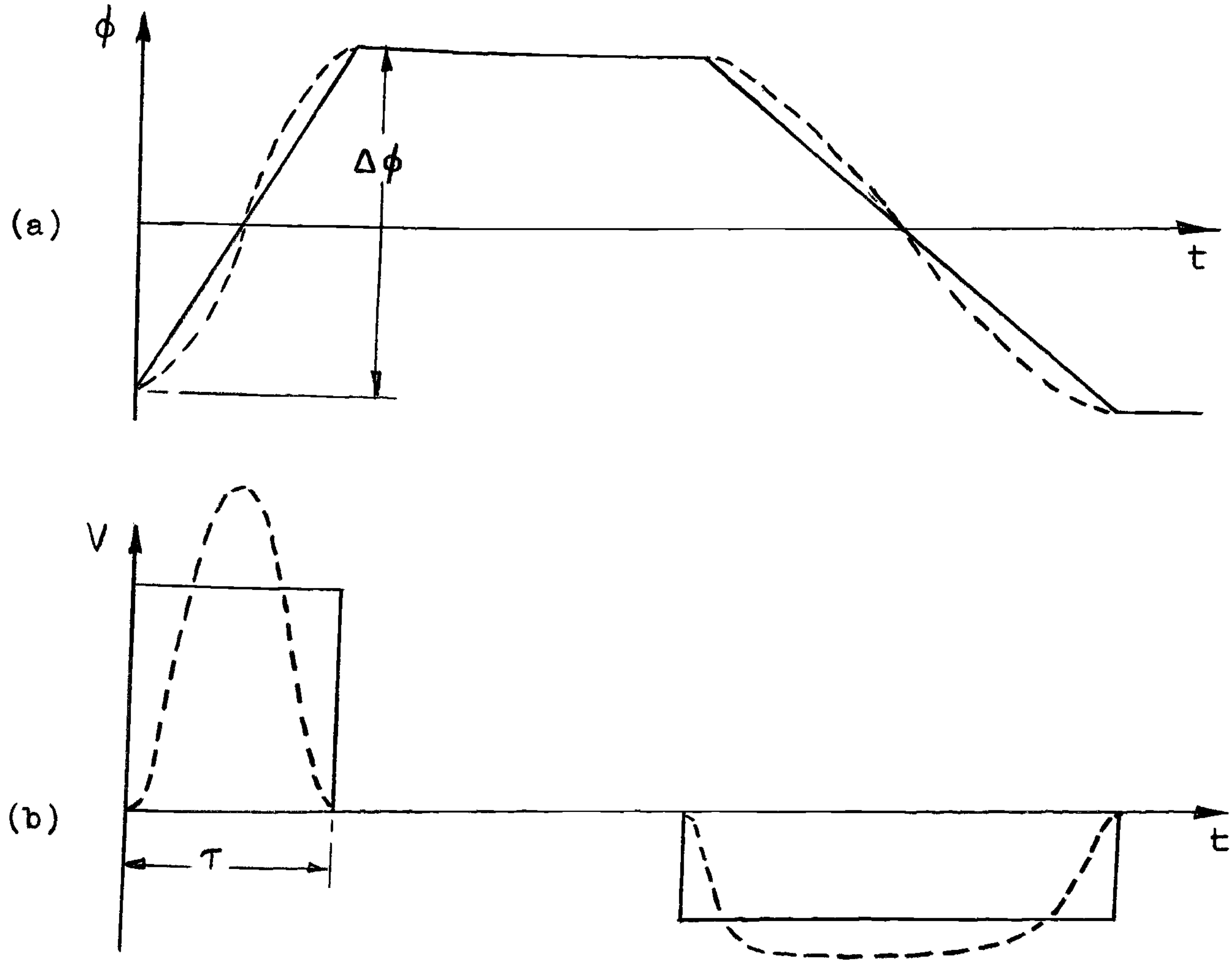


Figure 3.2.1

during switching and that the switching time is inversely proportional to the excess of applied m.m.f. over some critical value, which can be taken as the threshold switching value for most purposes.

By Faraday's Law, the e.m.f. across a winding, $V = Nd\phi/dt$ or, with our simplifying assumptions, $V = N\Delta\phi/\tau$, where τ is the time for switching from one fully saturated state to the other, and $\Delta\phi$ is the resultant flux change. Now $\int Vdt = \int Nd\phi$, so that for a given flux change, $\Delta\phi$, the ' volt-seconds' integral is independent of the actual speed of switching. Therefore, in fig. 3.2.1(b) the area under the 'assumed' rectangular pulse is made equal to that under the 'actual' pulse. This 'volt-seconds area' is an important parameter of the core.

The switching time, $\tau = s_W/(I - I_C)$ ...(3.2.1) where I_C is approximately the 'threshold' m.m.f. and I is the sum of the m.m.f.'s produced by the currents in all the windings. S_W, the 'Switching Constant', depends on the core material and dimensions and on the amount of flux which is switched. Equation 3.2.1 is obtained experimentally and it is found that most cores obey it reasonably well.

<u>3.3 Partial Switching</u>. The core can be used to store analogue rather than digital information by only partly switching it. Referring to fig. 3.3.1, suppose we start with the core at A and supply sufficient m.m.f. to

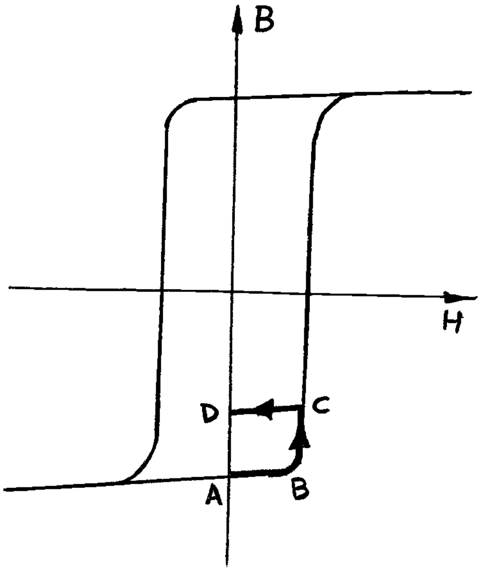


Figure 3.3.1

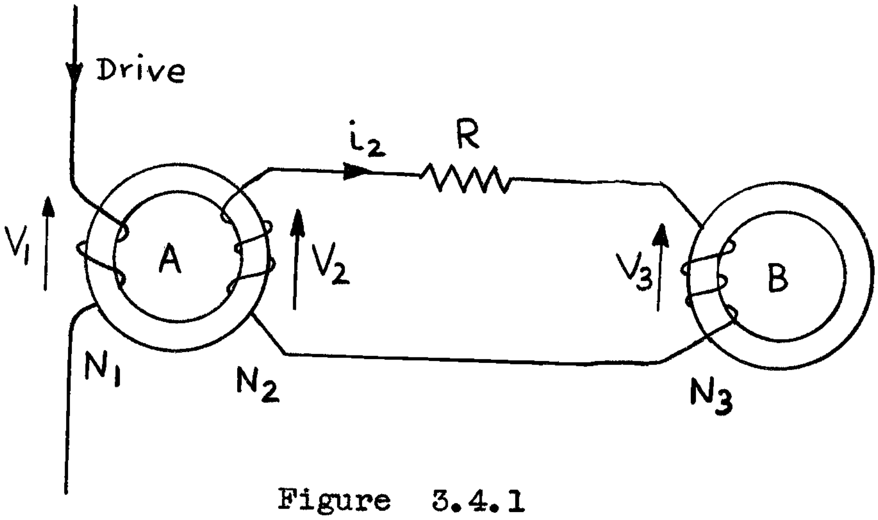


Figure 3.4.1

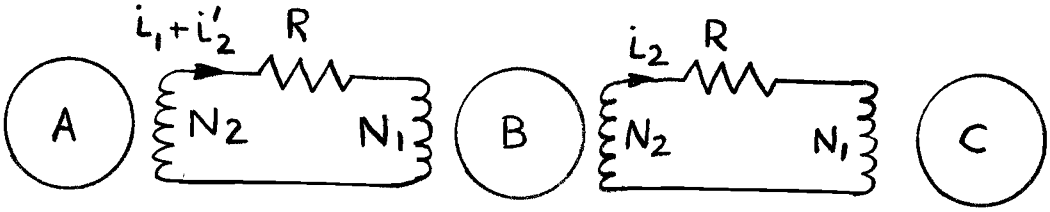


Figure 3.5.1

bring the core to B , at which point it starts switching. If the input is now removed before switching has been completed, at point C for instance, then the final state of the core is at point D. The quantity of flux switched is directly proportional to the 'volt-seconds' appearing across a winding on the core. For this reason it is apperently possible to make a reliable digital scaler using such a core , if the input is provided from another identical core which always switches completely and so always provide pulses of constant volt-time area whatever its switching speed. Temperature variations will affect both cores equally and so the count should not be affected. The scaling core must be reset as soon as it is saturated.

3.4 Cores in Circuits. In almost all machines using cores as computing elements, the arrangement is basically as shown in fig. 3.4.1, where information flows from A to B when the drive winding on A is energised. Core B may switch completely, partly, or not at all, depending on the turns ratio N_2/N_3, the resistance of the coupling windings R, and the magnitude of the drive current or voltage. The e.m.f. across N_2, $V_2 = N_2(d\phi/dt)_A$, and across N3, $V_3 = N_3(d\phi/dt)_B$, so that,

$$N2(d\phi/dt)_A = N_3(d\phi/dt)_B + i_2R \quad ....(3.4.1)$$

If B is to switch completely, then $(d\phi/dt)_B$

must be greater than or equal to $(d\phi/dt)_A$, so that B completes switching while V_2 is still present. A ratio N_3/N_2 greater than unity must therefore be provided if complete switching is wanted. However, partial switching will occur if, at any stage, i_2N_3 is greater than the threshold m.m.f. (I_c) for core B. If the value of R is big enough, then for a given drive voltage (assuming constant voltage drive), and hence constant $(d\phi/dt)_A$, core B will not switch at all. We have:

$$N_2(d\phi/dt)_A = (I_B/N_3)R + N_3(d\phi/dt)_B \qquad (3.4.2)$$

I_B is the m.m.f. on B. For no switching, $(d\phi/dt)_B = 0$ and so,

$$\frac{I_cR}{N_3} \geqslant N_2\left(\frac{d\phi}{dt}\right)_A \qquad ...(3.4.3)$$

or, $$R \geqslant \frac{N_3N_2}{I_c}\left(\frac{d\phi}{dt}\right)_A \qquad ....(3.4.4)$$

Where $(d\phi/dt)_A$ is the maximum value attained, so that switching would start once a critical value of it is reached.

3.5 Necessity for Isolation Between Stages. When toroidal cores are used as logic or counting elements it is generally necessary to provide some form of isolation between stages, for two main reasons:-

a) So that flow of flux in the forward direction switches only the required cores, and to the correct extent.

b) To avoid backward flow of information.

The most important ways of achieving this are by the use of semiconductor diodes or transistors or by using additional cores in the coupling loops. The last method will not be discussed, as considerable complication seems to be needed and this was not considered worth while with the limited facilities for core winding available.

Consider three cores connected as in fig. 3.5.1, wich is two of the stages of fig. 3.4.1 cascaded . When core A switches,

$$N_2\left(\frac{d\phi}{dt}\right)_A = (i_1 + i_2')R + N_1\left(\frac{d\phi}{dt}\right)_B \qquad \ldots(3.5.1)$$

It is assumed that C does not switch at all, as will be required generally. i_2' is the current in the second loop referred to the first loop. Now, $i_2 = (d\phi/dt)_B . N_2/R$ as there is no back e.m.f. due to $(N_1)_C$, so $i_2' = \frac{N_2^2}{N_1}\left(\frac{d\phi}{dt}\right)_B . \frac{1}{R}$, hence :

$$N_2\left(\frac{d\phi}{dt}\right)_A = i_1R + \left(\frac{N_2^2}{N_1} + N_1\right)\left(\frac{d\phi}{dt}\right)_B \qquad \ldots(3.5.2)$$

However, $(d\phi/dt)_B = (i_1 - i_o)/s$, or $i_1 = s(d\phi/dt)_B + i_o$ where s is a constant for the core (from eq. 3.2.1).

So,

$$N_2\left(\frac{d\phi}{dt}\right)_A = i_oR + \left(\frac{d\phi}{dt}\right)_B\left\{N_1 + \frac{N_2^2}{N_1} + s\right\} \qquad \ldots(3.5.3)$$

and,

$$\frac{d\phi}{dt}_B = \frac{(d\phi/dt)_A - i_oR/N_2}{(N_1/N_2 + N_2/N_1) + s/N_2} \qquad \ldots(3.5.4)$$

$(N_1/N_2 + N_2/N_1)$ cannot be less than 2 and so, even if $R = 0$, $(d\phi/dt)_B$ is less than $\frac{1}{2}(d\phi/dt)_A$, so core B would be less than half switched. The system would

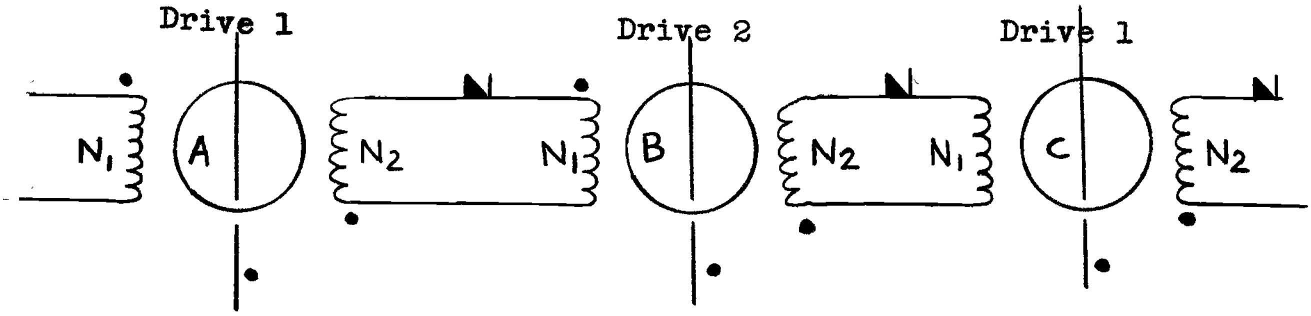


Figure 3.5.2

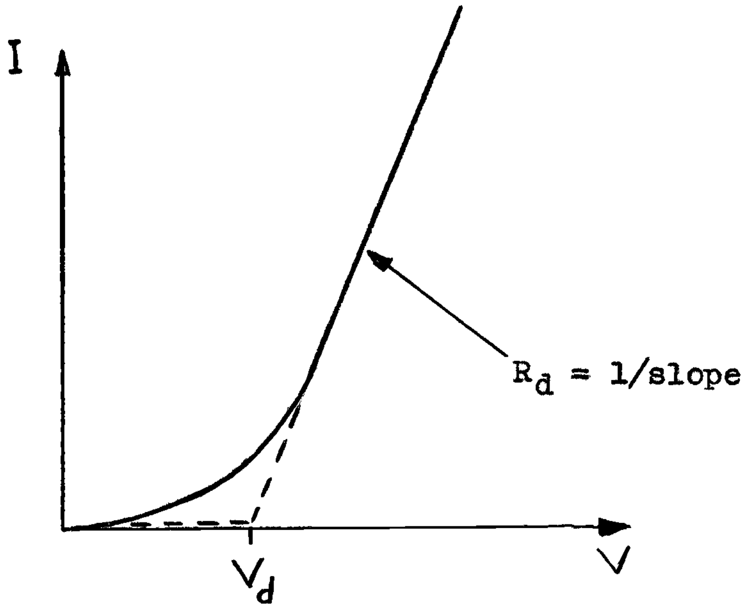


Figure 3.8.1

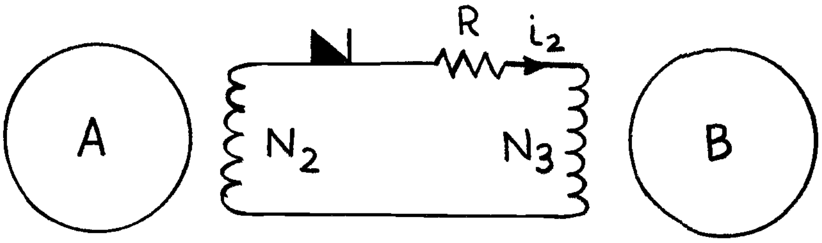


Figure 3.8.2

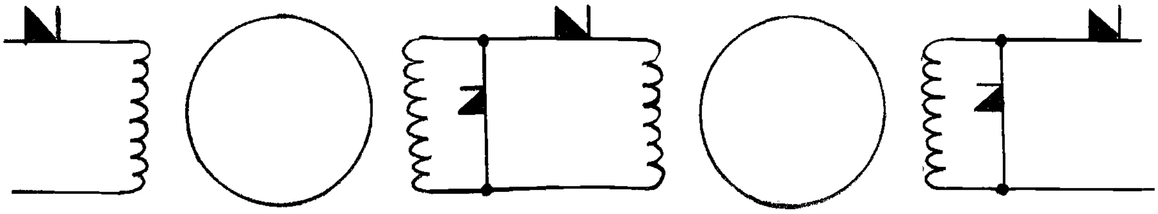


Figure 3.5.3

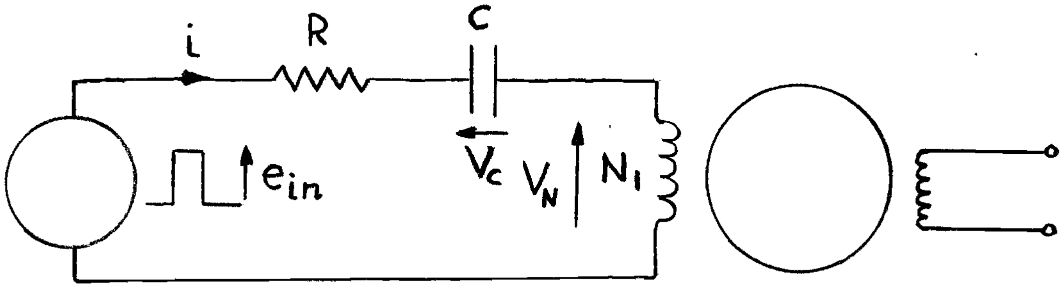


Figure 3.7.1

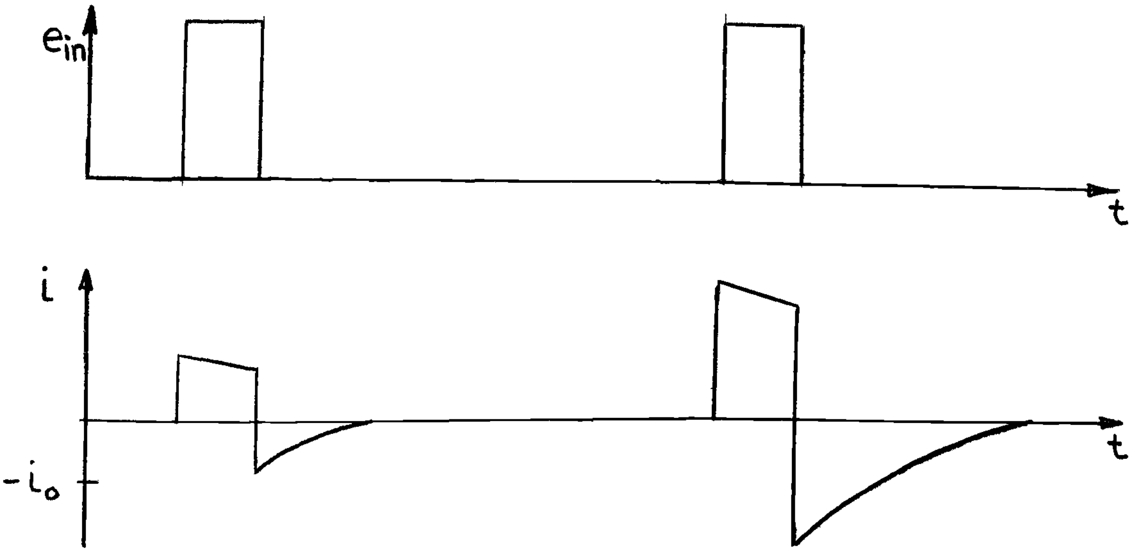


Figure 3.7.2

therefore not be regenerative as a shift register should be, but would attenuate input pulses. It is therefore essential that core C should beisolated from B while flux is being transferred from A to B. The simplest and most commonly used method is to place a diode in each coupling loop, as in fig. 3.5.2 . This is effective in preventing current from flowing in the secondary loop and so complete flux transfer is possible provided that a sufficient turns ratio, N_2/N_1, is provided to overcome winding resistances and diode losses.

Unfortunately, such a diode is not effective in preventing backward transfer. If a drive pulse is applied to core B, for example, while it is in the '1' state, then not only is there a current in the forward direction in the loop to the right of the core, but there is also a current induced in the loop to the left. This current is in the forward direction of the diode, but if the ratio N_2/N_1 is large enough this does not result in reverse flux transfer because the threshold of the previous core is not exceeded. However, in some cases, it has been found advisable to include another diode to short out this current, as in fig. 3.5.3 . This is not altogether satisfactory, since quite large currents can flow in this extra diode and the losses involved have to be supplied by the clock pulse and so the power consumption may rise very considerably.

3.6 Isolation in Counting Circuits. In the counting circuits employed in this computer, the conditions imposed are more stringent than for shift registers. For the scaler to count to a base greater than 2, the ratio N_2/N_1 must be less than unity, considerably less for a count of 10 scaler, so that backward propagation becomes an important problem. Power consumption also must be kept to an absolute minimum, so the two diode system is not satisfactory. It was found that only by using a transistor in the coupling loop was it possible to achieve acceptable performance.

Consider the arrangement of figP 3.5.2, in which each stage is identical. The condition for forward switching to start is:-

$$N_2\left(\frac{d\phi}{dt}\right)_A > \frac{I_c}{N_1} R \qquad \ldots.(3.6.1)$$

i.e. there must be enough voltage to exceed the threshold for any switcing at all to occur.

Similarly, for backward switching to start,

$$N_1\left(\frac{d\phi}{dt}\right)_B > \frac{I_c}{N_2} R \qquad \ldots.(3.6.2)$$

Now, $(d\phi/dt)_B = (d\phi/dt)_A$, as the conditions are the same for the cores in the two cases. Therefore, both the above equations are identical and if the counter works at all, i.e. eq. 3.6.1 is obeyed, then eq. 3.6.2 is also obeyed and there is bound to be backward flux transfer unless proper isolation is provided.

3.7 Counter Circuits Using Cores.[8] The simplest possible basic element which could be used in the machine is a count of two scaler. Such a scaler could be used directly as the mesh-point unit in a one-dimensional system, or several may be connected together suitably to form a unit for two and three dimensional systems.

Such a scaler can be made using one magnetic ferrite core as the counting element. A simplified diagram is shown in fig. 3.7.1 . The pulse generator provides unidirectional pulses of closely controlled volt-time area, as in fig. 3.7.2 . The first pulse reverses the core completely. In so doing a certain current must pass and so the capacitor C is charged to some voltage. When the pulse stops, this charge flows back through the pulse generator and the core winding. The values of C and the total series resistance must be so chosen that this reverse current is below the threshold current for the core, so that no reswitching can occur.

The second pulse is now fed in. As the core is saturated, no back e.m.f. will appear across N_1 and so a relatively large current will flow, thus charging up C to a considerably higher potential than for the first pulse. This time, therefore, there will be a larger reverse current when the pulse stops and if the circuit parameters are correctly chosen, it will completely reset the core. The counter is now ready for the next input pulse and the core is in its original state.

During the first pulse,

$$e_{in} = i R_g + \frac{1}{C}\int_0^t i \, dt + N_1 \frac{d\phi}{dt} \qquad ...(3.7.1)$$

The switching time is related approximately to the m.m.f. by the relation

$$\tau = \frac{S_w}{I - I_o} = \frac{s}{i - i_o} \qquad ...(3.7.2)$$

where $s = S_W/N_1$. This switching time is for complete switching of the core, the flux change being ϕ_c. So we can write :

$$V_{N1} = N_1 \frac{d\phi}{dt} \doteq N_1 \frac{\phi_c}{s}(i - i_o) \qquad ...(3.7.3)$$

hence, from eq. 3.7.1 ,

$$e_{in} = i\left(R + N_1 \frac{\phi_c}{s}\right) - i_o \frac{N_1\phi_c}{s} + \frac{1}{C}\int_0^t i \, dt \qquad ..(3.7.4)$$

Putting

$$A = \left(e_{in} + i_o \frac{N_1\phi_c}{s}\right) \; ; \; B = \left(R + \frac{N_1\phi_c}{s}\right) \qquad ..(3.7.5)$$

the solution to eq. 3.7.4 is

$$i = \frac{A}{B} e^{-t/BC} \qquad ..(3.7.6)$$

The potential across the capacitor at the end of the first pulse must not be large enough to cause the reverse current to exceed the threshold value for the core. So,

$$\frac{1}{C}\int_0^T \frac{A}{B} e^{-t/BC} < i_o \qquad \ldots(3.7.7)$$

T, the core switching time being equal to the input pulse length. Therefore the circuit parameters must be so chosen that:

$$A e^{-T/BC} < i_o \qquad \ldots(3.7.8)$$

A more approximate expression can be deduced if it can be assumed that during the pulse the current does not change very much. Then the capacitor charges to $i_{mean} T/C$, and so the condition for no reswitching is

$$RC > \frac{i_{mean}}{i_o} T \qquad \ldots(3.7.9)$$

which is useful as a guide in designing such a circuit.

At the end of the second pulse,

$$e_{in} = iR + \frac{1}{C}\int_0^T i\,dt \qquad \ldots(3.7.10)$$

the solution of which is

$$i = \frac{e_{in}}{R} e^{-t/RC} \qquad \ldots(3.7.11)$$

and

$$\int_0^T i\,dt = RC\left(1 - e^{-T/RC}\right)\frac{e_{in}}{R} \qquad ..(3.7.12)$$

So that the capacitor has a potential difference $e_{in}\left(1 - e^{-T/RC}\right)$ across it when the pulse is removed. Calling this V_0, we have, if reswitching now starts:-

$$V_o = i\left(R + N\frac{\phi_c}{S}\right) - i_o N\frac{\phi_c}{S} + \frac{1}{C}\int i\,dt \qquad \ldots(3.7.13)$$

where ϕ_c is the maximum possible flux change in the core.

Putting

$$D = \left(V_0 + i_0 N \frac{\phi_c}{s}\right), \quad B = \left(R + N \frac{\phi_c}{s}\right) \qquad ..(3.7.14)$$

the solution to eq. 3.7.13 is

$$i = \frac{D}{B} e^{-t/Bc} \qquad ...(3.7.15)$$

Re-switching will cease when i drops below i_0 , so the time for which the core is switching is given by:-

$$i_0 = i = \frac{D}{B} e^{-T_2/Bc} \qquad ...(3.7.16)$$

or, $$T_2 = Bc \ln \frac{D}{Bi_0} \qquad ...(3.7.17)$$

Therefore the flux change is:-

$$\Delta\phi = \int_0^{T_2} \frac{\phi_c}{s} (i - i_0)\, dt \qquad ...(3.7.18)$$

$$= \frac{\phi_c}{s} \int_0^{T_2} \left\{ \frac{D}{B} e^{-t/Bc} - i_0 \right\} dt$$

$$= \frac{\phi_c}{s} \left\{ Dc - i_0\, Bc\left(1 + \ln \frac{D}{Bi_0}\right) \right\}$$

or, $$\frac{\Delta\phi}{\phi_c} = \frac{c(V_0 - i_0 R) - i_0 T_2}{s} \qquad ...(3.7.19)$$

and values must be chosen so that this ratio is greater than one.

3.8 Effect of Winding and Diode Losses. The degradation of performance of the core used as a counter, due to the presence of resistance and semiconductor diode or transistor in the coupling loop, will now be considered.

The diode can be more easily treated analytically if its characteristic is divided into two straight-line portions , as in fig. 3.8.1 . It can then be treated as a resistance, R_d, in series with a constant voltage drop, V_d.

Referring to fig. 3.8.2, we have:-

$$N_2\left(\frac{d\phi}{dt}\right)_A = N_3\left(\frac{d\phi}{dt}\right)_B + R\,i_2 + V_d \qquad \text{...(3.8.1)}$$

R includes winding, diode and any additional resistance. Putting in the usual approximations, this becomes

$$N_2\frac{\Delta\phi_A}{T_A} = N_3\frac{\Delta\phi_B}{T_B} + R\,i_2 + V_d \qquad \text{...(3.8.2)}$$

hence, $i_2 = N_2\dfrac{\Delta\phi_A}{R\,T_A} - \dfrac{V_d}{R} - \dfrac{\Delta\phi_B N_3}{R\,T_A}$ ...(3.8.3)

But also, $i_2 = i_c + \dfrac{S}{T_B}$ ...(3.8.4)

and as T_B is the time required to switch B completely, we have $T_B = T_A\dfrac{\Delta\phi_{max}}{\Delta\phi_B}$ ...(3.8.5)

The cores are identical, so $\Delta\phi_{max.B} = \Delta\phi_A$ and so,

$$i_2 = i_c + \frac{\Delta\phi_B}{\Delta\phi_A}\cdot\frac{S}{T_A} \qquad \text{...(3.8.6)}$$

Therefore,

$$(n-1) = \frac{\Delta\phi_A}{\Delta\phi_B} = \frac{N_3 + SR/\Delta\phi_A}{N_2 - T_A(R\,i_c + V_d)/\Delta\phi_A} \qquad \text{...(3.8.7)}$$

where n is the base to which the scaler counts.

As far as performance is concerned, it is advantageous to make N_3 and N_2 large, so that it is their ratio which is the main factor in determining to what base the scaler counts, even when τ_A , R and V_d may vary. However, it is desirable that the number of turns in the windings should be kept within reasonable limits, both as regards the cost of winding the cores and the physical impossibility of winding more than a certain number of turns on the small cores used. The quantity which is most likely to vary in the expression for $\Delta\phi_A/\Delta\phi_B$ is the switching time, τ_A . This is due mainly to the fact that the pulse generators will be supplying a number of units, some of which may switch while others may not, so that the loading on the generator will vary and soits output is likely to vary. Supply voltage variations will also affect it. So matters must be arranged that the denominator of eq. 3.8.7 should vary little for the variations of τ_A likely to be encountered in practice. For example, if $N_2 = 8$ and $N_3 = 64$ and the scaler counts to 12, then the variation of the denominator must not be enough to affect this count, the limit being ± ½ a count. Then the maximum permissible variation in switching time is

$$\delta\tau_A < \frac{\left\{ \frac{N_2 \Delta\phi_A}{RI_c/N_3 + V_d} - \tau_A \right\}}{24} \qquad \ldots(3.8.8)$$

using the FX 1724 core, $I_c = 1.23$ AT, $\Delta\phi_A = 3.81 \times 10^{-7}$ wb., and with typical values of $v_d = 0.2$ v., $R = 3\Omega$, $\tau_A = 2\mu s$

$\delta \tau_A$ must be less than 0.4 μS. This value is not greatly affected by what the value of τ_A is , so it is an advantage, again , to make τ_A small normally in order that a larger percentage variation be allowable.

For a scaler investigated experimentally, $N_2 = 7$ and $N_3 = 40$,

τ_A	$\Delta\phi_A/\Delta\phi_B$	Theoretical n
1 μS	7.7	9
2	9.6	11
3	11.2	13

Experimentally it was found that a particular count was stable for about ± 10% changes in τ_A, but the exact count depended , in addition to the factors considered above, on the means used for triggering the transistor used for resetting the core when the count was completed. The actual count could vary by 1 or 2 from that predicted.

Fig. 3.8.3 shows the predicted flux change ratio for a turns ratio of 8. The flux ratio approaches the limiting vale as the switching time is decreased and the total number of turns is increased.

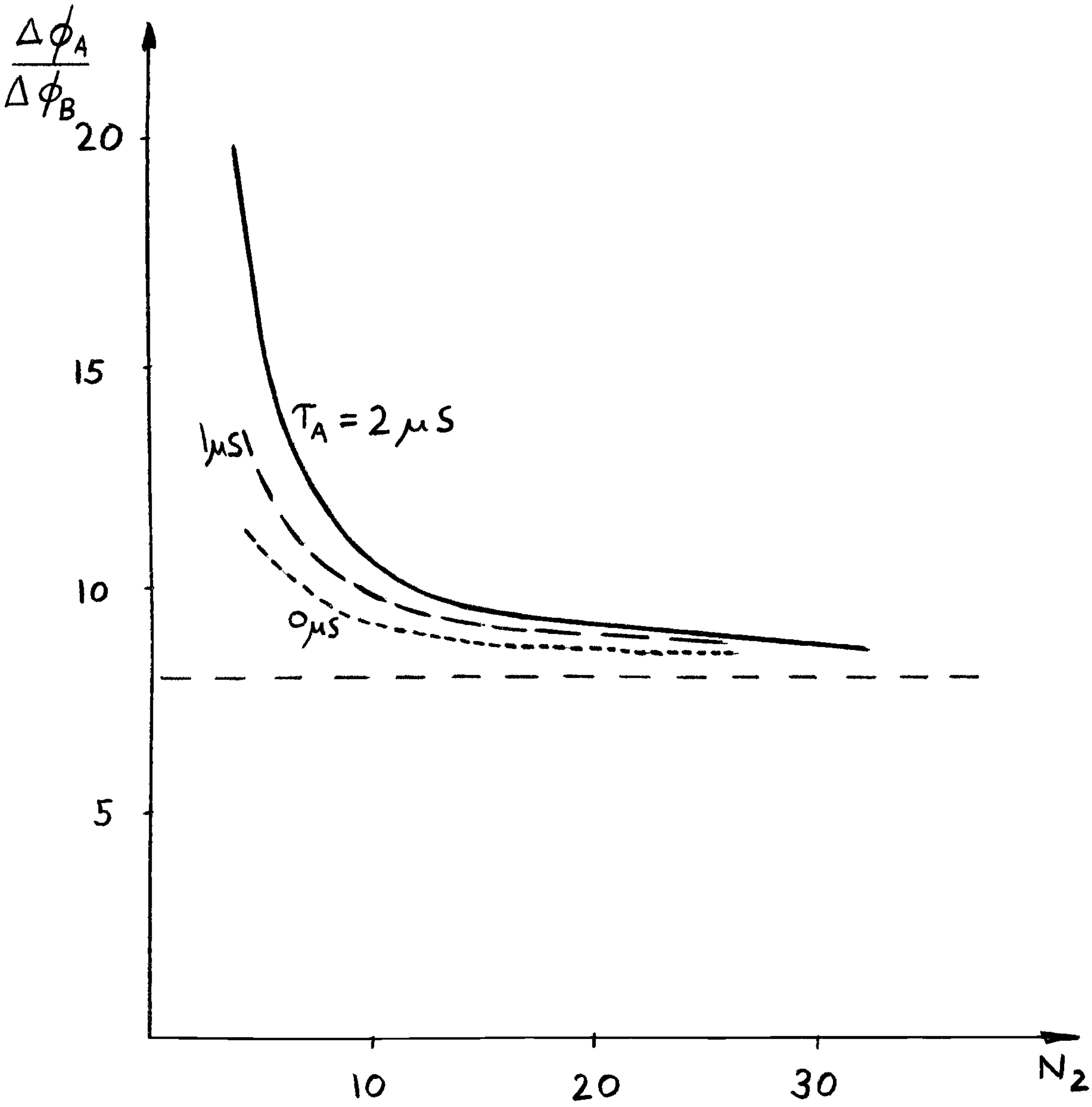

$\frac{\Delta\phi_A}{\Delta\phi_B}$
20
15
10
5
$\tau_A = 2\mu S$
1μS
0μS
10
20
30
N_2

Figure 3.8.3

4 DESIGN OF MESH-POINT UNITS

4.1 Requirements for Unit. The unit must count the total number of pulses arriving on its inputs and for every 2, 4, or 6 input pulses (depending on the number of dimensions in the problem, 1, 2, or3), give an output pulse at the appropriate clock phase. It must therefore be able to store pulses for an indefinite period, as it generally will not be the case that exactly the right number of pulses will arrive during each cycle for an output to be emitted in that cycle

For the moment, attention will be confined to units for the one-dimensional case, so that the scaler must count to two. In the design to be considered there two parts to the sub-unit, a scaler using a magnetic core as described previously, and an output circuit which incorporates another core, or cores. There may also be a register attached to the unit to record the total of pulses emitted, or the potential of the point. This unit will be considered separately.

4.2 Mode of Operation.The output core is 'set' when two pulses have passed into the counter, which must then be automatically reset. The output core is then reset by a clock current at the appropriate phase in the cycle, thus inducing ann e.m.f. on the output winding which is used as the input to the adjacent mesh-point units.

A circuit which works successfully using this system is shown in fig. 4.2.1 . An output pulse from unit n - 1 on clock phase '0' is fed via a diode to the input of the scaler section of unit n. This scaler is of the type which uses a capacitor to reset the core, the reset current flowing through the 15Ω resistor across the input. When the core resets it induces an e.m.f. in the secondary coil which turns on the transistor while the counter core resets. This sets the output core ready to emit a pulse to units n-1 and n+1 at clock phase 1. If the counter core was in the '0' state before the input pulse arrived on phase '0', or if it was in the '1' state and there was no input at phase '0', the core would not reset, the output core would not be set and there would be no output. The counter core can retain a '1' or a '0' indefinitely if there is no input.

These units operate reasonably well, as shown by the pulse shapes shown in fig. 4.2.2 . However, the drive current requirements are quite high and this is not helped by the presence of the 15 ohm resistors across the input of each scaler. Also, although such scalers using capacitor reset are quite easy to design and make for a count of two, they become increasingly unsuitable as the count is increased. This is the result of the fact that the energy of one pulse must be enough to charge the capacitor sufficiently for it to reset the core, which has been set with a number of similar

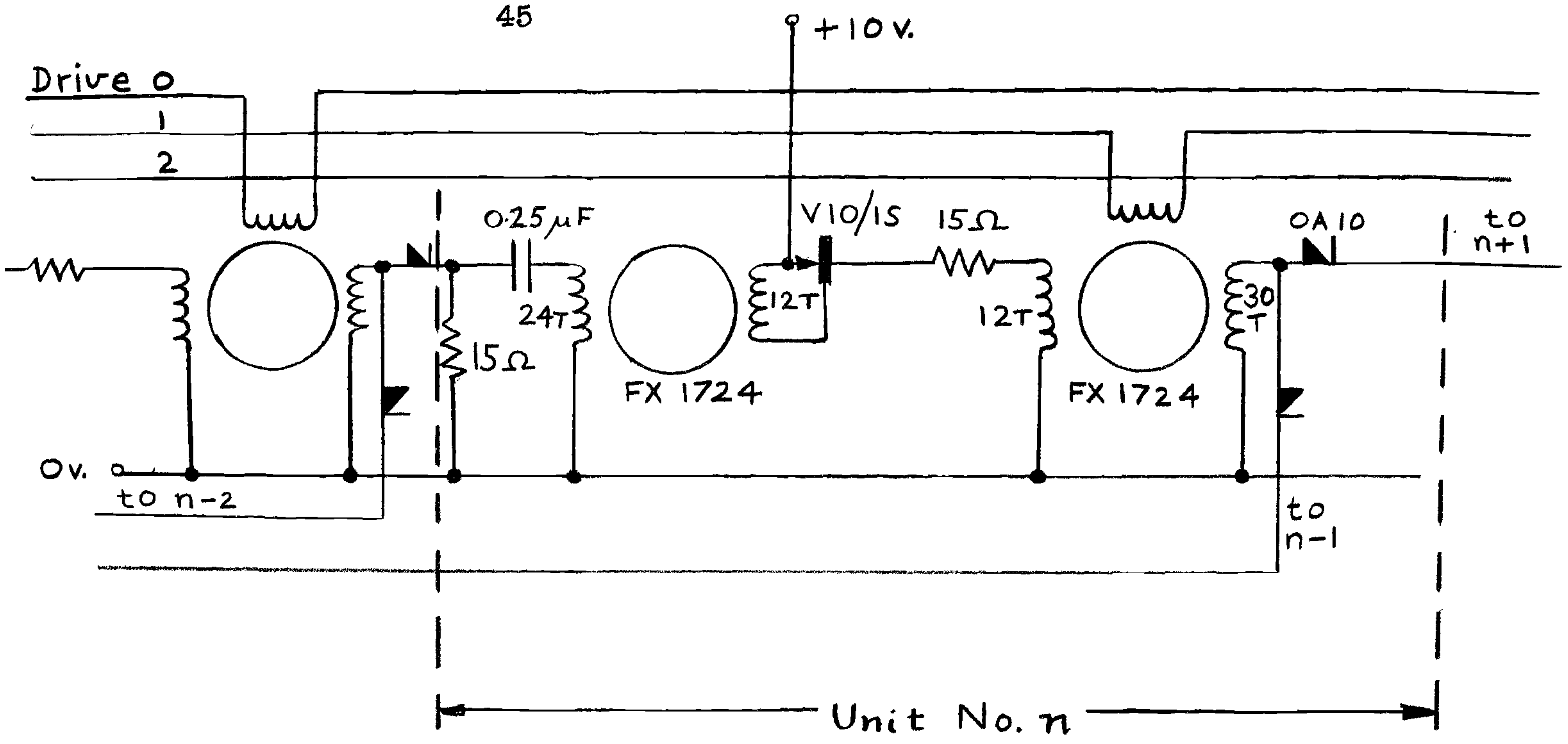
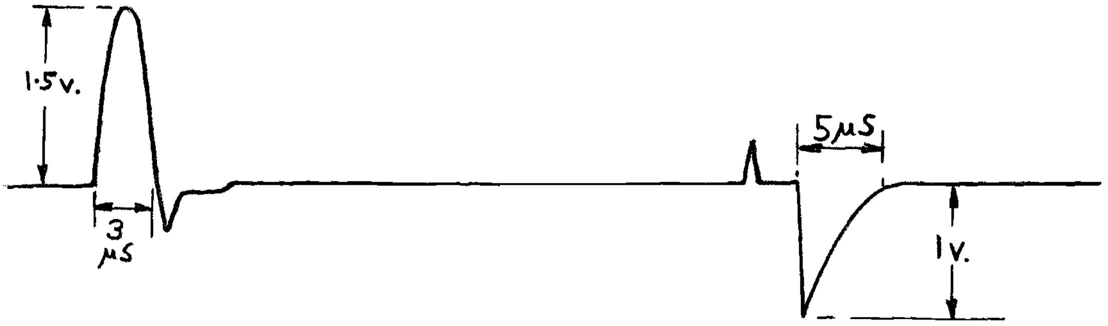
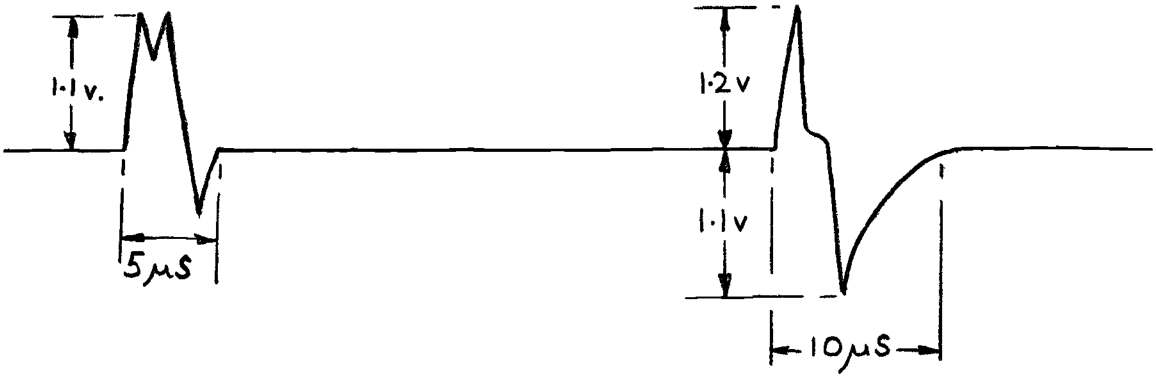


Figure 4.2.1

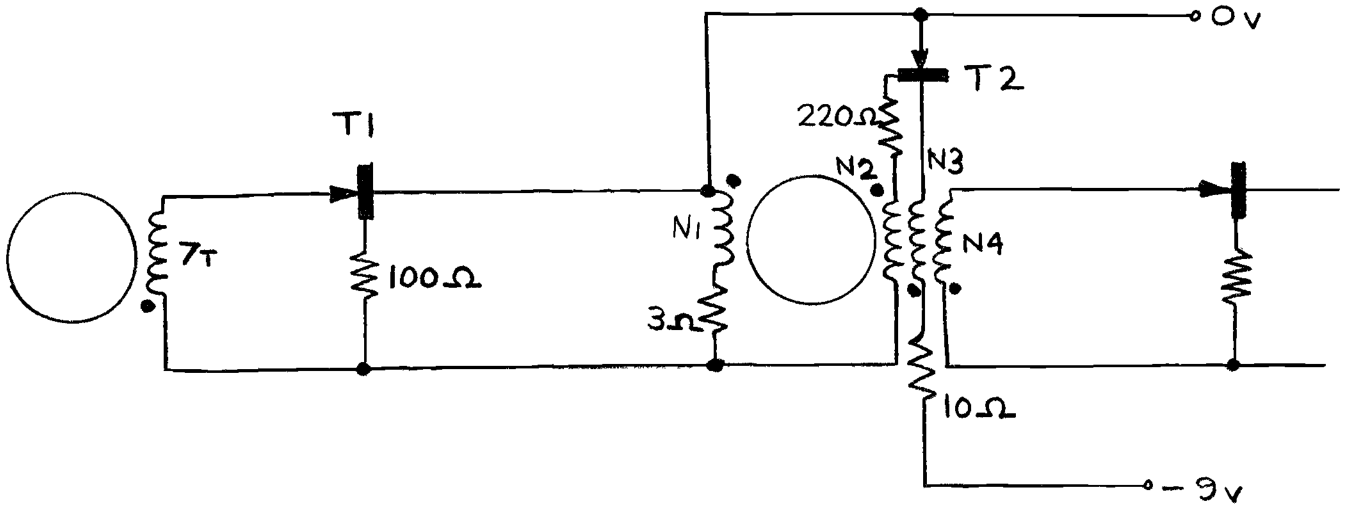


A.- both n-1 and n+1 in the same state



B. - n-1 and n+1 in opposite states

Figure 4.2.2 - Voltage waveforms across output of unit no. n.

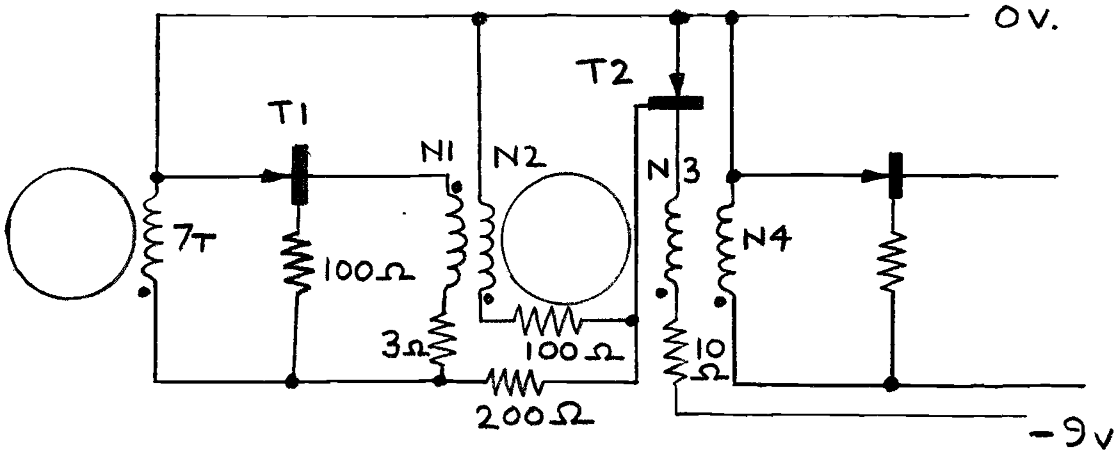


T1 and T2: V10/1S or OC 123

N1 = 40 T.   N2 = 80 T.

N3 = 20 T.   N4 = 7 T.

Figure 4.2.3



T1 and T2: V10/1S or OC 123

N1 = 40turns N2 = 80 turns

N3 = 20   N4 = 7

Figure 4.3.1

pulses. Large voltages and hence large numbers of turns on the windings are therefore needed.

A scaler which has been made to work fairly reliably up to a count of at least 10 is shown in fig. 4.2.3 . This uses two transistors per stage. The first transistor is used in order to completely isolate the previous core from the counter.It allows positive pulses to pass in the forward direction, but prevents negative pulses from going forward and both positive and negative ones from coming backwards.

4.3 Operation of Count of 10 Scaler. Pulses of constant volt-time area are fed into N_1 until the core is set. During this period the voltage induced in N_2 ensures that the base-emitter voltage of T_2 is positive, with the result that no current flows through the transistor and N_3. The first pulse that arrives after the core is set will not induce an e.m.f. in N_2 and furthermore a larger current will flow in N_1 due to the lack of any flux change in the core. This causes the base of T_2 to go negative and a current flows through the reset winding, N_3, the flux change in the core inducing an e.m.f. in N_2 which keeps the transistor turned on as long as the core is switching. The e.m.f. in N_1 is prevented from interfering with the state of the previous core by the presence of the transistor in the coupling loop. The output from N_4 can be used as the input to the next identical scaler.

As soon as the core has been reset it is ready to receive the next input pulse.

The scaler shown was designed to count to the base 10, and it does this for supply voltage variations of about ± 10%. However, the count is dependent to a considerable extent on the values of the resistors in the circuit, and if the transistors are replaced with other examples, of the same type even, the count will probably be changed. Also the operation of the circuit appears to deteriorate with time, several of these scalers refusing to operate correctly after being left for three months without being operated or, as far as is known, interfered with in any way. The critical factor seems to be the triggering of the reset transistor, T_2.

A modified circuit that should be more reliable is shown in fig. 4.3.1 . The emitter of T_1 has been earthed and the base circuit of T_2 has been modified in the hope that it will result in more definite turn-on and turn-off of T_2.

4.4 Drive Pulse Requirements. In the practical work done so far on the basic units and counters, only a few units have been in operation at one time and so drive power has been no problem. The pulse generator designed for this purpose is described in Appendix 3, and has an efficiency of only about 20%. However, in the complete computer this question is rather important.

In order to give reasonable voltage levels in the coupling circuits, so that semiconductor components work efficiently, it is necessary to switch cores in a time of about 1 or 2 microseconds if the number of turns in the core windings is to kept low enough to be a practical proposition. This leads to the requirements for drive current being quite stringent, currents of several amps with rise times of 1 or 2 microseconds being usual.

In development of the circuits so far, the drive has been of the constant voltage type. This means that a certain voltage is applied to the driven core winding during the pulse, and the current is limited principally by the back e.m.f. generated as the core switches. The current cannot, of course, rise to very high values as it is delibarately limited by the circuit design, in order to avoid damaging the drive output transistors. The object of using constant voltage rather than constant current drive is to ensure that the secondary voltage from the core is constant in voltage amplitude. This must be so in order that the inputs connected to this secondary winding shall always receive pulses of nearly the same volt-time area and shape. If this condition is not fulfilled, the counters for which the inputs are may not count reliably. Now, if one of the scaler cores is saturated, the next input pulse to it will be virtually short-circuited. A large secondary current will therefore flow and so a large drive current will be taken. This process is clearly wasteful, as this large current is doing nothing useful.

Referring to fig. 4.4.1, assuming that core 1 switches completely in 2 microseconds, then the e.m.f. across N_1 is about 0.2 volts/turn, and so if it is assumed that the minimum e.m.f. needed for the semiconductor diodes is about 2 volts, then N_1 should be about 10 turns. The four cores A,B,C,D are each to count to 4, as this is a 2-dimensional type basic unit, so that N_2 needs to be about 40 turns.

First, we shall see what drive current is needed if each of the four scaler cores is unsaturated.

The amp-turns needed to switch a core in τ μS is given by

$$I = I_0 + S_W/\tau \qquad \ldots(4.4.1)$$

These scaler cores must switch $\frac{1}{3}$rd as fast as core 1, so $\tau = 6$ μS

For an FX 1724 core,

$$I = 1.23 + 1.6/6 = 1.5 \text{ AT} \qquad \ldots(4.4.2)$$

So, current in each winding = 1.5/40 = 37.5 mA. and current in secondary of corê 1 = 0.15 A.

A convenient value for the drive voltage on core 1 is 10 volts, so there must be 10/0.15 = 67 turns on the drive winding. The drive amp turns must overcome the

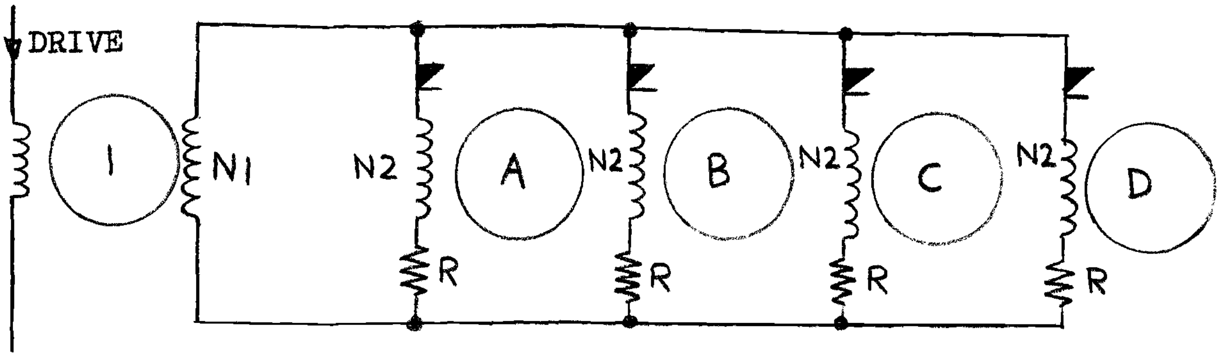


Figure 4.4.1

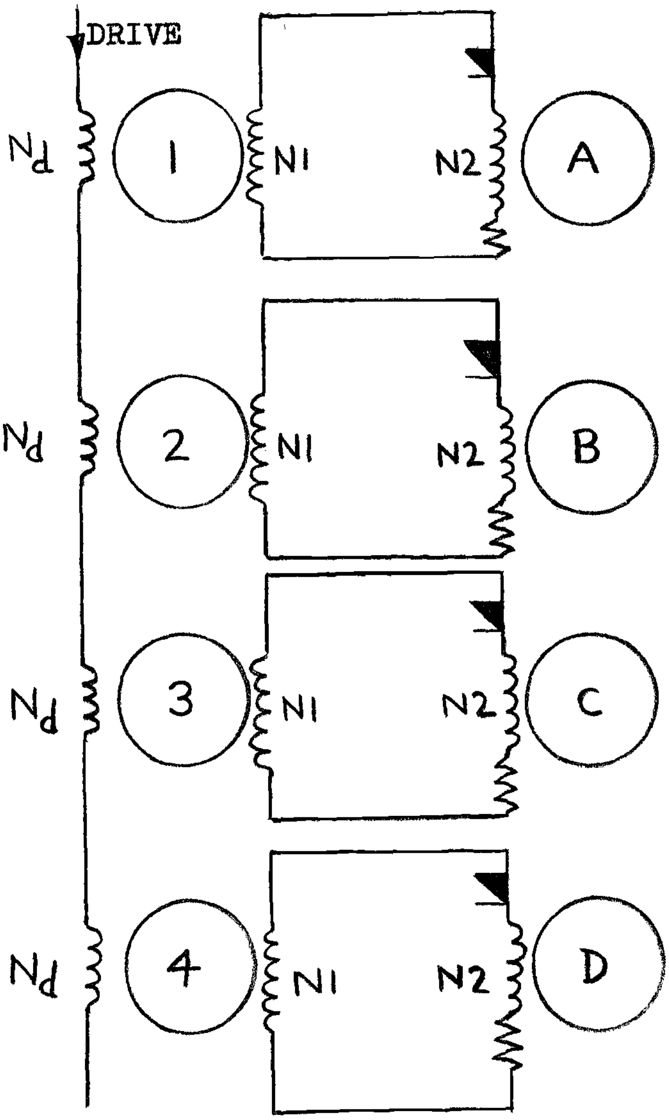


Figure 4.5.1

threshold amp turns, I_c , the secondary AT and S_w/π

So,

$$67\ i_d = 10 \times 0.15 + 1.23 + 1.6/2$$

$$i_d = 3.53/67 = 0.053\ \text{A} \qquad \ldots(4.4.3)$$

where i_d is the current in the drive winding. This current is quite satisfactory, and one OC 23 transistor, which can supply a 1 amp pulse with a fast enough rise time, can therefore supply 1/0.053 = 19 units. However, the situation is entirely different when any of the cores A,B,C,D is saturated so that there is no back e.m.f. and so the current is limited only by the resistance R, which is normally not greater than about 3 ohms. In this case the current becomes 2/3 = 0.67 A. If the worst case is taken in which all the cores are saturated, the secondary current becomes 2.67 A. SO, for the drive curent,

$$50i_d = 10 \times 2.67 + 1.23 + 1.6/2$$

$$i_d = 28.7/50 = 0.57\ \text{A} \qquad \ldots(4.4.4)$$

and so only two units could be supplied by one OC 23. This is rather few and would mean a rather large number of drive transistors, so it is worth trying to find a way of reducing the current needed.

4.5 A Modified Output Circuit. A possible solution lies in the use of separate driven cores for each output channel in conjunction with constant current drive, as shown in fig. 4.5.1 .For any particular driven core there will be two possible conditions when it is driven:

a) The following scaler core is not saturated and so is switching and creating a back e.m.f. The amplitude of the drive current must be such that for this condition the output core switches at the right speed.

b) The following scaler core is saturated. Hence a large current tends to flow in the coupling loop, and with the fixed drive current this means that the driven core tends to switch more slowly. The drive pulse length must therefore be great enough for the core to get completely switched. In addition, the winding (and any added) resistance,R, serves to limit this current and also prevents noise pulses from exceeding the threshold current for switching the scaler cores.

Using the same values of N_1, N_2 and switching times as before, the switching current in the counter windings will take the same values. So, in the case of an unsaturated scaler core, amp turns due to N_1 = 10 x .0375 = 0.375 AT. So, primary AT

$$= I_{sec} + I_c + S_w/\tau$$

$$= 0.375 + 1.23 + 1.6/2 = 2.4 \text{ AT} \qquad ..(4.5.1)$$

If the drive is a constant current pulse of 1A amplitude, then 3 turns are needed for each drive winding, N_d. So the back e.m.f.

$$\therefore \quad 3 \times \Delta\phi/\tau = \frac{3 \times 3.81 \times 10^{-7}}{2 \times 10^{-6}} = .556 \text{ v.} \qquad ..(4.5.2)$$

So that about 2.4 volts per unit is the maximum voltage drop. About 10 units could be fed in series from one constant current generator, as a total drop of 24 volts is reasonable.

Consider now the case of a saturated scaler core. The same analysis as in the constant voltage driven case cannot now be used, and we want to find the length of pulse necessary to ensure complete reversal of the driven core.

The secondary current,

$$i_s = N_1 \frac{d\phi}{dt} \Big/ R \doteqdot \frac{N_1}{R} \cdot \frac{\Delta\phi}{T}$$

Also, $$T = \frac{S_w}{i_d N_d - i_s N_1 - I_c}$$

where i_d is the drive current. Hence,

$$T = \frac{S_w}{i_d N_d - \frac{N_1^2}{R} \cdot \frac{\Delta\phi}{T} - I_c} \qquad \ldots(4.5.3)$$

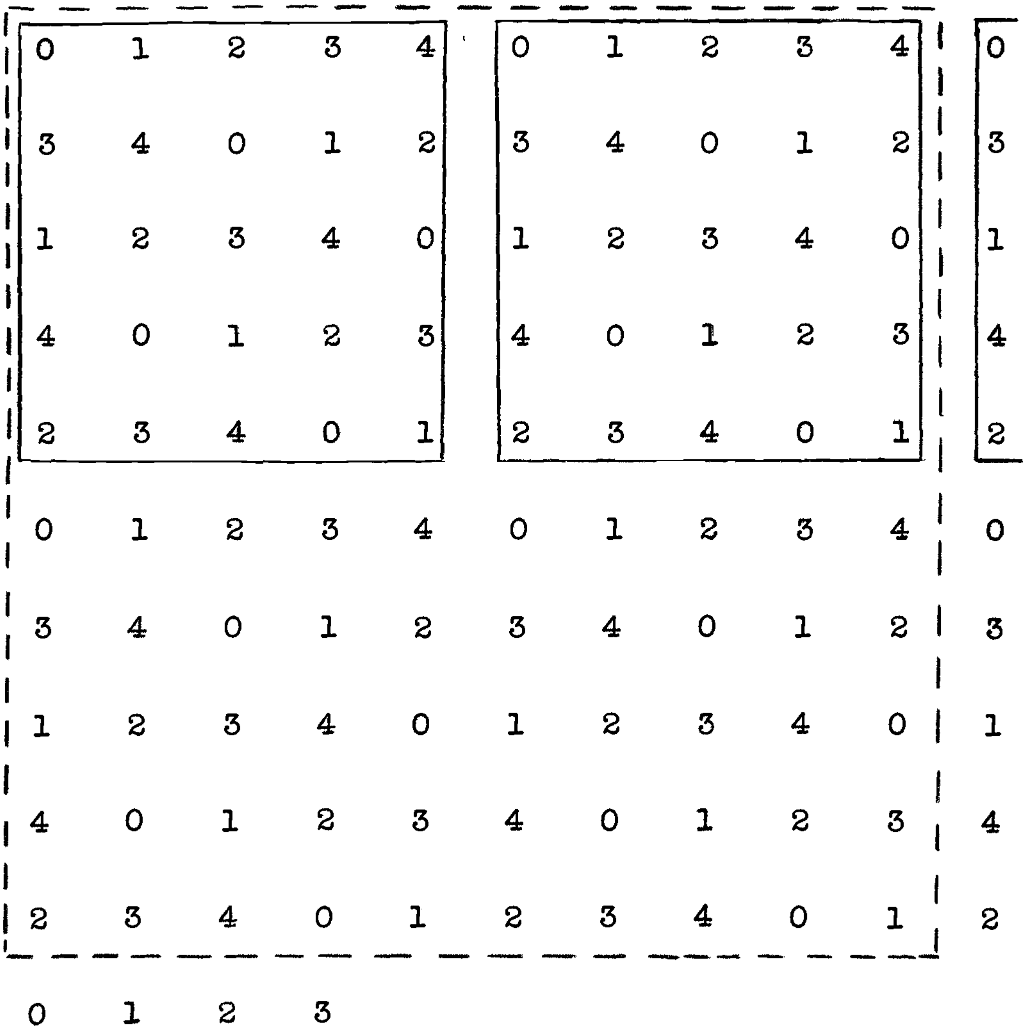
$$T\left(i_d N_d - I_c\right) = S_w + \frac{N_1^2}{R}\Delta\phi \qquad \ldots(4.5.4)$$

$$T = \frac{1.6 \times 10^{-6} + \frac{100}{3} \times 3.8 \times 10^{-7}}{3 - 1.23} = 7.26\mu S \ldots(4.5.5)$$

So the pulse length must exceed 7.26 microseconds. The power used if a pulse length of 10 μS is used in conjunction with a p.r.f. of 10 kc/s is therefore:

$$P = 10^4 \times 10 \times 10^{-6} \times 1 \times 24 = 2.4 \text{ watts} \qquad ..(4.5.6)$$

This is to supply 10 units, so for a small computer of 10 x 10 units the drive power required would be



0	1	2	3	4	0	1	2	3	4	0
3	4	0	1	2	3	4	0	1	2	3
1	2	3	4	0	1	2	3	4	0	1
4	0	1	2	3	4	0	1	2	3	4
2	3	4	0	1	2	3	4	0	1	2
0	1	2	3	4	0	1	2	3	4	0
3	4	0	1	2	3	4	0	1	2	3
1	2	3	4	0	1	2	3	4	0	1
4	0	1	2	3	4	0	1	2	3	4
2	3	4	0	1	2	3	4	0	1	2
0	1	2	3							

Figure 4.5.2

24 watts from 10 pulse amplifiers.

In view of the relatively small number of units which can be driven from one drive pulse amplifier, it seems logical to split up the computer into self-contained modules, say 5 x 5 units in size, incorporating the pulse amplifiers needed for these units. As shown in fig. 4.5.2, the 5 x 5 or 10 x 10 arrangements form convenient patterns, as the computer can be made up from a number of identical such modules. However, the 10 x 10 module would need more powerful drive pulse amplifiers in order that each could drive the 20 units comprising each phase.

4.6 Input Pulse Generators. It is not necessary to have a separate pulse generator for each boundary point. However, the time taken for solution of a problem will depend primarily on the time for setting and reading out the potential values. Therefore, a fair number of generators compared with the number of boundary points likely to be needed should be provided for the machine.

The requirements for the input generator are that it should deliver to a specified mesh-point unit in the machine a specified number of pulses. In a large machine it would probably be most convenient to list both these quantities on magnetic or punched paper tape and arrange that the generators obey instructions in this form.

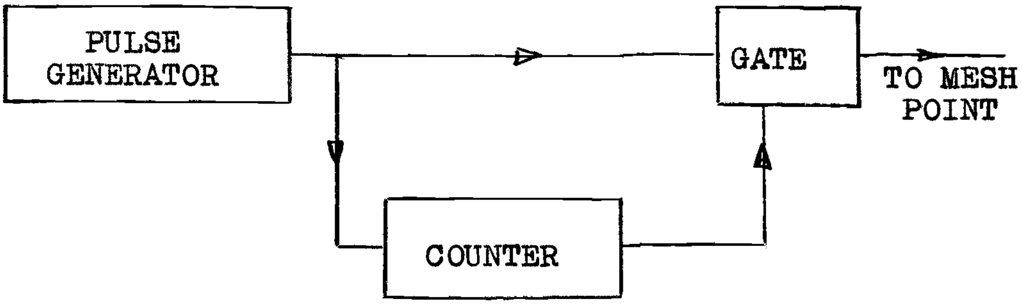


Figure 4.6.1

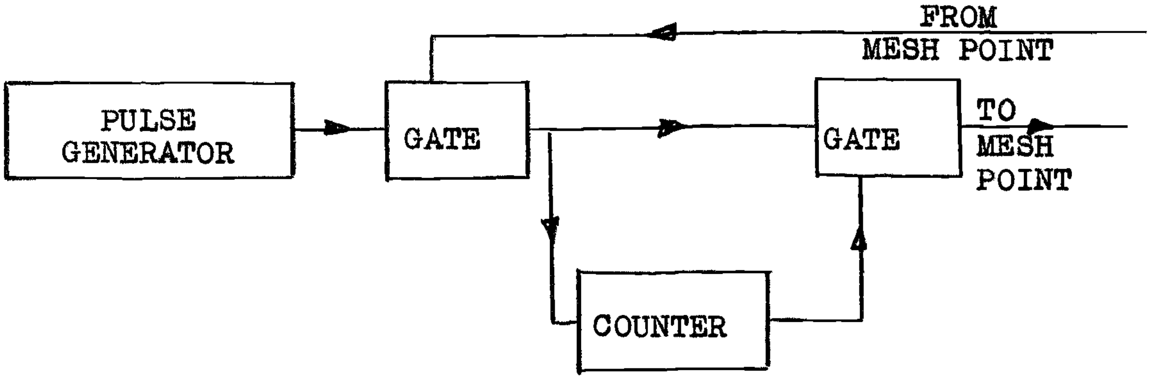


Figure 4.7.1

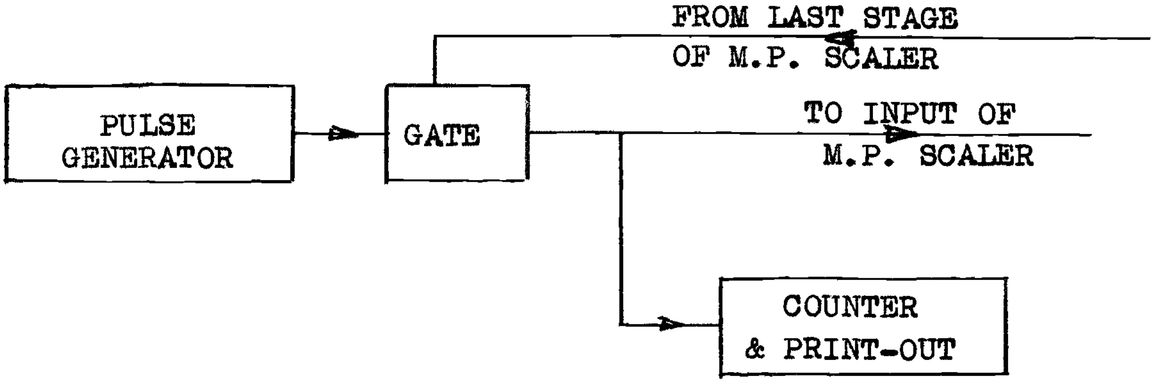


Figure 4.8.1

As no complete machine has yet been made, no pulse generators of this type have been constructed. A possible arrangement is, however, outlined in fig. 4.6.1. The pulse generator runs in synchronism with the central clock pulse generator. The number of pulses to be injected is set on the scaler, which may well use the scaler units developed for use in the mesh-point potential registers. Pulses from the generator are fed through the gate and to the scaler. When the scaler has counted the correct number of pulses the gate is closed, and the mesh-point selector is switched to the next point and the process repeated.

Selection of the appropriate point in the computer at which to feed in these pulses could be done manually in a small machine, or in a large machine by using the coincident-current principle as used in computer magnetic matrix stores.

4.7 Input for Poisson's Equation. In this case, a mesh-point in a region of charge density ρ will need extra pulses injected, the number being directly proportional to ρ , as we have for the finite-difference equivalent of Poisson's equation,

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = -\frac{\rho}{\epsilon} \qquad \ldots(4.7.1)$$

the equation

$$V_0 = \frac{1}{4}\left(V_1 + V_2 + V_3 + V_4 + \frac{h^2\rho}{\epsilon}\right) \qquad \ldots(4.7.2)$$

where h is the mesh size.

The situation is not as simple as the case where a point is to have a certain specified potential, because the unit must still operate in the normal way, accepting pulses from adjacent units and raising its potential accordingly. The input must therefore sense whether the mesh-point unit is to emit a pulse at the next appropriate clock phase. If the mesh-point unit is not going to emit one, them the input unit can put in a pulse, but may not do so if the mesh-point unit is to emit one. Extra circuitry is therefore needed between the selected mesh point and the input generator. This need not be complex as all that is added to the normal input unit is a gate on the output of the pulse generator which is closed when the selected mesh-point unit is emitting a pulse, as in fig. 4.7.1 It may be possible, in the interests of economy, to use the same conducting path for both feeding in pulses and for sensing if the mesh-point unit is emitting a pulse, if it is arranged that the input pulse is slightly delayed on the mesh-point output pulse.

4.8 Read-out of Stored Potentials. The system of read-out proposed for the storage registers at each mesh-point is that pulses should be fed into the input of the register until it is completely filled, this being indicated by an output from the final stage of the register. The complement of the number of pulses fed in is then the potential of that particular point. Selection of the point in the mesh may again be done by a coincident current system. Sensing the output of the final scaler stage of each unit could be done by a sense wire through all the units served by the read-out device. Again, only one such device for the whole computer is strictly necessary, but results would be read more quickly if a number of them were used, each for a particular region of the mesh.

A schematic of the read-out device is shown in fig. 4.8.1. The pulse generator need not work at the machine clock frequency, it should work at the maximum speed the mesh-point register is capable of accepting reliably. The gate closes as soon as the register has filled, and the counter then indicates the potential would preferably print out the value. The device would then be switched to the next mesh-point unit.

4.9 Stopping the Computation. Some means of telling when the computation is finished must be incorporated in the machine, so that the read-out process can be started. This would best be done by sensing whether, during a complete clock cycle, no pulses were emitted from any of the mesh-point unit output cores. A sense wire could be threaded through a number of output cores, and then to a system of 'or' gates. The voltage pulse produced when a core switches completely is about 20 times the amplitude of the pulse produced when the same drive is applied to a saturated core. This would limit the number of units that one wire could serve, which would depend also on the discrimination of the 'or' circuits used.

4.10 Cost of Mesh Point Units. The following is a very approximate estimate of the cost of the mesh point units described above:-

Input Scaler	£	s	d
1 wound core	0	8	0
5 resistors	0	2	6
2 transistors	0	16	0
Output Circuit			
4 wound cores	1	12	0
4 transistors	1	12	0
4 resistors	0	2	0
Potential Register			
5 scalers similar to Input Scaler	6	12	6
Printed circuit	0	5	0
Total	£ 11	10	0

The units could probably be produced rather more cheaply if they were made in large quantities, but in estimating the cost of a prototype computer it would be unwise to rely on the cost dropping very much below £ 10 per uhit in view of the modifications which will probably be found necessary.

5. ANALYSIS OF THE OPERATION OF THE FIELD COMPUTER

5.1 Propagation of Pulses in a One-Dimensional System.

To examine the behaviour of the individual units when connected together to form a computing network, it will be helpful to first consider the simplest possible situation. The units to be considered are of the one-dimensional type, which emit one output pulse for every two input input ones.

First consider the case where unit no. '0' is a boundary, and units numbered 1,2,3,4,..... extend infinitely to the right, with no right hand boundary. The residues in each unit are set initially to zero. Pulses are now fed in from unit'0 to unit'1' at the rate of one every clock cycle. For simplicity a two phase rather than three phase clock cycle will be assumed. Precisely the same conclusions are reached using a three phase system, the only difference being that the two phase system assumes that the units are capable of handling two simultaneous inputs.

The states of the first few units for the first few clock cycles are shown in fig. 5.1.1. The numbers in each 'box' show the state of the residue of the appropriate unit after each of the two clock phases in each cycle. Units 0,2,4 etc. emit pulses on phase '0' and units 1,3,5 etc. on phase '1'.

There is a certain pattern in the way in which pulses progress, and examination shows that

Unit No. Clock Cycle	1	2	3	4	5	6
1	1 1					
2	2 0	1				
3	1 1	1 1				
4	2 0	1 2				
5	2 0	0 1	1 1			
6	1 1	1 1	1 1			
7	2 0	1 2	1 1			
8	2 0	0 2	2 0	1		
9	2 0	0 1	1 1	1 1		
10	1 1	1 1	1 1	1 1		
11	2 0	1 2	1 1	1 1		
12	2 0	0 2	2 0	1 2		
13	2 0	0 2	2 0	0 1	1 1	
14	2 0	0 1	1 1	1 1	1 1	
15	1 1	1 1	1 1	1 1	1 1	
16	2 0	1 2	1 1	1 1	1 1	
17	2 0	0 2	2 0	1 2	1 1	
18	2 0	0 2	2 0	0 2	2 0	1
19	2 0	0 2	2 0	0 1	1 1	1 1
20	2 0	0 1	1 1	1 1	1 1	1 1
21	1 1	1 1	1 1	1 1	1 1	1 1
22	2 0	1 2	1 1	1 1	1 1	1 1

Figure 5.1.1

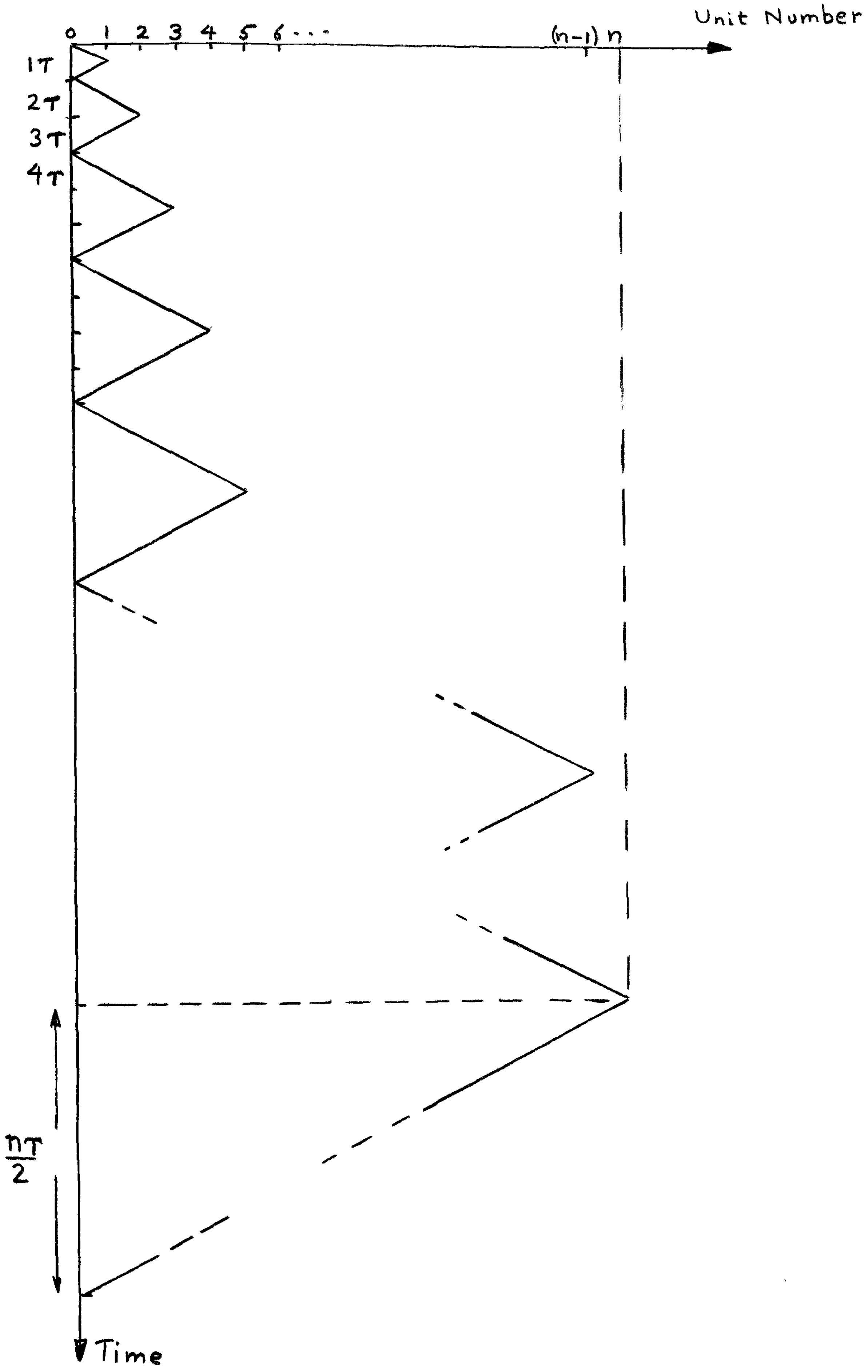


Figure 5.1.2

the time taken for the first pulse to appear in the residue of the nth. unit is n(n+1)/2 clock cycles. This time is that taken for all units to the left of n to revert to having 1 in their residues, as this simplifies calculation to some extent. The 2's in the residues are seen to propagate to the right at the rate of 2 units per clock cycle, but can only propagate if there are 1's in front of them. The total potential of any point at any particular time is found by adding the number of times it emits a pulse, i.e. the number of times a 2 appears in the residue.

Fig. 5.1.2 is a simplified form of the previous diagram. The triangles enclose regions where propagation of 2's is occuring. The valleys between the peaks are filled with 1's . All the rest of the residues are zero. The figure extends to the situation after the first pulse has reached unit no. n and all the units below that have a residue of 1. The 'potential' at any given instant of any point may be found by drawing a vertical line at that point on the diagram. The potential is directly proportional to the sum of the lengths of line which lie inside the triangles.

5.2 Potentials in the One-Dimensional System. These rise in a similar manner to that in which the pulses propagate. Consider the instant when a 1 has arrived at point n (and the system has settled down so that there are no 2's present) i.e. at time n(n + 1) T/2, where T is the period of one clock cycle. The potentials at each point are :-

$$V_n = 0$$

$$V_{n-1} = 1$$

$$V_{n-2} = 2 + 1$$

$$V_{n-3} = 3 + 2 + 1$$

$$V_{n-4} = 4 + 3 + 2 + 1$$

....

....

$$V_{n-p} = p + (p-1) + (p-2) + \ldots + 1 = p(p+1)/2$$

$$V_0 = n + (n-1) + (n-2) + \ldots + 1 = n(n+1)/2 \quad \ldots(5.2.1)$$

The potential of unit '0' is confirmed by the fact that n(n+1) clock cycles have been completed, and unit '0' has emitted one pulse in each cycle.

In the general case,

$$V_{n-p} = p(p+1)/2 \quad \text{when } t = n(n+1)T/2 \quad ..(5.2.2)$$

or,

$$V(q,t) = \frac{(n-q)(n-q+1)}{2} \quad \ldots(5.2.3)$$

with $n = -\frac{1}{2} \pm (2t/T + \frac{1}{4})^{\frac{1}{2}}$ ...(5.2.4)

where q is the number of the unit and t the time. For t much greater than T and n much greater than 1, we obtain :

$$V(q,t) \fallingdotseq (\sqrt{2t/t} - q)^2/2 \quad \ldots(5.2.5)$$

5.3 The Difference Equation Solved by the Computer.

The important factors in the operation of a unit are the residue at any instant, and whether it emits a pulse on any particular clock phase. Let $P(n,sT)$ represent the output from unit number n at clock cycle number s, and $R(n,sT)$ the residue of the point immediately after this unit's clock phase. If these expressions have the value 1 when a pulse is present and otherwise 0, then:-

$$P(n,sT) = 1, \text{ if } R(n,\overline{s-1T}) + P(n-1,\overline{s-1T}) + P(n+1,\overline{s-1T}) = Q(n,sT) \geqslant 2 \quad \ldots(5.3.1)$$

$$P(n,sT) = 0, \text{ if } Q(n,sT) < 2 \quad \ldots(5.3.2)$$

i.e. an output is obtained if R is 1 initially and a pulse is received from either, or both adjacent units or if pulses are received from both adjacent units and R is 0 initially. Similarly,

$$R(n,sT) = 1, \text{ if } Q(n,sT) = 1 \text{ or } 3 \quad \ldots(5.3.3)$$

$$R(n,sT) = 0, \text{ if } Q(n,sT) = 0 \text{ or } 2 \quad \ldots(5.3.4)$$

These expressions can be combined by noting that:-

$$\begin{aligned} R(n,sT) + 2P(n,sT) &= Q(n,sT) \\ &= R(n,\overline{s-1T}) + P(n-1,\overline{s-1T}) \\ &\quad +P(n+1,\overline{s-1T}) \qquad \ldots(5.3.5) \end{aligned}$$

The same expression could have been obtained by considering the fact that pulses are 'conserved', i.e. that for every pulse entering a unit, one will eventually be emitted, the difference between the numbers of those received and emitted being, at any particular time, the residue at that point. Therefoee,

$$R(n,sT) = R(n,\overline{s-1}T) + P(n-1,\overline{s-1}T) + P(n+1,\overline{s-1}T) - 2P(n,sT) \qquad \ldots(5.3.6)$$

The term 2P(n,sT) arises because when a unit gives an output, it goes to two adjacent points and so counts as two pulses.

The total potential at any point is the sum of the pulses emitted from that point; let this be V(n,sT). To obtain this quantity, we write down eqn. 5.3.5 for all clock cycles and add them together:-

$$\begin{array}{lllll}
R(n,sT) - R(n,\overline{s-1}T) & = P(n-1,\overline{s-1}T) & - 2P(n,sT) & + P(n+1,\overline{s-1}T) \\
R(n,\overline{s-1}T)-R(n,\overline{s-2}T) & = P(n-1,\overline{s-2}T) & -2P(n,\overline{s-1T-}) & +P(n+1,\overline{s-2}T) \\
\cdots\cdots & & & \\
\cdots\cdots & & & \\
\cdots\cdots & & & \\
R(n,2T) - R(n,T) & = P(n-1,T) & -2P(n,2T) & +P(n+1,T) \\
R(n,T) - R(n,0) & = P(n-1,0) & - 2P(n,T) & + P(n+1,0)
\end{array}$$

So that;

$$R(n,sT) - R(n,0) = V(n-1,\overline{s-1}T) - 2V(n,sT) + V(n+1,\overline{s-1}T) \qquad \ldots(5.3.7)$$

5.4 Possibility of Solving the Diffusion Equation

If, in equation 5.3.7 , R can be assumed small compared with other values, this is the difference equation corresponding to the Diffusion Equation;

$$\frac{\partial V}{\partial t} = \frac{1}{2T}\frac{\partial^2 V}{\partial n^2} \qquad \ldots(5.4.1)$$

and so the computer would solve this equation as well as Laplace's and Poisson's Equations.

However, on closer examination it becomes evident that ignoring R is not permissible. R can only have the values 0,1 or2, but because of the method of operation of the computer, the values of $V(n,sT) - V(n,\overline{s-1}T)$ can only be 0 or 1 and therefore

$$\left\{V(n,\overline{s-1}T) - V(n-1,\overline{s-1}T)\right\} - \left\{V(n+1,\overline{s-1}T) - V(n,\overline{s-1}T)\right\}$$

can only have the values 0, ±1 or ±2. Therefore the computer in its present form cannot solve the Diffusion Equation , as to do this it would be necessary that the differential values could take up a wide range of values. However, the principle of the field computer could be applied to a computer which would solve the Diffusion Equation, but the method of operation of the units would need to be different.

5.5 Demonstration that the Computer can Solve Laplace's Equation.

Consider equation 5.2.3 ;

$$V(q,sT) = \frac{(n-q)(n-q+1)}{2}$$

Then the finite-difference form of $\frac{\partial V}{\partial x}$ is:

$$\frac{\Delta V}{h} = \frac{(n-q)(n-q+1) - (n-q-1)(n-q)}{2h}$$

$$= \frac{n-q}{h} \qquad \ldots(5.5.1)$$

and, for $\frac{\partial^2 V}{\partial x^2}$:

$$\frac{\Delta^2 V}{h^2} = \frac{\frac{n-q+1}{h} - \frac{n-q}{h}}{h} = 1/h^2 \qquad \ldots(5.5.2)$$

So that in the case where equation 5.2.3 applies, $\partial^2 V/\partial x^2$ is not exactly zero and so Laplace's Equation is not solved exactly. The error is small and may be negligible if the total potential in the region is large. It is, however, an appreciable error at or near the front of the pulse propagation pattern. It is important, therefore, to eliminate this constant error.

The result of equation 5.5.2 is that which would be expected if there were a uniform negative charge density throughout the region considered. Therefore it seems likely that a more accurate solution would result if, initially, uniform positive charge were imposed everywhere. The amount needed would be a residue of +1 at each mesh-point. From a consideration of fig. 5.1.1 this is a reasonable thing to do, as the units seem most likely to settle down to this state at the end of a computation. It is also the mean value of the states of the residue.

Figure 5.5.1 shows the uniform progress of the pulse pattern through a field with the residues initially set to 1. Conditions are otherwise as in fig. 5.1.1 and 5.1.2. The potential at any point,q, where q is less than n, is given by:-

$$V(q,sT) = n - q \qquad \ldots(5.5.3)$$

with sT = nT. So;

$$\frac{\Delta V}{h} = \frac{(s-q) - (s-q+1)}{h} = \frac{-1}{h} \qquad \ldots(5.5.4)$$

and,

$$\frac{\Delta^2 V}{h^2} = \frac{\{(s-q-1) - (s-q)\} - \{(s-q) - (s-q+1)\}}{h^2}$$

$$= 0 \qquad \ldots(5.5.5)$$

So that in this particular case Laplace's Equation is solved. However, this is not a steady state case, as V is rising steadily at each point to the left of poin number n. If unit '0' were to cease emitting pulses, the pulse pattern would continue to move to the right indefinitely, but the potential of all points would rise steadily and then stop when it had reached the final potential of unit '0'. This would then give another trivial solution of Laplace's Equation.

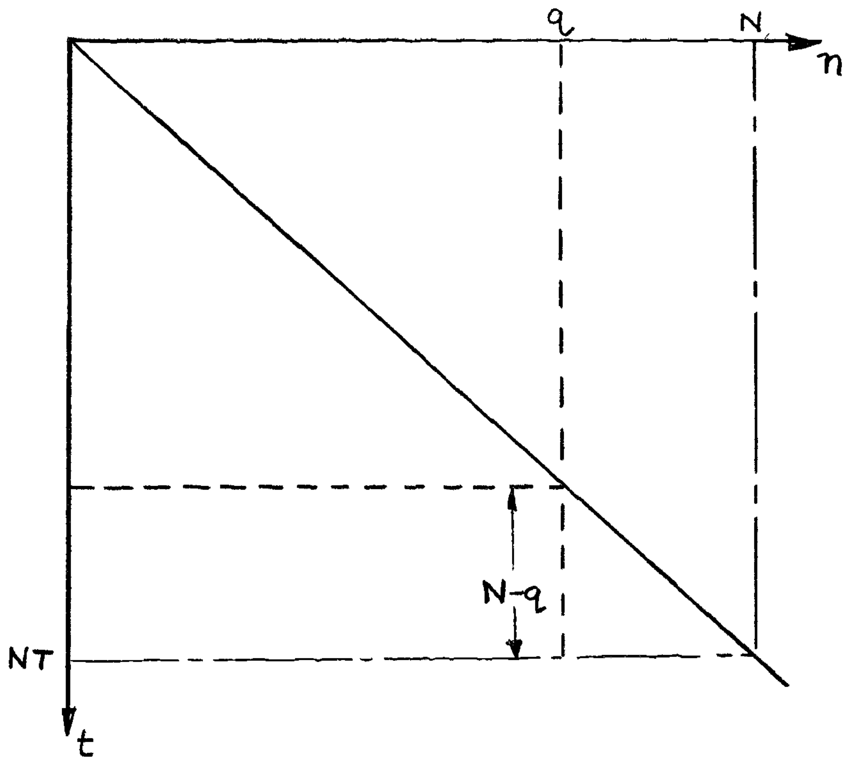


Figure 5.5.1

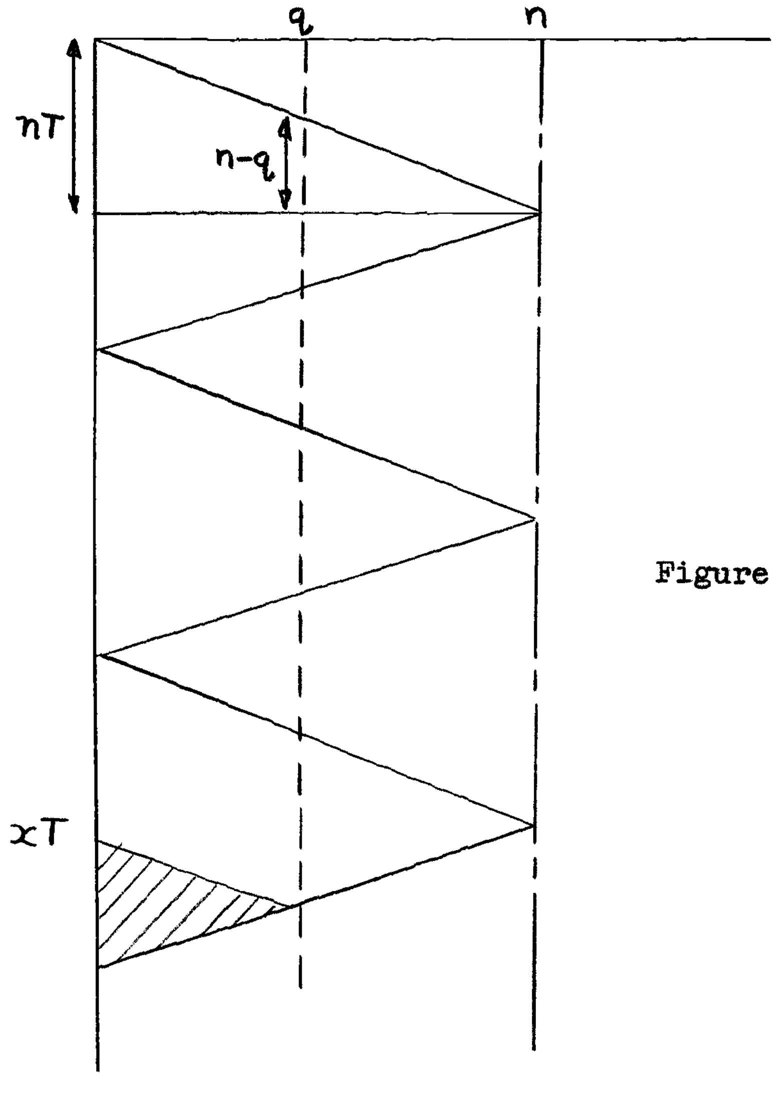


Figure 5.6.1

5.6 A One-Dimensional System with a Boundary at Each End. Suppose we take the next simplest case, with a boundary condition that point no. n shall be always at zero potential. This means that unit n receives pulses but never emits any. The residues are all assumed to be 1 initially. The pattern of fig. 5.6.1 is the result, if unit '0' emits pulses at the rate of one per clock cycle. The potential is:

$$V(n,sT) = \frac{sT}{2nT} . 2(n-q)$$

$$= s(1- q/n) \qquad \ldots(5.6.1)$$

$$\frac{\Delta V}{h} = \frac{s(n-q)/n \;-\; s(n-q+1)/n}{h} = \frac{-s}{nh} \qquad ..(5.6.2)$$

$$\frac{\Delta^2 V}{h^2} = \frac{s(n-q-1) - s(n-q)}{nh^2} = \frac{s(n-q) - s(n-q+1)}{nh^2}$$

$$= 0 \qquad \ldots(5.6.3)$$

and so Laplace's Equation is satisfied. Now,

$$\frac{\Delta V}{T} = \frac{s(1-q/n) \;-\; (s-1)(1-q/n)}{T}$$

$$= \frac{n-q}{nT} \qquad \ldots(5.6.4)$$

which is constant as long as pulses continue to be injected by unit '0'. When these pulses stop, Laplace's Equation will be solved exactly if it so happens that the instant of stopping coincides with one of the minima of the sawtooth pattern of fig. 5.6.1. If it stops at any other point, then a certain amount of inaccuracy arises. If it stops at time xT, as in fig. 5.6.1, then the shaded area of the last triangle has no 2's in it and so the potentials of the points concerned will turn out to be low. As it would be difficult in general to arrange for the input to stop at just the right moment, in fact this would be impossible in more complex problems,it is necessary to allow a large number of complete pulse patterns to form. In this way, the errors will be insignificant compared with the total potentials. The number of sawteeth is directly proportional, in this case, to s/n and so the larger the computer network the larger the number of pulses which must be fed in at the boundaries in order to achieve comparable accuracy.

In the above cases, it has been assumed for simplicity that the boundary pulses were sent in in a steady stream. The results are not materially affected if the boundary pulses are fed in at longer intervals, or in a random manner, provided always that they are emitted at the rate of not more than one per clock cycle. The only difference in such a case would be the obvious one that there is then no direct relation between time and the number of pulses emitted.

5.7 Extension to Two-Dimensional Systems. It seems to be rather unprofitable to attempt to extend the previous method of analysis to systems having two dimensions because of the large amount of work involved and the likelihood of making mistakes. The simplest case, that of the potential distribution between two infinite parallel plates, is of little interest since it is precisely equivalent to the one-dimensional system already considered. All that can be concluded from such attempts is that the residue at each point should initially be set to some value greater than 0 if the same sort of results as in the one-dimensional system are to be obtained. With an initial residue of 3 in the parallel plate case, injection of 1 pulse at a boundary will result in propagation of the disturbance to the other boundary, the first few steps being shown in fig. 5.7.1. If the initial residue is 2, however, fig. 5.7.2 shows that this sort of propagation does not occur. 2 is the mean of the possible values of the residue and so seems likely to be the mean value of all the residues at the end of a computation. Therefore, at this stage it is rather doubtful which value should be used in a general problem .

$$\left|\begin{array}{cccc} 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 \end{array}\right.$$

$$\left|\begin{array}{cccc} 4 & 3 & 3 & 3 \\ 4 & 3 & 3 & 3 \\ 4 & 3 & 3 & 3 \end{array}\right.$$

$$\left|\begin{array}{cccc} 2 & 4 & 3 & 3 \\ 2 & 4 & 3 & 3 \\ 2 & 4 & 3 & 3 \end{array}\right.$$

$$\left|\begin{array}{cccc} 3 & 2 & 4 & 3 \\ 3 & 2 & 4 & 3 \\ 3 & 2 & 4 & 3 \end{array}\right.$$

$$\left|\begin{array}{cccc} 4 & 3 & 2 & 4 \\ 4 & 3 & 2 & 4 \\ 4 & 3 & 2 & 4 \end{array}\right.$$

$$\left|\begin{array}{cccc} 2 & 4 & 3 & 2 \\ 2 & 4 & 3 & 2 \\ 2 & 4 & 3 & 2 \end{array}\right.$$

Figure 5.7.1

$$\left|\begin{array}{cccc} 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 \end{array}\right. \qquad \left|\begin{array}{cccc} 3 & 2 & 2 & 2 \\ 3 & 2 & 2 & 2 \\ 3 & 2 & 2 & 2 \end{array}\right.$$

$$\left|\begin{array}{cccc} 4 & 2 & 2 & 2 \\ 4 & 2 & 2 & 2 \\ 4 & 2 & 2 & 2 \end{array}\right. \qquad \left|\begin{array}{cccc} 3 & 3 & 2 & 2 \\ 3 & 3 & 2 & 2 \\ 3 & 3 & 2 & 2 \end{array}\right.$$

$$\left|\begin{array}{cccc} 4 & 3 & 2 & 2 \\ 4 & 3 & 2 & 2 \\ 4 & 3 & 2 & 2 \end{array}\right. \qquad \left|\begin{array}{cccc} 3 & 4 & 2 & 2 \\ 3 & 4 & 2 & 2 \\ 3 & 4 & 2 & 2 \end{array}\right.$$

Figure 5.7.2

5.8 Simulation of the Field Computer on a Digital Computer. In order to discover whether the machine behaves satisfactorily and solves Laplace's Equation, it is necessary either to build a fair-sized example of the field computer or to simulate it somehow. Theoretical methods seem unable to provide the information required and so it was decided to simulate the operation of the computer on the Pegasus digital computer. By causing the computer to print out frequently it should be possible to check that the field computer would behave as predicted, and the final result of the computation should give some idea of the accuracy of the field computer. Another important result is the time taken for the field computer to settle down to the final solution after all the boundary condition pulses have been fed in.

The scheme adopted for the programme for the digital computer was to scan all points in the mesh in series in a five-phase cycle. These phases correspond to the five clock phases needed in the two-dimensional problem. The nett effect, with the system of phasing used, is the same as if the processes in any given phase happened simultaneously, as in the field computer, rather than sequentially as in the digital computer. This is because the pulses emitted by a unit in its phase cannot affect another unit having the same phase, and neither can two units of the same phase simultaneously give a pulse to a unit of different phase.

At each mesh point, the process is the same as in the field computer. The input store, or residue, is examined; if it is greater or equal to 4 a pulse is emitted to the four adjacent points,i.e. one is added to the residue of each. One is also added to the total potential store at that point. The programme is arranged so that one word in the computer contains the total potential and the residue at the point together with information as to whether the point is on a boundary or is internal. In order to do this, the programme had to be written in machine language for Pegasus.

At a boundary point the procedure was somewhat different, as the potential of that point was initially put into the store there, one subtracted from it every clock cycle and one added to the residues of the adjacent internal points until the potential store was exhausted. At a boundary where the potential was specified all incoming pulses were, of course, ignored as in the field computer. This situation corresponds to that in the field computer in which all boundary points are fed at the same time by separate pulse generators. It would be quite easy to modify the digital computer programme so that it could simulate the case where pulses are fed in at different times at various boundaries, but this has not yet been done as it seems that no very significant problems would be resolved by so doing.

A print out subroutine was written which printed in tabular form the potential and residue at each point in the mesh. Initially the programme was set to print out once in every cycle of 5 clock phases. This could be increased for cases where a large number of cycles had to be gone through before the computation settled down to give final values of potentials. The programme in addition printed out all these values at the end of the computation, this stage being sensed by checking whether any pulses had flowed in the last clock cycle.

The complete programme is given in Appendix 2.

5.9 Application of Simulation to Specific Cases.

To test the programme, it was applied first to the case of the infinite parallel plate capacitor. As the computer cannot properly take account of the edges, C & D in fig. 5.9.1, of the section of the capacitor which is considered, a uniform potential gradient along these two edges was inserted as boundary conditions.

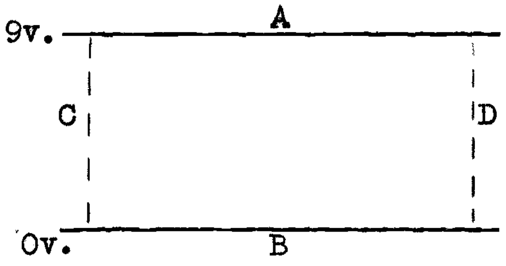


Figure 5.9.1

Figures 5.9.2, 5.9.3 and 5.9.4 show the results after 1, 5 and 10 clock cycles for the cases where the initial residues were 0, 2 and 3 respectively. In all cases the 10th cycle produced the final solution.

8,3	8,2	8,2	8,2	8,I	8,3	8,2	8,2	8,2	8,I	8,3	8,2	8,2
7,I	0,2	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,2	7,4
6,3	0,I	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,2	0,I	0,2
5,I	0,I	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,I	5,4
4,2	0,I	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,I	4,I
3,4	0,I	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,I	3,3
2,I	0,I	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,I	2,4
I,3	0,I	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,I	I,2
0,I	0,I	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,I	0,2
0,2	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I

4,3	4,3	4,2	4,2	4,I	4,3	4,2	4,2	4,2	4,I	4,4	4,3	4,2
3,2	3,I	I,5	I,3	I,3	I,3	I,3	I,3	I,3	I,4	2,I	3,I	3,5
2,4	2,I	0,3	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,3	I,5	2,2
I,I	I,4	0,I	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,I	I,3	I,4
0,2	I,3	0,I	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,I	I,3	0,I
0,3	I,2	0,I	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,I	I,2	0,2
0,I	I,0	0,I	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,I	I,0	0,2
0,I	0,3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,3	0,I
0,I	0,I	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,I	0,I
0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I

Fig. 5.9.2

0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I
0,I	6,2	5,I	4,3	4,2	4,2	4,2	4,2	4,2	4,3	5,I	6,2	0,I
0,2	4,I	2,3	I,3	I,2	I,2	I,2	I,2	I,2	I,3	2,3	4,2	0,I
0,I	2,5	I,0	0,2	0,I	0,I	0,I	0,I	0,I	0,2	I,I	3,I	0,I
0,I	2,0	0,3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,3	2,I	0,I
0,I	I,3	0,I	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,I	I,3	0,I
0,I	I,0	0,I	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,I	I,0	0,I
0,I	0,3	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,3	0,I
0,I	0,I	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,I	0,I
0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I

8,3	8,3	8,2	8,2	8,I	8,3	8,2	8,2	8,2	8,I	8,3	8,3	8,2
7,2	I,0	0,4	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,4	I,0	7,3
6,3	0,4	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,4	6,2
5,I	0,3	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,3	5,4
4,2	0,3	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,3	4,I
3,4	0,3	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,3	3,3
2,I	0,3	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,3	2,4
I,3	0,3	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,3	I,2
0,I	0,3	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,3	0,2
0,2	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I

4,3	4,3	4,2	4,2	4,I	4,4	4,3	4,2	4,2	4,I	4,4	4,3	4,2
3,2	3,0	4,4	4,3	4,2	4,2	4,I	3,5	3,4	3,4	4,2	3,0	3,5
2,4	4,3	4,I	4,0	3,4	3,3	3,2	3,I	3,0	2,5	3,3	4,2	2,2
I,I	3,5	3,3	3,2	3,I	3,0	2,4	2,3	2,2	2,2	3,0	3,5	I,4
0,3	3,3	3,0	2,4	2,3	2,2	2,I	2,0	I,4	I,4	2,2	3,2	0,I
0,3	2,5	2,2	2,I	2,0	I,4	I,3	I,2	I,I	I,I	I,4	2,4	0,2
0,I	2,I	I,4	I,3	I,2	I,I	I,0	0,4	0,3	0,4	I,I	2,I	0,2
0,I	I,4	I,I	I,0	0,4	0,3	0,3	0,2	0,2	0,2	0,4	I,3	0,I
0,I	I,0	0,4	0,3	0,2	0,2	0,2	0,2	0,2	0,2	0,3	I,0	0,I
0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I

0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I
0,I	8,2	8,2	8,2	8,2	8,2	8,2	8,2	8,2	8,2	8,2	8,2	0,I
0,I	7,2	7,2	7,2	7,2	7,2	7,2	7,2	7,2	7,2	7,2	7,2	0,I
0,I	6,2	6,2	6,2	6,2	6,2	6,2	6,2	6,2	6,2	6,2	6,2	0,I
0,I	5,2	5,2	5,2	5,2	5,2	5,2	5,2	5,2	5,2	5,2	5,2	0,I
0,I	4,2	4,2	4,2	4,2	4,2	4,2	4,2	4,2	4,2	4,2	4,2	0,I
0,I	3,2	3,2	3,2	3,2	3,2	3,2	3,2	3,2	3,2	3,2	3,2	0,I
0,I	2,2	2,2	2,2	2,2	2,2	2,2	2,2	2,2	2,2	2,2	2,2	0,I
0,I	I,2	I,2	I,2	I,2	I,2	I,2	I,2	I,2	I,2	I,2	I,2	0,I
0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I

Fig.
5.9.3

8,3	8,3	8,2	8,2	8,I	8,4	8,3	8,2	8,2	8,I	8,4	8,3	8,2
7,2	I,2	0,6	0,5	0,5	I,I	I,I	0,5	0,4	0,5	I,I	I,2	7,5
6,4	I,2	I,I	I,0	0,4	0,4	0,4	0,3	0,3	0,3	0,4	0,5	6,2
5,I	0,6	0,5	0,4	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,4	5,4
4,3	I,2	I,I	0,5	0,4	0,3	0,3	0,3	0,3	0,3	0,3	0,4	4,I
3,5	I,3	I,2	I,I	I,0	0,4	0,3	0,3	0,3	0,3	0,3	0,5	3,3
2,2	I,2	0,6	0,5	0,4	0,3	0,3	0,3	0,3	0,3	0,4	I,0	2,5
I,4	I,2	I,I	I,0	0,4	0,3	0,3	0,3	0,3	0,3	0,3	0,5	I,2
0,I	0,5	0,4	0,4	0,3	0,3	0,3	0,3	0,3	0,3	0,3	0,4	0,2
0,2	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I

4,3	4,3	4,2	4,2	4,I	4,4	4,3	4,2	4,2	4,I	4,4	4,3	4,2
3,2	5,2	4,6	4,5	4,5	5,I	5,I	4,5	4,4	4,4	5,I	5,2	3,5
2,4	5,2	5,I	5,0	4,4	4,4	4,3	4,2	4,I	3,6	4,3	4,5	2,2
I,I	4,6	4,5	4,4	4,3	4,2	3,6	3,5	3,4	3,4	4,I	4,4	I,4
0,3	5,I	5,0	4,4	4,3	4,3	4,2	4,I	3,5	3,4	3,5	4,3	0,I
0,3	4,3	4,3	4,2	4,2	4,2	4,2	4,2	4,I	4,0	3,5	4,I	0,2
0,I	3,3	3,3	3,3	3,3	3,3	3,3	3,3	3,3	3,3	3,2	3,3	0,I
0,I	2,3	2,3	2,3	2,3	2,3	2,3	2,3	2,3	2,3	2,3	2,3	0,I
0,I	I,3	I,3	I,3	I,3	I,3	I,3	I,3	I,3	I,3	I,3	I,3	0,I
0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I

Fig. 5.9.4

0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I
0,I	8,3	8,3	8,3	8,3	8,3	8,3	8,3	8,3	8,3	8,3	8,3	0,I
0,I	7,3	7,3	7,3	7,3	7,3	7,3	7,3	7,3	7,3	7,3	7,3	0,I
0,I	6,3	6,3	6,3	6,3	6,3	6,3	6,3	6,3	6,3	6,3	6,3	0,I
0,I	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	5,3	0,I
0,I	4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3	4,3	0,I
0,I	3,3	3,3	3,3	3,3	3,3	3,3	3,3	3,3	3,3	3,3	3,3	0,I
0,I	2,3	2,3	2,3	2,3	2,3	2,3	2,3	2,3	2,3	2,3	2,3	0,I
0,I	I,3	I,3	I,3	I,3	I,3	I,3	I,3	I,3	I,3	I,3	I,3	0,I
0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I	0,I

With residues of 2 and 3, the final states of the residues are the same as the initial ones, in this particular case, and the solution is exactly correct. In a more complex problem it would be most unlikely that all the residues would return to their initial values, and, as 2 is the mean value of the states of the residue, it was decided that this would be used in the next computation. However, it can be seen from fig. 5.9.4 that with an initial residue of 3 the solution is approached more quickly at first. This question of inital residues is, of course, not at all important when large potentials are involved.

In all three cases the number of cycles taken was the same and the time for settling down after the last pulse had been fed in was negligible. The time taken on Pegasus was about 3 minutes in each case, the majority of this time being taken in punching the results, this taking about 15 seconds per cycle compared with about 2 seconds computing time.

A more complex problem, that of the electron gun in fig. 5.9.5, was next tried. This had been solved previously by relaxation and so the solution obtained by the computer could be readily checked.The relaxation solution is shown in fig. 5.9.6 an the computer solution in fig. 5.9.7. Good agreement between the two is shown, and no residues (in the usual relaxation sense) of greater than 2 could be found in the computer solution. The only discrepancy

330	330	330	330	330	330	330	330	330	330	330	330
287	287	286	285	283	281	279	277	276	275	275	275
245	245	243	240	236	232	228	225	223	221	220	220
204	204	201	196	189	182	176	172	169	167	166	165
165	165	161	153	142	130	122	117	114	113	111	110
130	130	125	113	95	72	64	60	58	56	55	55
100	100	94	79	52	0	0	0	0	0	0	0
77	77	71	57	34	0						
60	60	54	43	26	0	0	0	0	0	0	0
47	47	43	36	26	16	12	11	10	10		
38	38	36	32	28	24	22	21	20	20		
30	30	30	30	30	30	30	30	30	30		

figure 5.9.6

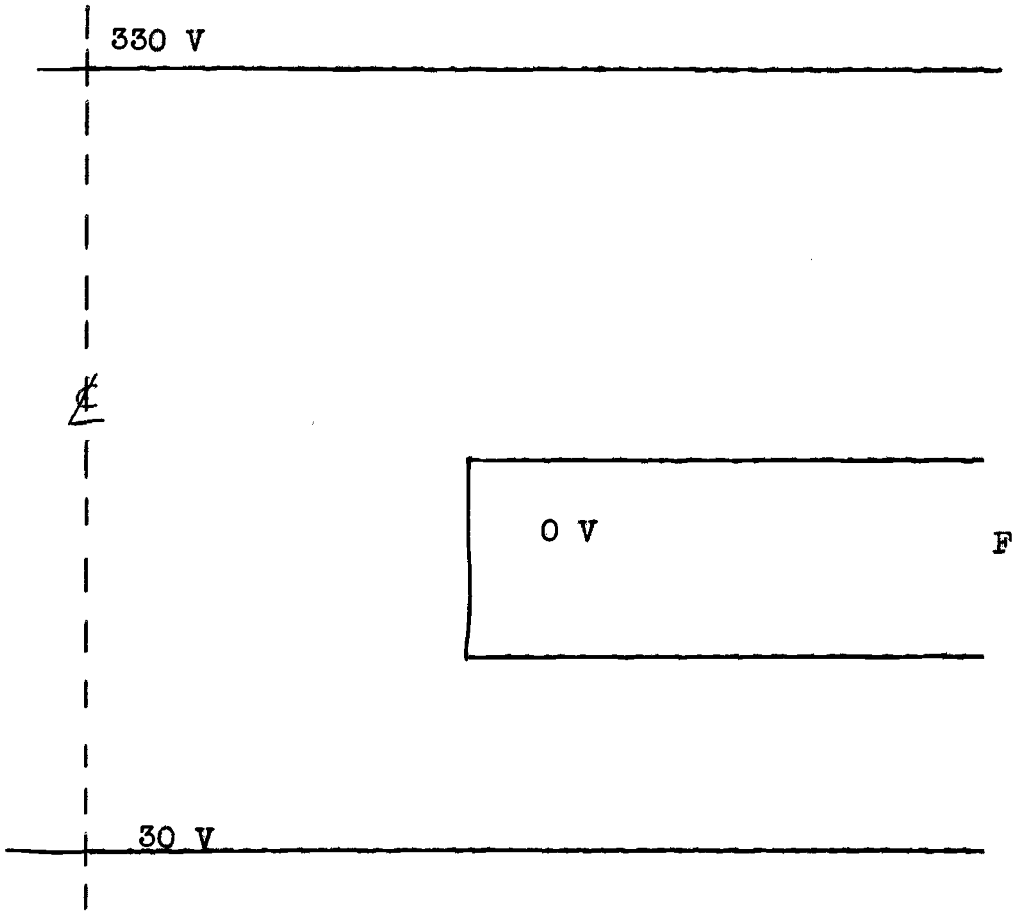


Figure 5.9.5

o	o	o	o	o	o	o	o	o	o	o	o	o
o	287	286	285	283	281	279	277	276	276	275	o	o
o	244	242	239	235	231	227	224	222	221	220	o	o
o	203	200	194	187	180	175	171	168	166	165	o	o
o	164	160	151	140	128	121	116	113	111	110	o	o
o	129	123	111	93	71	63	59	57	56	55	o	o
o	99	92	77	51	o	o	o	o	o	o	o	o
o	76	69	55	33	o	o	o	o	o	o	o	o
o	59	55	42	25	o	o	o	o	o	o	o	o
o	47	43	36	26	16	12	11	10	o	o	o	o
o	38	36	33	28	24	22	21	20	o	o	o	o
o	o	o	o	o	o	o	o	o	o	o	o	o

Figure 5.9.7

between the two was that the computer solution tended to be slightly lower than the relaxation one. This might have been avoided if initial residues of 3 had been used, but this has not yet been tried.

The time taken for this computation was 20 minutes and although this time is not excessive it is probably slower than using more normal digital computer methods for solving such a problem. The time for settling down at the end of the computation was less than 30 clock cycles, and so was small compared with the time taken for the reading in of pulses. The time taken in the actual field computer would be much less, as the time for each cycle would be much smaller, and constant however large the machine were. This does not mean that the time for solving a problem on a large machine will be the same as for a small machine. In the large one, in order to take advantage of the greater accuracy possible due to the greater number of mesh points, larger potentials will be used on the boundaries, with a consequent increase in computing time. The time for settling down at the end of the computation will increase in the larger machine, but may stay at a fixed fraction of the total computing time because of the increase in the latter.

6 FURTHER DEVELOPMENT OF THE FIELD COMPUTER

6.1 Feasibility of the Computer. It has been shown that the computer will behave as expected and solve Laplace's Equation, and is a practical proposition if some further development work is done on the circuits needed. It is not suggested that the circuits described are the only ones possible, but they show that it is economically possible to build such a computer using existing circuit components. Whatever circuit is finally developed for the basic unit, a large number of identical units will be needed so that quantity production will simplify their manufacture and tend to reduce their cost.

Apart from the basic units, there are other aspects of the computer which need further consideration, principally concerning the imposition of various types of boundary conditions.

6.2 Neumann Boundary Conditions. With such conditions, the normal derivative of potential at the boundary is specified, rather than the potential itself. In fig. 6.2.1, if the normal derivative is known at point a on the surface S, then the potential difference between a and the next point out, b, can be deduced. The computer must then ensure that the correct p.d. exists between a and b at all times. This will involve special units at points like a which will initially have to emit enough pulses to bring its potential to the right amount above that of b and thereafter maintain this potential difference by emitting one pulse to b for every pulse received from b.. These requirements are simplified, of course, if $\frac{\partial V}{\partial \underset{\sim}{n}} = 0$ because then the initial emission of pulses becomes unnecessary, as a and b will normally be at the same potential at the start of a calculation.

The boundary units needed for Neumann conditions are, then, more complicated than for Dirichlet conditions but should be realizable in a not too complex form. It has not yet been possible to investigate designs for such a unit.

6.3 The Effect of an Electrically Isolated Conductor.

This case seems likely to be rather troublesome in realization. The essential requirement is that all points on the conducting surface must be always at the same potential. When the potential of the whole conductor rises by one unit, each point on the surface emits a pulse to its neighbours outside the surface. But before this can happen the conductor must have received one pulse on every input connected to mesh point units on its surface or, as will generally be the case, a number of pulses equal to the number of inputs (but not necessarily one on each input). This requirement is based on the principle that pulses are conserved in the system, as many pulses leave as enter a point and so as many pulses leave an area as enter it (assuming no pulse sources are enclosed).

What is required, then, is an enlarged version of a mesh point unit with a large number of inputs, and capable of counting to this number. It may be possible to connect the idle units in the interior of the conductor to do this, or at least to help with it, but this has not been investigated.

6.4 Special Units Needed at Dielectric Interfaces.

We shall consider only the simplest case, that of a straight boundary between two dielectrics with different permittivities, μ and M, the boundary being along a line of nodes as in fig. 6.4.1. Allen[12] has shown that the residue in a relaxation calculation at node '0' on the interface is:-

$$F_0 = \frac{2M}{\mu+M}\Phi_1 + \phi_2 + \frac{2\mu}{\mu+M} + \phi_4 - 4\phi_0 \quad \ldots(6.4.1)$$

where ϕ's are potentials on the left of the interface and Φ's potentials to the right. Hence ϕ_1 and Φ_3 are fictitious and have been eliminated in the above equation.

It is therefore necessary to interpose special units between units 3 and 0 and between 1 and 0, to multiply the number of pulses reaching point 0 from 3 and 0 by $2\mu/(\mu+M)$ and $2M/(\mu+M)$ respectively. The outgoing pulses from point 0 should be unaffected.

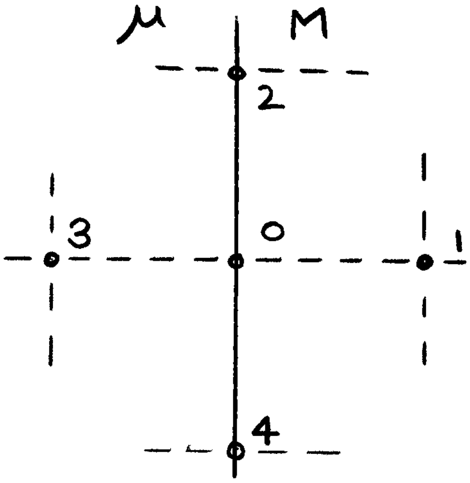


Figure 6.4.1

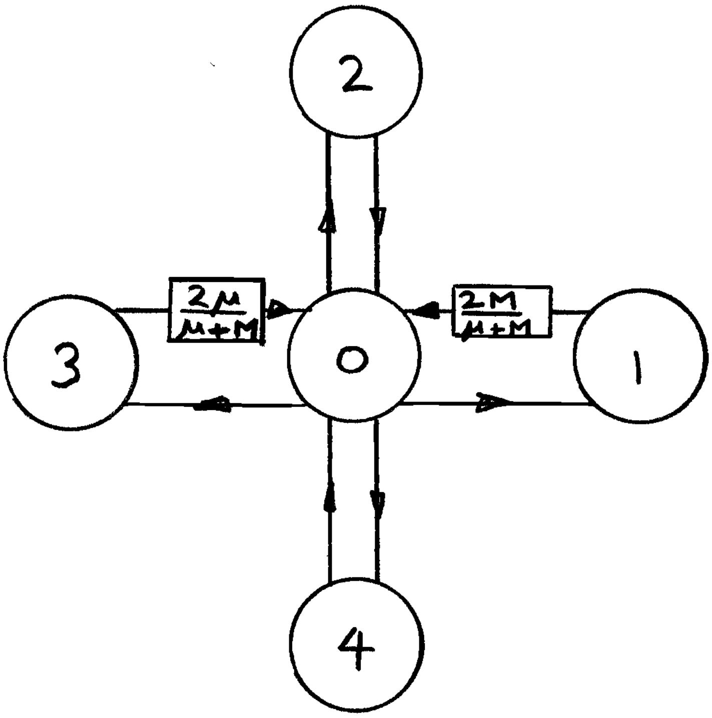


Figure 6.4.2

Fig. 6.4.2 shows which paths will be affected, plain arrows indicating unaffected paths.

These units will be quite difficult to design because $\mu/(\mu + M)$ or $M/(\mu + M)$ will in general not be a small integer, but a fraction. As the basic unit normally only divides by 2, 4 or 6 , these interface units must be considerably more complex than normal basic units. However, it is very important that al units should be kept as simple and reliable as possible. The following is only a suggested general scheme of operation.

The ratio of output to input number of pulses will be some number like 2.546, say. This number should be stored semi-permanently in the unit, but should be capable of being read nondestructively. Transfluxor cores might be used for this, with partial switching as in the basic units, but with the additional facility of nondestructive readout.

In order to achieve the non-integral multiplication of pulses, normally for every one input pulse there will be 2 output pulses. However, when there have been 10 input pulses, an extra 5 pulses must be emitted at the 10th input (for the ratio 2.546). Similarly, at every 100th pulse an extra 4 pulses must be emitted (a total of 2 + 5 + 4 pulses on every 100th pulse). At every 1000th pulse an extra 6 are emitted, and so on as in fig. 6.4.3.

Total no. of pulses in	Actual ratio of o/p:i/p	Total no. of pulses out	No. of o/p pulses per i/p pulse
1	2	2	2
2	2	4	2
3	2	6	2
⋮	⋮	⋮	⋮
9	2	18	2
10	2.5	25	7(=2+5)
11	2.46	27	2
12	2.42	29	2
⋮	⋮	⋮	⋮
19	2.26	43	2
20	2.50	50	7
21	2.48	52	2
⋮	⋮	⋮	⋮
30	2.5	75	7
⋮	⋮	⋮	⋮
99	2.46	243	2
100	2.54	254	11(=2+5+4)

Figure 6.4.3

Now, as more pulses come out than go in, trouble will arise if there is an input on successive clock cycles, as there will not be time to get rid of the output pulses. Therefore an output store will be needed, where the output pulses can wait their turn to be put out. An input store is also needed, so that the device knows when the 10th, 100th etc. pulse has come in.

Input and output registers could be of the same type as the potential registers used at each mesh-point. The output register will need to be more complex, in fact, as it will need to count both ways, backwards when a pulse is emitted to an adjacent point. The means of producing 2+5+4 pulses etc. seems likely to be the biggest problem, as this must be done in between computer clock cycles and so must be done rather quickly. This problem might be removed if, in the example chosen, 3 pulses were normally emitted for every input pulse normally and output suppressed at every 10th etc. pulse for the appropriate number of pulses.

6.5 Possibility of Improved Field Computer.

The existing design of Digital Field Computer solves only Laplace's and Poisson's Equations. For such a complex device this seems a rather small range and so it is of interest to inquire whether it is possible to extend the range of application of the computer, preferably with little extra complication. It has been shown that the computer does not solve the Diffusion Equation because can only take on a few discrete values. However, if the computer did its operations on the total potentials at each point, things would be rather different, as the required derivatives could be available. The total potential is in any case stored at each point so this would not cause any further complication. However, rather more complex arithmetic units would be needed. The arithmetical operations are not very complicated, division by 2, 4 or 6 and addition, but they involve the total potential and so the units would be more complex than the existing design.

Consider the Diffusion Equation;

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = A \frac{\partial V}{\partial t} \qquad \ldots(6.5.1)$$

or, in finite difference form;

$$\frac{V_1 + V_2 + V_3 + V_4 - 4V_o}{h^2} = \frac{A(V\delta - V_o)}{T}$$

$$\text{or,} \quad V_o' = V_o + \frac{T(V_1+V_2+V_3+V_4-4V_o)}{Ah^2} \qquad \ldots(6.5.2)$$

where V_o' is the new value and V_o the present value of the potential at the central point of the 5-point finite-difference star. If scale factors are arranged so that $T = Ah^2$, then the arithmetic is quite simple.

With more complication it is also possible to solve the wave equation;

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = B \frac{\partial^2 V}{\partial t^2} \qquad \ldots(6.5.3)$$

in finite difference form:-

$$\frac{V_1 + V_2 + V_3 + V_4 - 4V_o}{h^2} = \frac{B(V_o^+ + V_o^- - 2V_o)}{T^2}$$

or,

$$V_o^+ = 2V_o - V_o^- + \frac{T^2}{Bh^2}(V_1+V_2+V_3+V_4-4V_o) \qquad \ldots(6.5.4)$$

where V_o^+ is the new potential at point 'O', V_o the existing potential and V_o^- the previous potential. In this case it is necessary to store at each point the potential at the previous clock cycle as well as the existing potential.

These cases have not been investigated in further detail as even the Laplace computer is not yet made, but they show that the principle of the Digital Field Computer can be extended to differential equations other than Laplace's.

APPENDIX 1

THE SOLUTION OF A PIERCE GUN PROBLEM FOR RELATIVISTIC ELECTRON FLOW, USING A DIGITAL COMPUTER

A1.1 Derivation of the Differential Equation.

The problem is to design the focusing electrodes for a gun producing a planar electron beam with relativistic electron velocities. The equations governing the electron flow are:-

Poisson's Equation,

$$\frac{d^2V}{dx^2} = -\frac{\rho}{\epsilon} \qquad \text{...(A1.1.1)}$$

The kinetic energy equation,

$$c^2(m - m_o) = eV \qquad \text{...(A1.1.2)}$$

Where m is the mass and m_o the rest-mass of an electron.

$$m = \frac{m_0}{(1 - v^2/c^2)^{\frac{1}{2}}} \qquad \text{...(A1.1.3)}$$

where v is the velocity of an electron (uniform at any given cross-section).

From A1.1.2 and A1.1.3,

$$v^2 = c^2 \left\{ \frac{(eV/c^2 + m_o)^2 - m_o^2}{(eV/c^2 + m_o)^2} \right\}$$

Putting $U = eV/m_oc^2$ ...(A1.1.4)

$$v = \frac{c\ (U^2 + 2U)^{\frac{1}{2}}}{U + 1}$$

therefore, from A1.1.1 ;

$$\frac{d^2U}{dx^2} = \frac{eJ}{\epsilon m_o c^3} \cdot \frac{(U+1)}{(U^2+2U)^{\frac{1}{2}}}$$

Putting $eJ/\epsilon m_o c^3 = K$, where J is the current density in the beam,

$$\frac{d^2U}{dx^2} = K.\frac{U+1}{(U^2+2U)^{\frac{1}{2}}} \qquad \text{...(A1.1.5)}$$

or, $$\frac{d}{dU}\left(\frac{dU}{dx}\right)\frac{dU}{dx} = \frac{K(U+1)}{\sqrt{U^2+2U}}$$

integrating,

$$\frac{1}{2}\left(\frac{dU}{dx}\right)^2 = K(U^2+2U)^{1/2}$$

$$\frac{dU}{dx} = \sqrt{2K}\,(U^2+2U)^{1/4}$$

$$x = \frac{1}{\sqrt{2K}}\int_0^U \frac{dU}{(U^2+2U)^{1/4}} \qquad \text{...(A1.1.6)}$$

This equation gives the variation of potential along the beam edge, and from this we wish to find the necessary electric field outside the beam for this to be so.

A1.2 Application of Complex Variable Theory.

As the system is 2 dimensional, the results of complex variable theory can be applied to the solution of Laplace's Equation outside the beam. Any analytic function of a complex variable is a solution of Laplace's Equation. So, if we take equation A1.1.6 and replace $\sqrt{2K}x$ by z, = x+jy, and the potential function, U, by a "complex potential" U, = V + jW, and then integrate with respect to U, we shall obtain a description of the required electric field. Integrating with W = 0 is equivalent to integrating along the beam edge. Integrating along lines of constant V in the complex U plane will give equipotentials directly as complex numbers in the x + jy plane. In other words, the required electrode shapes are given directly .

<u>A1.3 Method of Computation.</u> The integral to be evaluated is

$$z = \int_0^U \frac{dU}{(U^2+2U)^{1/4}} \qquad .(A1.3.1)$$

over the complex U plane.

Although this integral can be expressed in terms of elliptic integrals and so can, in principle be solved analytically, such a process is extremely long and tedious. It was therefore decided to attempt a computer solution.

For purposes of computation, the complex U plane was split up into rectangles, n_2 in the V direction and m_2 in the W direction. The potentials were computed at each intersection, the paths of integration being along W=0 and then along V= const. For the numerical integration process, each of these steps was further divided into n_1 and m_1 divisions respectively.

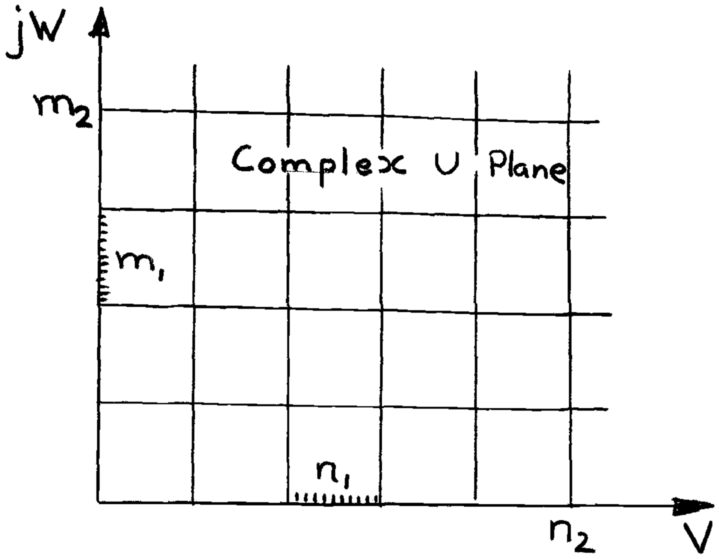


Figure A1.3.1

Al.4 Computer Programming. The computer used was an Englüsh Electric Deuce and the programme was written in Deuce Alphacode. This interpretive scheme incorporates instructions for manipulating complex numbers and for numerical integration using Simpson's Rule, both of which were used in the programme.

Al.5 Results Obtained. The results for the most accurate solution obtained are plotted in fig. Al.5.1. From this it is clear that the integration is not accurate enough, because the electric flux lines and equipotentials do not cross perpendicularly at the cathode electrode. There is, however, an improvement over an earlier , cruder result shown in fig. Al.5.2 and further improvement could therefore reasonably be expected if more steps of integration were used. However, the computation was discontinued at this stage, as about 8 hours computer time would be needed to get a significantly more accurate result. It was thus not considered worth while to continue in view of the expense, or alternitively the re-writing of the programme in Deuce machine language. The computation should be much more rapid on a larger more modern machine.

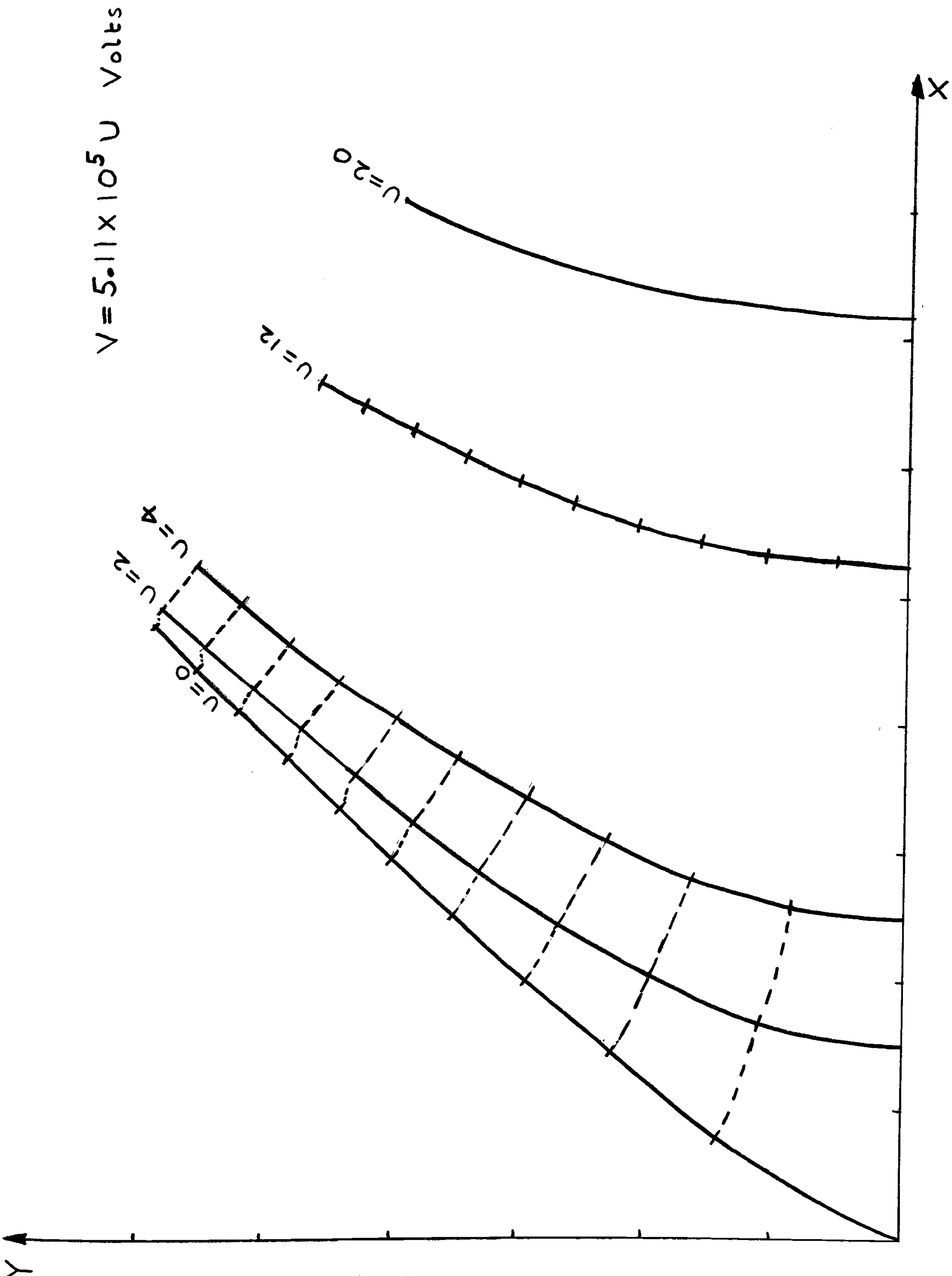


Figure A1.5.1

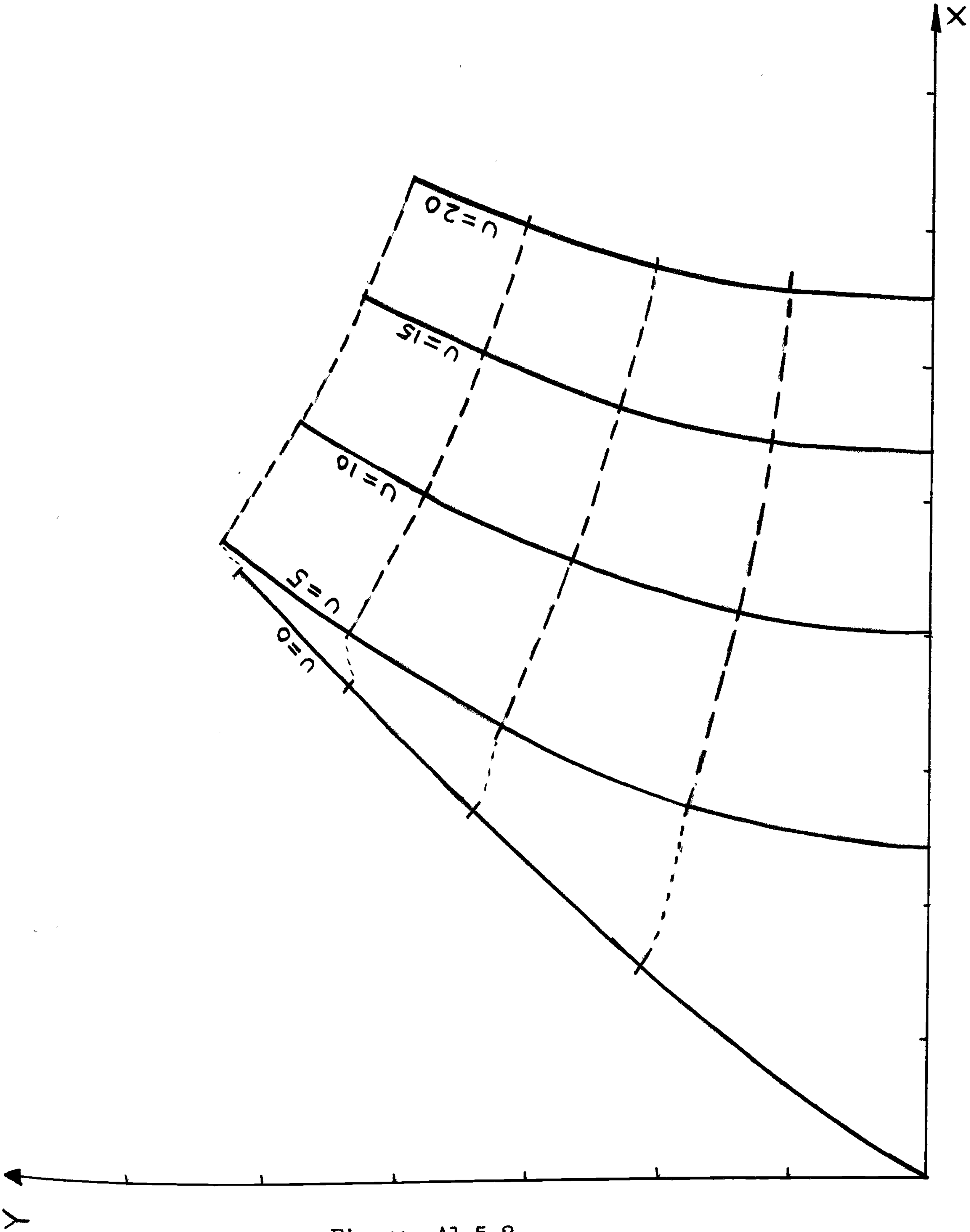


Figure A1.5.2

Programme for integrating $dU/(U^2+ 2U)^{1/4}$ in the complex U plane, using Deuce Alphacode.

No.	r	R	A	B	FUNCTION	C	D	P^0_s	NOTES
1	23			4	DATA	X1			Read Re. Data
2	54			3	Q DATA	X5			Read Im. Data
3	23			1	DATA	N1			n_1
4	23			1	DATA	N2			n_2
5	23			1	DATA	N3			m_1
6	23			1	DATA	N4			m_2
7	1		N13		+	1			
8	49		X703		Q +	X5		PO	
9	1		N10	1	+	1			
10	1		N12		+	N10			
11	1		T1		+	0			
12	1		T2		+	X4			
13	5				JUMP		S1		
14	49		X101		Q +	T3			
15	1		X12		+	X4			
16	1		X11		+	0			
17	1	1	X12	X12	+	X2			
18	49	3	T1		Q +	X11			T1=U
19	5				JUMP		S1		
20	12		N6		MODIFY				
21	49		X103		Q +	T3			$X103=(U^2+2U)^{-\frac{1}{4}}$
22	10			N6	UP TO	N3	R1		
23	1		N7	N3	+	1			$N7=m_1+1$
24	1	2	N9	N8	+	N8			
25	12		N8	N9	MODIFY				
26	0		X101	X101	MOVED				
27	12		N8	N9	MODIFY				
28	0		X301	X102	MOVED				
29	10			N8	UP TO	N7	R2		
30	1		X100		+	X2			
31	43		X21	N3	INTEGRAL	X100			X21=∫ReIΔW
32	1		N11	N12	+	N10			
33	12		N11	N12	MODIFY				
34	1		X702	X702	+	X21		PO	X702=Sum of ∫ReIΔW
35	1		X300		+	X2			
36	43		X22	N3	INTEGRAL	X300			X22=∫ImIΔW
37	12		N11	N12	MODIFY				
38	2		X701	X701	-	X22		PO	
39	55			1	Q RESULT	X11	1		
40	12				MODIFY	N11			
41	55			1	Q RESULT	X701	1		
42	1		N12	N12	+	N10			
43	12			N3	MODIFY				
44	0		X101	X101	MOVED				
45	12			N3	MODIFY				
46	0		X102	X301	MOVED				
47	10			N13	UP TO	N4	R1		
48	1		N12	N12	+	N10			
49	9			N14	>	0	R4		
50	1	N13			+	1			
51	1		N15	N4	+	1			

No.	r	R	A	B	FUNCTION	C	D	P^o_s	NOTES
52	12		N15		MODIFY				
53	49		X701		Q +	X7		Po	
54	1		N19		+	1			
55	12		N15		MODIFY				
56	49		X703		Q +	X9		PO	
57	1		N12	N12	+	N10			
58	1		N14		+	1			
59	49		X11		Q +	X3			
60	1		X501		+	X3	1		
61	5				JUMP		R1		
62	1	4	X11	X11	+	X1			
63	1		T1		+	X11			
64	1		T2		+	0			
65	5				JUMP	S1			
66	12		N23		MODIFY				
67	1		X502		+	T3			
68	10			N23	Up TO	N1	R4		
69	1		X500		+	X1			
70	43		X13	N1	INTEGRAL	X500			
71	L		N16	N2	+	1			
72	1		N17	N4	+	1			
73	1		N18	N19	÷	1			
74	3		N20	N17	x	N18			
75	3		N17	N19	x	N17			
76	1		N18	N20	+	N20		P	
77	1		N17	N17	+	N17		P	
78	12		N18	N17	MODIFY				
79	1		X701	X701	+	X13			
80	24			1	RESULT	X11	1		
81	12				MODIFY	N18			
82	24			1	RESULT	X701	1		
83	12			N1	MODIFY				
84	0		X501	X501	MOVED				
	01		X12		+	0			
85	10			N19	UP TO	N16	R1		
86	14				STOP				
87	19	S1			SUBROUTINE				
88	49		T3	T1	Q +	T1			
89	51		T1	T1	Q x	T1			
90	49		T3	T3	Q +	T1			
91	56		T3		Q ROOT	T3			
92	56		T3		Q ROOT	T3			
93	52		T3	1	Q ÷	T3			
94	20				END OF		S1		
95	18				FINISH				

APPENDIX 2

COMPUTER PROGRAMME FOR SIMULATING THE FIELD COMPUTER

A2.1 Organization of Data. At each mesh point, the total potential and the state of the mesh point counter (the 'residue') must be stored. In addition, in order to treat the data at a given point correctly it must be known whether the point is a boundary one or not and also some means of deciding when the end of the mesh has been reached must be incorporated. All the data for one point were stored in one word in the machine. The sign digit('a') was set to 0 (+) for all points at which computation was required, and set to 1 to indicate the end of the mesh. The last digit in the word (b) was set to 1 for a boundary point, or one outside the boundaries, and to 0 for an internal point. The state of the 'residue' was stored in binary places 35, 36 and 37 and the total potential in the places directly above these. As a two dimensional system was being considered, the maximum possible state of the residue was 7 (= 3 from the previous phase + a possible maximum of 4 in the current phase) and so needed 3 bits.

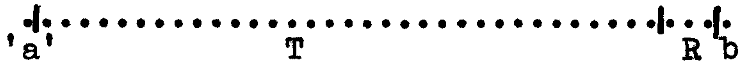


Figure A2.1.1

A2.2 Data Input. Boundary values were fed in in the form of pseudo orders and an Input Interlude rearranged these into the form indicated above. The modifier position contained the potential of the point, and 1 was put in the counter position and so provided the correct value for 'b'. A boundary point at zero potential could be more simply represented by a +1, and all internal points were set to have the desired residue, +0 for a zero residue, +4 for a residue of 2, and +6 for a residue of 3. The input interlude operated only on the number in the modifier position, so these latter numbers would be unaffected by it.

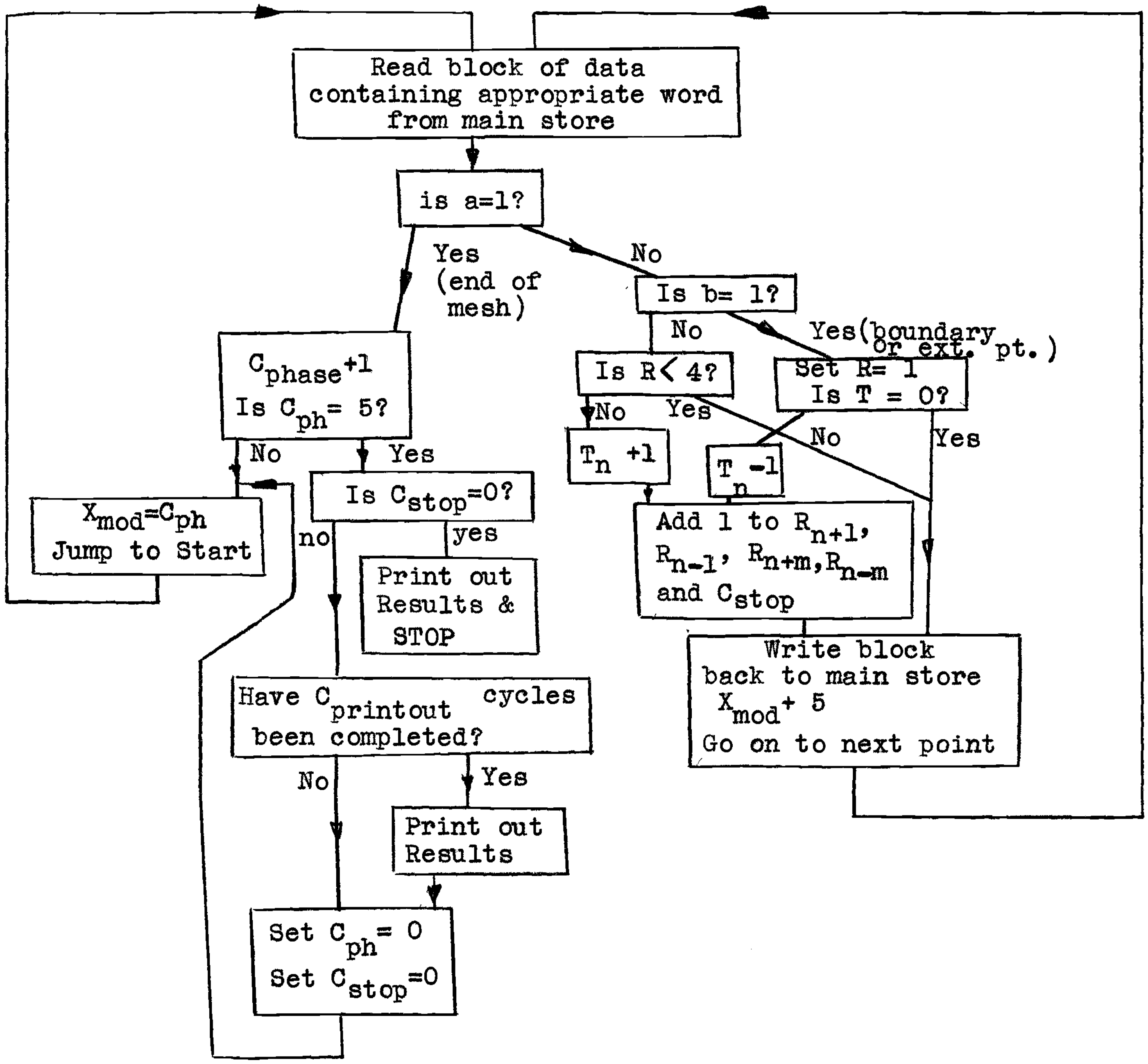
A2.3 The Computing Process. The programme scans through the mesh 5 times in a clock cycle. A cycle starts with the first point in the mesh, then jumps to the fifth, then to the tenth and so on to the end of the mesh. This completes the first phase. The next phase starts at the second point in the mesh, and so on for the 5 phases. At each internal point the residue, R, is checked and if it is greater than 4 then 1 is added to the residues of the 4 adjacent points and to the potential register of the point, the residue being reduced by 4 at the same time.

At a boundary point, a 1 is added to the adjacent residues and 1 subtracted from the potential register at each cycle until the register contains zero.

In order to determine when to stop the computation, 1 is added to a register (C_{stop}) every time any unit emits a pulse. C_{stop} is reset to zero at the end of each cycle unless it is already zero, in which case the computation stops and the final results are printed out.

A2.4 Print Out. Print out can be arranged to occur every time a given number of cycles have been gone through. The total potential together with the residue at each point are printed out in tabular form corresponding to the mesh of the Field Computer.

FLOW DIAGRAM FOR MESH POINT PROGRAMME


Read block of data containing appropriate word from main store
is a=1?
Yes (end of mesh)
No
Is b= 1?
No
Yes(boundary or ext. pt.)
C_{phase}+1 Is C_{ph}= 5?
No
Yes
X_{mod}=C_{ph} Jump to Start
Is C_{stop}=0?
no
yes
Print out Results & STOP
Is R< 4?
No
Yes
Set R= 1 Is T = 0?
No
Yes
T_n +1
T_n -1
Add 1 to R_{n+1}, R_{n-1}, R_{n+m},R_{n-m} and C_{stop}
Write block back to main store X_{mod}+ 5 Go on to next point
Have $C_{printout}$ cycles been completed?
No
Yes
Print out Results
Set C_{ph}= 0 Set C_{stop}=0

F.C.S. INPUT INTERLUDE

DATE ....

T 37.0

PAGE 1

0+

0.0	1.2	5	00		Set modifier and counter in X5
	2	5	72	5	Read block of Data
0.1	5.0	3	00	5	
	3	2	00		
0.2	0.5	3	63		Jump if a=1
	1.1	2	05		T only in X2
0.3	2	3	03		R,b only in X3
	21	2	53		T moved down 21 places
0.4	2	3	01		X3 = T + R + b
	5.0	3	10	5	N → V5
0.5	0.6+	5	66		X_m+1, Jump if not start of block
	1	5	73	5	
0.6	2	5	72	5	
	0.1	5	67		X_c-1, Jump if not end of Data
0.7	1.0	1	10		
	0				

1+

1.0	+0				Link for return to Initial Orders
1.1	1023	-	-0	0.	
	0				
1.2	0	-	50	0.	5 in mod. posn
	161				no. of numbers in data
.3					
.4					
.5					
.6					
.7					

.0				
.1				
.2				
.3				
.4				
.5				
.6				
.7				

F.C.S. MASTER PROGRAM

T 41.0

DATE

PAGE 2

0 +

0.0	40 0.7	4 5	72 00		Constants → U4 Set Modifier
0.1	42 0.0	0 0	72 60		Jump to Mesh Point Program
0.2	+0				
0.3	+0				
0.4	+0				
0.5	+0				
0.6	+0				
0.7	2 0	–	50	0.	Xmod indicates start of Data in Main Store

.0					
.1					
.2					
.3					
.4					
.5					
.6					
.7					

.0					
.1					
.2					
.3					
.4					
.5					
.6					
.7					

F.C.S. MESH POINT PROGRAM DATE..........

T 42.0

PAGE 3

0+

	0.0	0	7	73		Store X's in B0
5+.4+		1+	1	72		
	0.1	2+	2	72		
		3+	3	72		
	0.2	0				
		0	5	72	5	
3+.1	0.3	5.0	2	00	5	
		0.5	2	62		Jump for a ≠ 1
	0.4	4+	0	72		
		0.0	0	60		Jump to end of mesh prog.
4+.0	0.5	1	2	45		
		3.3	2	61		Jump for ext. pt. or boundary
3+.3	0.6	⑧	3	40		
		5.0	3	05	5	
	0.7	3.0	3	60		Jump if R < 4
3+.0		5.0	3	11	5	T_n+1 , R_n-4

1+

	1.0	4.5	4	00		Put 7 in mod. posn. of X4
3+.7+		5	4	05		Collate X_m with 7
	1.1	②	2	40		
		1.5	4	60		Jump if $P_m = 0$
	1.2	4.7	2	11	5	$R_{n-1}+1$
		1.7+	4	66		Jump if $P_m \neq 7$
	1.3	0	1	70	5	} $R_{n+1}+1$ for $P_m = 7$
		2	1	01		
	1.4	0	1	71	5	
		2.0	0	60		
	1.5	35	5	03		} R_{n-1} +1 for $P_m = 0$
		0	0	70	5	
	1.6	2	1	01		
		0	0	71	5	
	1.7	35	5	01		
		5.1	2	11	5	$R_{n+1}+1$

2+

2.0	5	3	00		
	4.3	3	03		(Mod. part of X3) − m
2.1	4.3	5	01		(" " X5) + m
	0	0	70	3	}
2.2	1	2	00		
	0	0	70	5	
2.3	1	4	00		
	2	1	00		
2.4	②	1	41		$R_{n-m}+1$ and $R_{n+m}+1$
	②	4	41		
2.5	0	0	71	3	
	4	1	00		
2.6	0	0	71	5	
	4.3	5	03		
2.7	①	1	40		
	4.7	1	11		

PAGE 4

```
0+.7         3+

0+.2+  3.0   0     5   73   5    Write back old block to Drum
             4.6   5   01        Modifier +5
       3.1   0.4   0   60
             0
0+.5+  3.2   0
             0
       3.3   ⑮     2   40
             5.0   2   05   5    R b → X2
       3.4   5.0   2   04   5    X2 = T only
             2     3   00        preserve T in X3
       3.5   3     2   41        T↓1 in X2
             5.0   2   10   5    Write back N to US
       3.6   3.0   3   60        Jump if T=0
             16    2   43        Tn-1
1+.0   3.7   5.0   2   10   5
             1.0   0   60

0+.4+        4+

       0.0   5+    1   72
             0     5   73   5
       0.1   4.2   3   00
             ①     3   41        Cphase +1
       0.2   5     2   40
             3     2   03        5 - Cph → X2
       0.3   1.1   2   61        Jump if Cph ≠ 5
             1.6   1   00        Set link in X1
       0.4   33    0   72
             0.0   0   60        Jump to Print Out
       0.5   0     3   00        Cph = 0
             5+    1   72
       0.6   4.7   1   00        Cstop → X1
             1.0+  1   61
       0.7   34    0   72
             0.0   0   60        Jump to Final Print Out & Stop

             5+

       1.0   0     0   77
             4.7   0   10        Reset Cstop
       1.1   4.2   3   10
             0     7   72        Reset X's
       1.2   4.2   3   00
             ㉕    3   52
       1.3   3     5   01        X mod + Cph
             0
0+.0+  1.4   0+    0   72
             0.0   0   60
       1.5   0
             0
       1.6   46    0   72        Link
             05    0   60
       1.7   0
             0
```

DATE

T 28.0

PAGE 5

0+

0.0	39	3	72		Print Out Constants → U3
	㉚	6	40		
0.1	16	6	10		Punch CR
	⑬	6	40		
0.2	16	6	10		Punch LF
	37	1	10		Store Link to M.P. Prog. in U3
0.3	0	7	72		Set X's
	1+	1	72		
0.4	2+	2	72		
	0	5	72	5	Read results to U5
0.5	5.0	2	00	5	Read one result → X2
	2	3	00		Preserve N
0.6	⑮	3	45		
	①	3	53		X3 = R only
0.7	④	2	53		X2 = T only
	⑩	7	40		X7 = 10 initially

1+

1.0	①	4	40		X4 = 1 initially (no. of dec. digits)
	7	2	03		T-10, T-100 etc.
1.1	1.3+	2	63		Jump if T<10, 100 etc.
	7	2	01		Restore T
1.2	3.4	7	20		Form 100, 1000 etc. in X7
	①	4	41		Increase X4 by 1
1.3	1.0+	0	60		
	7	2	01		Restore T
1.4	7	2	26		T/10, T/100 etc. → X7
	①	7	41		Correct for binary error in division
1.5	32	2	00		
	4	2	03		Subtract no. of digits from max in Print Out
1.6	⑭	1	40		
	16	1	10		Print spaces before number
1.7	1.6	2	67		
	3+	0	72		

2+

2.0	3.4	7	20		T/100 etc. X10	
	16	6	10		Punch dec. digit of T	
2.1	2.0	4	67			
	㉛	1	40		Comma	Omitted when only T is wanted
2.2	16	1	10			
	16	3	10		Punch R	
2.3	0					
	0					
2.4	0					
	3.3	2	00		C_p (no. of printed columns) → X2	
2.5	0.3	2	67		C_p-1, Jump if not end of printed row	
	㉚	6	40			
3+.3 2.6	16	6	10		Punch CR	
	⑬	6	40			
2.7	16	6	10		Punch LF	
	0.0	0	60			

DATE

PAGE 6

3+

0.0	39	3	70		Reset Cp
	3.3	1	10		
0.1	0				
	0				
2&5 0.2	0				
	0.3+	0	60		
0.3	3.3	2	10		Return Cp → U3
	3.1	2	00		Read Cf (no. of columns in field) → X2
0.4	4+	1	72		
	1.3+	2	67		Cf -1, Jump if not end of field row
0.5	33	6	40		
	16	6	10		
0.6	13	6	40		
	16	6	10		
0.7	39	1	70		
	3.1	1	10		

1.0	39	3	70	
	3.3	1	10	
1.1	3.0	3	00	
	1.2+	3	67	
1.2	3.7	0	60	
	3.0	3	10	
1.3	1.4	0	60	
	3.1	2	10	
1.4	1.5	5	66	
	0	5	72	5
1.5	0+	0	72	
	0			
1.6	1+	1	72	
	0.5	0	60	
1.7				

.0				
.1				
.2				
.3				
.4				
.5				
.6				
.7				

ADDENDA

DATE ....

PAGE 7

33

0 .0	0.7	1	10		
	4.0	2	00		
0 .1	0.4	2	67		Jump if not ($C_{P.O.}$)th cycle
	40	0	70		
0 .2	4.0	1	10		Reset $C_{print\ out}$
	0.7	1	00		Link → X1
0 .3	28	0	72		
	0.0	0	60		Jump to Print Out
0 .4	4.0	2	10		
	0				
0 .5	46	0	72		
	0.5	0	60		Jump back to Mesh Point Prog
0 .6	0				
	0				
0 .7	0				
	0				

34

0 .0	0.3	1	00		Link → X1
	0				
0 .1	28	0	72		
	0.0	0	60		Jump to Print Out
0 .2	0	0	77		Stop
	0				
0 .3	34	0	72		
	0.2	0	60		
.4					
.5					
.6					
.7					

.0				
.1				
.2				
.3				
.4				
.5				
.6				
.7				

DATE

PAGE 8

39

3.0	+12				C_{Row}, no. of Rows in Field
3.1	+13				C_{field}, no. of columns in Field
3.2	+4				(Max no. of digits in T) +1
7.3	+13				C_p, no. of columns on Paper
3.4	+10				10 for decimal point calculation
3.5	+0				
3.6	+0				
3.7	+0				Link to Mesh Point Prog.

40

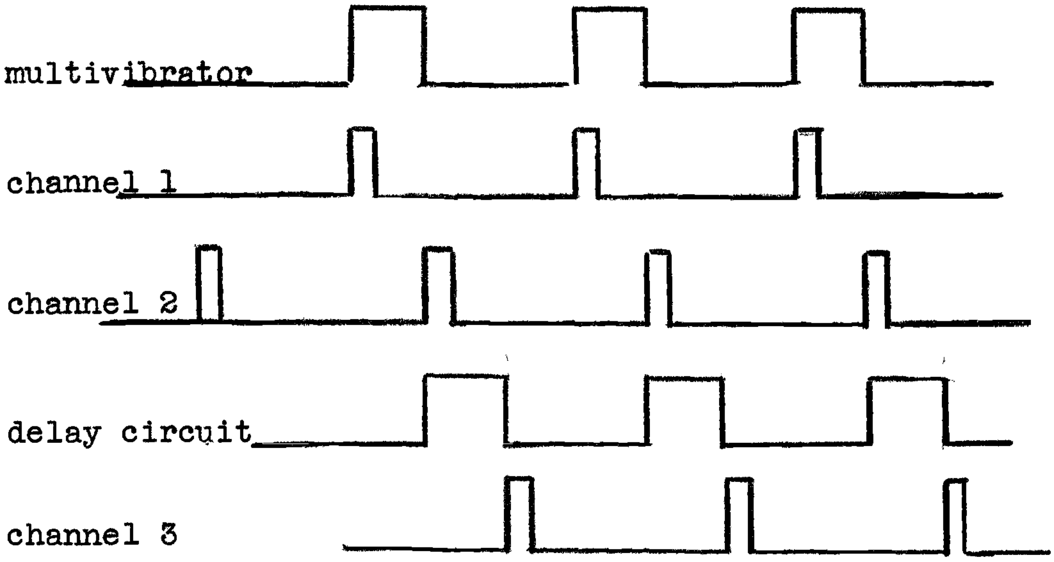
4.0	+30				$C_{print\ out}$, no. of cycles per print out
4.1	+0				
4.2	+0				C_{phase}
4.3	1 0	−	50	0.	m, no. of columns, put into mod. posn
4.4	0 0	−	30	0.	
4.5	0 0	−	70	0.	
4.6	0 0	−	50	0.	
4.7	+0				C_{stop}

.0					
.1					
.2					
.3					
.4					
.5					
.6					
.7					

APPENDIX 3

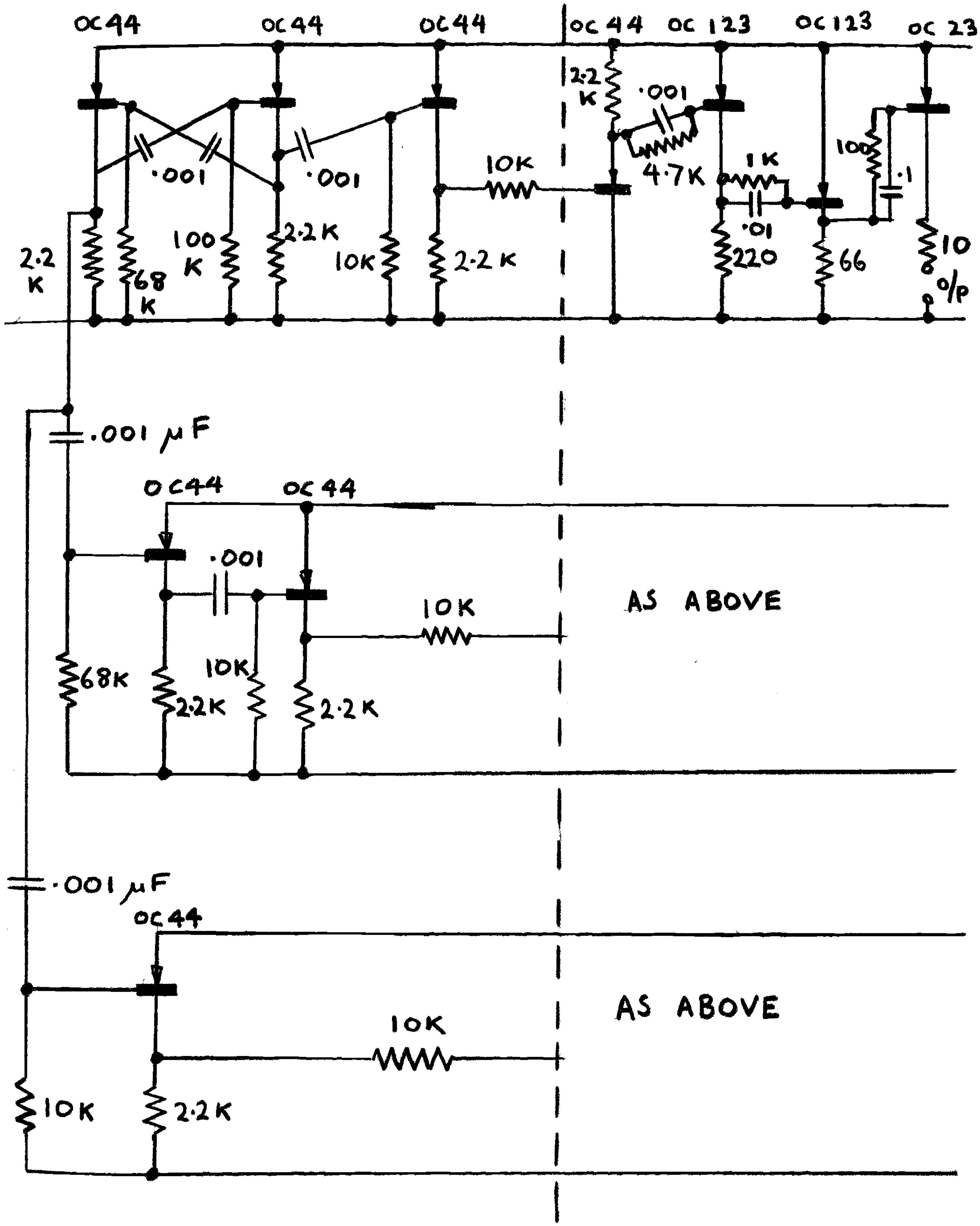
EXPERIMENTAL THREE PHASE PULSE GENERATOR

A3.1 Timing Circuits. A multivibrator is used as the basis of the timing of the pulses. It produces a rectangular waveform with a mark/space ratio of 1:2. The leading edge of this pulse triggers a monostable circuit which is 'on' for the required duration of the output pulse. This pulse is amplified for output channel 1. The trailing of the pulse triggers another similar monostable circuit which is connected to output amplifier 2. This edge also triggers a different monostable circuit, the 'delay circuit', and the trailing edge of the pulse thereby produced is used to trigger a monostable circuit of the former type and so give channel 3.



A3.2 Pulse Amplifier. This is quite straightforward, and the amplitude of the output pulses can be varied from about 10 volts to 30 volts by varying the supply voltage to the output stage. The 10 ohm resistor in the collector circuit of this stage is to limit the maximum current through the transistor to avoid damaging it, but normally the output pulses have a nearly constant-voltage characteristic. For an output pulse of 1 amp the rise time of the pulses is about 1 microsecond.

This pulse generator is fairly crude and an improved one, with a better timing system and faster rise time is needed for a complete computer. The efficiency of such a generator could be much greater than for this one, about 80% should be easily attainable.


OC44
OC44
OC44
OC 44
OC 123
OC 123
OC 23
2·2 K
·001
4·7K
1 K
·01
220
66
100
·1
10
o/p
·001
·001
10K
2.2 K
68 K
100 K
2·2K
10K
2·2 K
.001 μF
OC44
OC 44
·001
10K
AS ABOVE
68K
2·2K
10K
2·2K
·001 μF
OC44
AS ABOVE
10K
10K
2·2K

REFERENCES

1 Kirstein, P.T., Microwave Lab. Report No. 440, Stanford University (1958).

2 Pierce, J.R., J. App. Phys., 11, 548 (1940).

3 Kyhl, R.L. and Webster, H.F., General Electric (New York) Reports No. 56-RL-1556 (1956) and No. 57-RL-1800 (1957).

4 Epsztein, M.B., Comptes Rendus, 242, 1425 (1956).

5 Meltzer,B. and Brown,I.F., Nature, 181, 1384 (1958)

6 Brown, I.F., 'A Design Study for a Digital Computer for Solving Field Equations', Edinburgh University Postgraduate Report No. EDR 37 (1957).

7 Gianola, U.F., B.S.T.J., 39, 295 (1960).

8 'Digital Applications of Magnetic Devices', p.480, Meyerhoff, A.J. (Editor), Wiley (New York), and 'Solid State Magnetic and Dielectric Devices', Katz., H.W., (Editor), Wiley (New York), 1959, p. 357.

9 Radley, D.E., J. Electron & Control, 4, 125 (1958).

10 Forsythe, G.E., and Wasow, W.R., 'Finite Difference Methods for Partial Differential Equations', Wiley (N.Y.), 1960, pp. 11-14.

11 Meltzer, B., and Brown, I.F., Nature, 181, 1384, (1958), and Electronic Engineering, 31, 590, (1959).

12 Allen, D.N.de G., 'Relaxation Methods in Engineering and Science', McGraw Hill, 1954, p 198.

AKNOWLEDGEMENTS

I wish to thank my supervisors, Dr. B. Meltzer and Dr. J.V.Oldfield, for their help and suggestions during the work for this thesis.