63-4-1

TR-1072

DIODELESS MAGNETIC CORE LOGIC CIRCUITS

Dale L. Hamilton

15 August 1962

DIAMOND ORDNANCE FUZE LABORATORIES
ORDNANCE CORPS ● DEPARTMENT OF THE ARMY
WASHINGTON 25, D. C.

HARRY DIAMOND LABORATORIES

Robert W. McEvoy
LtCol, Ord Corps
Commanding

B. M. Horton
Technical Director

MISSION

The mission of the Harry Diamond Laboratories is:

(1) To perform research and engineering on systems for detecting, locating, and evaluating targets; for accomplishing safing, arming, and munition control functions; and for providing initiation signals: these systems include, but are not limited to, radio and non-radio proximity fuzes, predictor-computer fuzes, electronic timers, electrically-initiated fuzes, and related items.

(2) To perform research and engineering in fluid amplification and fluid-actuated control systems.

(3) To perform research and engineering in instrumentation and measurement in support of the above.

(4) To perform research and engineering in order to achieve maximum immunity of systems to adverse influences, including countermeasures, nuclear radiation, battlefield conditions, and high-altitude and space environments.

(5) To perform research and engineering on materials, components, and subsystems in support of above.

(6) To conduct basic research in the physical sciences in support of the above.

(7) To provide consultative services to other Government agencies when requested.

(8) To carry out special projects lying within installation competence upon approval by the Director of Research and Development, Army Materiel Command.

(9) To maintain a high degree of competence in the application of the physical sciences to the solution of military problems.

DIAMOND ORDNANCE FUZE LABORATORIES
ORDNANCE CORPS WASHINGTON 25, D. C.

DA-5N06-01-014
OMS Code 5530.11.62400
DOFL Proj 46345

TR-1072

15 August 1962

DIODELESS MAGNETIC CORE LOGIC CIRCUITS

Dale L. Hamilton

FOR THE COMMANDER:
Approved by

Robert S. Hoff

R. S. Hoff
Chief, Laboratory 400

CONTENTS

ILLUSTRATIONS

ABSTRACT

This report shows that it is possible to perform all the basic logic functions associated with memory, "and" gates, "or" gates, and "not" gates by using circuits composed only of square loop magnetic cores and interconnecting wire. A design procedure is given for all the basic logic circuits, a coincidence detector, and a shift code counter. These circuits were developed using miniature ferrite cores that have switching times on the order of microseconds.

1. INTRODUCTION

During the last few years much attention has been given to the field of ferromagnetic materials and devices. This attention has arisen mainly because of the ability of ferromagnetic devices to store information indefinitely. Initially their use was confined to computer memory elements, but more recently they have been used for performing logic functions as well. The first logic circuits merely used magnetic cores as another circuit element. Gradually many of the other circuit elements were eliminated because it was found that it is possible to use magnetic cores in such a way that active devices are not necessary. This was highly desirable because of the high reliability of magnetic cores. These circuits were gradually simplified to the point where only magnetic cores and diodes were necessary to perform logic operation and consequently much attention has been given to this type of circuit (ref 1). Several years ago it was speculated that it might be possible to build entire logic systems with circuits that consist of nothing but magnetic cores and interconnecting wire. The investigation presented in this report was started so that logic systems such as these could be built. There are essentially two basic methods by which magnetic logic circuits may be constructed. The first involves the use of simple toroidal magnetic cores and the second employs multiaperture cores. While both elements are capable of performing all magnetic logic, each type possesses certain advantages over the other. For example, the multiaperture elements require fewer cores to perform a given logic function than their toroidal counterpart. The toroidal cores possess such advantages as size and simplicity of analysis. This latter point is particularly significant. The flux configuration in multiaperture cores is extremely complicated and not very well understood. The flux configuration in toroids is relatively simple since only one flux path exists. For these and other less important reasons the toroidal core circuits were selected for analysis. Simple shift registers were constructed in the all magnetic form several years ago in both multiaperture and toroidal form (ref 2, 3). The extension of the basic toroidal shift register circuits into logic circuits is considered in detail in the remainder of this report. To the knowledge of the author this is the first complete report on a system utilizing toroidal magnetic cores that can perform any logic function. As anticipated, work on multiaperture cores was continued elsewhere and was recently found to be successful (ref 4).

In a previous report, a method of constructing various all magnetic shift registers was described (ref 5). This report, which is essentially

a continuation of the previous one, describes a method of magnetically reading information out of a shift register and subsequently performing various logic operations. To utilize the logic techniques developed, several systems using all magnetic logic were successfully designed and constructed. The most outstanding was a six stage shift code counter with readout. As a refinement to the basic system a method of replacing two of the four drive pulses with a d-c bias is described.

The materials used in this investigation are all miniature, square loop, ferrite toroids. These materials were used for the following reasons:

(a) Extreme miniaturization potential

(b) Ruggedness

(c) Reliability

(d) Commercial availability at nominal cost

2. LOGICAL FUNCTIONS

2.1 Branching

To effectively utilize information in a shift register, a method of extracting information must be found. This is known as branching and simply refers to a device with a single input and identical multiple outputs. The number of outputs determines the branching factor. In the circuits to be described a branching factor of two was the greatest value obtained.

As an example, consider the block diagram in figure 1 below. If information is progressing across stages 1, 2, and 3 and then branches to stages 4 and 6, then stage 3 is a stage of a shift register that exhibits a branching factor of two. To describe in detail how this is accomplished, a thorough knowledge of the improved shift register in section 3, DOFL Report R320-60-25 is assumed. A schematic diagram of the branching circuit is illustrated in figure 2. The dash-outlined boxes

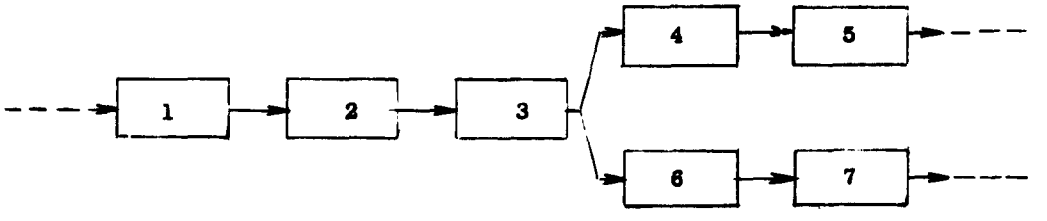


Figure 1. Logical branching diagram.

numbered 3, 4, and 6 correspond to the numbered boxes in the block diagram of figure 1. To simplify the diagram, all drive and bias windings are represented by an arrow toward a zero or a one to indicate the direction in which the current tends to drive the core (current into a

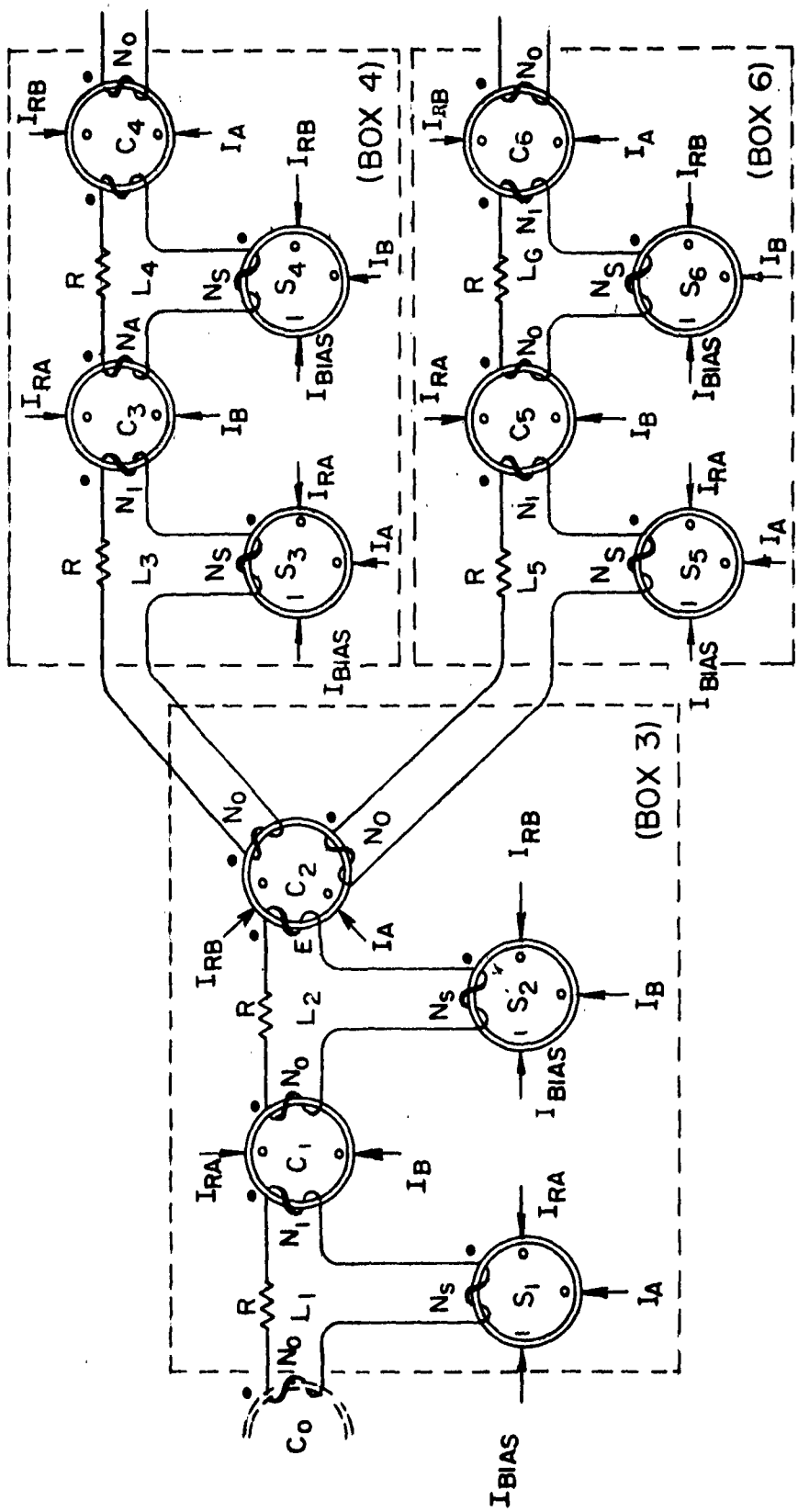


Figure 2. Branching circuit.

dot drives a core toward the zero state). As before, the pulse sequence is I_A, I_{RA}, I_B, I_{RB}, and hereafter a full pulse sequence will be referred to as a clock sequence.

To understand the operation of the branching circuit, consider the condition existing when core S_1 is in the one state and all other cores are in the zero state. Upon application of the I_A pulse, core S_1 is driven toward the zero state, causing the core to switch and producing a counterclockwise current in coupling loop L_1. This current tends to drive core C_o to the zero state and core C_1 to the one state. Since both cores were originally in the zero state, only core C_1 switches producing a counterclockwise current in loop L_2. This current tends to drive cores C_2 and S_2 to the one state. Since core C_2 is held in the zero state by the I_A pulse, only core S_2 switches. Thus after the I_A pulse is completed, cores C_1 and S_2 are in the one state and all other cores are in the zero state. During the next pulse (I_{RA}), core C_1 is driven back to the zero state. This switching produces a counterclockwise current in loop L_1 and a clockwise current in loop L_2. The loop L_1 current drives C_o toward the zero state and S_1 toward the one state. Since S_1 is held in the zero state by I_{RA}, it does not switch. The current in coupling loop L_2 drives C_2 and S_2 toward the zero state. Since C_2 is already in the zero state it is not affected, but this current tends to switch S_2 since it is in the one state. Since the switching of S_2 is undesirable, the amplitude of the I_{RA} pulse must be limited so that the resulting loop current is not strong enough to switch S_2. This amplitude limiting of I_{RA} accounts for the slow switching of the coupling cores and thus for the relatively slow operation of this type of circuit. Therefore, after the I_A and I_{RA} pulses, a one has been transferred from core S_1 to S_2. The sequence just described is identical with that of the improved shift register described in the previous report (ref 5).

During the I_B and I_{RB} pulses, the actual branching function is accomplished. On the application of I_B, core S_2 is switched toward the zero state producing a counterclockwise current in loop L_2. This current does not affect C_1 but causes C_2 to switch to the one state, resulting in counterclockwise currents in coupling loops L_3 and L_5. These currents tend to switch C_3, S_3, C_5, and S_5 toward the one state, but C_3 and C_5 are held in the zero state by the I_B pulse. Thus I_B has transferred a one from S_2 to C_2, S_3, and S_5. During the I_{RB} pulse, core C_2 is driven to the zero state causing counterclockwise current in L_2 and clockwise currents in L_3 and L_5. The L_2 loop current tends to switch S_2, but it can not because of the overriding I_{RB} pulse. Loop currents L_3 and L_5 tend to drive S_3 and S_5 to the zero states. As with the I_{RA} pulse, I_{RB} is amplitude limited so as to cause small loop currents that do not affect cores S_3 and S_5. Therefore, after a full clock sequence (I_A, I_{RA}, I_B, I_{RB}) a one has been transferred from S_1 to S_3 and S_5. During the next clock sequence the ones in S_3 and S_5 progress along their respective registers in the manner just described. It should be noted that core S_2 drives three cores instead of two as used in a regular shift register, so it must be driven much harder during the transfer operation. This is accomplished by using more turns on the I_B winding of S_2. A similar statement can be made about the I_{RB} winding of core C_2. It does not drive more cores but contains more output windings

in which the output current can divide. Also, for effective operation, the number of turns on the loop L_2 winding of C_2 must be modified. These modifications are indicated in the circuit analysis section of the report.

2.2 "And" Gate

To perform logical operations, an "and" gate is one of the necessary functions. The block diagram of figure 3 shows two two-stage shift registers feeding an "and" gate. Thus it is necessary to have a one present in both shift registers in order to have an output. The circuit details of stages 2, 4, and the "and" gate are shown in figure 4. The dotted boxes labeled 2, 4, and "and" gate correspond to those of the block diagram of figure 3. To understand the operation

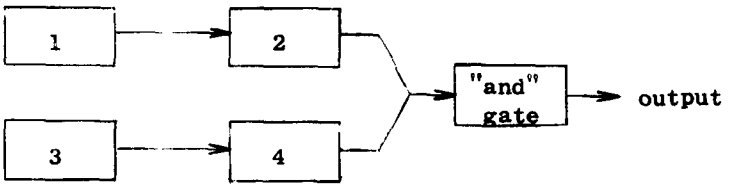


Figure 3. Logical "and" gate diagram.

of the circuit, consider the case where cores S_1 and S_3 contain ones and all other cores are in the zero state. This corresponds to the case where a one exists in stages 2 and 4 and we wish to detect their simultaneous presence. On the application of the I_A pulse, the one in core S_1 is transferred to cores C_1 and S_2 and the one in S_3 to C_4 and S_4. The I_{RA} pulse returns C_1 and C_4 to the zero state. Thus as before, after the I_A and I_{RA} pulse, the ones in S_1 and S_3 have been transferred to S_2 and S_4, respectively. During the I_B pulse the one in S_2 is transferred to C_2 and S_5 and that in S_4 to C_5 and S_6. The I_{RB} pulse returns C_2 and C_5 to the zero states. Therefore after one clock sequence, the ones in S_1 and S_3 have been transferred to storage cores S_5 and S_6 in the "and" gate. At this point the operation markedly deviates from that previously described. Upon application of the next I_A pulse, the ones in S_5 and S_6 are driven out producing a clockwise current in L_5 and a counterclockwise current in L_6. It should be observed that neither loop current affects C_2 or C_5, but both currents drive C_6 to the one state producing a voltage on the output winding. During the I_{RA} pulse, C_6 is driven back to the zero state producing another voltage of opposite polarity on the output winding. Thus it is seen how the presence of ones in stages 2 and 4 caused an output after two clock sequences. If a one had been present in only one of the stages, say stage 2 (core S_1), then we would not want an output to occur. To consider this case, let us again start with a one in S_1 and after a single clock sequence the one would be transferred to S_5. During the next I_A pulse, the one in S_5 is driven out so as to cause a clockwise current in loop L_5. This current tends to drive C_6 to the one state, but the turns on the I_A winding of S_5 along with the N_F windings of C_6 are adjusted so that the loop current is just short of that which would cause

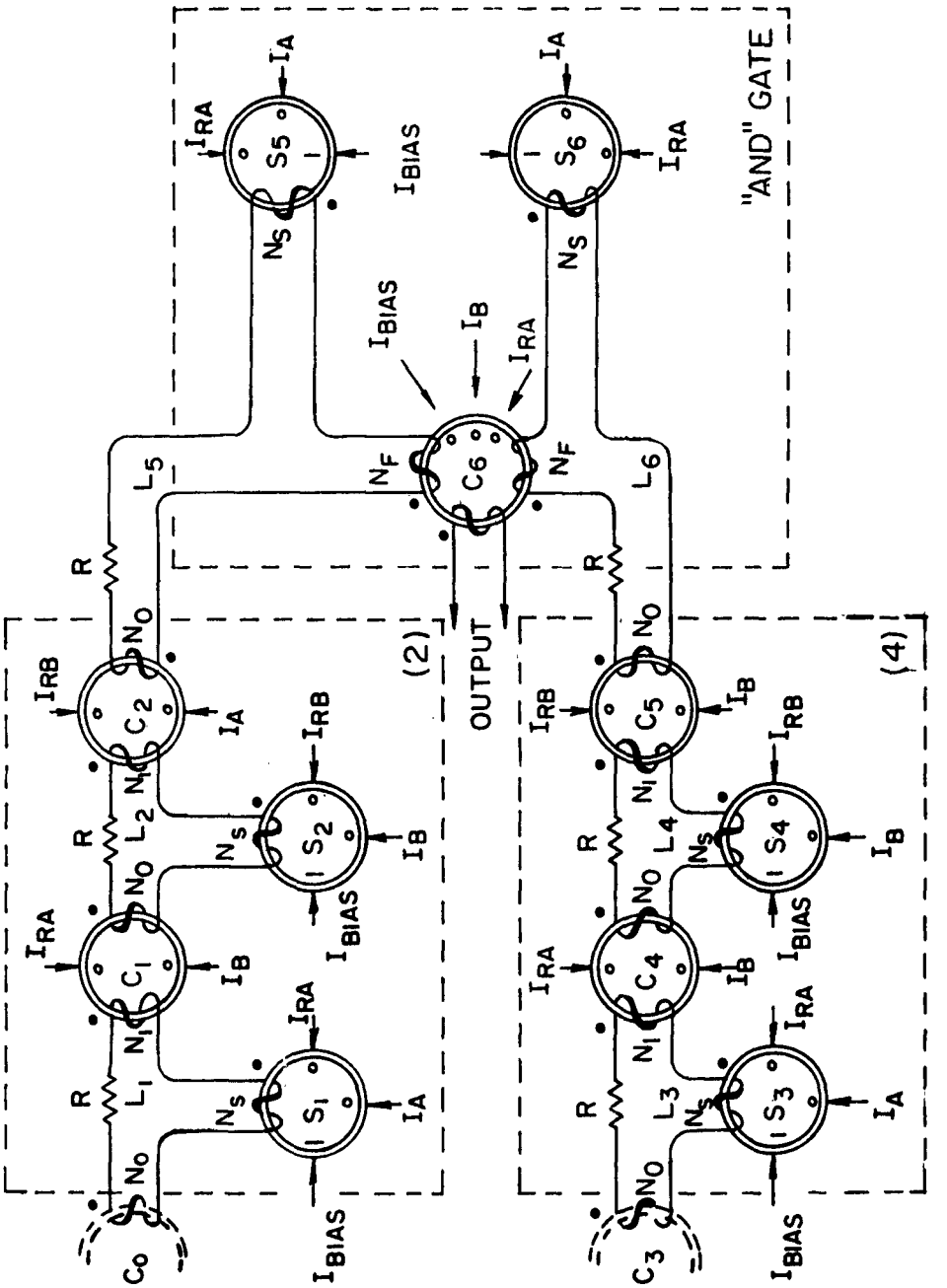


Figure 4. "And" gate circuit.

C_6 to change states. Thus core C_6 operates on a coincident current basis, and its operation is one of the most critical parts of the circuit. To help alleviate this critical condition, note that core C_6 has been biased toward the zero state. A typical hysteresis loop for the materials used in this report is illustrated in figure 5. If the

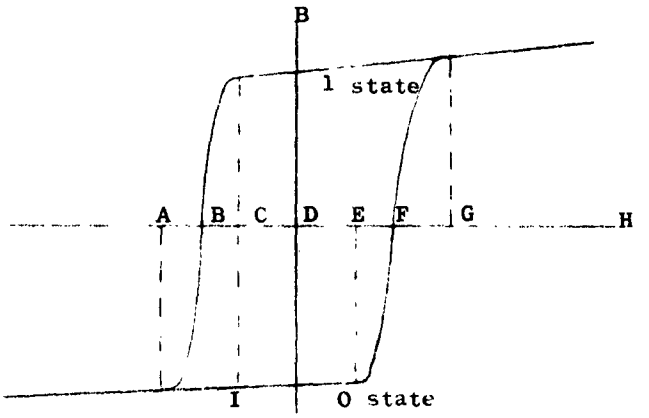


Figure 5. Typical hysteresis loop.

core were not biased then the magnetomotive force (mmf) caused by just one loop current would be limited to a value equal to or less than DE. Then if inputs from both shift registers occurred, the mmf would be twice DE and the core would change states. To improve the operation consider the core to be biased to point I (fig. 5). Here the mmf due to one loop current could be as much as CE and that due to two loop currents twice CE. Thus it is seen that biasing the core to the zero state results in a greater stability of operation. It should be noted that in the "and" gate just described the final output from the gate did not work into a load. This was done because it was not needed for the particular application at hand. The author feels that it would be a relatively simple task to contrive such a circuit. It should also be noted that all other logic devices described in this report do work into loads and thus are more compatible with a complete system.

2.3 "Or" Gate

The construction of a multiple input "or" gate can be performed in a rather straightforward manner. To understand the operation of such a gate, consider the block diagram of figure 6 and the

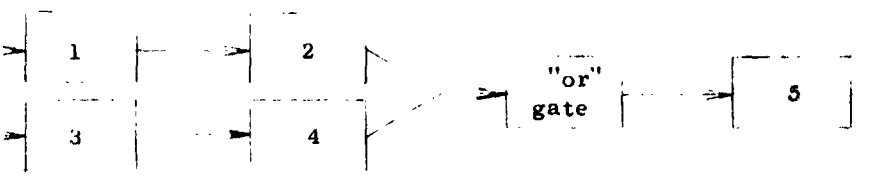


Figure 6. Logical "or" gate diagram.

corresponding schematic diagram of figure 7. If a one exists in core S_1 of stage 2, then after the I_A and I_{RA} pulse it will be transferred to core S_2. The application of the I_B pulse will cause a counterclockwise current in L_2 and a clockwise current in L_3. It should be observed that the current in L_3 will not affect C_7, and thus the usual transfer operation will not be affected by the presence of this core. The same result will occur if a one exists in stage 4 or even if a one started in both stages 2 and 4. In this latter case the loop current L_3 produced by the shifting of ones out of cores S_2 and S_6 will result in a larger current that causes S_3 to switch much faster. Information in stages 2 and 4 ends in the "or" gate without affecting the other stage. It should be noted that one intricate difficulty occurs in the "or" gate. If a one is transferred during the I_B pulse from both cores S_2 and S_6 to cores C_2, C_7, and S_3, then during the reset operation (I_{RB}) both cores C_2 and C_7 must be reset. When the cores are reset simultaneously, both cores contribute current to loop L_3, resulting in a higher value. Recalling that it is just this loop current that must be limited to avoid destroying the information in core S_3, then this is an area of possible trouble. One way to circumvent this trouble is to observe that during the transfer of a one from both cores S_2 and S_6 to C_2, C_7, and S_3, core S_3 switches much faster than normal because of the greater current in loop L_3. Thus S_3 switches before the other cores and then essentially puts a "short" across its loop L_3 winding. This puts a very large load on the other cores and greatly slows down their switching process. If the drive pulse were limited in width so that it would be sufficient for a normal transfer but not sufficient to complete this so-called loaded transfer, then the cores C_2, C_7, S_2, and S_6 would not be completely switched. Now during the following I_{RB} reset pulse the resulting currents would be smaller because the cores only have to be partially switched to return them to the zero state. It should be noted that during reset the loop L_3 current increases only slightly for each additional coupling core in the loop. Thus it is necessary to limit the drive pulse width only when an "or" gate with many inputs is being used. None of the "or" gates that were tested required this narrow drive pulse. One advantage of the "or" gate over the "and" gate is that, except for the minor restrictions just described, it can be used with multiple inputs. As will be shown later, a five input "or" gate was successfully operated.

f7

2.4 Inhibit Function

To perform many logic operations such as a "not" gate and an "exclusive or" gate a device to inhibit or block another stage is a necessity. An inhibit device is shown in block diagram form in figure 8. If a one exists in both stages 2 and 4, then the inhibit stage does not give an output. If a one exists only in stage 4, then it is merely passed through the inhibit stage, and if a one exists only in stage 2, it is merely lost during the next clock sequence. A detailed schematic diagram of stages 2, 4, and the inhibit gate is shown in figure 9. To understand the operation of the circuit, consider the case where a one exists in cores S_1 and S_4 and all other cores are in the zero state. After one clock sequence, the ones in S_1

f8

f9

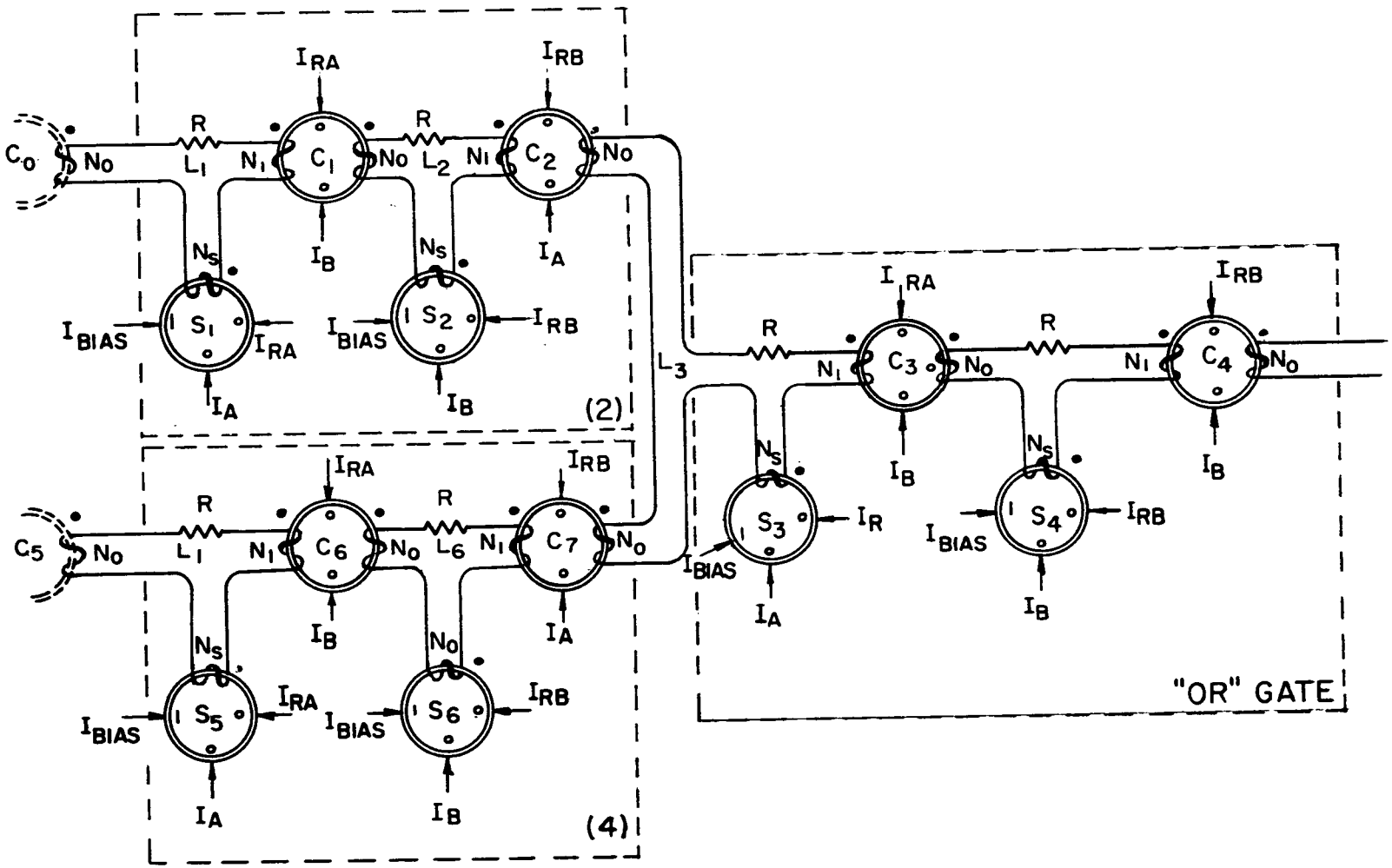


Figure 7. "Or" gate circuit.

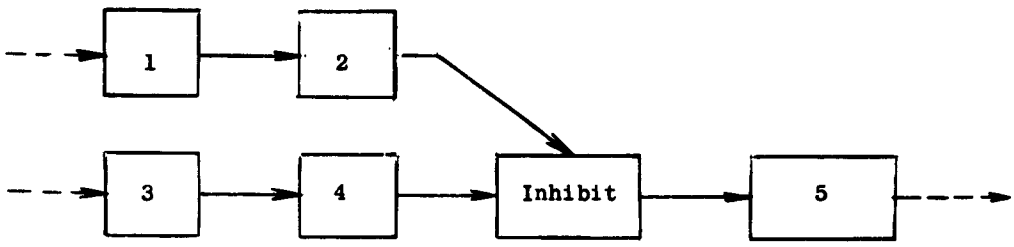


Figure 8. Logical inhibit diagram.

and S_4 have been transferred to S_3 and S_6 in the usual manner. The next I_A pulse causes S_3 and S_6 to switch toward the zero state producing counterclockwise currents in L_3 and L_6. The current in L_3 tends to drive C_5 to the zero state and the current in L_6 tends to drive it to the one state. Provided the current and number of turns in the loops are identical, this bucking action produces no net switching of core C_5. Thus after the entire sequence of pulses has been completed, all cores are in the zero state and the effective inhibiting of the one in core S_4 by that in core S_1 has been accomplished. It should be observed that to complete an effective inhibit action all parameters in loops L_3 and L_6 must be identical. That is, all resistances and numbers of turns must be equal. To complete the description of the inhibit gate consider the case where a one exists only in core S_4. After a clock sequence the one has progressed to core S_6. During the next I_A pulse, core S_6 tends to switch to the zero state producing a counterclockwise current in loop L_6. This current drives C_5 to the one state producing a counterclockwise current in loops L_7 and L_3. The current in L_3 drives C_2 to the zero state and S_3 toward the one state, but S_3 is held in the zero state by the overriding I_A pulse. Thus loop L_3 acts as a low impedance load on C_5 when a one is being driven out of S_6 into C_5 and S_7. This low impedance load means that core S_6 has to be driven very hard to complete the transfer of the one. To accomplish this, the I_A winding of core S_6 must have more turns than the other I_A windings. After the passage of the one from S_6 to C_5 and S_7, C_5 is reset to the zero state in a normal manner for a transfer core with three loops tied to it (see reset description for branching circuit). The last and almost trivial case occurs when a one exists in S_1 and all other cores are in the zero state. After one clock sequence the one is transferred to core S_3. During the next I_A pulse the one is driven out of S_3 producing a counterclockwise current in loop L_3 that tends to drive C_2 and C_5 to the zero state. Since both cores are already in the zero state no switching occurs, and the one that existed in S_3 is simply driven out and lost. This is precisely the desired result since there was not any one from stage 4 to inhibit.

2.5 "Not" Gate

Basically a "not" gate is a simple extension of the inhibit function described in the previous section. The diagram for such a gate is represented in block form in figure 10 and schematically in figure 11. If all the cores in figure 11 are in the zero state then, after

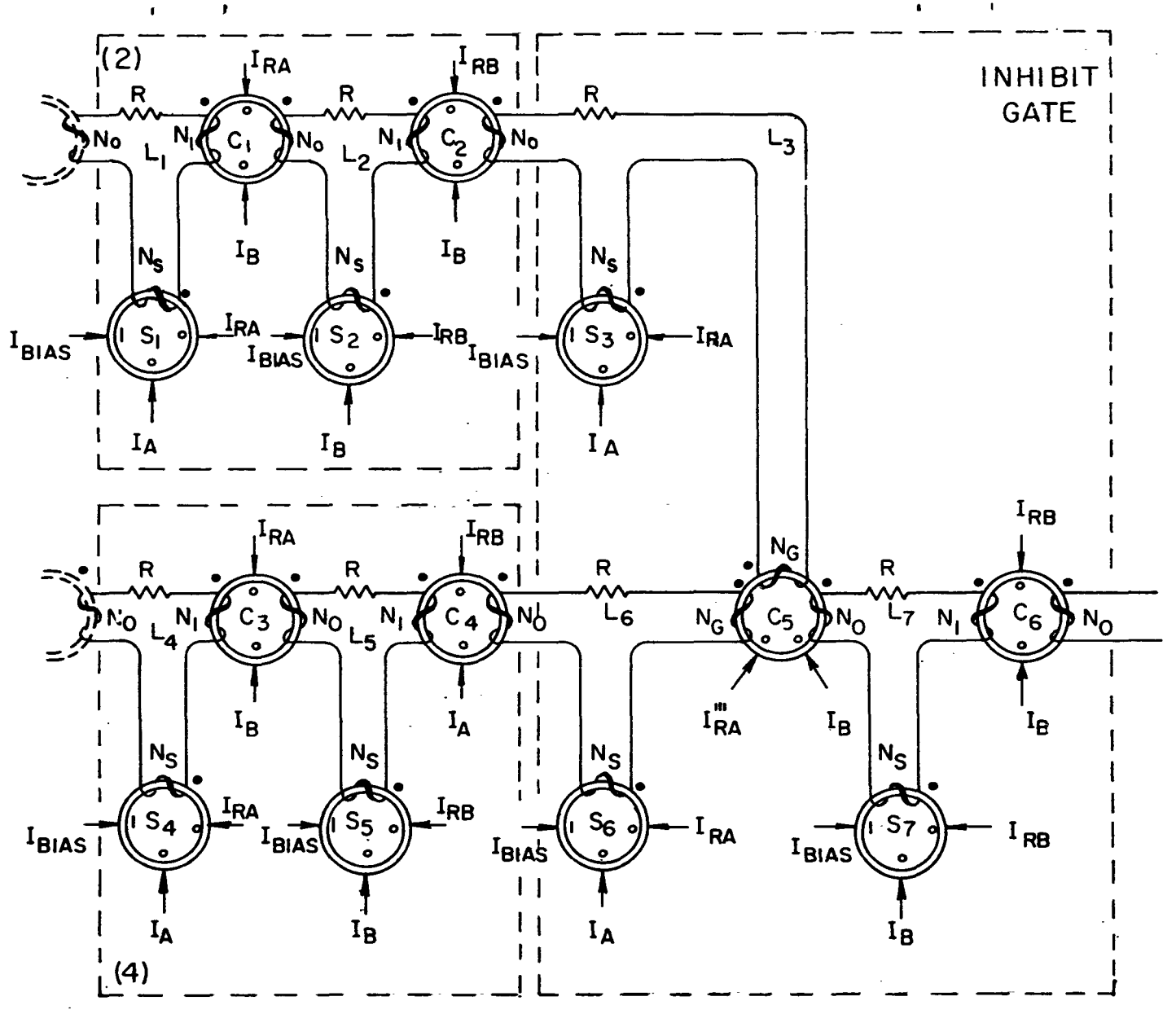


Figure 9. Inhibit schematic diagram.

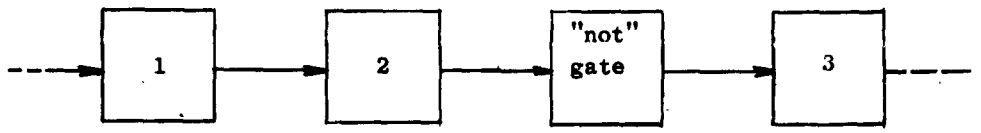


Figure 10. Logical "not" gate diagram.

the I_A, I_{RA}, and I_B pulses, they are still in the zero state. During the I_{RB} pulse, core S_4 is driven toward the one state producing a clockwise current in loop L_4 that does not affect core C_3. During the next I_A pulse, the one in S_4 is driven out into cores C_3 and S_5 in the same manner as described under the inhibit function. Succeeding pulses merely transfer the one down the line in the usual manner. Now consider the case where a one exists in core S_1 and all other cores are in the zero state. During one clock sequence, the one in S_1 is transferred to core S_3 and another one is injected into S_4. On the following I_A pulse, the resulting currents in loops L_3 and L_4 tend to drive C_3 in opposite directions. Prcvided the loops are properly balanced for inhibiting, C_3 is not switched and thus the ones are driven out of S_3 and S_4 and are lost. Thus since no output occurs when a one is injected into the gate, and also since an output occurs when the gate does not receive any input, the operation of the "not" gate is complete.

3. DESIGN CRITERIA

To build all-magnetic logic systems it is necessary to have some sort of design criteria to determine the number of turns, loop resistance, and driving currents for all the building blocks described in section 2. There are several methods that can be used to solve magnetic core circuits but, for circuits that consist entirely of magnetic cores, one method is particularly useful. This method involves the representing of a core during switching by an equivalent circuit. Thus in a circuit consisting of multiple cores, each core can be replaced by its equivalent circuit and the resulting circuit can be analyzed by standard circuit analysis techniques. The remainder of this section will be devoted to the detailed description of this technique and its subsequent application to the circuits described in section 2.

Typical switching characteristics of a simple toroidal core are shown in figures 12 and 13. Figure 12 is a plot of the peak output voltage per turn versus the driving amp-turns, and figure 13 is a plot of the reciprocal of the switching time versus the driving amp-turns. The data for these curves were obtained in the manner illustrated in figure 14. In this setup a positive current pulse is applied to the core under test producing an output pulse similar to that shown on the oscilloscope screen. By noting the amplitude of the current drive pulse, the peak output voltage and the switching time of the output

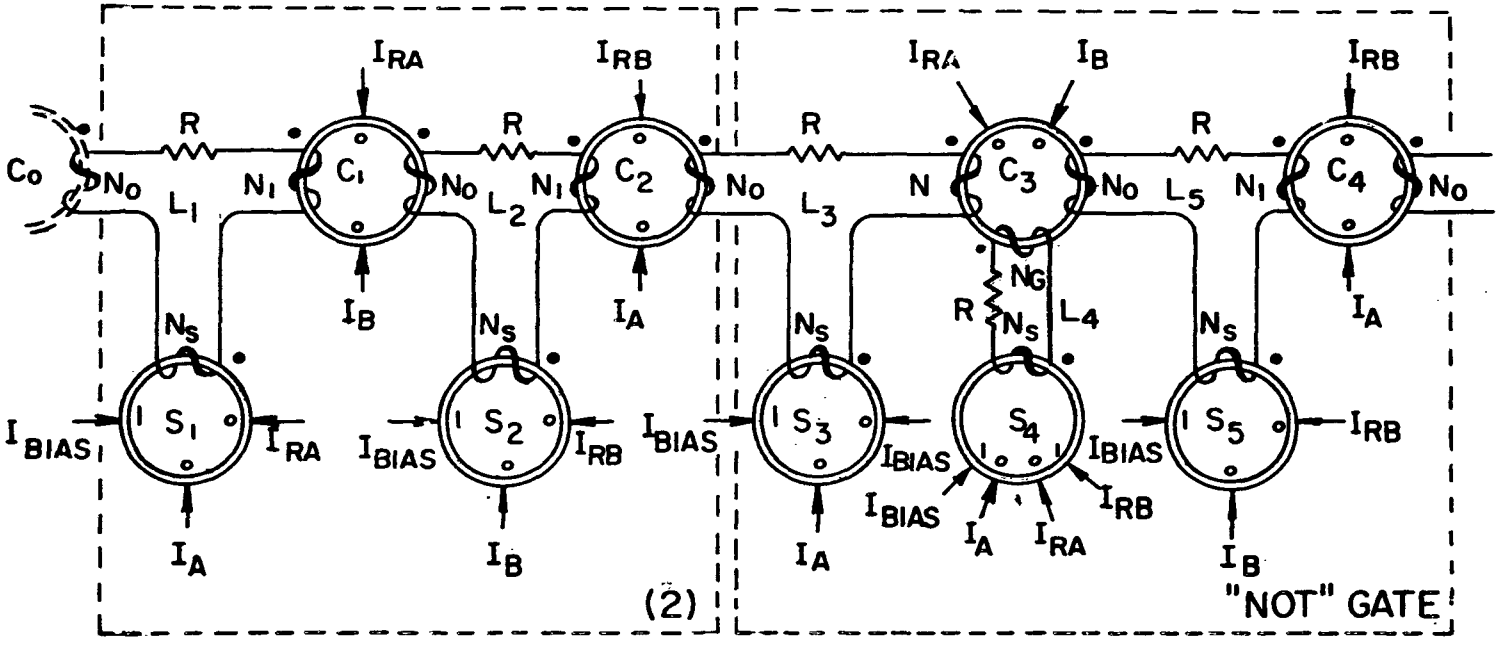


Figure 11. "Not" gate circuit.

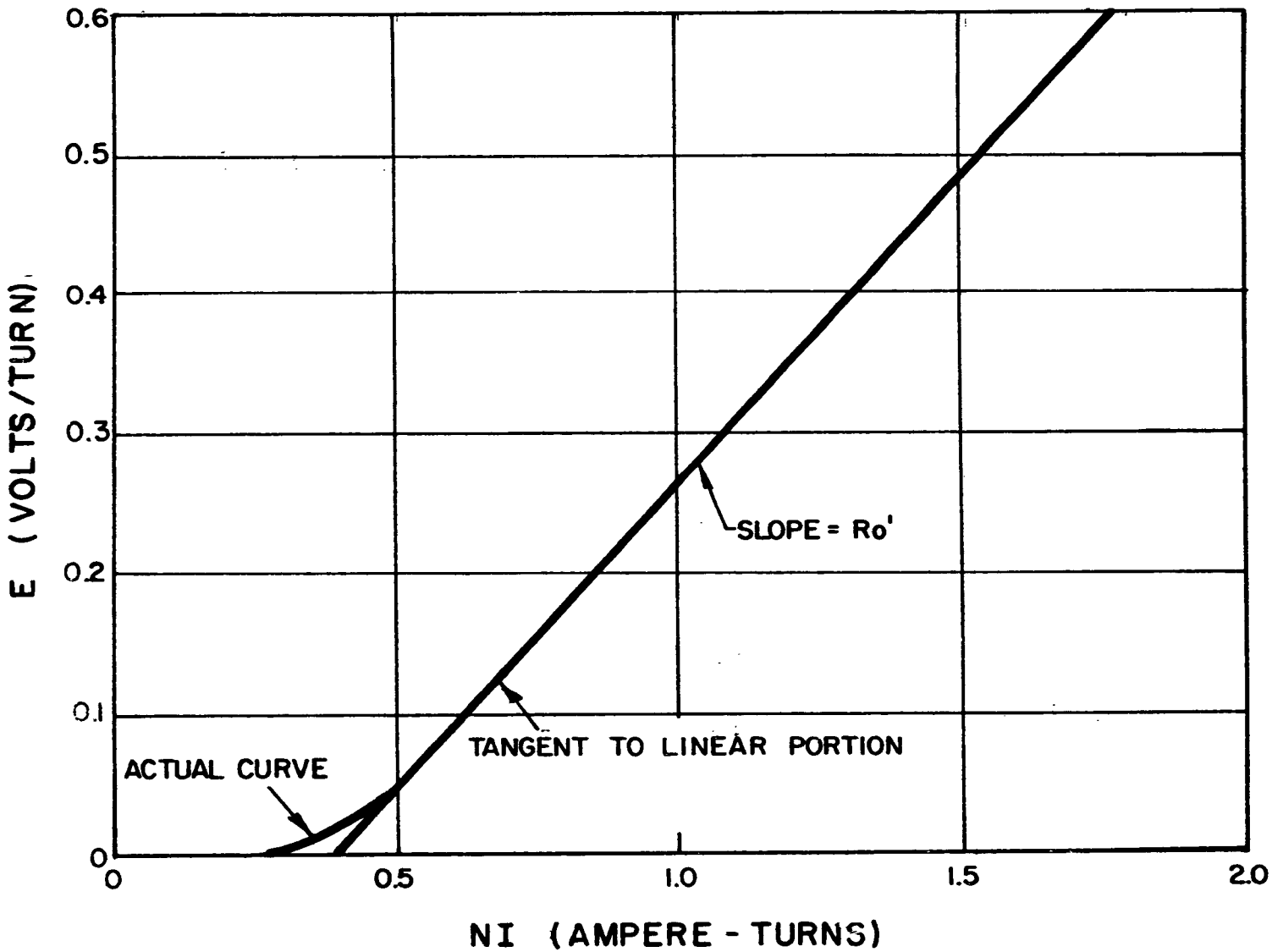


Figure 12. Output voltage versus ampere turns for General Ceramics F-426-S4 cores.

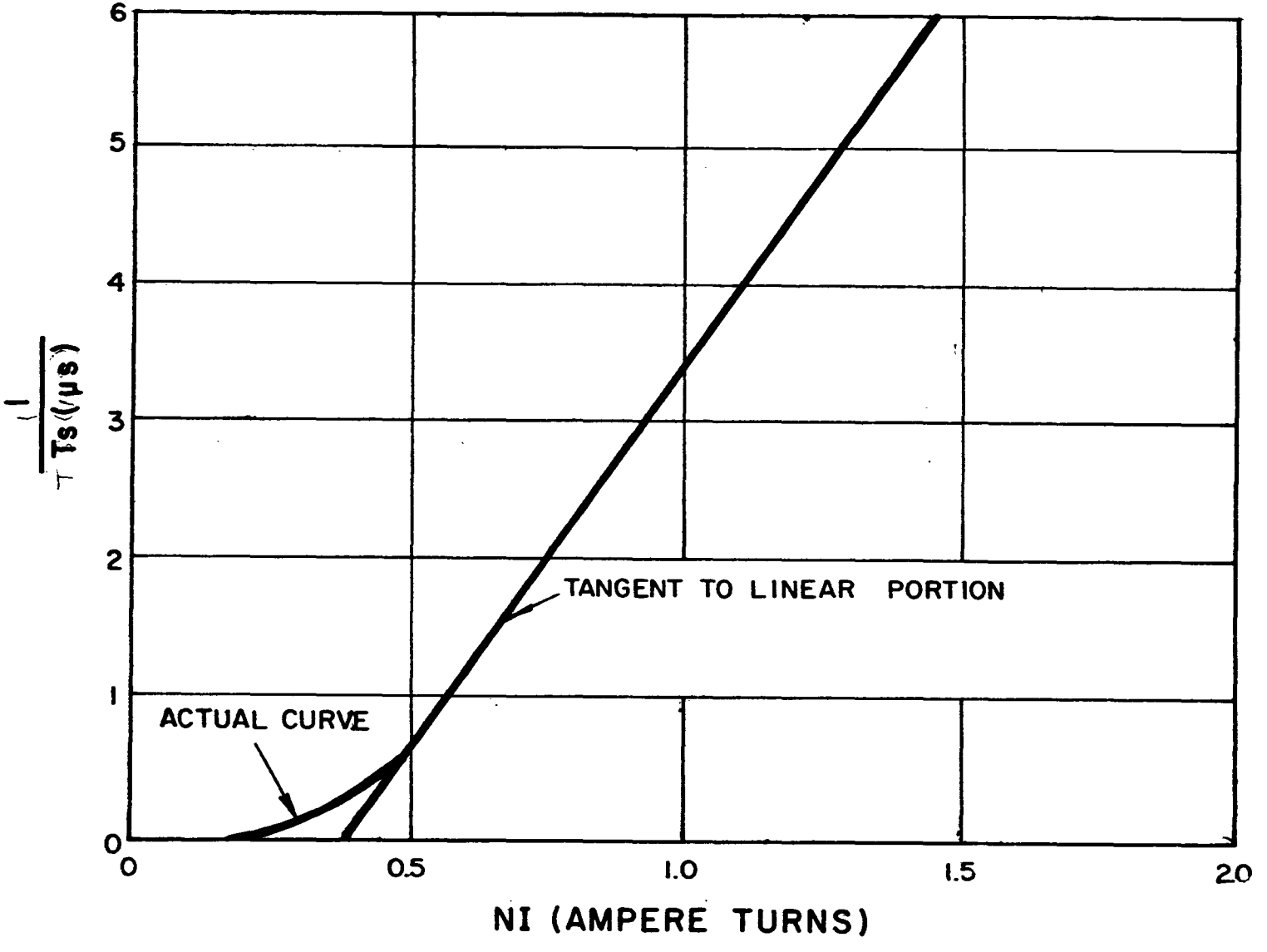


Figure 13. Reciprocal of switching time versus ampere turns for General Ceramics F-426-S4 cores.

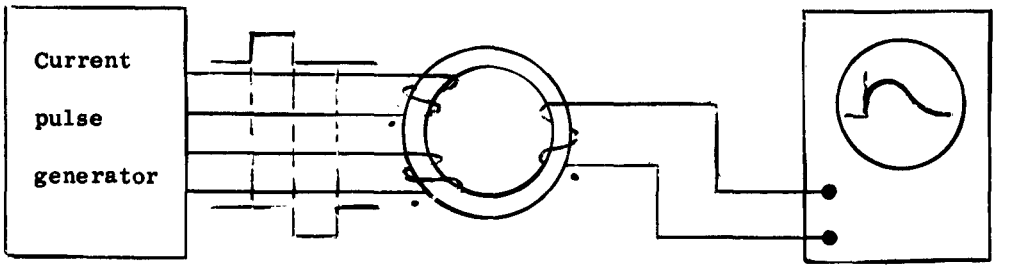


Figure 14. Core testing apparatus.

pulse (defined as the time interval between the 10 percent points of the output pulse), the data necessary for the plotting of the graphs can be obtained. Note that both curves are linear throughout most of the graph and that the tangent to the linear portion of both graphs intersects the abscissa at the common point, hereafter to be denoted as NI min. NI min is defined as the minimum value of ampere-turns that will completely switch the core. It corresponds to point G on the hysteresis loop of the core (fig. 5).

With the use of figure 12 an equivalent circuit for a single toroidal magnetic core during switching can be determined. Since the slope of the graph has the dimensions of resistance per turn squared, which will be denoted R_o^1, a core can be represented by a resistance, a current generator, and an ideal transformer in the following manner.

Note that the slope of the graph in figure 12 is R_o^1 and the resistance in the equivalent circuit is denoted as R_o. The difference between R_o^1 and R_o is attributed to the basic assumption that all the pulses in a magnetic circuit have a square waveshape instead of the actual quasi-semisinusoidal shape (fig. 14). This seems like a rather drastic assumption but several workers in the field have performed complicated analyses of core circuits with semisinusoidal pulses, and they have come to the conclusion that the procedure does not yield more accurate results (ref 2). The assumption of a square pulse, which accounts for the difference between R_o and R_o^1, greatly simplifies the analysis. The value of R_o listed by various researchers runs from about 0.6 R_o^1 to 0.7 R_o^1 (ref 2, 6). The value of 0.6 R_o^1 will be used in this report. The remainder of this section will be devoted to the analysis of the building blocks of section 1. All the analysis is merely some type of combination of the equivalent core circuits of figure 15.

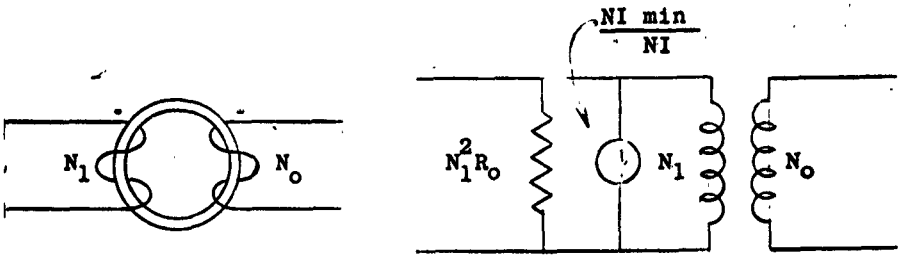


Figure 15. Equivalent core circuit.

3.1 Basic Shift Register

Stage 2 of the basic shift register circuit shown in figure 4 can be represented, during the switching of a one from core S_1 to C_1 and S_2, by the circuit shown in figure 16.

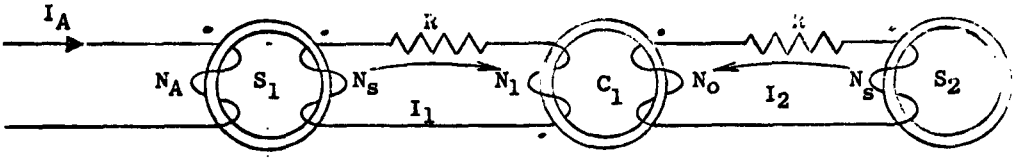


Figure 16. Circuit showing a one going from S_1 to S_2.

This representation is possible since cores C_0 and C_2 (fig. 4) merely act as shorts and therefore do not affect cores S_1, C_1, or S_2 during switching. The equivalent circuit for these cores is shown in figure 17.

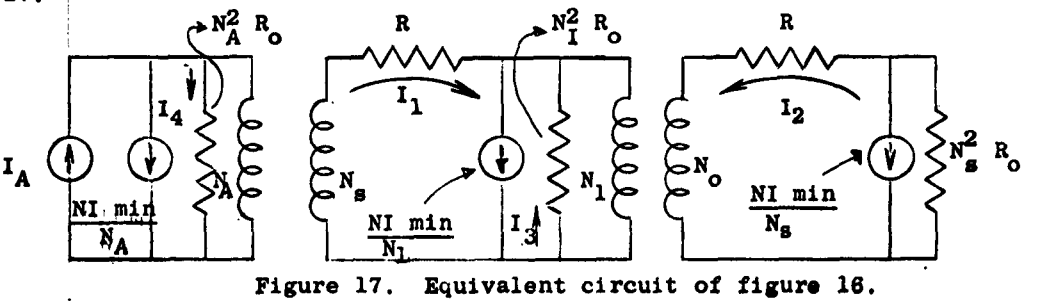
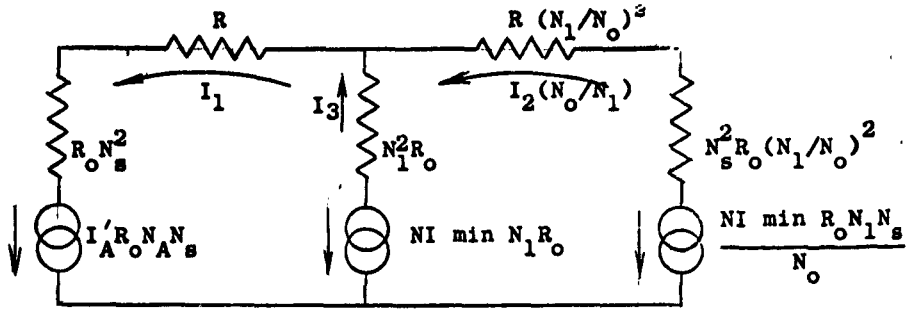


Figure 17. Equivalent circuit of figure 16.

By simple circuit analysis techniques this circuit can be reduced to the equivalent circuit shown in figure 18. With this equivalent circuit we are now in a position to derive some of the design equations. One of the most important considerations in the design of magnetic circuits is to insure that a driving core does not switch before a receiving core. This is necessary because the receiving core must be completely switched so that it can transmit information to the following stage without attenuation or loss of flux down the register. For example, in stage 2 of figure 4, core S_2 must be switched slightly before core C_1, and C_1 must be switched slightly before S_1. Let F_1



where $I'_A = I_A - NI\ min/N_A$ (1)

Figure 18. Equivalent circuit of figure 17.

be defined as the fraction of flux in core C_1 switched during the switching time of core S_2 (T_{s2}). Thus,

$$F_1 = \left.\frac{\Delta \phi_{c1}}{\Delta \phi_{s2}}\right|_{T_{s2}} \tag{2}$$

Since the voltage waveshape was assumed to be square,

$$V = N \frac{d\phi}{dt} = \text{constant} \tag{3}$$

Therefore,

$$V_{s2}\ T_{s2} = N_s\ \Delta \phi_{s2} \tag{4}$$

$$V_{c1}\ T_{s2} = N_1\ \Delta \phi_{c1}\ \Big|_{T_{s2}} \tag{5}$$

Substituting (4) and (5) into (2) yields,

$$F_1 = \frac{V_{c1}\ N_s}{V_{s2}\ N_1} \tag{6}$$

From the equivalent circuit (neglecting NI min),

$$V_{c1} = I_3\ N_1^2\ R_o \tag{7}$$

$$V_{s2} = I_2\ N_s^2\ R_o \tag{8}$$

Substituting (7) and (8) into (6) gives,

$$F_1 = \frac{N_1\ I_3}{N_s\ I_2} \tag{9}$$

Now using the equivalent circuit we can solve for the ratio in (9). This turns out to be,

$$F_1 = \frac{R + N_s^2 R_o}{N_o N_s R_o} \tag{10}$$

To insure that core C_1 switches before S_1, then

$$F_2 = \frac{\Delta \phi_{s1}}{\Delta \phi_{c1}} \tag{11}$$

must also be somewhat less than unity. Using the same procedure used in arriving at (9),

$$F_2 = \frac{N_A I_4}{N_1 I_3} \tag{12}$$

Solving the equivalent circuit for this ratio gives,

$$F_2 = \frac{N_1}{N_s} + \frac{R R_o (N_o^2 + N_s^2) + R^2}{N_1 N_s R_o (R + R_o N_s^2)} \tag{13}$$

Since core S_2 switches before core C_1 and since C_1 switches before core S_1, then this automatically assures that core S_2 switches before S_1. This can easily be observed from the following:

Let
$$F_3 = \frac{\Delta \phi_{s1}}{\Delta \phi_{s2}} \tag{14}$$

and as before,

$$F_3 = \frac{I_4 N_A}{I_2 N_s} \tag{15}$$

From (9) and (12),

$$F_3 = F_1 F_2 \tag{16}$$

Since F_1 and F_2 are both less than one, then F_3 is less than one.

Now that the proper switching of the cores has been established, a suitable value of drive current must be selected. This is done by picking a value of I_2 that gives a desirable switching time for core S_2 and then determining the proper drive current by solving for $N_A I_A$ in terms of I_2 in the equivalent circuit. The result is:

$$N_A I_A = \frac{I_2(N_1/N_o)(R + N_s^2 R_o)[R + R_o(N_s^2 + N_1 + N_o^2)]}{R_o^3 N_1^2 N_s} \qquad (17)$$

+ NI min terms

This equation may seem to be somewhat complicated, but since the second term usually amounts to about 10 percent of the first term, it may be omitted if desired.

For some applications it may be valuable to know the input impedance of the circuit. This can be determined from the equivalent circuit by transferring all impedances to the input terminals. The result is

$$\frac{Z}{N_A^2} = \frac{R_o^2 R(N_1^2 + N_s^2 + N_o^2) + R_o(R^2 + N_1^2 N_s^2 R_o^2)}{R_o(R + R_o N_s^2)(N_1^2 + N_s^2 + N_o^2) + R(R + N_s^2 R_o)} \qquad (18)$$

During the resetting operation I_{RA} (or I_{RB}) drives the appropriate coupling core back to the zero state. When this process occurs it is necessary to insure that the storage cores adjacent to the coupling cores do not switch (fig. 4). Here they behave as shorts so that the resultant circuit during reset is shown in figure 19.

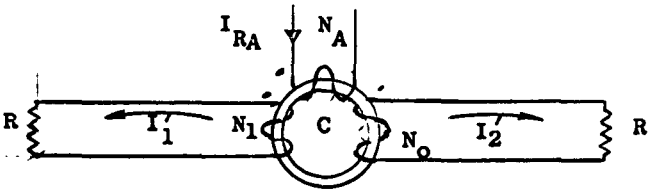


Figure 19. Circuit for C core during reset.

The equivalent circuit for this diagram is shown in figure 20.

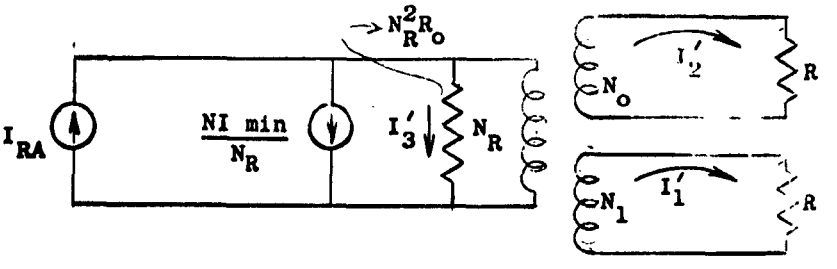


Figure 20. Equivalent circuit for figure 19.

Since I_2' is the current that must be amplitude limited, the reset drive current in terms of I_2' can be determined from the equivalent circuit. The result is

$$I_{RA} N_R = \frac{I_2' [R + R_o(N_1^2 + N_o^2)]}{N_1 R_o} + NI \text{ min} \qquad (19)$$

Choosing a proper value for I_2' and inserting it into (19) results in a particular value for the reset ampere turns. In the operation of the actual circuit, it is often found that the value of reset ampere-turns differs markedly from that predicted by (19). This can easily be explained from the fact that during reset the coupling core is being switched very slowly with an $I_{RA} N_R$ on the order of NI min. From figure 12 it can be seen that in the vicinity of NI min the actual switching curve departs considerably from its linear tangent in this region. This means that R_o is greatly in error and also that NI min is not exactly the smallest value of mmf that will completely switch the core. Thus NI min is only a first approximation to this value. This means that the linear equivalent circuit approach tends to break down in this region. For this reason, instead of using (19) to find $I_{RA}N_R$, it is just as accurate, and much simpler, to merely select a drive mmf roughly equal to NI min. Slight adjustments of this value are usually necessary in order to operate an actual circuit.

Knowledge of all the design equations for a basic shift register circuit enables a definite design procedure to be set up. The steps in the design of such a circuit are as follows:

(1) Pick the core to be used in the shift register and determine R_o and NI min from curves similar to those of figures 12 and 13.

(2) Pick N_o and N_s, choose F_1 to be slightly less than one, say .95, and solve (10) for R.

(3) Pick N_1 and solve (13) for F_2. From experience F_2 should be about 0.88 to 0.98. If (13) does not give such a value, then another value of N_1 must be tried. If this does not work, then new values of N_o and N_s must be chosen and step 2 must be repeated. Juggling of (10) and (13) usually results in appropriate values without too much trouble.

(4) Select the desired switching time of the receiving cores and from figure 13 determine the ampere-turns necessary to switch a core at that rate. Since I_2 is the chosen value of ampere-turns divided by N_s, substitute I_2 in (17) to determine the driving ampere-turns for the circuit. Selection of a suitable value of N_A (eq 17), perhaps dictated by the desired input impedance (eq 18), gives the proper value of drive current. The width of the drive current pulse must be equal to or greater than the switching time of the receiving cores.

(5) Choose the desired value of I_2' during reset, which would be about equal to 1/3 to 1/2 of NI min/N_s, and substitute it into (19).

This gives the proper value of reset ampere-turns. As mentioned previously it is nearly as accurate to choose $I_{RA}N_R$ equal to NI min with the intention of making slight adjustments in the operating circuit. The width of the reset pulse must be long enough to insure complete resetting.

(6) Choose the bias ampere-turns which, for best operation, should lie somewhere between 1/3 and 1/2 of NI min.

If these steps are followed, a circuit can be built that can be expected to operate with only slight adjustments of drive amplitude, reset amplitude, or bias current.

3.2 Branching Circuit

The design of a branching circuit (fig. 2) uses many of the design equations developed for the basic shift register. For example, the shifting of a one out of core S_1 and into core C_1 and S_2 along with the subsequent resetting of core C_1 occurs in exactly the same manner as for the basic shift register. For this reason that portion of the circuit will not be considered any further.

The driving of a one from core S_2 to cores C_2, S_3, and S_5 does occur in a different manner. The cores involved in such a transfer can be redrawn as shown in figure 21.

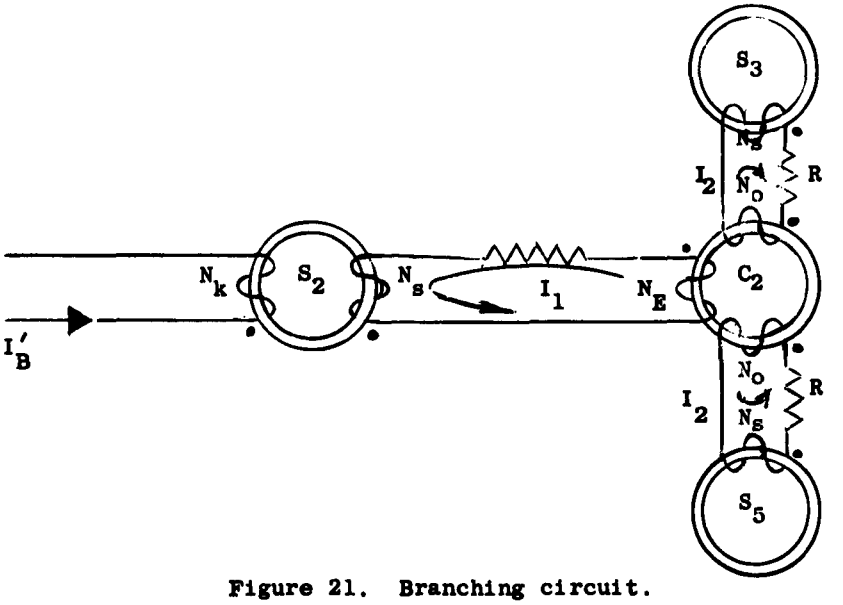


Figure 21. Branching circuit.

The equivalent circuit for these cores is shown in figure 22.

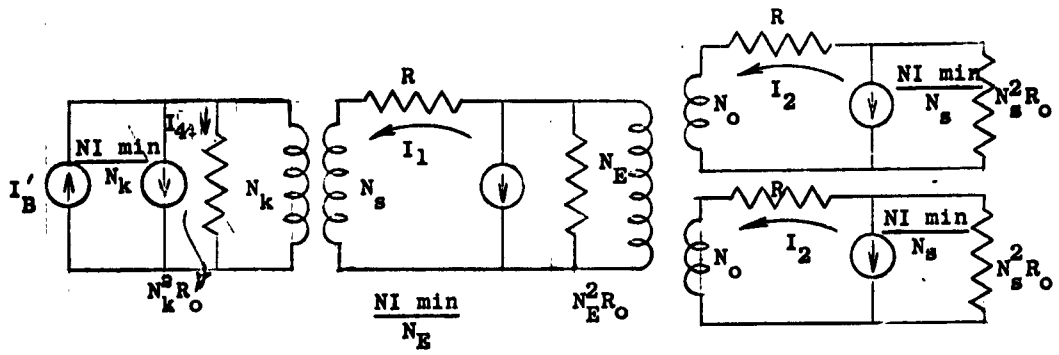
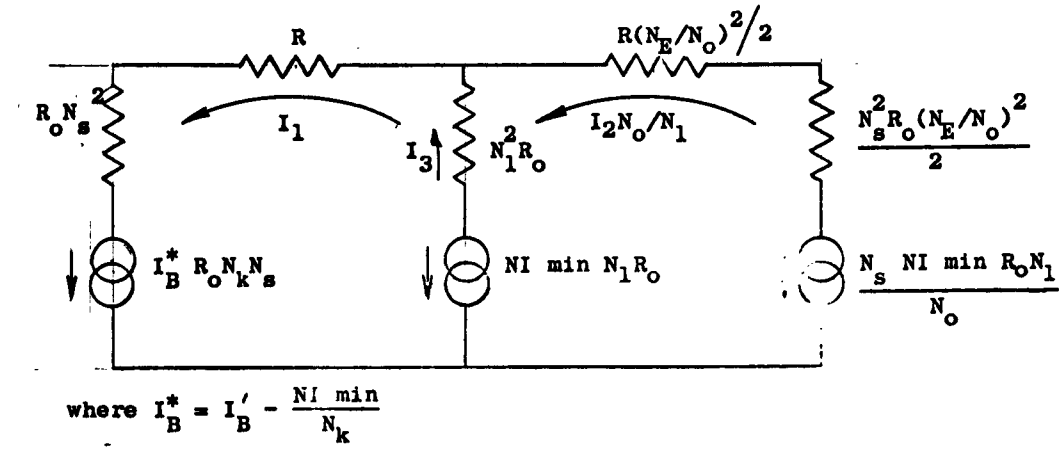


Figure 22. Equivalent circuit of branching circuit.

Using simple circuit analysis techniques this circuit can be reduced to the following equivalent circuit shown in figure 23.



where $I_B^* = I_B' - \frac{NI\ min}{N_k}$

Figure 23. Equivalent circuit of figure 22.

To insure correct operation of this circuit, cores S_3 and S_5 must shift before core C_2, and C_2 must shift before S_2. If, in the same manner as previously used, one solves F_{c2-s3} and F_{c2-s5} one would find them to be identical and equal to $F_{1/2}$(eq 10). Thus the added circuitry only changes F_{s2-c2}. Let

$$F_4 = F_{s2-c2} = \frac{\Delta \phi_{s2}}{\Delta \phi_{c2}}\bigg|_{\Delta t_{c2}} = \frac{I_4\ N_K}{I_3\ N_E} \qquad (20)$$

Solving the equivalent circuit for this ratio (neglecting NI min) yields,

$$F_4 = \frac{R\,R_o\,(2N_o^2 + N_s^2) + R^2}{N_E\,N_s\,R_o\,(R + R_o\,N_s^2)} + \frac{N_E}{N_s} \tag{21}$$

Sometimes in the evaluation of F_4 it is difficult to pick an N_E that will result in an F_4 below unity. To facilitate the choosing of F_4, let

$$A = \frac{R\,R_o\,(2N_o^2 + N_s^2) + R^2}{R_o\,(R + N_s^2\,R_o)} \tag{22}$$

Normally in the evaluation of F_4 all parameters involved in A are known. Thus to solve for the minimum value of F_4,

$$F_4 = \frac{N_E}{N_s} + \frac{A}{N_E\,N_s} \tag{23}$$

$$\frac{dF_4}{dN_E} = \frac{1}{N_s} - \frac{A}{N_E^2\,N_s} \tag{24}$$

Thus from (24), $N_E = \sqrt{A}$. This is the value that will make (21) a minimum. This fact simplifies the choosing of N_E.

Solving the last equivalent circuit for $I_B\,N_K$ in terms of I_2 yields,

$$I'_B{}_K = \frac{I_2\,(N_E/N_o)(R + N_s^2\,R_o)[R + R_o(N_s^2 + N_E^2 + \quad^2)]}{2R_o^2\,N_E^2\,N_s} \tag{25}$$

+ NI min terms

Since core C_2 has three loops connected to it, the value of reset mmf must be changed. The part of this circuit involved in this resetting operation is shown in figure 24.

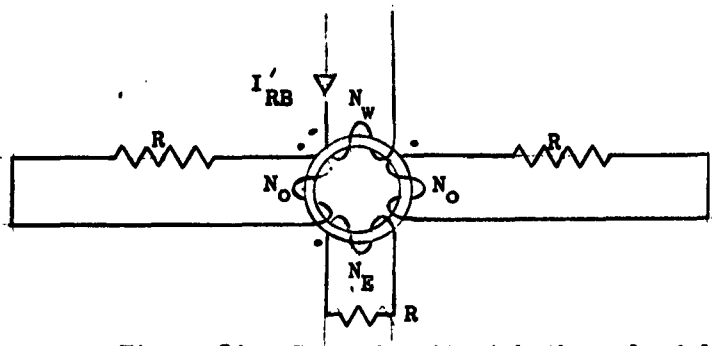


Figure 24. Core circuit with three load loops.

The equivalent circuit for this diagram is shown in figure 25.

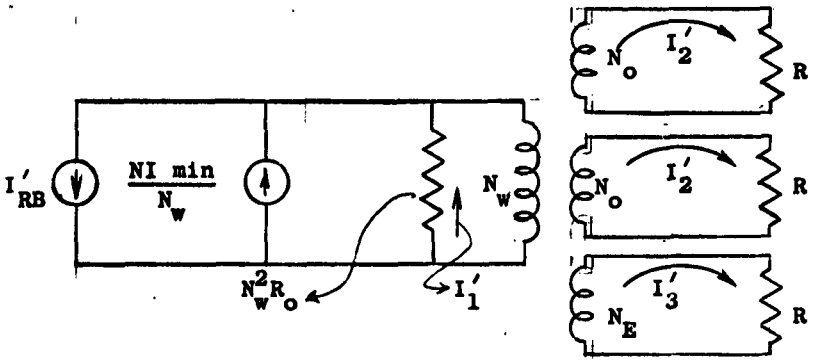


Figure 25. Equivalent circuit for figure 25.

Solving this circuit for I_{RB} in terms of I'_2 gives the following result

$$I'_{RB} N_w = \frac{I'_2 [R + R_o(N_E^2 + 2N_o^2)]}{N_o R_o} + \text{NI min terms} \qquad (26)$$

As with the basic shift register reset operation, this equation (for reasons mentioned before) might be considerably in error. If it is, then the error is in the same direction as that for the basic shift register, and thus it at least tells how much larger the mmf must be for branching reset than for basic reset. For this reason it is still a valuable equation.

With the above equations a shift register with a branching circuit can now be designed. The steps for this procedure are as follows:

(1) First satisfy all the design criteria for the basic shift register.

(2) Completion of step 1 gives all the values necessary to evaluate A (eq 22). After evaluating A find N_1 from $N_1 = \sqrt{A}$ and then substitute it into (21). If F_4 does not turn out to be less than one, then step 1 must be repeated until it gives a value for A that will satisfy (21).

(3) Pick the desired switching time of the receiving cores and from a curve of the type shown in figure 13 determine the mmf necessary to achieve this result. From this mmf and N_s determine the value for I_2 and use it in (25) to obtain $I'_R N_K$.

(4) Choose the desired value of I'_2, which should be about 1/3 to 1/2 of NI min/N_s, and substitute it in (26) to determine $I'_{RB} N_w$. After this is completed, adjust N_w with respect to N_R to make up for the larger mmf of $I_{RB} N_w$.

(5) Use the same bias current established for the basic shift register circuit.

Completing these steps should result in a branching circuit that operates with only a slight modification of the driving currents.

3.3 "And" Gate

The additional equations necessary to design an "and" gate are relatively simple. Since the shifting of ones (fig. 4) out of cores S_2 or S_4 into core S_5 or S_6 is still like that of the basic shift register, they will not be considered here. The shifting of ones out of cores S_5 or S_6 into C_6 involves another technique and therefore will be considered. The shifting of cores S_5 and S_6 into C_6 involves the following part of the circuit as shown in figure 26.

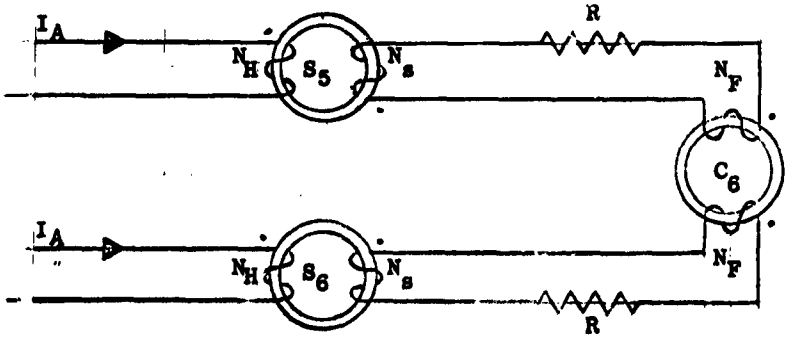


Figure 26. Circuit for shifting ones from cores S_5 and S_6 into core C_6.

The equivalent circuit for this diagram is shown in figure 27.

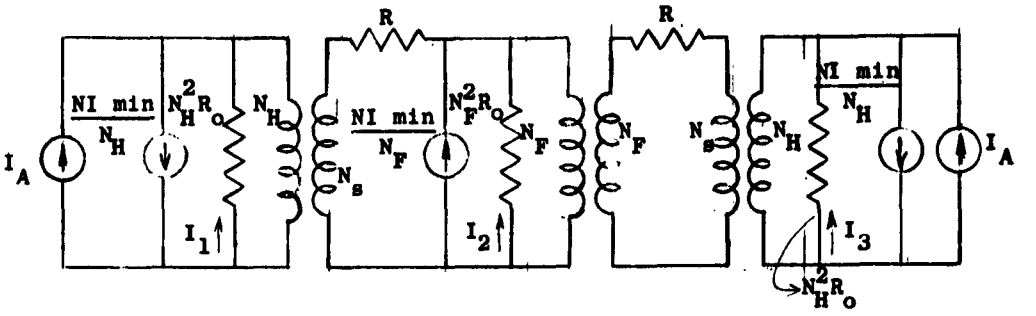


Figure 27. Equivalent circuit for figure 26.

Let

$$F_5 = \frac{\Delta \phi_{s5}}{\Delta \phi_{c6}} = \frac{\Delta \phi_{s6}}{\Delta \phi_{c6}} = \frac{N_H I_1}{N_F I_2} = \frac{I_3 N_H}{I_2 N_F} \tag{27}$$

The solution of the equivalent circuit for F_5 yields,

$$F_5 = \frac{R + 2N_F^2 R_o}{2R_o N_F N_s} \tag{28}$$

The solution for $I_A N_H$ yields

$$I_A N_H = I_2 \left[\frac{2N_F^2 R_o + R + R_o N_s^2}{2R_o N_s} \right] \tag{29}$$

Use of the above equations allows for the complete design of an "and" gate section. The following steps should be used in the design of an "and" gate.

(1) Complete all the steps described for the basic shift register circuit.

(2) Pick an N_F that satisfies (28), i.e., $0.88 < F_5 < 0.98$.

(3) Choose an I_2 about equal to or slightly less than NI min/N_F and solve (29) for $I_A N_H$.

(4) Choose the number of turns on the I_{RA} winding (N_z) to be any value that will provide an mmf larger than NI min. Note that this mmf cannot be too large since the resetting of C_6 cannot affect any of the other cores.

The completion of these steps should result in a practical "and" gate logic block.

3.4 "Or" Gate

The design equations for an "or" gate will not be considered here because they are exactly the same as those used in the basic shift register circuit. Inspection of figure 7 should clarify this point.

3.5 "Inhibit" Function

The design of an "inhibit" function circuit differs considerably from those previously considered. For example, during the switching of a one out of core S_6 into C_5 and S_7, core C_5 has a severe load on it (fig. 9). The cores involved in this transfer consist of the circuit as shown in figure 28.

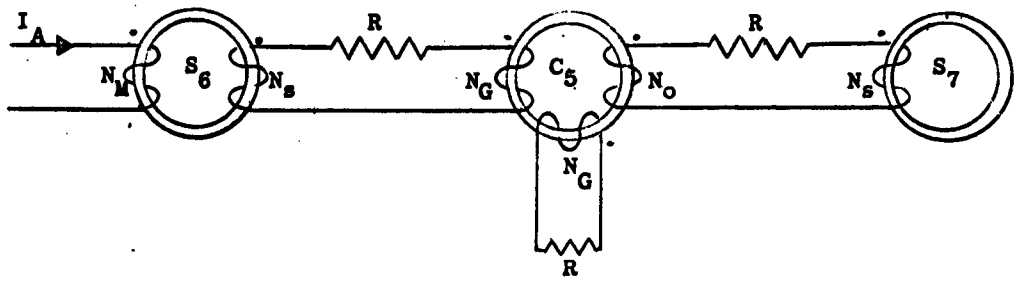


Figure 28. Transfer circuit for inhibit function.

The equivalent circuit for this diagram is shown in figure 29.

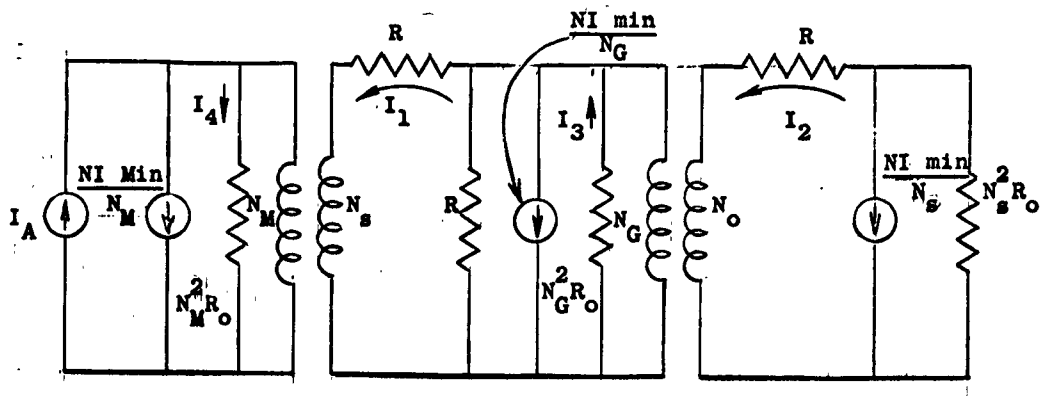


Figure 29. Equivalent circuit for figure 28.

With the use of simple circuit analysis techniques this circuit can be reduced to the circuit shown in figure 30.

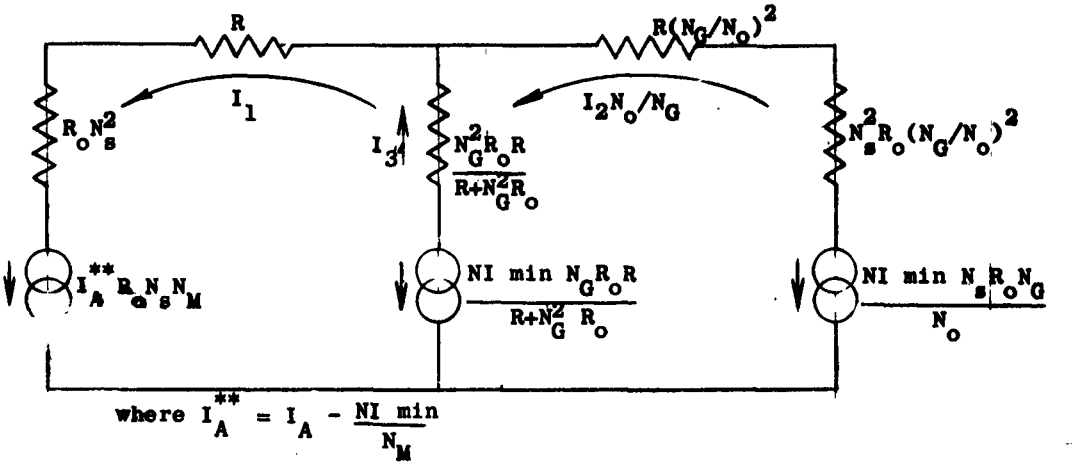


Figure 30. Equivalent circuit for figure 29.

The flux transfer ratio for cores C_5 and S_7 is identical with that of cores C_1 and S_2 of the basic shift register circuit (fig. 4, stage 2) and thus is the same as eq (10). Let F_6 be defined as the flux transfer ratio for cores S_6 and C_5. Thus neglecting NI min,

$$F_6 = \frac{\Delta\phi_{s6}}{\Delta\phi_{c5}} = \frac{I_4\ N_M}{I_3\ N_G} \tag{30}$$

Solving the equivalent circuit for this ratio (neglecting NI min) yields,

$$\left(\frac{R}{R+R_G^2 R_o}\right) F_6 = \frac{R^2(R + R_o N_o^2) + R_o R^2(N_G^2 + N_s^2) + R_o^2 R N_G^2 N_s^2}{N_G N_s R_o (R + R_o N_s^2)(R + R_o N_G^2)} + \frac{1}{N_s} \frac{R N_G}{(R + R_o N_G)} \tag{31}$$

Solving the same equivalent circuit for $I_A N_M$ in terms of I_2 yields,

$$I_A N_M = I_2 \frac{N_G}{N_o} \left[\frac{(R+R_o N_s^2)\left[(N_o^2+N_G^2) R_o R + (R+N_G^2 R_o)(R+R_o N_s^2)\right]}{R_o^2 N_G^2 N_s R} \right]$$

$$+ \text{ NI min terms} \tag{32}$$

The part of the circuit involved during the resetting of core C_5 consists of the circuit shown in figure 31.

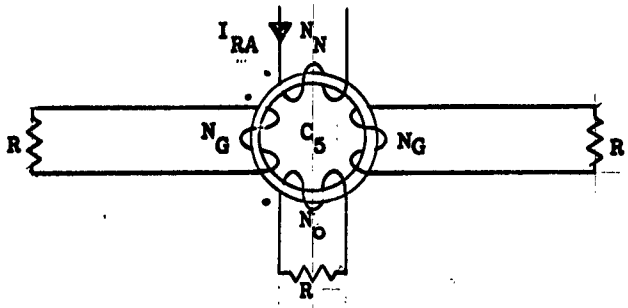


Figure 31. Circuit for resetting core C_5.

The equivalent circuit for this core during switching is shown in figure 32.

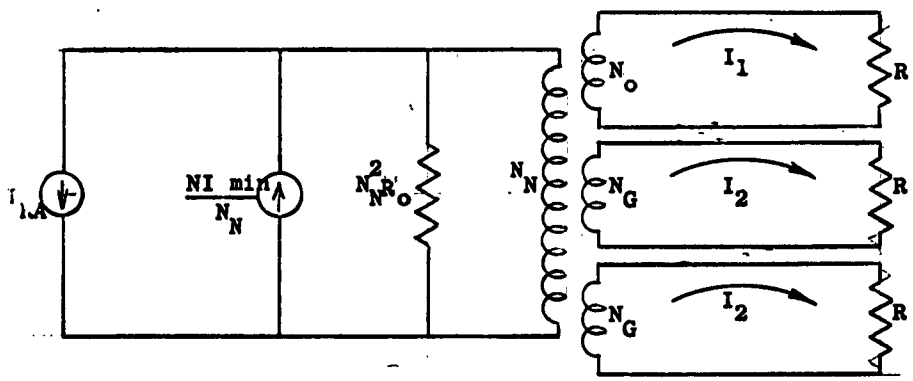


Figure 32. Equivalent circuit for figure 31.

The solution of this circuit for $I_{RA}\ N_N$ in terms of I_1 yields

$$I_{RA}\ N_N = \left(\frac{R + N_o^2 R_o + 2N_G^2 R_o}{N_o R_o} \right) I_1 + \text{NI min terms} \qquad (33)$$

The above equation combined with those of the basic shift register circuit constitutes enough information to design a circuit containing an inhibit function. The steps in the design consist of the following:

(1) Complete all the steps described for the basic shift register circuit.

(2) Pick a value of N_G that will satisfy (31), i.e., $0.88 < F_6 < 0.98$.

(3) Choose an I_2 that will give a desirable switching time for core S_7. To do this it will be necessary to refer to the characteristic curves for the core (fig. 13). Then substitute I_2 into (32) to solve for $I_A N_M$.

(4) Choose a reasonable value of I_1 during reset (about 1/3 to 1/2 of NI min/N_S) and substitute it into (33) to find $I_{RA} N_N$.

Completion of these steps should result in a circuit that operates with only a slight adjustment of parameters.

3.6 "Not" Gate

Since a "not" gate merely consists of a basic shift register circuit combined with an inhibit function, the completion of the steps listed under section 3.5 should result in a practical "not" gate circuit.

4. PRACTICAL ALL-MAGNETIC CIRCUITS

To utilize some of the techniques already developed, several magnetic systems were designed and constructed. These circuits consisted of a coincidence detector to determine the simultaneous presence of ones in two parallel shift registers and a shift code counter. These circuits were devised to obtain frequency division so that they could be used in electronic timing applications. In addition to the logical circuits a system to replace both reset pulses by a direct current is also described in the following.

4.1 Coincidence Detector

One method to obtain frequency division is to drive two shift registers of unequal length in parallel and to detect the simultaneous presence of ones by the use of an "and" gate with one input from each register. Provided the number of stages in each shift register is not divisible by the same integer, this system will give a frequency division equal to the product of the number of stages of each register. The particular system used consisted of a five-and a six-stage register connected to an "and" gate. This system is illustrated in block diagram form in figure 33 and schematically in figure 34. If ones are inserted in stages 1 and 1_a, then after 30 clock sequences the "and" gate will produce an output. Stages 1 through 5 and 1_a through 4_a are basic shift register stages, while stages 6 and 5_a are branching circuits. For

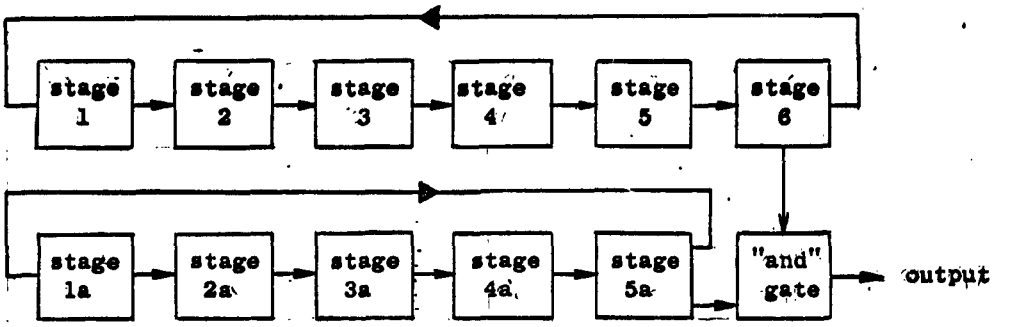


Figure 33. Logical parallel coincidence detector.

simplicity some of the details of the basic shift register stages are omitted because, as indicated on the diagram, they are identical with those shown. Table I lists the system parameters along with the design criteria of section 2. This circuit was constructed using General Ceramics F-426-S6 cores and was found to operate extremely well.

TABLE I. PARALLEL COINCIDENCE DETECTOR DESIGN CRITERIA

N_A (I_A winding)	8 turns	Core-General Ceramics
N_B (I_B winding)	8 turns	F-426-S6
N_J (I_A winding)	20 turns	R_o = .375 ohms
N_R (I_{RA} winding)	4 turns	
N_T (I_{RB} winding)	4 turns	F_1 = .945 (from eq 10)
N_V (I_{RA} winding)	6 turns	F_2 = .951 (from eq 13)
N_o	17 turns	F_3 = .898 (from eq 16)
N_s	15 turns	F_4 = .985 (from eq 21)
N_1	11 turns	F_5 = .930 (from eq 28)
N_E	5 turns	
N_F	7 turns	$I_A N_A = I_B N_B \cong 3$ at* (from eq 17)
N_H (I_B winding)	1 turn	$I_{RA} N_R = I_{RB} N_T \cong .2$ at (from eq 19)
		$I_A N_J = I_B N_k \cong 5.88$ at (from eq 25)
		$I_{RA} N_V = I_{RB} N_W \cong .3$ at (from eq 26)
		$I_B N_H = I_A N_H \cong .5$ at (from eq 29)

NOTE: All values of drive mmf are only approximate since the circuit will work over a wide range.

* ampere turns

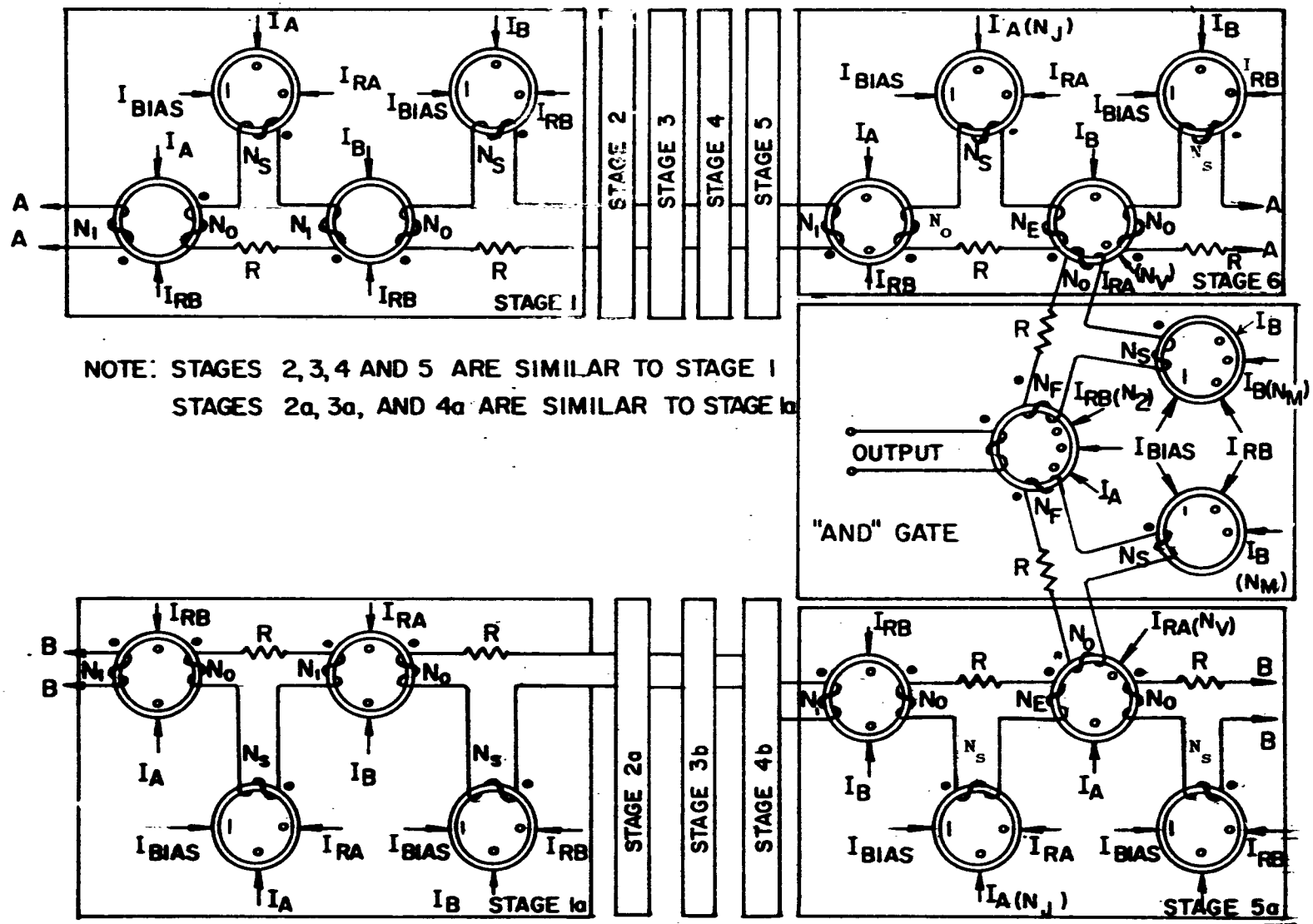


Figure 34. Parallel coincidence detector circuit.

4.2 Shift Code Counter

A shift code counter is a device that effectively utilizes all the possible states of a shift register. In other words, instead of a shift register containing just a single "one," a shift code counter contains various combinations (in fact all combinations except one) of ones and zeroes. A block diagram of a simple 3-stage shift code counter is shown in figure 35. As indicated in the figure, a shift code counter is a shift register with an "exclusive or" gate tied to two stages of the counter. The logical operation of a 3-stage shift counter is listed in table II and the truth table for an "exclusive or" gate is shown in table III. These two tables should provide enough information

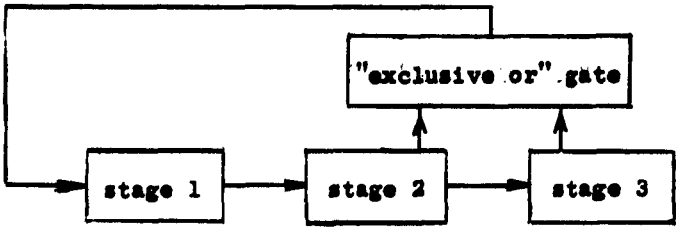


Figure 35. Three-stage shift code counter.

TABLE II. THREE-STAGE SHIFT CODE COUNTER SEQUENCE

Clock Sequence	Stage 1	Stage 2	Stage 3
0	1	0	0
1	0	1	0
2	1	0	1
3	1	1	0
4	1	1	1
5	0	1	1
6	0	0	1
7	1	0	0

TABLE III. TRUTH TABLE FOR "EXCLUSIVE OR" GATE

Stage 2	Stage 3	Output
0	0	0
0	1	1
1	0	1
1	1	0

to describe completely the operation of the shift code counter. For example, starting with a one in stage one and zeroes in all other stages, then after the first clock sequence the one will be shifted

to stage 2. From the truth table it can be seen that with a one in stage 2 and a zero in stage 3 an output occurs. Thus on the second clock sequence a one is inserted in stage 1 and the one in stage two is transferred to stage 3. Following this procedure it will be found that on the seventh clock sequence the original pattern reappears. This indicates that the shift code counter has passed through one complete cycle. This method of counting produces a cycle length of 2^n-1 counts where n is the number of stages. Thus for this counter the cycle length is 7 as verified by table II. A more detailed description of shift code counters appears in another DOFL report (ref 7).

An all magnetic "exclusive or" gate is shown logically in figure 36 and in schematic form in figure 37. Note that in figure 37 some of the details of driving pulses have been omitted for clarity.

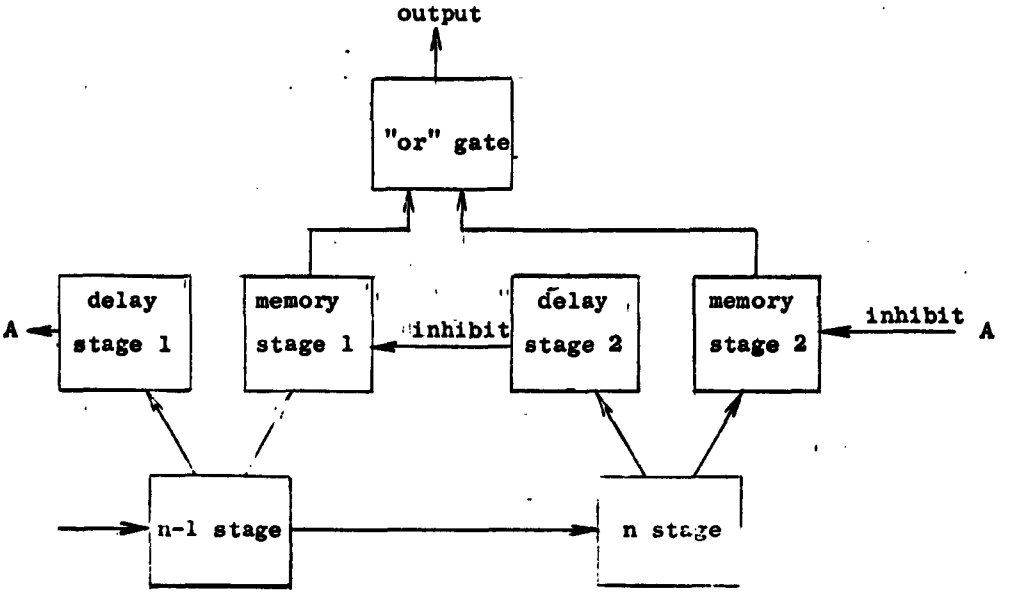


Figure 36. Logical "exclusive or" gate diagram.

These details are included in the "exclusive or" section of figure 38.. As indicated in figure 37 the "exclusive or" gate basically consists of a combination of the gates considered in section 1. To compare the operation of this circuit to the truth table for an "exclusive or" gate (table III) consider the first, almost trivial, case where zeroes exist in both stages 2 (n-1) and 3 (n). Here, no inputs merely result in no outputs. In the second case where a one exists in core S_3 of stage 3 then after the I_A and I_{RA} pulses the one has been transferred to cores S_4 and S_7. During the I_B and I_{RB} pulses the ones are transferred to cores S_{10} and S_8. During the application of the next I_A and I_{RA} pulse the one in S_{10} is driven out and lost and that in S_8 is transferred to S_{11}. The I_B and I_{RB} pulses of the second pulse sequence

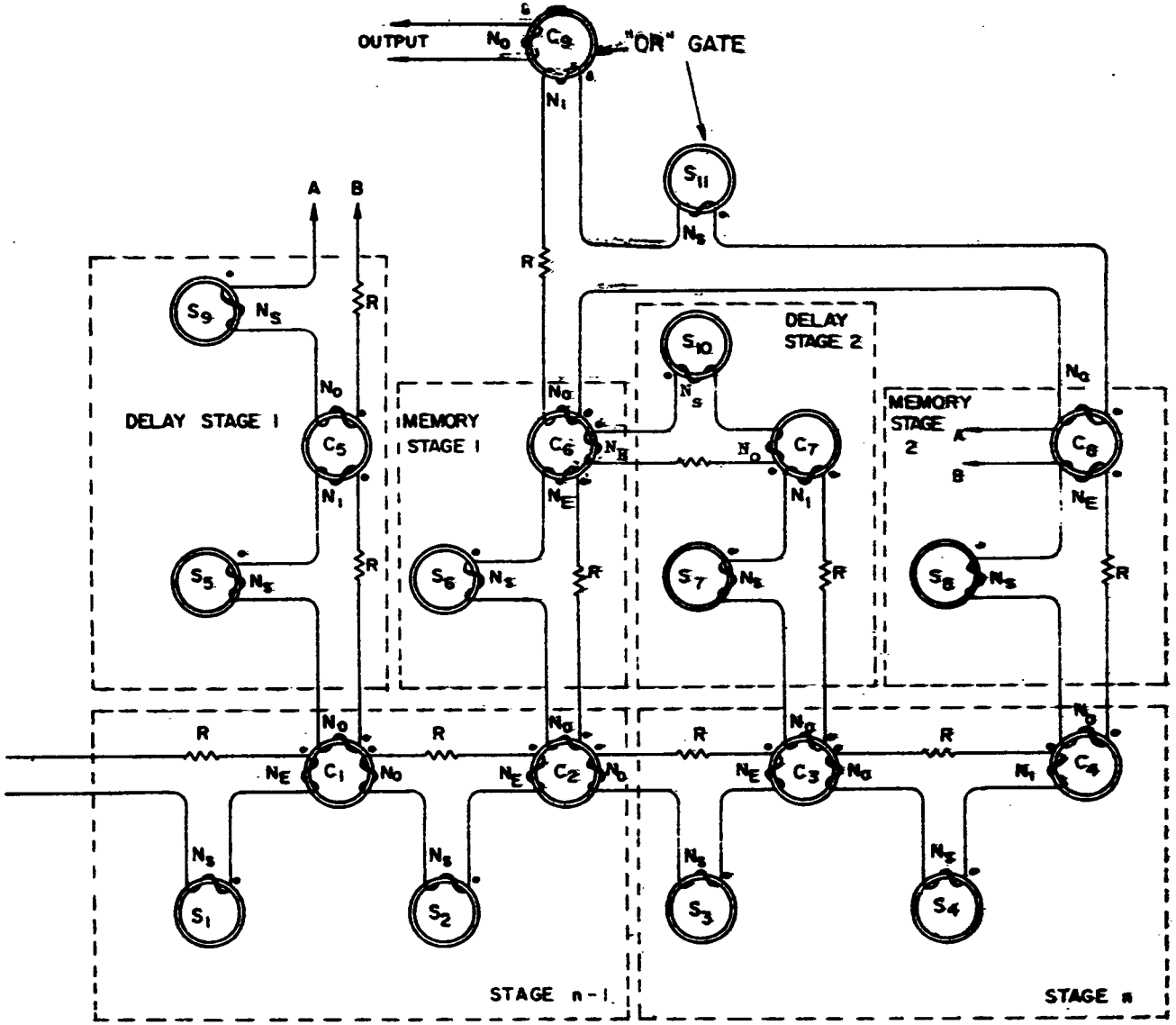


Figure 37. "Exclusive or" gate, circuit diagram.

drive the one out of S_{11} and into any desired load. Note that it required two full pulse sequences to get the information out of the shift register and through the logic into the load. This means that the logic circuitry acts as another stage because it took one full clock sequence to digest the information and deliver an output. Thus, for example, in addition to the "exclusive or" logic circuitry a six-stage shift code counter requires only five basic shift register stages. Returning to the problem of verifying the truth table (table III) consider the third case where a one exists in stage 2 (n-1) and a zero in stage 3 (n). Thus in a manner similar to case 2 the one can be followed through the logic and it will give an output after two full pulse sequences. The fourth and final case is when a one exists in both cores S_1 and S_3 [stages2 (n-1) and 3 (n)]. Here, the desired result is not to have any output. After the I_A and I_{RA} pulses the one in S_1 has been transferred to cores S_5 and S_2 and the one in S_3 to cores S_4 and S_7. After the I_B and I_{RB} pulses, the one in S_5 goes to S_9, the one in S_7 to S_{10}, the one in S_2 to S_3 and S_6, and the one in S_4 goes to S_8. During the next I_A pulse the one in S_9 "inhibits" the one in S_8 and the one in S_{10} inhibits that in S_6. Thus, except for the ones in the main shift register stages, the ones are lost and no output occurs. Since this is the desired result for the final case, the "exclusive or" gate of figure 37 operates in a manner prescribed by the truth table for such a gate.

The particular circuit constructed and operated in the laboratory consisted of a six-stage shift code counter (fig. 38) with readout circuitry. This readout circuitry is for determining when the counter has completed a full cycle. Stages 1, 2, 3, 4, and 5 are the shift register stages which, in addition to the memory stages, delay stages, and the "or" gate, constitute the basic shift code counter circuit. The readout circuitry is designed to detect the presence of all zeroes in the five shift register stages. Therefore, an output is produced when a one is contained in the logic circuitry and all other stages are in the zero state. The fact that a one exists in the logic circuitry when all the other stages contain a zero is known to be true, since the counter goes through all possible combinations of ones and zeroes except for the case of all zeroes. As can be seen from the 6-stage shift code counter sequence table (table IV) this condition occurs at 63 counts, i.e., (2^n-1) counts.

Figure 38 shows that cores S_5, S_6, S_{13}, C_5, and C_6 of stage 3 constitute a branching circuit. Thus any information that enters stage 3 follows two paths. One path goes on to stages 4 and 5 and the other goes to stages 4_a and 5_a. This means that stages 4 and 4_a and 5 and 5_a contain identical information. This is necessary because, as mentioned in the first part of the report, the maximum branching factor obtained was two. Since all the elements in stages 4 and 5 already use this maximum branching factor, no other loads can be tied to them. Since each stage must be branched out to the output circuitry, stages 4_a and 5_a were added so that they could withstand the loads. Thus the last coupling cores of stages 1, 2, 3, 4_a, and 5_a feed into the storage

TABLE IV. SIX-STAGE SHIFT CODE COUNTER SEQUENCE CHART

Count	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5	Stage 6
1	1	0	0	0	0	0
2	0	1	0	0	0	0
3	0	0	1	0	0	0
4	0	0	0	1	0	0
5	0	0	0	0	1	0
6	1	0	0	0	0	1
7	1	1	0	0	0	0
8	0	1	1	0	0	0
9	0	0	1	1	0	0
10	0	0	0	1	1	0
11	1	0	0	0	1	1
12	0	1	0	0	0	1
13	1	0	1	0	0	0
14	0	1	0	1	0	0
15	0	0	1	0	1	0
16	1	0	0	1	0	1
17	1	1	0	0	1	0
18	1	1	1	0	0	1
19	1	1	1	1	0	0
20	0	1	1	1	1	0
21	1	0	1	1	1	1
22	0	1	0	1	1	1
23	0	0	1	0	1	1
24	0	0	0	1	0	1
25	1	0	0	0	1	0
26	1	1	0	0	0	1
27	1	1	1	0	0	0
28	0	1	1	1	0	0
29	0	0	1	1	1	0
30	1	0	0	1	1	1
31	0	1	0	0	1	1
32	0	0	1	0	0	1
33	1	0	0	1	0	0
34	0	1	0	0	1	0
35	1	0	1	0	0	1
36	1	1	0	1	0	0
37	0	1	1	0	1	0
38	1	0	1	1	0	1
39	1	1	0	1	1	0
40	1	1	1	0	1	1
41	0	1	1	1	0	1
42	1	0	1	1	1	0
43	1	1	0	1	1	1

TABLE IV. SIX-STAGE SHIFT CODE COUNTER SEQUENCE CHART (Continued)

Count	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5	Stage 6
44	0	1	1	0	1	1
45	0	0	1	1	0	1
46	1	0	0	1	1	0
47	1	1	0	0	1	1
48	0	1	1	0	0	1
49	1	0	1	1	0	0
50	0	1	0	1	1	0
51	1	0	1	0	1	1
52	0	1	0	1	0	1
53	1	0	1	0	1	0
54	1	1	0	1	0	1
55	1	1	1	0	1	0
56	1	1	1	1	0	1
57	1	1	1	1	1	0
58	1	1	1	1	1	1
59	0	1	1	1	1	1
60	0	0	1	1	1	1
61	0	0	0	1	1	1
62	0	0	0	0	1	1
63	0	0	0	0	0	1
64	1	0	0	0	0	0

TABLE V. SHIFT CODE COUNTER DESIGN CRITERIA

N_A (I_A winding)	10 turns
N_B (I_B winding)	10 turns
N_J (I_A winding)	25 turns
N_K (I_B winding)	25 turns
N_R (I_{RA} winding)	4 turns
N_T (I_{RB} winding)	4 turns
N_V (I_{RA} winding)	6 turns
N_W (I_{RB} winding)	6 turns
N_M (I_A winding)	25 turns
N_P (I_B winding)	25 turns
N_N (I_{RA} winding)	6 turns
N_D (I_{RB} winding)	6 turns
N_S	8 turns
N_O	9 turns
N_1	6 turns
N_E	5 turns
N_G	5 turns

F_1 (from eq 10) = .896

F_2 (from eq 13) =1.00

F_3 (from eq 10) = .896

F_4 (from eq 21) = .80

F_6 (from eq 31) = .1

$I_A N_A = I_B N_B \cong 4.68$ at* (from eq 17)

$I_A N_J = I_B N_B \cong 15$ at (from eq 25)

$I_{RA} N_R = I_{RB} N_T \cong .6$ at (from eq 19)

$I_{RA} N_V = I_{RB} N_W \cong .6$ at (from eq 26)

$I_A N_M = I_B N_P \cong 18$ at (from eq 32)

$I_{RA} N_N = I_{RB} N_P \cong .77$ at (from eq 33)

Core: General Ceramics
S-6 F426
R_o = .36 ohms
NI min = .40 at
R = .21 ohms (loop resistance)

* ampere turns

cores of the output circuit. Logically the output circuit consists of a 5 input "or" gate that feeds a "not" gate. The cores involved in the "or" gate are those not contained in any of the dotted boxes (fig. 38). If a one exists in any of the stages 1, 2, 3, 4_a, or 5_a, then after the next I_B or I_{RB}, the one (or ones) will be transferred to the appropriate storage cores (S_{11}, S_{12}, S_{14}, S_{23}, S_{27}) of the "or" gate. After the next I_A and I_{RA} pulse any one that is contained in any of these storage cores will be transferred to core S_{28} and, at the same time, a one will be put into core S_{29}. On the next I_B pulse the one in core S_{29} will be inhibited by the one in core S_{28}. Thus the only time that an output will occur is when core S_{28} does not contain a one. This will be the case only when stages 1, 2, 3, 4_a and 5_a do not contain ones. Thus an output occurs only for the case where the "exclusive or" circuit contains a one and all the other stages contain zeroes.

The parameters used in this circuit, as well as the design criteria of section 2, are listed in table V. Sometimes the values of drive current found to operate the actual circuit differ somewhat from those listed in the table. The difference between these values can readily be traced to the cores used in the circuit. The cores were General Ceramics F-426-S4 cores. These cores have extremely fast switching times of about 0.1 to 0.3 μs. With these fast switching times the distributed inductance and capacitance of the circuit just begin to become important. Another, even more important factor, that was not considered is skin effect which at these high switching rates may raise the effective value of loop resistance. Since none of this was taken into account in the design equations, one would naturally expect some error in the results.

When one considers the above factors, the shift code counter worked reasonably well. Judging from the fact that very close agreement between equation results and actual operating values for slower materials was obtained, it is readily assumed that the agreement between theory and practice would have been much better if slower cores had been used for the shift code counter. In summary, the successful operation of this circuit with its multiple logic functions readily proves the feasibility of all magnetic logic systems.

4.3 Pulse Elimination Technique

The use of direct current to reset the coupling cores is an excellent method to simplify the driving apparatus. Since most circuits usually have dc available somewhere in the circuit it is not necessary to provide a separate source because the reset operation draws an extremely small amount of power. A basic shift register stage utilizing dc for reset is illustrated in figure 39. The driving pulses consist only of two pulses, I_A and I_B, that must be separated in time by an amount equal to or greater than the reset time of the coupling cores. The dc for reset must be adjusted to a

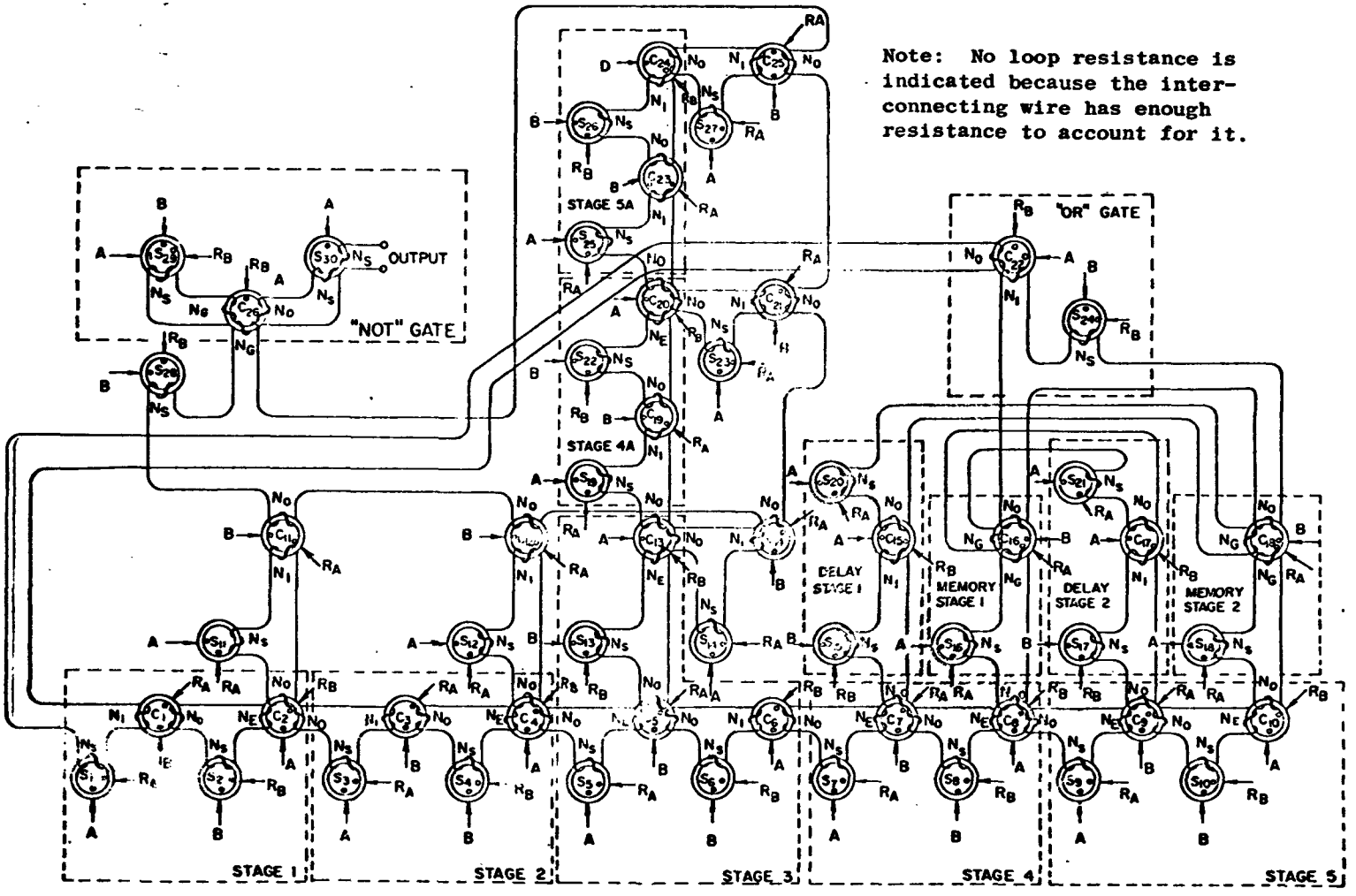


Figure 38. 6-stage shift code counter.

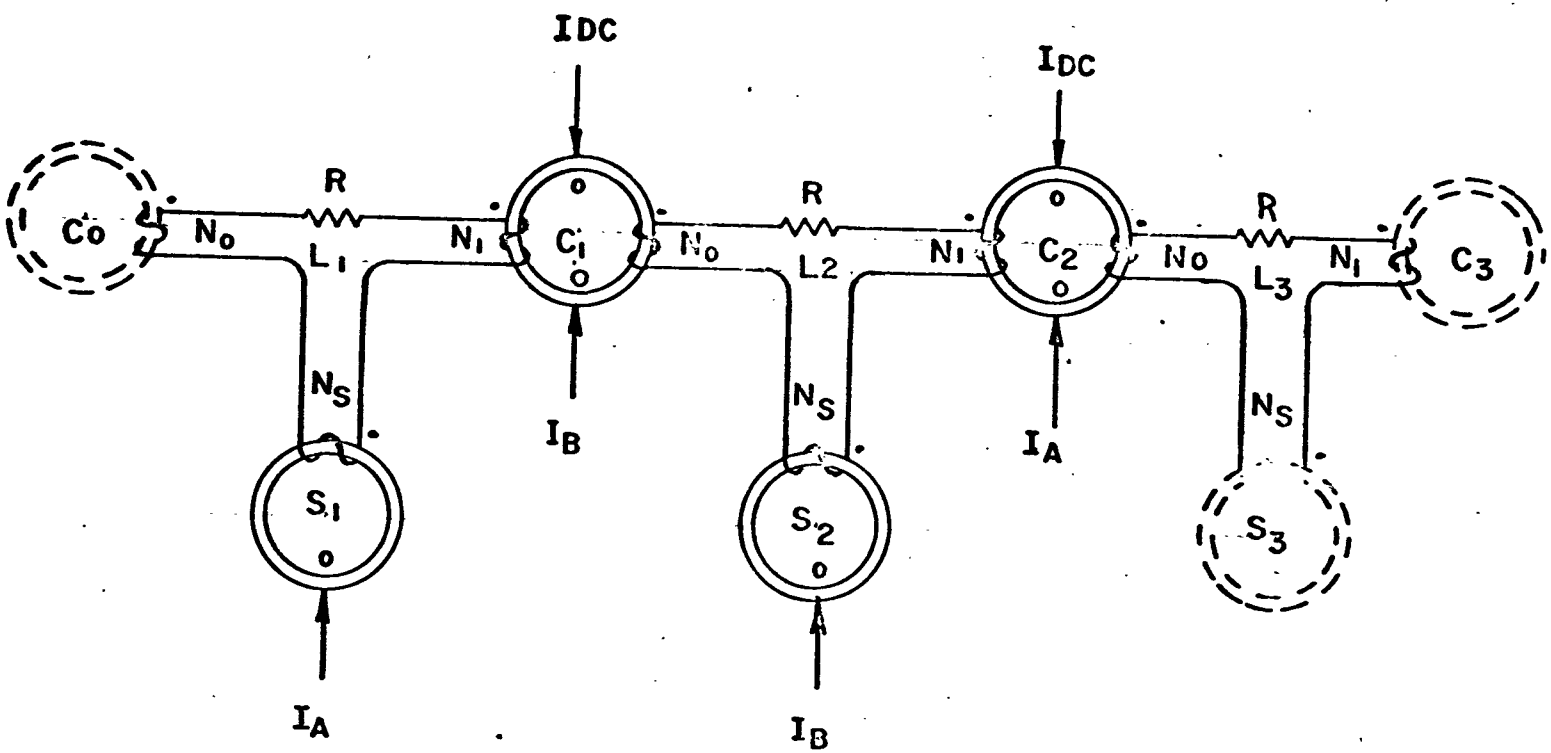


Figure 39. Basic two-pulse shift register.

value about equal to NI min/N_R, where N_R is the number of turns on the reset winding. This is the smallest value of current that will reset the core completely. During the I_A pulse, core S_1 switches producing a counterclockwise current in loop L_1. This current tends to switch core C_1 to the one state while I_{DC}, which is always present, tends to keep it in the zero state. At first glance it might appear that core C_1 might not switch because of the bucking mmf fields, but this is not the case since the mmf due to the current in loop L_1 is much more than that due to I_{DC}. Thus core C_1 does switch causing a counterclockwise current in loop L_2 that switches core S_2. Immediately after the transition occurs, core C_1 switches back to the zero state because of the ever-present I_{DC}. As in the case where the coupling cores were reset with pulses, the resetting process produces currents in loops L_1 and L_2 that tend to put a one in core S_1 and a zero in core S_2, both of which are undesirable. To insure that information is not inserted in core S_1, the width of the I_A drive pulse must be wide enough to hold core S_1 in the zero state. To insure that the one in core S_2 is not lost, I_{DC} must be amplitude limited so as not to cause too large a current in coupling loop L_2. Note that this has already been done by making I_{DC} about equal to NI min/N_R. During the I_B pulse, the same procedure occurs driving the one to the first core in the second stage (core S_3).

It is interesting to note that the design equations of section 2.1 are only slightly modified because of I_{DC}. Since $N_A I_{DC}$ is on the order of NI min, and also since all the terms involving NI min amounted to less than 10 percent of the final value in all the design equations, then no appreciable error will result from using these design equations without regarding I_{DC}.

A system using dc for reset was built and tested using General Ceramics F-726-S4 cores. It was tested on a basic shift register circuit and also on one that included a branching circuit. In all cases it was found to operate satisfactorily.

5. CONCLUSIONS

This investigation has shown that it is possible to perform all the basic digital functions associated with memory, "and" gates, "or" gates, and "not" gates by using circuits composed only of magnetic cores and interconnecting wire. One of the most important features of this report is the presentation of a complete design procedure for all types of logic circuits. As indicated in section 4 this analysis worked extremely well for cores with switching times on the order of microseconds and served as an excellent guide for cores whose switching times are on the order of one-tenth microsecond. The great success achieved by designing and constructing a variety of these circuits indicates their possibilities as an excellent method to perform logic functions. The obvious advantages of extreme simplicity of components, nominal cost, and small size should make systems of this type applicable to a great variety of circuitry.

The only problem remaining for all magnetic circuits or more specifically ferrite cores is that of temperature. None of the circuits described in this report was subjected to any temperature tests (ref 6). The effect of temperature on all magnetic systems using ferrite cores is to be the subject of a subsequent investigation.

6. BIBLIOGRAPHY

(1) A. T. Meyerhoff, "Digital Applications of Magnetic Devices," John Wiley & Sons, New York.

(2) L. S. Onyshkevych, "Analysis of Circuits with Multiple Hole Magnetic Cores," M. S. Thesis, Massachusetts Institute of Technology, 1957.

(3) L. A. Russel, "Diodeless Magnetic Core Circuits," RCA Review, 1957.

(4) H. D. Crane and E. K. Van DeRiet, "Design of an All-Magnetic Computing System—Part I," IRE Transactions on Electronic Computers, June 1961.

(5) D. L. Hamilton, "Diodeless Magnetic Core Counter and Logic Circuits," DOFL Report R-320-60-25, August 1960.

(6) E. A. Sands, "An Analysis of Magnetic Shift Register Operation," Proc IRE 41, 993 (1953).

(7) G. W. Kinzelman and H. J. Rosenberg, "Feedback Shift Register for Use in a Programmer," DOFL TR-982, October 1961.

DISTRIBUTION

Office of the Director of Defense Research & Engineering
The Pentagon, Washington 25, D. C.
Attn: Technical Library (rm 3E1065)—2 copies

Commanding Officer
AMC Detachment #1
Temporary Building I
Washington 25, D. C.
Attn: ORDTB (Research Br)

Director
Research & Development
AMC Detachment #1
Temporary Building I
Washington 25, D. C.
Attn: Director, Developments

Commanding General
U.S. Army Ordnance Missile Command
Redstone Arsenal, Alabama
Attn: Technical Documents Library

Commanding General
Frankford Arsenal
Philadelphia 37, Pennsylvania
Attn: Reference Librarian

Commanding General
Aberdeen Proving Ground, Maryland
Attn: Tech Library, Br No. 3, Bldg 400 D&P Services

Commanding Officer
Picatinny Arsenal
Dover, New Jersey
Attn: ORDBB-TW1, L. Marino
Attn: Library

Commanding General
Ordnance Weapons Command
Rock Island, Illinois
Attn: Technical Library

Commanding General
Ordnance Special Weapons-Ammunition Command
Dover, New Jersey

Commander
U.S. Naval Ordnance Laboratory
White Oak, Silver Spring 19, Maryland
Attn: Tech Library

DISTRIBUTION (Continued)

Commander
Armed Services Technical Information Agency
Arlington Hall Station
Arlington 12, Virginia
Attn: TIPDR (10 copies)

Director
National Security Agency
Fort George G. Meade, Maryland
Attn: CREF-121, Tech Documents Section
Attn: Mr. Dale L. Hamilton

Internal

Horton, B. M./McEvoy, R. W., Lt. Col.
Apstein, M./Gerwin, H. L./Guarino, P. A./Kalmus, H. P.
Spates, J. E./Schwenk, C. C.
Hardin, C. D., Lab 100
Sommer, H., Lab 200
Hatcher, R. D., Lab 300
Hoff, R. S., Lab 400
Nilson, H., Lab 500
Flyer, I. N., Lab 600
Campagna, J. H./Apolenis, C. J., Div 700
DeMasi, R., Div 800
Landis, P. E./Horsey, E. F., Lab 900
Seaton, J. W., 260
Piper, William, 450
Kinzelman, Gerald, 450
Rosenberg, Harry, 450
Marcus, Ira, 450
DeHasanta, Osvaldo, 450 (20 copies)
Technical Reports Unit, 800
Technical Information Office, 010 (10 copies)
DOFL Library (5 copies)

(Two pages of abstract cards follow.)

AD_____ Accession No._____

Diamond Ordnance Fuze Laboratories, Washington 25, D. C.

DIODELESS MAGNETIC CORE LOGIC CIRCUITS — Dale L. Hamilton

TR-1072, 15 August 1962, 33 pp text, 39 illus., Department of the Army Proj 5B06-01-014, OMS CODE 5530.11.62400, DOFL Proj 46345, UNCLASSIFIED Report

This report shows that it is possible to perform all the basic logic functions associated with memory, "and" gates, "or" gates, and "not" gates by using circuits composed only of square loop magnetic cores and interconnecting wire. A design procedure is given for all the basic logic circuits, a coincidence detector, and a shift code counter. These circuits were developed using miniature ferrite cores that have switching times on the order of microseconds.

Magnetic cores

Logic circuits

AD_____ Accession No._____

Diamond Ordnance Fuze Laboratories, Washington 25, D. C.

DIODELESS MAGNETIC CORE LOGIC CIRCUITS — Dale L. Hamilton

TR-1072, 15 August 1962, 33 pp text, 39 illus., Department of the Army Proj 5B06-01-014, OMS CODE 5530.11.62400, DOFL Proj 46345, UNCLASSIFIED Report

This report shows that it is possible to perform all the basic logic functions associated with memory, "and" gates, "or" gates, and "not" gates by using circuits composed only of square loop magnetic cores and interconnecting wire. A design procedure is given for all the basic logic circuits, a coincidence detector, and a shift code counter. These circuits were developed using miniature ferrite cores that have switching times on the order of microseconds.

Magnetic cores

Logic circuits

AD_____ Accession No._____

Diamond Ordnance Fuze Laboratories, Washington 25, D. C.

DIODELESS MAGNETIC CORE LOGIC CIRCUITS — Dale L. Hamilton

TR-1072, 15 August 1962, 33 pp text, 39 illus., Department of the Army Proj 5B06-01-014, OMS CODE 5530.11.62400, DOFL Proj 46345, UNCLASSIFIED Report

This report shows that it is possible to perform all the basic logic functions associated with memory, "and" gates, "or" gates, and "not" gates by using circuits composed only of square loop magnetic cores and interconnecting wire. A design procedure is given for all the basic logic circuits, a coincidence detector, and a shift code counter. These circuits were developed using miniature ferrite cores that have switching times on the order of microseconds.

Magnetic cores

Logic circuits

AD_____ Accession No._____

Diamond Ordnance Fuze Laboratories, Washington 25, D. C.

DIODELESS MAGNETIC CORE LOGIC CIRCUITS — Dale L. Hamilton

TR-1072, 15 August 1962, 33 pp text, 39 illus., Department of the Army Proj 5B06-01-014, OMS CODE 5530.11.62400, DOFL Proj 46345, UNCLASSIFIED Report

This report shows that it is possible to perform all the basic logic functions associated with memory, "and" gates, "or" gates, and "not" gates by using circuits composed only of square loop magnetic cores and interconnecting wire. A design procedure is given for all the basic logic circuits, a coincidence detector, and a shift code counter. These circuits were developed using miniature ferrite cores that have switching times on the order of microseconds.

Magnetic cores

Logic circuits