BASIC THEORY OF DIGITAL COMPUTERS

1 January 1957

VOLUME 3

IBM®

BASIC THEORY OF DIGITAL COMPUTERS

VOLUME III

Contract No.

AF 30(635)-1404

1 January 1957

MILITARY PRODUCTS DIVISION
INTERNATIONAL BUSINESS MACHINES CORPORATION
KINGSTON, NEW YORK

BASIC THEORY OF DIGITAL COMPUTERS

DC1

PART 4

COMPUTER COMPONENTS

Draft No. 2

INTERNATIONAL BUSINESS MACHINES CORPORATION
KINGSTON, NEW YORK

UNCLASSIFIED

PART 4

CHAPTER 1

ARITHMETIC COMPONENTS

1.1 LOGICAL CIRCUITS

1.1.1 General

The three logical operations AND, OR and NOT are introduced in Part 2, Chapter 3. The operations may be characterized in terms of binary numbers as follows:

a. Assume that a number of inputs $x_1, x_2, x_3, \ldots, x_n$ are applied to the input terminals of an AND block. Then if each of these inputs is 1, the output of the block is 1. If, on the other hand, any of the inputs is 0, the output of the block is 0.

b. Assume that a number of inputs $x_1, x_2, x_3, \ldots, x_n$ are applied to the input terminals of an OR block. Then, if any of these inputs is 1, the output of the block is 1. If all the inputs are 0, then the output is 0.

c. Assume that an input, x, is applied to the input terminal of a NOT block. Then, if x is 1, the output is 0. On the other hand, if x is 0 the output is 1.

Circuits which perform the AND and OR functions are presented in this section. A NOT circuit isn't developed. However, a study of the flip-flop circuit developed in the following section (Section 1.2) reveals that a NOT function can be obtained from this circuit by proper choice of output terminals. The inverter, which is presented in Chapter 3 of this Part, also performs the NOT function.

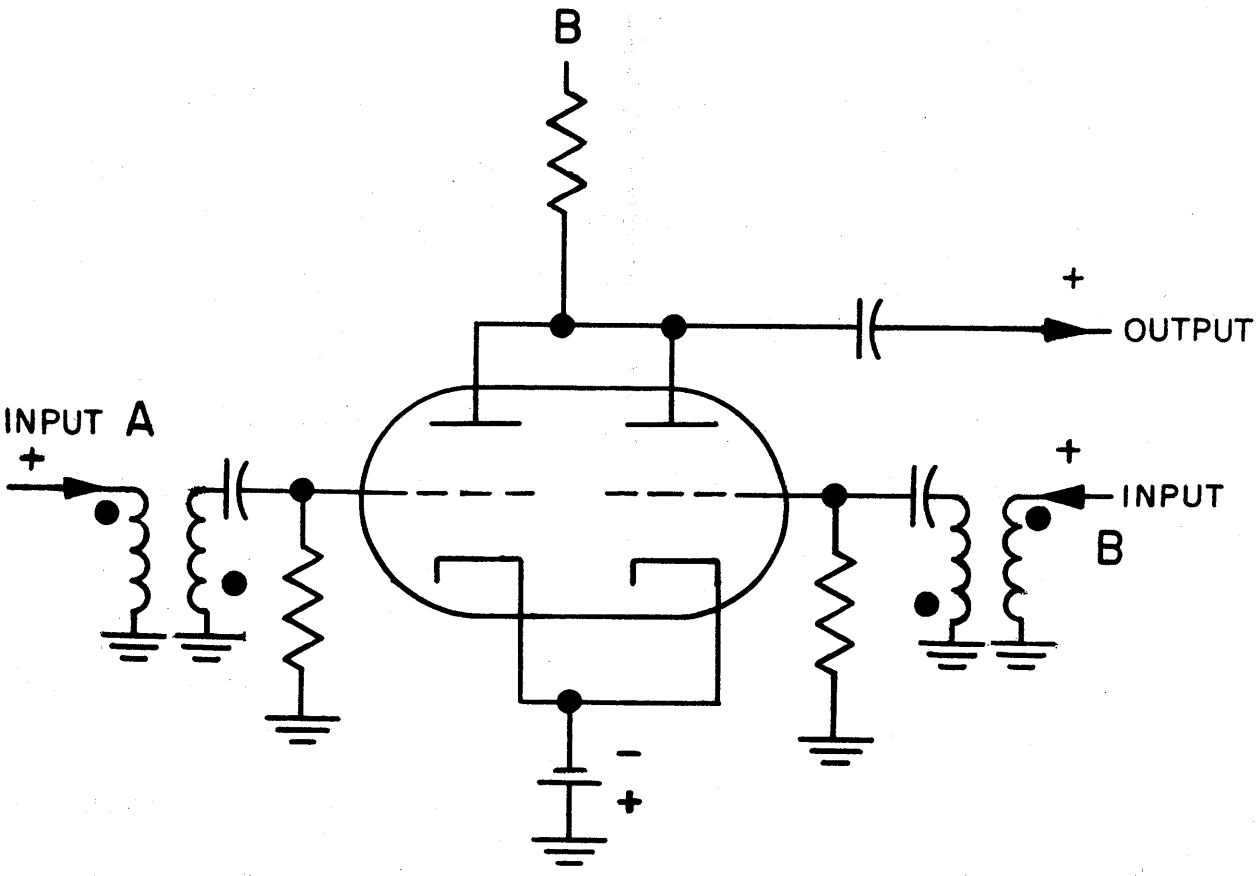


Figure 4-1

1.1.2 Positive and Negative Logic

In the logical circuits developed in this chapter, 1's and 0's are represented either by steady-state voltage levels or by the presence or absence of pulses at particular instants. The representation of a 1 by a positive voltage level and a 0 by a negative voltage level is defined as positive logic. Also, the representation of a 1 by a positive pulse appearing at a particular instant, or of a 0 by the absence of a pulse at that instant is defined as positive logic. On the other hand, the representation of a 0 by a positive voltage level and a 1 by a negative voltage level, or the representation of a 0 by a positive pulse appearing at a particular instant or of a 1 by the absence of a pulse at that instant is defined as negative logic. Unless otherwise specified positive logic is assumed in the discussion which follows.

1.1.3 AND Circuits

An AND circuit employing a twin troide is illustrated in Figure 4-1. The two sections of the tube share a common plate load resistor and a common cathode supply. The grid of each section of the tube is returned to ground through a resistor. Since the cathode supply is negative with respect to ground, plate current is drawn through both sections of the tube in the absence of an input to either grid. Inputs to the grids are applied through phase-inverting input transformers. A positive pulse applied to the input of either section thus appears as a negative pulse on the grid of the section causing the section to be cut off. However, the plate and battery supplies and

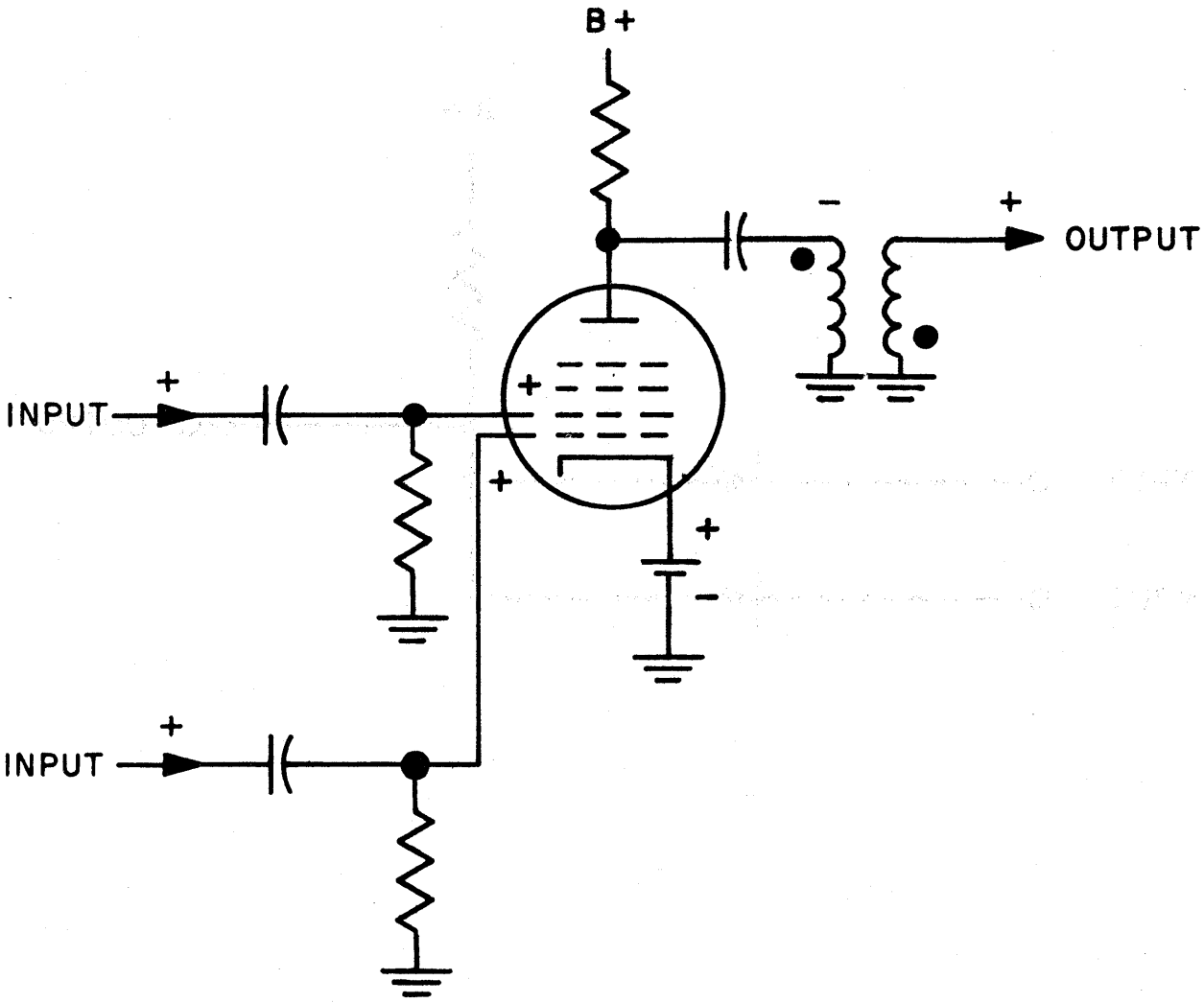


Figure 4–2

plate load resistance are so chosen that either section of the tube alone is capable of drawing approximately the same magnitude of current as can be drawn by both sections conducting simultaneously. Thus, an input applied to one section of the tube, causing that section to be cut off, has a negligible effect on the total current through the circuit and therefore the voltage drop through the plate load is only slightly affected. If, on the other hand, both sections of the tube are simultaneously cut off (by the simultaneous appearance of input pulses on both input lines) then the plate voltage rises to the level of the plate supply, producing a positive pulse on the output line. In terms of positive logic, then, the circuit performs the AND function; that is, a 1 appears at the output if and only if 1's are applied to both inputs simultaneously.

An AND circuit employing a single multi-grid tube is illustrated in Figure 4-2. Both grids are biased negatively with respect to the cathode so that the tube is normally cut off. A positive signal applied to either one of the grids is not sufficient to cause conduction through the tube. However, if positive pulses appear simultaneously at both inputs, the tube conducts, causing a voltage drop through the plate load. The resultant negative signal at the plate is coupled through a capacitor and a phase-inverting output transformer so that it appears as a positive pulse on the output line. Thus, in terms of positive logic, the circuit performs the AND function; that is, a 1 appears at the output if and only if 1's are simultaneously applied to both inputs.

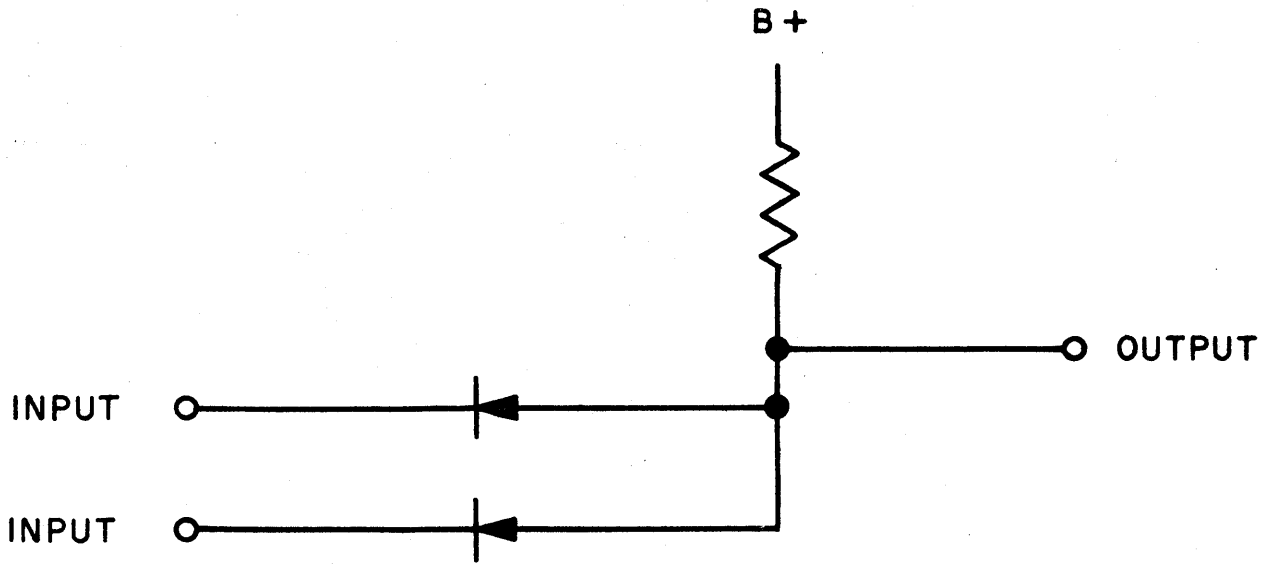


Figure 4-3

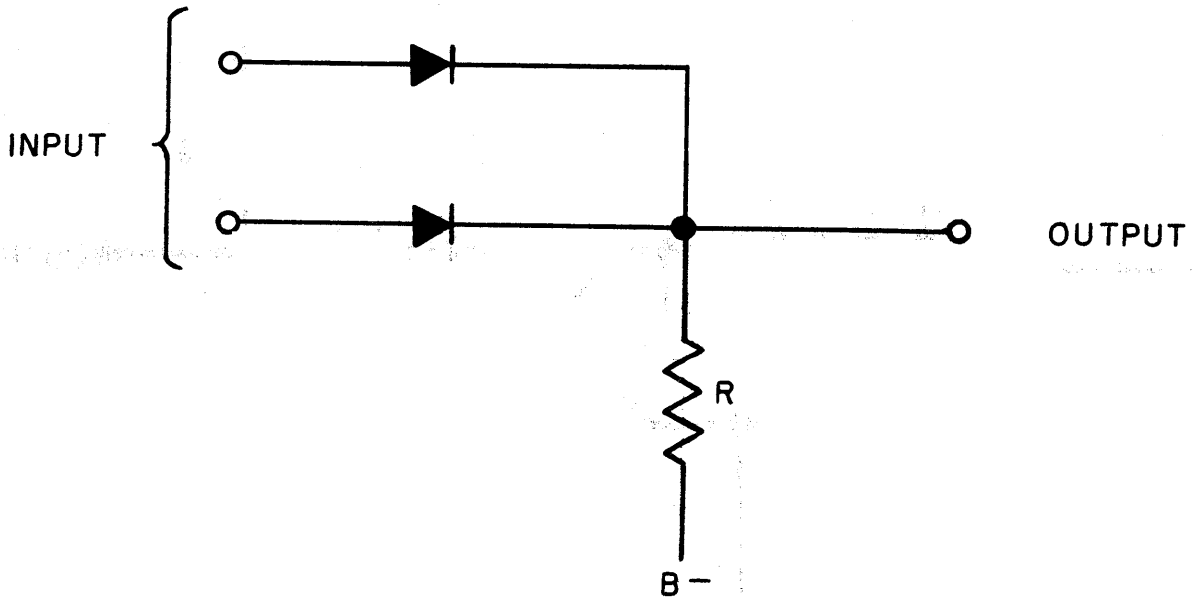


Figure 4-4

The AND circuits of Figures 4-1 and 4-2 are RC coupled. Thus they supply transient outputs in response to transient inputs. An entirely different kind of AND circuit is illustrated in Figure 4-3. This circuit comprises two diodes and a dropping resistor. If both of the inputs are positive (1's), then the output is positive (1). However, if either of the inputs is negative (0), current is drawn through the dropping resistor and through the corresponding diode. Since the forward resistance of the diode is small compared to the dropping resistance, the output falls virtually to the level of the negative input (0). Current from the positive input terminal is blocked by the associated diode. If both inputs are negative, current is drawn through both diodes and the effect is the same. This circuit, therefore, is capable of generating a steady-state AND output in response to steady-state inputs.

1.1.4 OR Circuits

As noted above, the circuits of Figure 4-1 through 4-3 are defined in terms of positive logic as AND circuits. It should be understood that in terms of negative logic they are OR circuits. For example, in terms of negative logic, the circuit of Figure 4-3 generates a steady-state 1 output (negative voltage level) in response to a steady-state 1 input (negative voltage level) on either of its input lines.

A diode network which functions as a positive OR circuit is shown in Figure 4-4. Comparing this network with the circuit of Figure 4-3, it can be seen that the direction of the diodes has been reversed and that, moreover, the polarity of the reference voltage applied to the dropping resistor has been re-

versed. If a 1 (positive voltage level) appears on either of the input lines, then a 1 (positve voltage level) appears on the output line. This follows from the fact that a positive voltage on either of the input lines causes current to flow between the input terminal and the reference supply. Since the forward resistance of the diodes is very small with respect to the dropping resistor, the output terminal is raised to essentially the potential of the input terminal. If both input terminals are positive, the effect is substantially the same. If both input terminals are negative, no current flows through either diode and the output terminal assumes the potential of the reference supply (i.e. a negative potential). Thus the circuit satisfies the definition of the OR function. It should be understood that in terms of negative logic, the circuit of Figure 4-4 is an AND circuit just as the circuit of Figure 4-3 is an OR circuit.

1.1.5 Adders, Subtractors and Multipliers

The AND and OR circuits discussed in the preceding sections provide the means for implementing the blocks of the half-adders, full-adders, multipliers and so on that were discussed in block form in Chapter 3 of Part 2. However, there is more to these combinations than just the logical circuits. The diode AND and OR circuits for example are passive elements, that is they dissipate rather than generate power. This implies the need for amplifiers to be used in conjunction with them. The dual triode and multi-grid logical circuits are active elements; however, their transient action implies the need for subsidiary timing

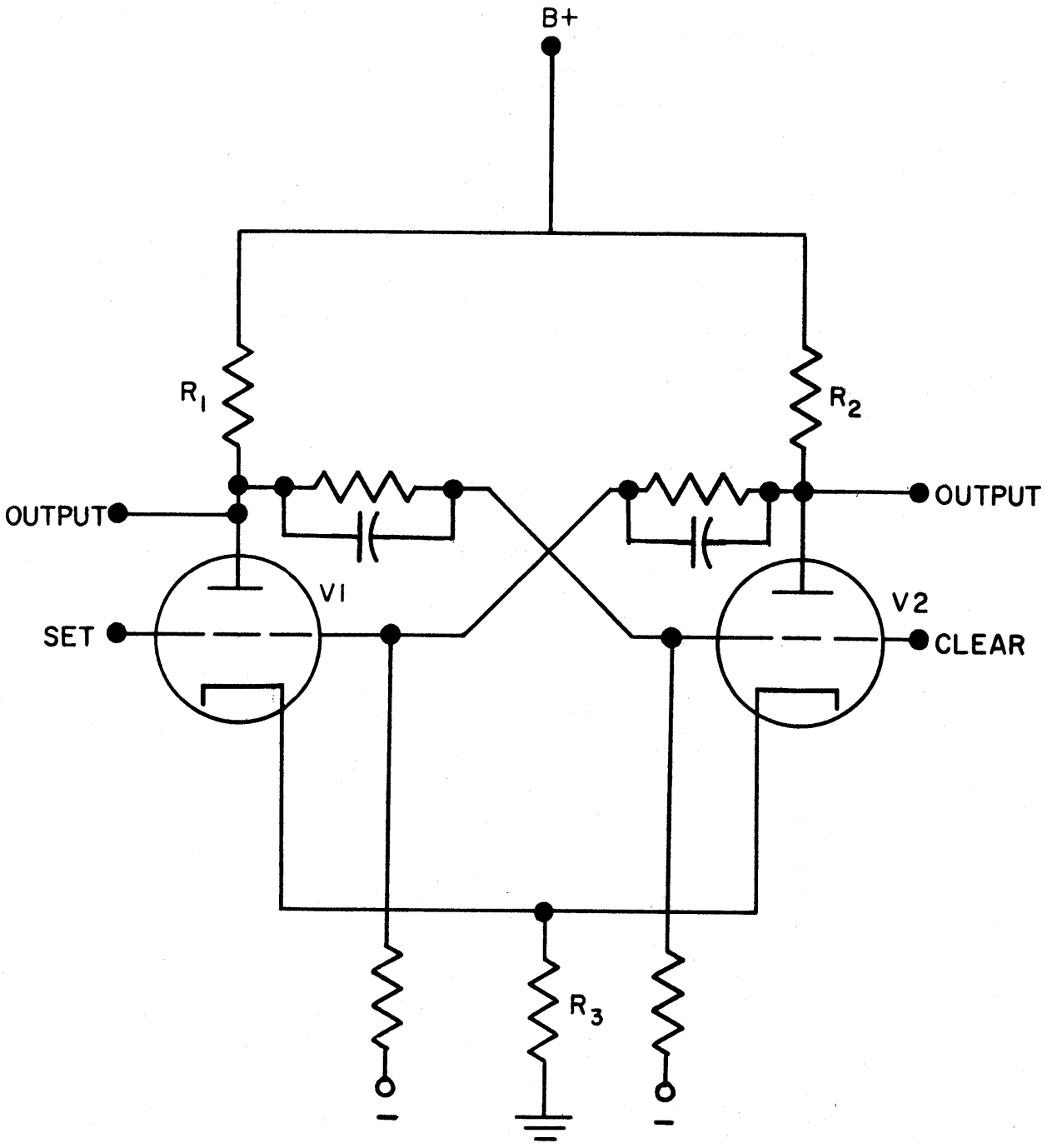


Figure 4-5

circuitry. The voltage and power amplification circuits that are required for use in conjunction with passive elements are discussed in Chapter 3 of this Part. A timing and ordering circuit is presented in the succeeding section. Timing is considered in more general terms in Part 5.

1.2. FLIP-FLOP CIRCUITS

1.2.1. General

The flip-flop is a bi-stable multivibrator; i.e. it is a circuit which has two stable states. This implies that an external input signal is required to drive it from one state to the other. The basic circuit which is shown in Figure 4-5, comprises two vacuum tubes (or two tube sections in a single envelope) and its stable states are characterized by the condition that one of the tubes is cut off and other is conducting. By associating one of the stable states of a flip-flop with 1 and the other with 0, the circuit can be employed to provide representation of a single binary digit or bit. A group of flip-flops, each one associated with a particular order of significance (i.e. 2^0, 2^1, 2^2 etc.), can be used to represent a binary number.

Such a group is called a register. The condition of a flip-flop can be sensed in terms of the voltage levels at the plates of either one or both of its tubes. For example, if the condition characterized by tube v^1 cut off and tube v^2 conducting is associated with a 1; then a positive voltage at the plate of V^1 or a negative voltage at the plate of V^2 is interpreted as a 1, while a positive voltage at the plate of V^2 or a negative

voltage at the plate of V^1 is interpreted as a 0. This corresponds to the fact that the plate voltage of a tube is lowered when it conducts, by virtue of the voltage drop through the plate load resistance. Thus the terms positive and negative are used above in the relative sense, that is the two voltage levels are positive and negative with respect to each other, but not necessarily with respect to ground.

As already noted, the two stable states of the flip-flop are characterized by the condition that one tube is conducting and the other tube is cut off. In order to make the condition characterized by both tubes conducting and unstable one, the plate of each tube is coupled to the grid of the other tube as shown in Figure 4-5.

To understand the operation of the circuit, assume that V1 is conducting and V2 is cut off. Assume further that a negative input pulse is applied (through the Set input) to the grid of V1. This causes a decrease in the plate current through V1 which appears as an increase of potential on the plate of the tube. This positive going signal is capacitively coupled to the grid of V2 allowing V2 to conduct. As plate current starts to flow through V2, the plate potential of the tube decreases. This negative going signal is capacitively coupled to the grid of V1 where it causes a further decrease in the plate current. In the limit, this unstable condition causes V1 to be driven to cut off. If, now, a negative signal is applied to the grid of V1 it will have no effect since the tube is already cut off. If, on the other hand, a negative signal is applied to the

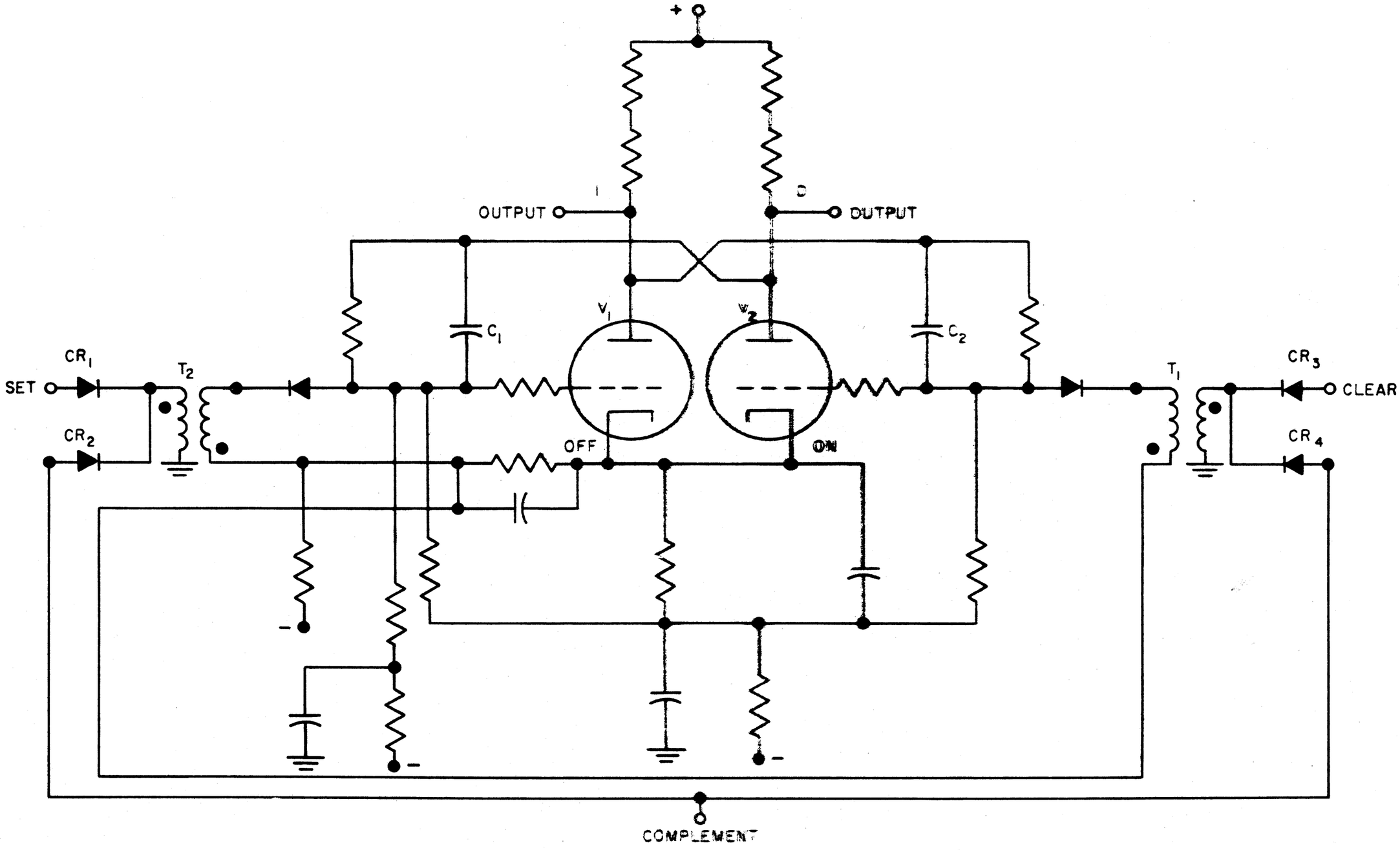

+
OUTPUT
1
0
OUTPUT
C_1
V_1
V_2
C_2
CR_1
SET
T_2
CR_2
OFF
ON
T_1
CR_3
CLEAR
CR_4
COMPLEMENT

Figure 4-6

grid of V2, the circuit will pass through the condition of instability described above and will arrive at the opposite stable condition (i.e. V1 conducting and V2 cut off). Thus the circuit can be driven back and forth between its two stable states by applying a negative pulse first to the grid of one tube and then to the grid of the other. An alternative method of driving the circuit back and forth between its two stable states is to use only one of the input grids but to alternate the polarity of the input signal applied to that grid. Assume for example, that V1 is conducting. The application of a negative pulse to its grid will reverse the state of the circuit as noted above. Thus V1 will be driven to cut-off. If, now, a positive signal is applied to the grid of V1 the circuit will be driven through the condition of instability to its opposite stable state; that is to say, the application of a positive pulse to the tube which is cut off is equivalent to the application of a negative pulse to the grid of the tube which is conducting.

1.2.2. Set, Clear and Complement Inputs

A flip-flop circuit with three input terminals is shown in Figure 4-6. This circuit is designed to accept only positive input pulses. However these pulses are coupled to the grids of the tubes through input transformers which provide a phase inversion. Thus the pulses applied to the grids are always negative. It is assumed that for the circuit of Figure 4-6, the state in which V1 is cut off and V2 is conducting represents a 1. With this convention established, the three input terminals

can be defined as the Set, Clear and Complement inputs.

A positive pulse applied to the Set input of the circuit passes through diode CRI and appears across the primary of transformer T2. This causes a negative pulse to appear at the grid of V1. The pulse applied to the Set line is blocked by diode CR2 so that it does not reach the primary of transformer T1 which is the input transformer for the other side of the circuit. Thus a positive pulse applied to the Set line reverses the state of the Flip-flop if and only if it is storing a o, that is if and only if V1 is conducting. Just the opposite is true, if a positive pulse is applied to the Clear input of the circuit. This pulse reaches input transformer T1 through CR3 but is blocked from reaching input transformer T2 by CR4. Thus, it causes a negative pulse to appear on the grid of V2 (by virtue of the phase inversion through T1) so that the state of the circuit is reversed if and only if it is storing a 1, that is if and only if V2 is conducting.

A positive pulse applied to the Complement input of the circuit is passed by diode CR2 and by diode CR4 so that it appears across the primary windings of both T1 and T2. Thus negative pulses appear simultaneously on both grids. The negative pulse arriving at the grid of the tube which is cut off has no effect; however the negative pulse arriving at the grid of the tube which is conducting causes the circuit to reverse its state regardless of which state it is in. This corresponds to the rule given in Part 2 for forming the 1's complement of

a binary number, i.e. change all 0's of the number to 1's and all 1's of the number to 0's.

1.2.3 1 and 0 Outputs

As already noted, the condition of a flip-flop can be sensed in terms of the voltage levels on the grids of either one or both of its tubes. For the circuit of Figure 4-6, a positive voltage on the plate of V1 or a negative voltage on the plate of V2 indicates that the circuit is storing a 1, while the reverse conditions indicate that the circuit is storing a 0. In terms of the conventions concerning polarity of logic which are introduced in Section 1.1.2 of this Part, the output from V1 represents the contents of the circuit in terms of positive logic while the output from V2 represents the contents of the circuit in terms of negative logic. The positive logic output line is called the 1 output while the negative logic output line is called the 0 output.

Sometimes only one of the output lines of a flip-flop is used. This is called single line transfer. Sometimes, on the other hand, both lines are used. This is called double line transfer.

1.2.4. Registers

1.2.4.1 General

As noted above, a group of flip-flops used to store the bits of a single number are called a register. Each flip-flop in a register is associated with a particular order or column. In a computer using flip-flop registers, all kinds of essentially non-numeric as well as numeric information is represented

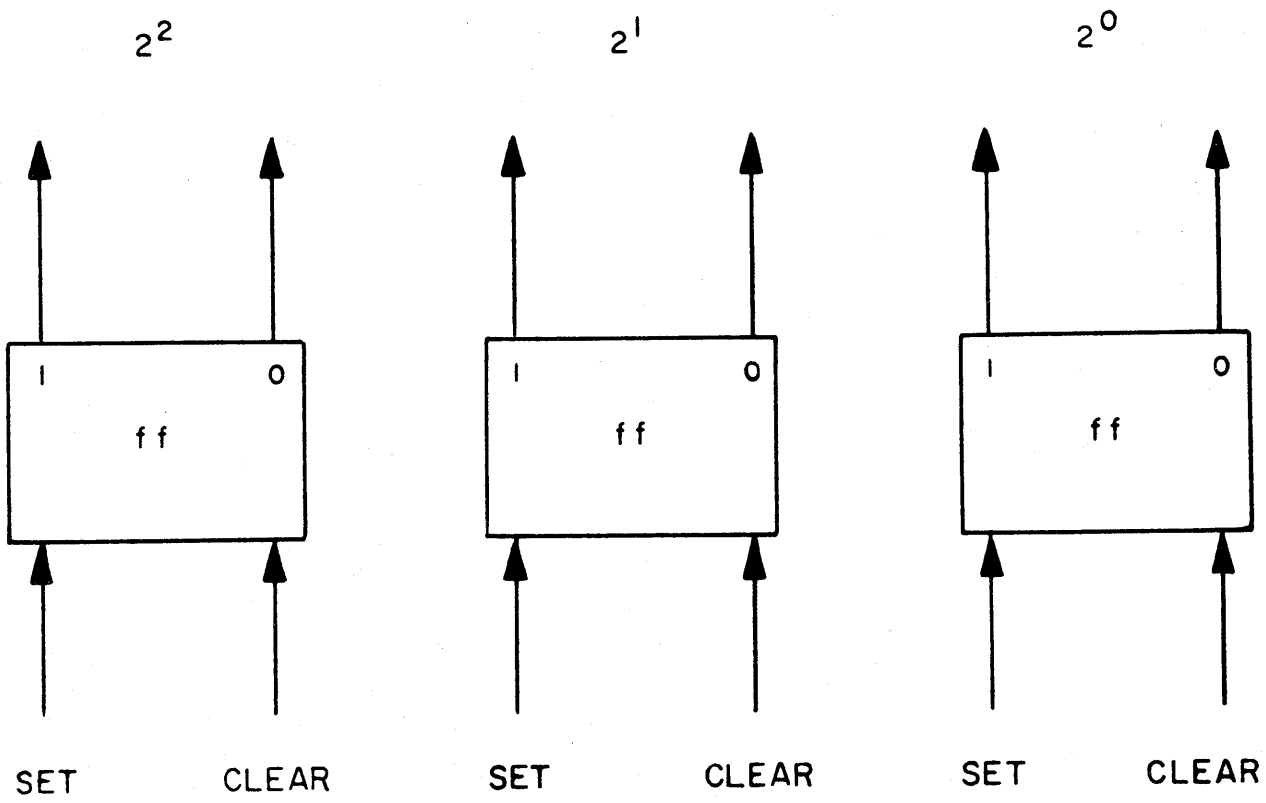


Figure 4-7

by binary codes. Thus a unit of information is usually called a word rather than a number, in order to indicate that it may represent either a number of an item of non-numeric information. In order to conform to this convention it can be stated that each register is associated with a bit position of a word rather than with an order of a number. However, when discussing operations upon numbers, it is more convenient to speak in terms of numbers and orders.

The bi-stable character of the flip-flop circuit makes it capable of storing a single bit of information. For example, when a positive pulse is applied to the Set input of the circuit of Figure 4-6, the circuit is driven to the state representing 1, if it is not already in that state, and remains in that state (i.e. stores a 1) until it receives a pulse on its Clear or Complement input, the simplest register is a storage register. Here, a flip-flop is provided for each required bit position as shown in Figure 4-7. In order to write a word into the register, a positive pulse is applied to the Set input of each flip-flop which is to store a 1 and to the Clear input of each flip-flop which is to store a 0. Another way to write into a register is first to clear each flip-flop and then to apply inputs to the Set lines of those flip-flops which are to hold 1's. This is the method employed when single line transfer of information into the flip-flop is desired.

If suitable interconnections are provided between the flip-flops of a single register, the register can be used to perform

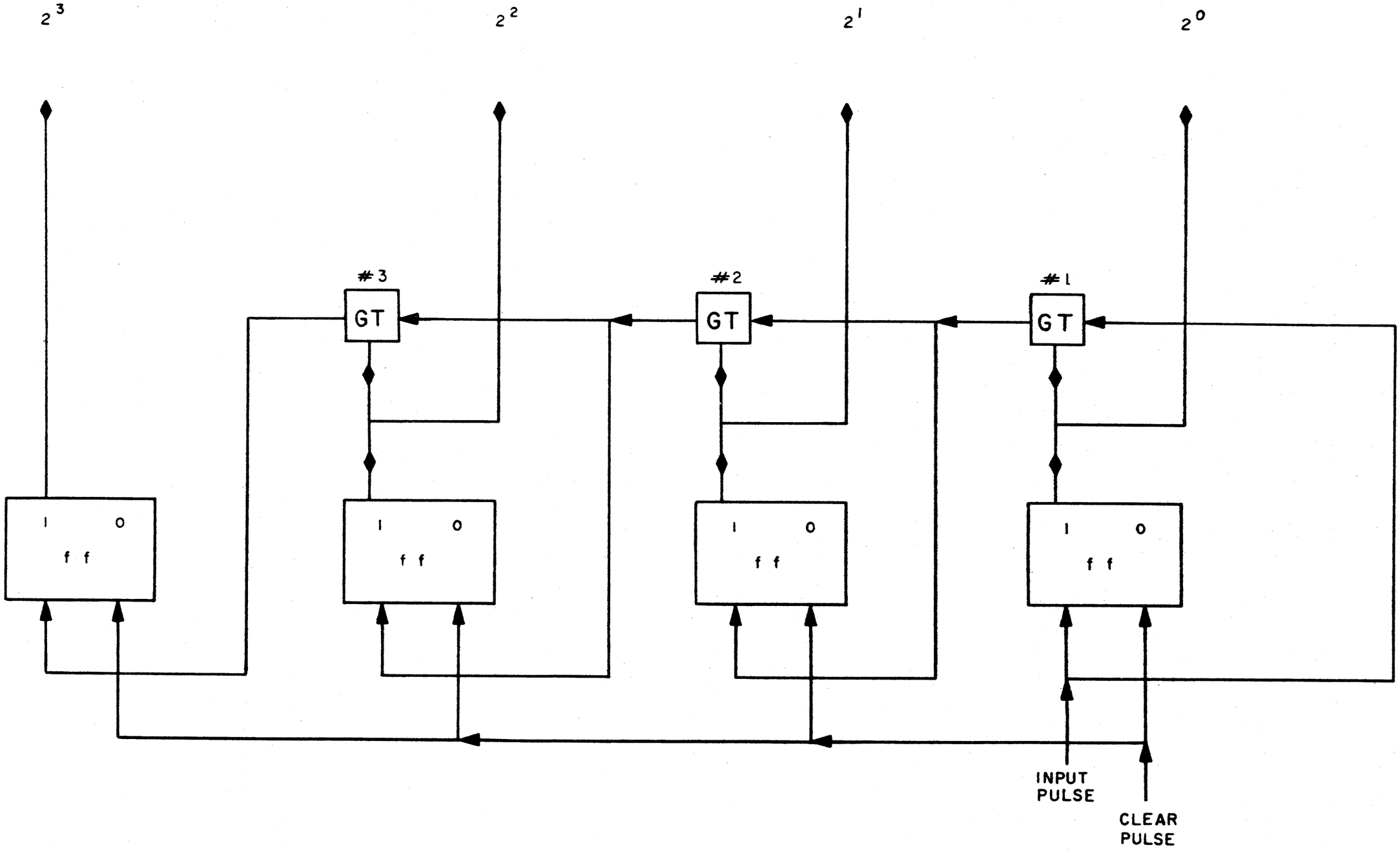

2^3
2^2
2^1
2^0
#3
#2
#1
GT
GT
GT
I
O
f f
I
O
f f
I
O
f f
I
O
f f
INPUT PULSE
CLEAR PULSE

Figure 4-8

counting and shifting functions in addition to the storage function. Counting and shifting registers are discussed in the succeeding sections.

1.2.4.2 Counting Registers

A binary counting register is shown in Figure 4-8. Here, each of the blocks marked FF is assumed to be a flip-flop circuit such as is illustrated in Figure 4-5.

Each of the blocks marked GT is a gate tube. Gate tubes are discussed in some detail in Section 1.3 of this Part. However, in order to understand the action of the counter it is necessary only to understand that a gate will pass a positive pulse if and only if it is receiving a steady-state positive signal at the instant when the positive pulse arrives. Since each gate in the counting register is connected to the 1 output of a flip-flop, this means that it will pass a positive pulse only if the contents of the flip-flop at the instant that the pulse arrives is a 1.

The counting input to each of the flip-flops of the register is to its complement input line. Assume that each of the flip-flops of the register is in the 0 condition (i.e. that the register is storing 0000). Then a positive pulse applied to the input pulse line appears simultaneously at the complement input of the 2^0 flip-flop and at the pulse input of the #1 gate. Since, at the instant when the pulse arrives, the flip-flop is storing a 0, the pulse is not passed through the gate. However, it does cause the 2^0 flip-flop to reverse its state (i.e. to store a 1). When a second pulse appears on the input pulse line, it is passed by gate #1, since the 2^0 flip-flop is storing a 1 at the instant

when the pulse arrives. Thus the second pulse reaches simultaneously the #2 gate and the complement input line of the 2^1 flip-flop. Since the 2^1 flip-flop is storing a 0 at the instant when the pulse arrives, the pulse does not pass through the #2 gate. However, it does cause the 2^1 flip-flop to reverse its state. Since the 2^0 flip-flop complement input line receives every input pulse directly, it also reverses its state. Thus, after the second pulse, the condition of the register is 0010 which is binary two. A third pulse is not passed by gate #1, since the 2^0 flip-flop is storing a 0 at the instant when it arrives. Thus, the third pulse merely reverses the state of the 2^0 flip-flop. The condition of the register is now 0011 which is binary three. A fourth pulse is passed by both gate #1 and gate #2 and thus reverses the state of the first three flip-flops. The condition of the register is now 0100 whi h is binary four. The count continues in this manner until the register is storing 1111 which is binary fifteen. When the sixteenth input pulse arrives it is passed by all three gates so that it reverses the state of all four registers; that is, the count is returned to 0. Thus the register is a modula 2^4 binary counter that is, it can store any of the dictinct numbers 0000 through 1111.

The register can be cleared (that is, made to store 0000) at any time by applying a pulse to the clear pulse line. This line is connected to the Clear input of each flip-flop. Thus, each flip-flop is driven to its 0 state (if it is not already in that state) when a pulse appears on the clear pulse line.

1.2.4.3 Shifting Registers

As discussed in Part 2, Chapter 3, a shift left operation in terms of binary arithmetic corresponds to a multiplication by two while a shift right operation corresponds to a division by two. These operations are required as a part of the routines connected with more general multiplication and division operations.

A register capable of providing a shift to the left is illustrated in Figure 4-9. The 1 and 0 outputs of each flip-flop of this register are coupled to the Set and Clear inputs respectively of the flip-flop on the left through gate tubes. If the 2^0 order flip-flop is storing a 1 when the shift pulse is applied, a pulse is passed to the Set input of the 2^1 flip-flop. On the other hand, if the 2^0 order flip-flop is storing a 0, then a pulse is passed to the Clear input of the 2^1 flip-flop. Thus, the bit initially held in the 2^0 flip-flop is shifted to the 2^1 flip-flop. Subtituting 2^n for 2^0 and $2^n + 1$ for 2^1, the above remarks can be generalized to apply to any two flip-flops of the register. Thus the register of Figure 4-9 performs a shift left operation for each shift pulse it receives. Notice that the shift pulse is applied directly to the reset input of the 2^0 order must contain 0 after a shift left has been performed. It is also possible to connect the circuit so as to shift the bit initially held in the left-hand flip-flop into the 2^0 flip-flop. This is called a cycling operation.

A register which shifts right rather than left can be formed by connecting the outputs of each flip-flop to the inputs of the flip-flop on its right rather than the flip-flop on its left. The connection is made through gate tubes just as in the case of the left shift register. For this register a shift right occurs each time a pulse is received.

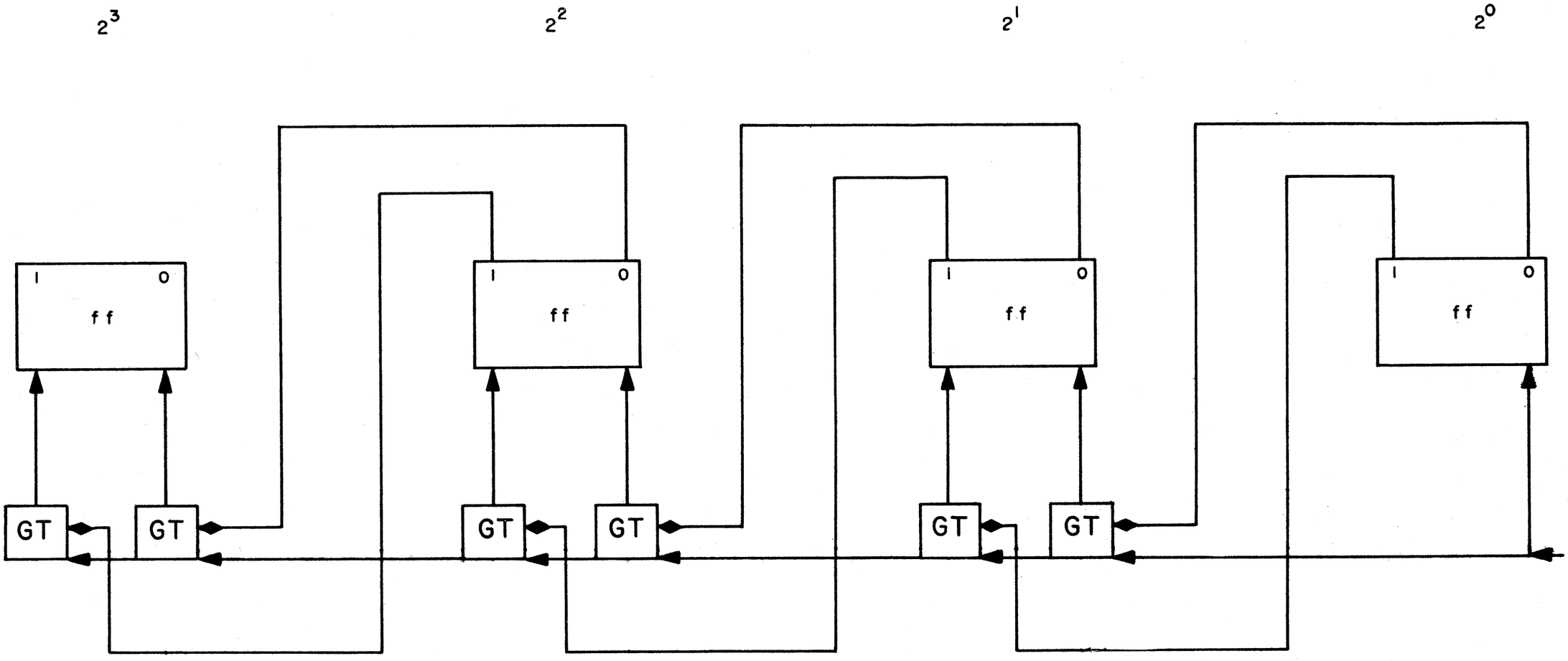


Figure 4–9

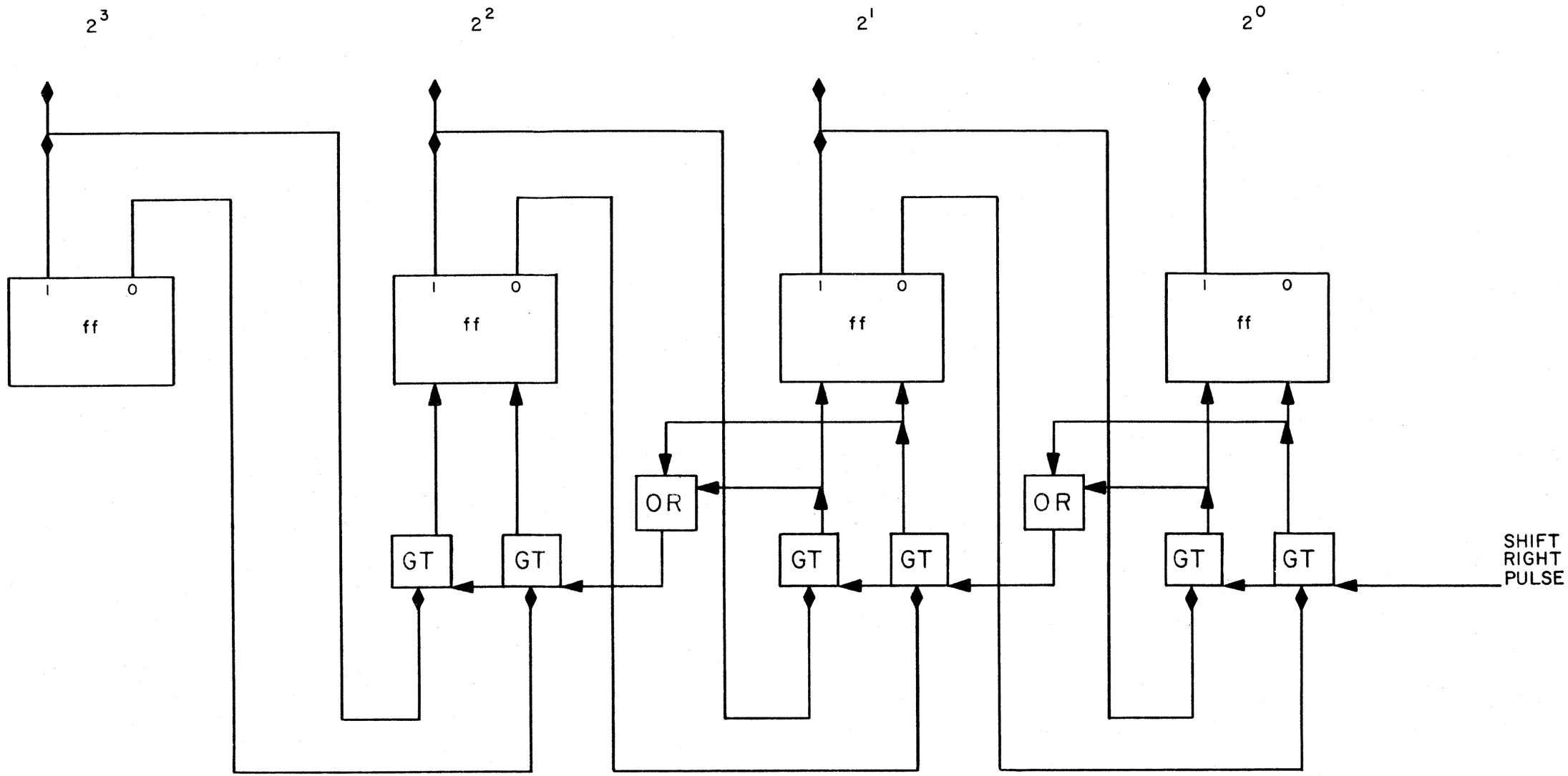


Figure 4-10

The register of Figure 4-9 provides a simultaneous shift of the contents of each flip-flop to the flip-flop on its left in response to a shift pulse. Another type of shift, called ripple shift is illustrated by the register shown in Figure 4-10. The particular register shown in this figure happens to provide a shift to the right rather than to the left. However, ripple shift register like simultaneous shift register can be designed to provide a shift in either direction. The ripple shift proceeds as follows: The shift signal is applied only to the gates between the two input lines of the 2^0 flip-flop and the output lines of the 2^1 flip-flop. Thus, a pulse is passed onto one input line or the other of the 2^0 flip-flop (depending upon whether the 2^1 flip-flop is storing a 1 or a 0). This shifts the contents of the 2^1 flip-flop to the 2^0 flip-flop. At the same time, regardless of which input line of the 2^0 flip-flop the pulse appears on, it is applied to the input of an OR circuit whose output provides a shifting pulse to the gates between the input lines of the 2^1 flip-flop and the output lines of the 2^2 flip-flop. As a result of this, the contents of the 2^2 flip-flop is shifted to the 2^1 flip-flop. The pulse on the input line of the 2^1 flip-flop, in turn, is applied through an OR circuit to the gates between the input lines of the 2^2 flip-flop and the output lines of the 2^3 flip-flop. Thus the shift right pulse is said to ripple through the register from right to left, each flip-flop receiving the contents of the flip-flop on the left an instant after its own contents

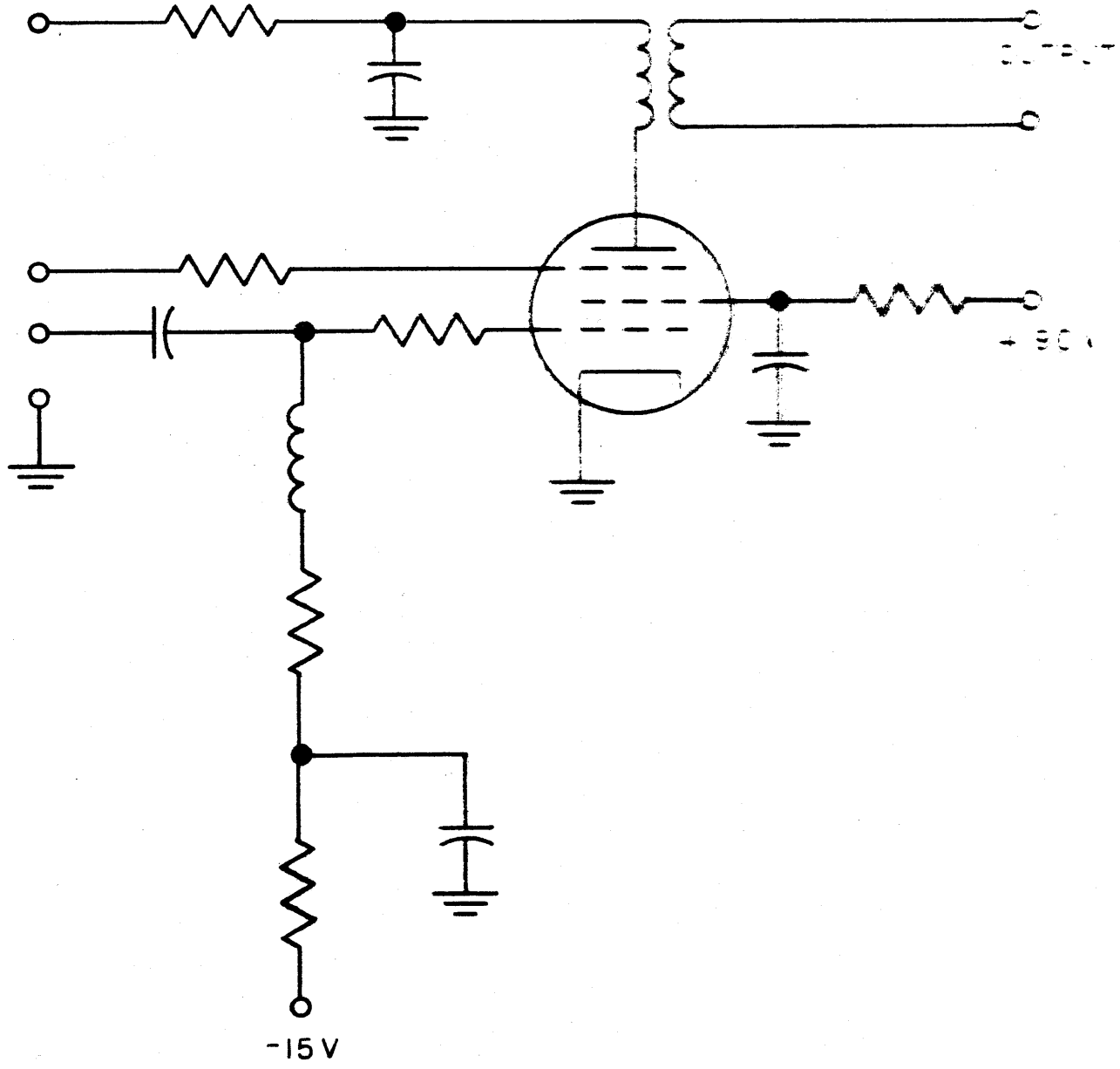


Figure 4-11

have been transferred to the flip-flop on its right.

It may appear that the simultaneous shift saves time. However, it turns out that, in certain situations, the ripple shift is the faster of the two. This results from the fact that a ripple shift can be initiated at the same time that some other signal (such as a carry signal) is propogating through the register. A simultaneous shift, on the other hand, cannot be initiated until all transient effects of a previous signal have been allowed time to die out.

1.3 GATE TUBE

Gate tubes are required to operate the shift register of the preceding section. As noted, their function in this application is to connect the outputs of each flip-flop in a register to the flip-flop on the left (or on the right depending upon the type of shift required) in response to a shift pulse. It is by a variety of command pulses such as this one that a computer executes instructions. Thus, the execution of an instruction requires the setting up of a specific set of signal paths. These paths may not be completed simultaneously but may be specified in an ordered sequence. The gate tube is the primary electronic switch which is used to complete specified signal paths.

A pentode gate tube circuit is shown in Figure 4-11. The circuit accepts one steady-state input which is directly coupled to the screen grid of the tube and another transient input which is RC-coupled to the control grid. The tube is biased so that it is normally cut off. It conducts only if a positive transient input is received at a time when the steady-state input is positive. The

transformer coupled output of the tube circuit produces a positive pulse in response to the transient plate current through the tube.

The gate tube is a special case of the AND circuit. Thus, assuming positive logic, it generates a 1 pulse at its output only if it receives a 1 pulse AND a 1 level. In a computer which uses diode AND and OR circuits, it is important to be able to distinguish a diode AND circuit from a gate tube circuit when they are represented on block-level diagrams, since the one provides steady-state logic and the other provides transient logic. For such a computer the diode circuit is usually represented on block diagrams by the name AND while the gate tube circuit receives some such designation as GT. This corresponds to the fact that it is the diode circuit which usually provides the AND functions required by adders, multipliers, etc., while the gate circuit applications are more in the nature of control functions such as ordering and timing the occurrence of sequences of operations.

PART 4

CHAPTER 2

STORAGE COMPONENTS

2.1 GENERAL

A digital computer provides problem solutions in a step-by-step manner. Thus initial data, intermediate results and instructions defining the sequence of steps must be stored during the course of a computation.

Instructions and data, whether numeric or non-numeric in character, are represented within a computer in essentially numeric form. Moreover, regardless of the number system upon which a computer operates, the representation of numbers within the computer can be inplemented by essentially binary devices. A decimal digit, for example, can be represented by the presence of a signal on a particular line. To represent a decimal order, then, a set of ten such lines is required, corresponding to the fact that the order may contain any one of the digits 0 through 9. Notice, however, that the presence or absence of any one of the digits is represented by a binary phenomenon, that is, the presence or absence of a signal on the line associated with that digit.

A storage element, then, must have the capability to accept and retain either a 1 or a 0. Thus it must be capable of assuming either one of two distinguishable and stable states. Since speed of operation is a primary concern in computer operation, a storage component must be capable of passing from one stable state to the other almost instantaneously.

The storage capability of flip-flop registers is discussed in Section 1.2 of the preceding chapter. Their description is included in Chapter 1 rather than here because they perform important arithmetic functions in addition to their storage function.

The types of storage devices discussed in this chapter may be classified as magnetic, electro-static, sonic and mechanical. They are discussed in that order in the following sections.

2.2 MAGNETIC STORAGE

2.2.1 General

The operation of magnetic storage devices depends upon the following properties of magnetism which have been discussed in some detail in Chapter 2, Part 4.

a. Certain so-called magnetic materials become magnetized when placed in a strong magnetic field, and retain a high value of remanent flux when the magnetizing field is removed. In some cases, the remanent flux is not decreased substantially even when the material is placed in a field opposite in polarity to the original field and of half the original field intensity.

b. A magnetic field surrounds any conductor through which current is flowing. When a conductor is wound to form a solenoid, individual flux loops are linked setting up a stronger magnetic field than is formed around a straight conductor. The strength of the field is a function of the number of turns of the solenoid and of the amount of current flow. If an iron core is inserted in the solenoid, more flux lines are linked resulting in a stronger

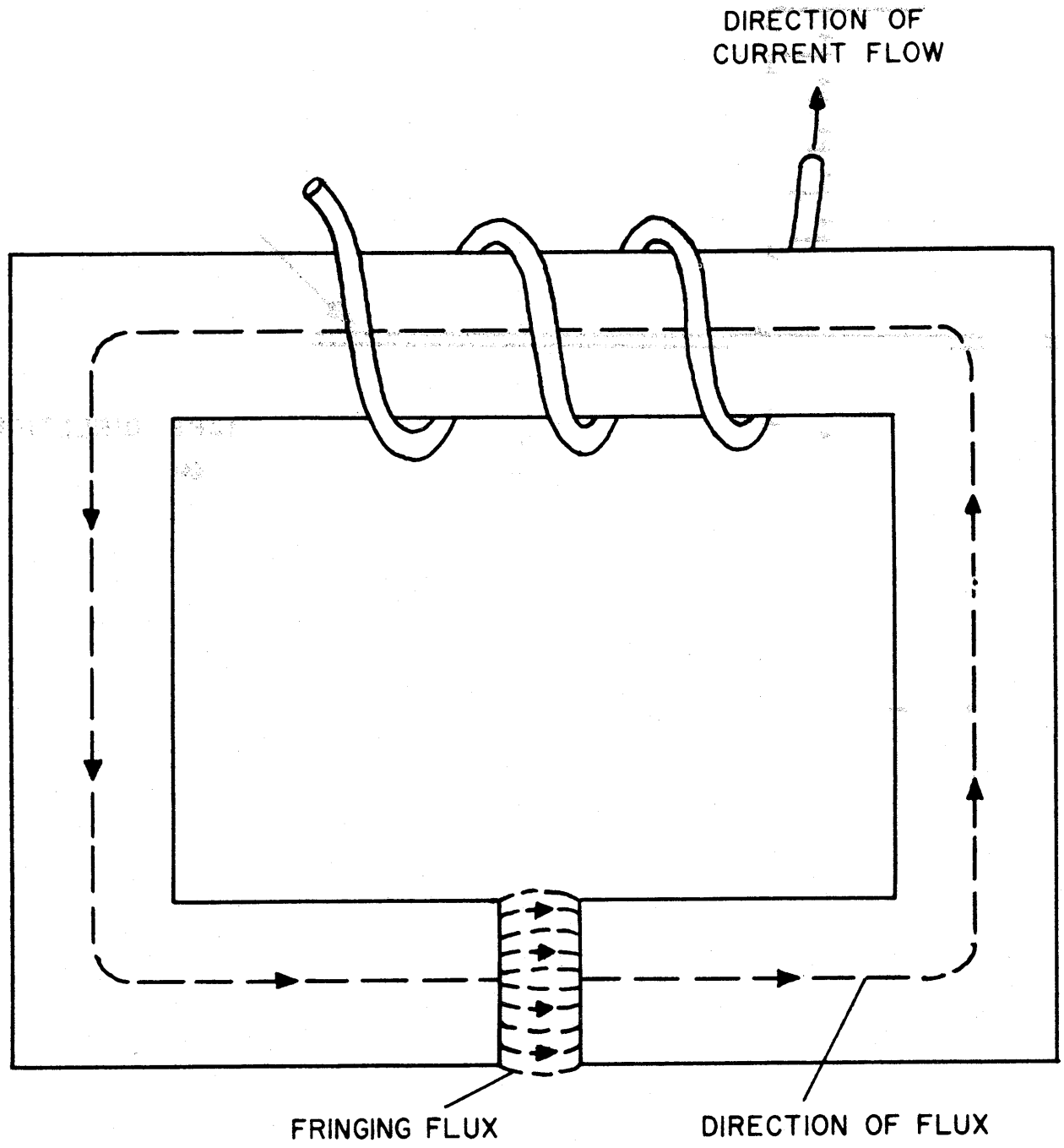


Figure 4-12

magnetic field.

c. If a conductor is moved across a magnetic field so that the lines of force of that field cut across the conductor, then an emf is induced in the conductor. If the conductor is part of a closed electrical circuit, then current will flow in the circuit in response to the induced emf.

2.2.2 Magnetic Writing

Figure 4-12 shows a magnetic circuit comprising a rectangular iron core with a solenoid wound on one leg and an air gap in the opposite leg. When current is driven through the solenoid, flux is set up in the core as shown in the figure. Notice that flux passes through and fringes around the air gap. With a very small gap, fringing flux is reduced to a minimum. However, as the gap is enlarged, fringing flux spreads to cover larger area.

If the gap in the rectangular core is placed adjacent to a piece of magnetic material, fringing flux passes through the magnetic material by virtue of the fact that it offers a lower reluctance than the surrounding air. Thus the magnetic material is placed in a strong magnetic field so that its molecules align themselves to the lines of force of the field. If the material has a squarish hysteresis loop characteristic, it retains the magnetism produced by the fringing flux even after the exciting current producing that flux has been removed. Thus a magnetic spot has been impressed or written on the magnetic material. The direction of the flux in this spot or impression depends upon the direction of the fringing flux around the air gap of the core. Since the fringing flux can be reversed by reversing the direction

of current through the solenoid, impressions can be written on a spot adjacent to the air gap in either one of two opposite directions. Remanent flux in one direction can be interpreted as a 1, while remanent flux in the other direction can be interpreted as a 0. Thus binary information can be written upon localized spots of a magnetizable medium. A magnetic circuit such as that shown in Figure 4-12 is called a writing head when it is used to impress information upon a magnetic medium.

2.2.3 Magnetic Reading

In the preceding section it has been shown that a spot of magnetism can be produced on a magnetic medium adjacent to an air gap in a magnetic circuit. This magnetic circuit comprising a rectangular core with an air gap in one leg and a solenoid wound around the opposite leg is shown in Figure 4-12. Assume that this circuit has been used to impress a state of magnetism on an adjacent magnetic medium. Then this magnetized spot in turn induces flux in the magnetic core and in the coil. However, once established, this flux remains constant so that it does not induce any voltage in the coil. On the other hand, if the magnetized spot is moved relative to the magnetic circuit then the flux in the coil varies in intensity, thus inducing a voltage in the coil. As any particular magnetized spot approaches the air gap, the field it produces in the magnetic circuit expands cutting across the coil in one direction; as the spot departs from the air gap, the field contracts cutting across the coil in the opposite direction. Thus a complete sine wave of voltage is induced in the coil as the spot approaches and departs from the air gap. The phase of this sine wave is a function of the

polarity of the magnetized spot; that is, it depends upon whether the spot is magnetized in the direction which represents a 1 or in the direction which represents a 0. The induced voltage, then, provides a reading of the magnetic state of any spot which passes the air gap. A magnetic circuit such as that shown in Figure 4-12 is called a reading head when it is used to sense information stored on a magnetic medium. Notice, that the passage of the magnetized spot past the air gap does not affect the magnetic state of the spot. Thus, this manner of reading from a magnetic medium is said to be nondestructive.

2.2.4 Magnetic Tape

Magnetic tape provides a convenient medium for the long-term storage of large blocks of information. Information is recorded on the oxide-coated plastic tape in the form of small magnetized areas, each area containing one bit. A reproducing head is usually placed in very close proximity to the tape surface, and when a positive pulse is passed through the winding of the magnetic head the molcules in the area are aligned in one direction. This magnetic impression on the tape surface represents the binary bit 1. If a negative pulse is passed through the winding of the magnetic head than a magnetic impression is recorded on the tape in a reversed direction. This represents the binary bit 0.

Three methods of magnetic tape recording are in use today. They are as follows:

a. The perpendicular method which applies the magnetizing

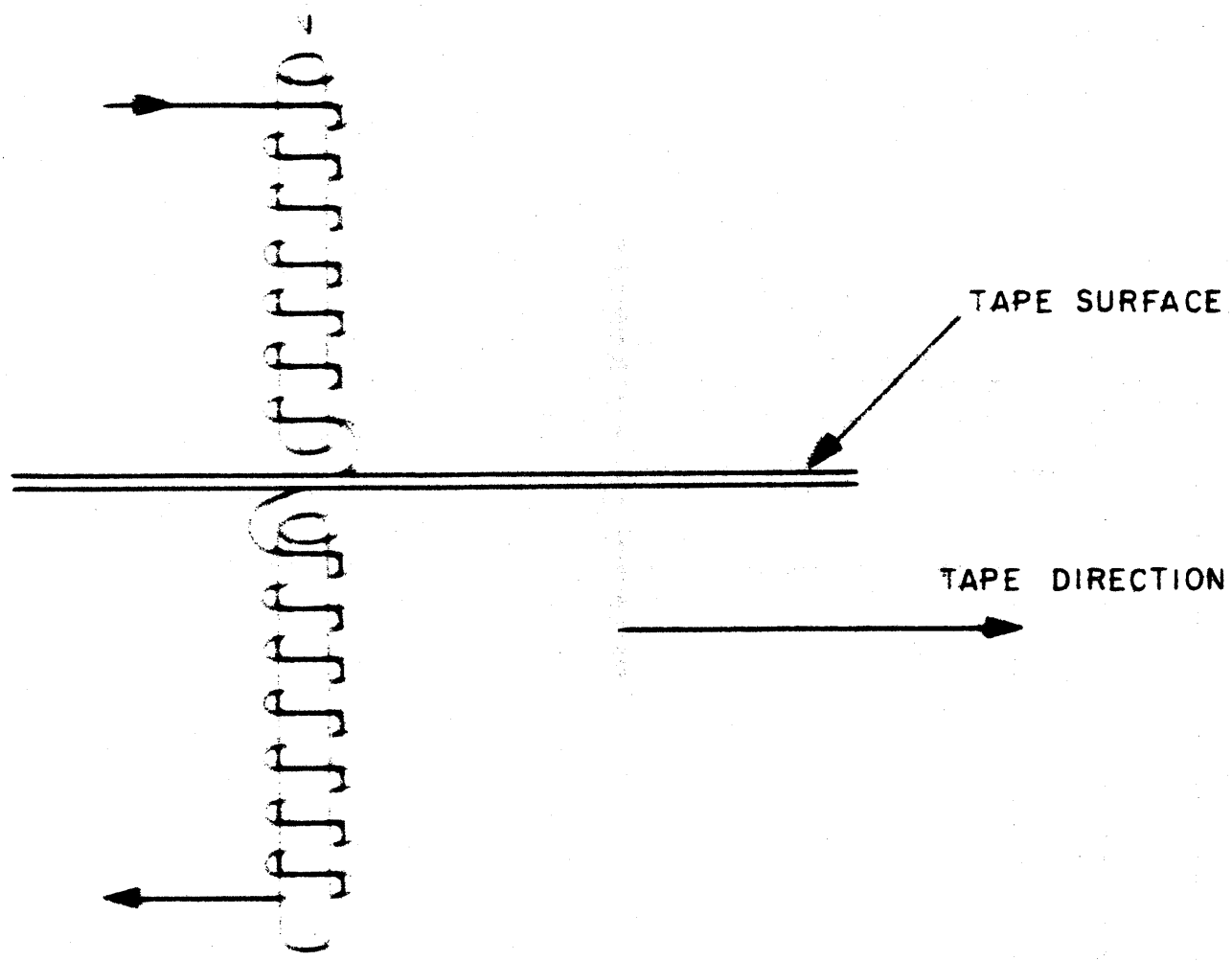


Figure 4-13

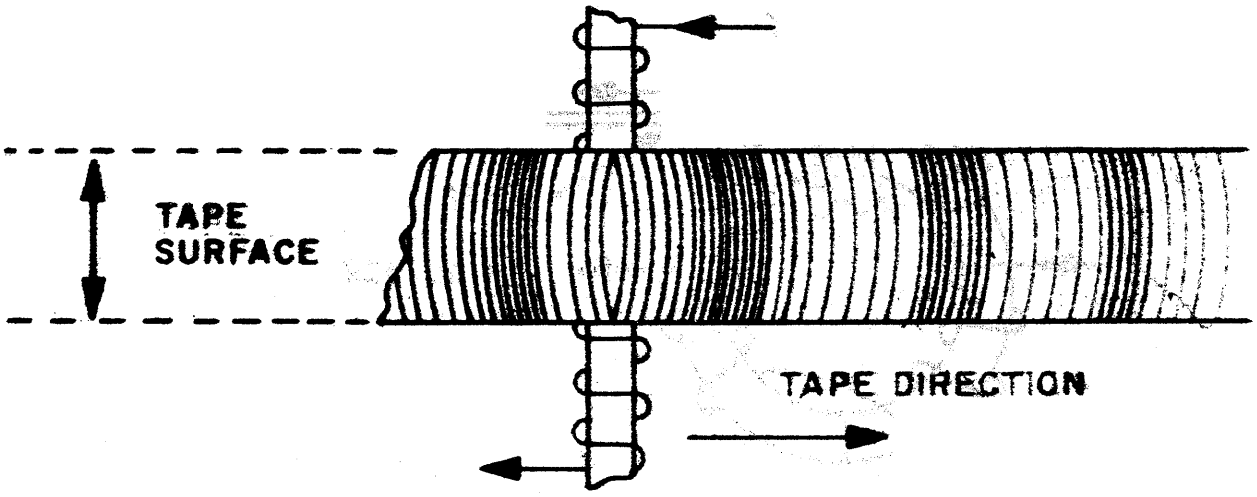


Figure 4-14

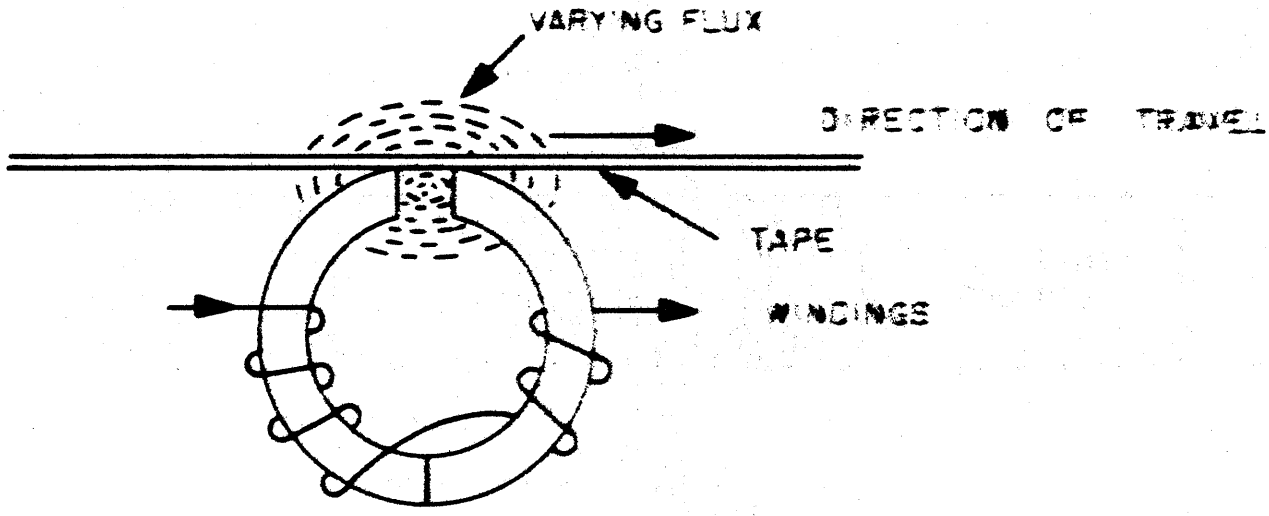


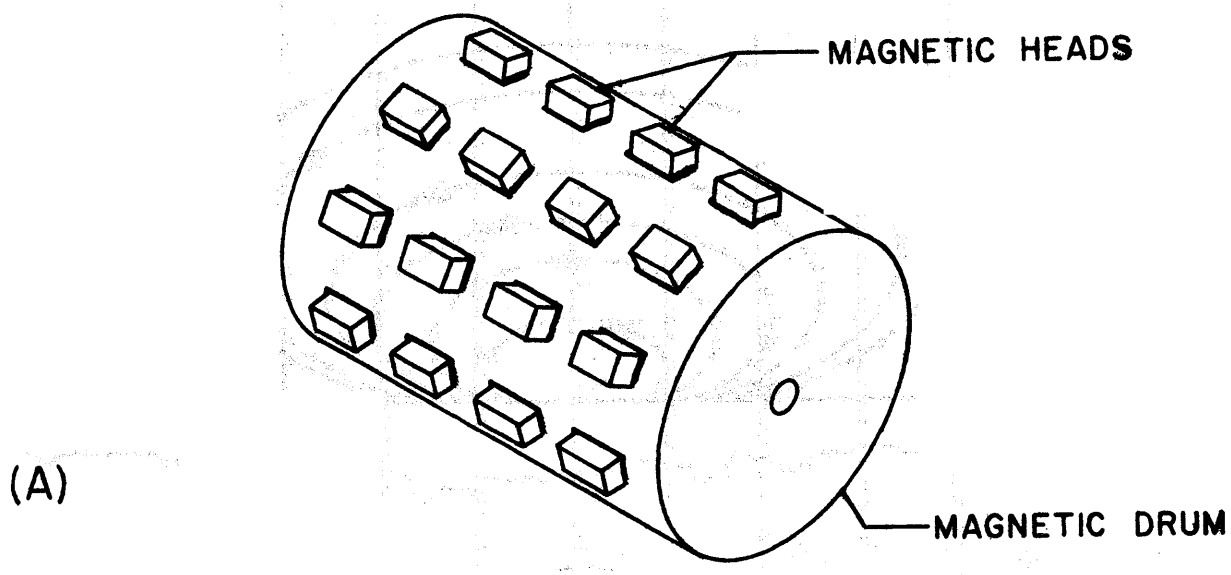
Figure 4-15

flux at right angles to the motion of the tape surface as shown in Figure 4-13. One side of the tape has a north polarity and the other side a south polarity. Such a tape is a permanent magnet in strip form.

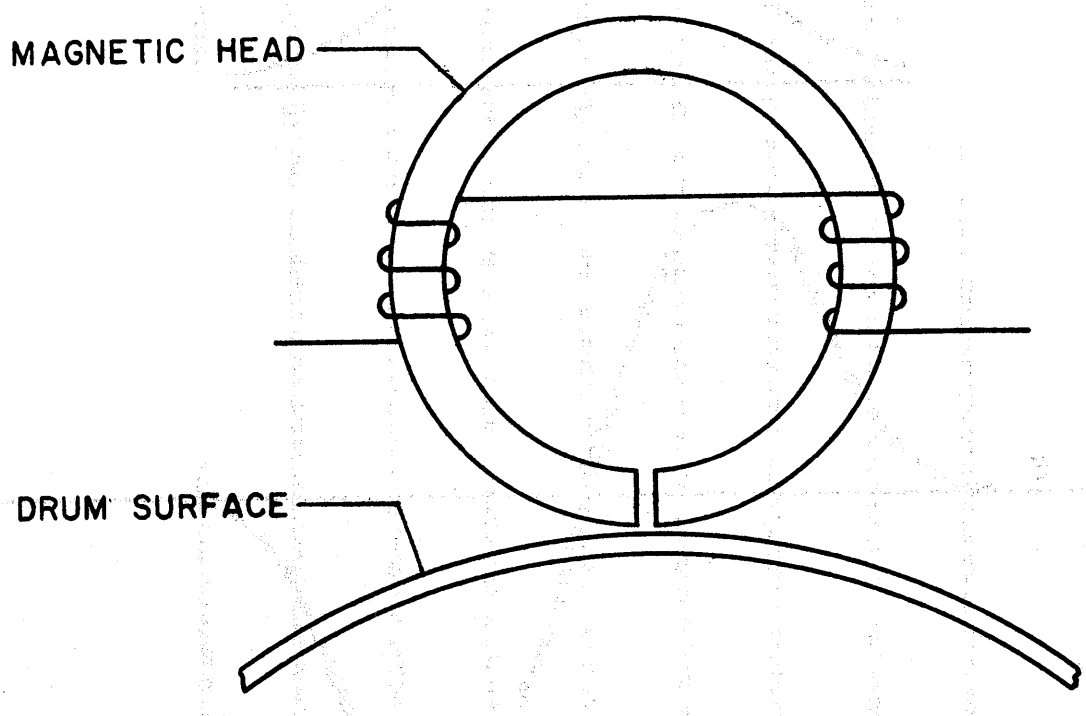
b. The transverse method (Figure 4-14) in which the pole pieces are placed at opposite edges of the tape rather than perpendicular to its surface. One major difference between transverse and perpendicular recording is the greater distance between recording head poles in the transverse method. Therefore, the transverse method requires a greater magnitude of the modulating signal.

c. The longitudinal method where magnetization is parallel to the motion of the tape, as shown in Figure 4-15. This is the method used today in most computer tape storage devices.

When a large quantity of information is to be stored, and a relatively long time (seconds) for its access is permissible, then magnetic tape provides a reliable means for storage. As many as six or more channels across a quarter-inch width of tape, and 100 magnetized spots to an inch of length is a realizable objective. Because tape may be of indefinite length and may be easily loaded onto and unloaded from writing and reading device it is an excellent medium for storing large quantities of information (provided that rapid access to that information is not required). Another advantage is that information read-out from tape does not destroy the information on the tape. Magnetic tape is also very durable, because the reading and writing



(A)



(B)

Figure 4-16

heads of this system are never in direct contact with the tape itself.

One of the greatest disadvantages of magnetic tape storage is the difficulty involved in making corrections or additions to stored information. It is uneconomical to re-write all the information on a length of tape in order to make additions or changes. Blank spots can be left between original entries in the file to accommodate additional information. However, if these spots are not used at a later date, and there is no assurance that they will be, this can prove to be a most wasteful arrangement.

2.2.5 Magnetic Drums

As previously pointed out, recording on tape has a decided advantage for certain applications. However, when information is to be written, read, and erased at frequent intervals, and in random order the magnetic drum provides much faster access. The magnetic drum is a form of cyclic storage device which is particularly adaptable for use with the larger, intermediate speed memory of present day computers. The drum is a rotating cylinder which is made in various sizes. One such drum which can store 16,384 numbers of 30 binary digits each, measures about 34 inches in diameter and is 10 inches long. Drums rotate at different speeds ranging from about 1,800 to about 7,200 rpm.

The drum is usually constructed of brass or aluminum with a surface coating of a magnetizable compound. One or more reproducing heads are placed in very close proximity to the surface of the drum as illustrated in Figure 4-16.

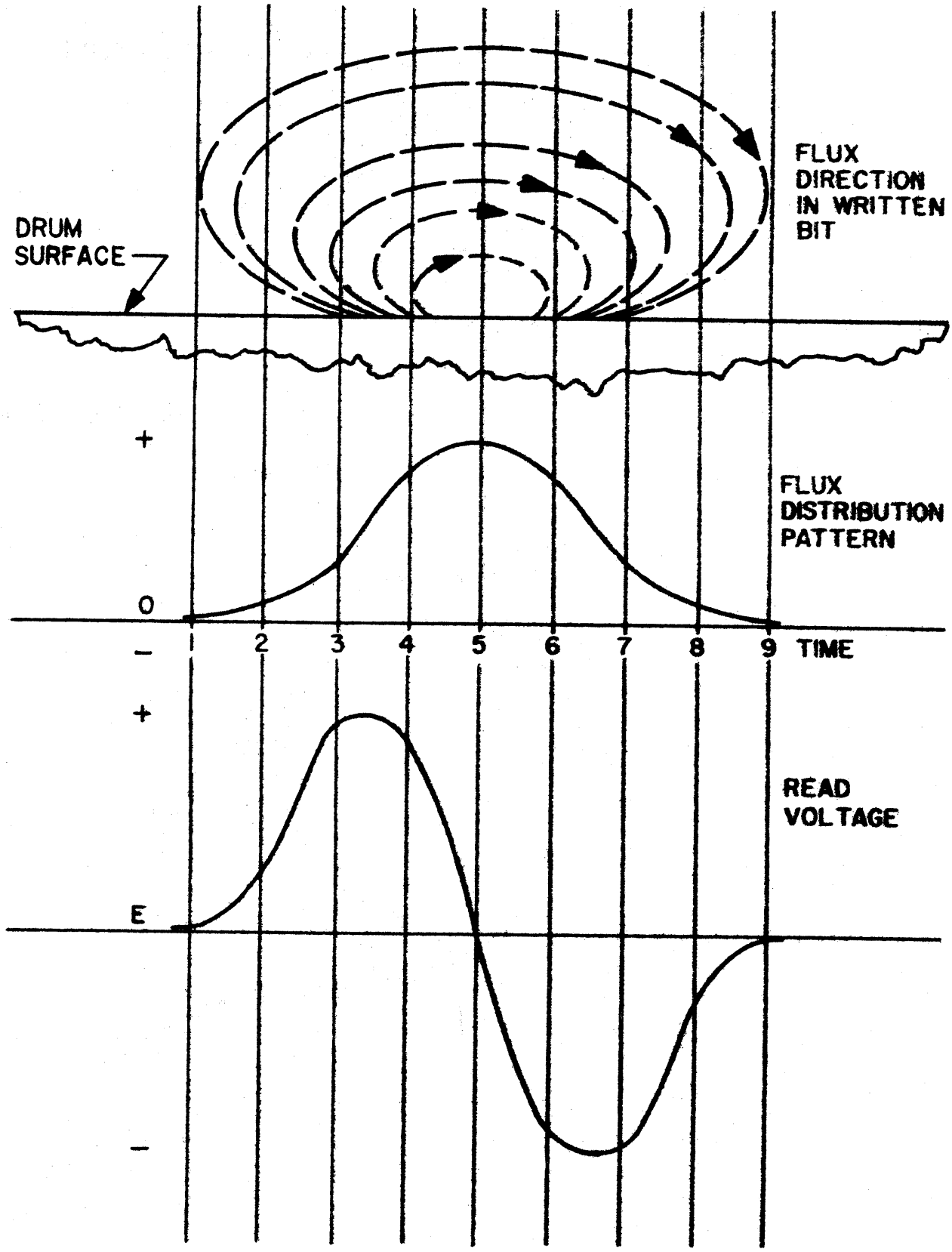

FLUX
DIRECTION
IN WRITTEN
BIT
DRUM
SURFACE
+
FLUX
DISTRIBUTION
PATTERN
0
−
1
2
3
4
5
6
7
8
9
TIME
+
READ
VOLTAGE
E
−

Figure 4-17

If the drum surface is completely demagnetized and direct current flows through a drum head, fringing flux at the gap magnetizes the spot on the magnetic drum surface that is adjacent to the gap. The process of inducing flux in the magnetic drum surface is called writing. If the magnetic drum is set in rotation, the magnetized spot remains on the drum and a voltage is developed across the coil in the head every time the magnetized spot (bit) passes under the air gap. The process of inducing voltage across the coil by moving the bit past the head is called reading. Since the magnetized spot is not affected by reading, this type of reading is called nondestructive.

The relationship of the flux distribution pattern written on the drum surface and the voltage induced in the head with respect to time is illustrated in Figure 4-17. Before the bit reaches the gap at time 1, flux (and therefore induced voltage) is zero. From time 1 to time 2, fringe flux is small and induces a small voltage. From time 2 to time 3, flux increases more rapidly, inducing a larger voltage. Time 3 to 4 represents the greatest area of flux change and consequently the period of maximum induced voltage. From time 4 to time 5 the rate of flux change is smaller, although the amount of flux continues to increase until it reaches a maximum value at time 5. Since the rate of flux change from time 4 to time 5 is approximately similar to that from time 2 to time 3 (although in the opposite direction), the voltage induced is approximately equal.

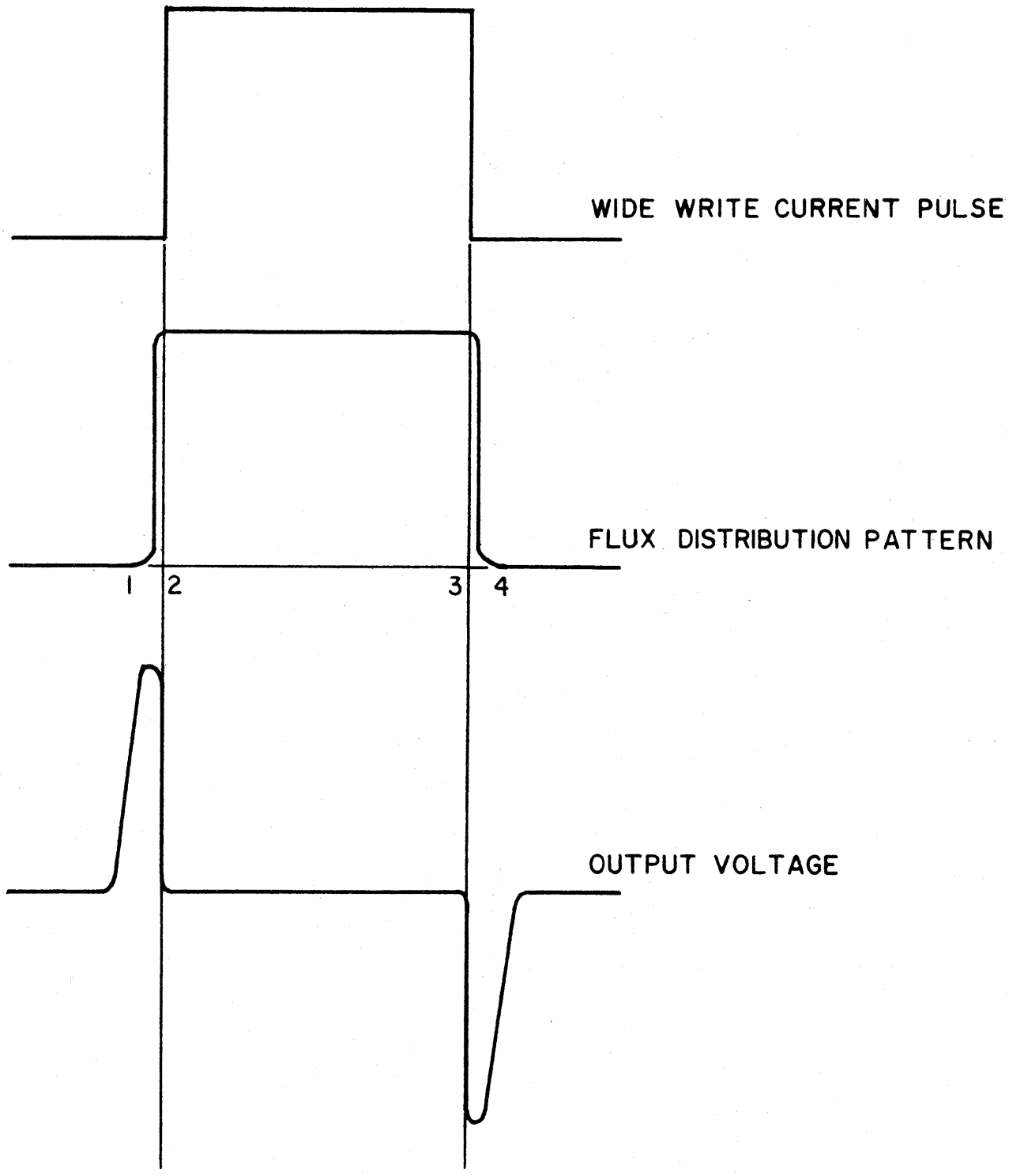

WIDE WRITE CURRENT PULSE
FLUX DISTRIBUTION PATTERN
1
2
3
4
OUTPUT VOLTAGE

Figure 4-18

Preceding time 5, the maximum flux point on the bit approaches the center of the air gap and at time 5 is exactly under the gap. The rate of flux change is zero at time 5. Therefore, the voltage induced in the head is also zero. Time 5 is also the voltage polarity crossover point. From time 5 to time 9 (as the magnetized bit recedes from the gap) flux decreases, producing a voltage similar to that produced from time 1 through time 5, but opposite in polarity. If the direction of current flow through the writing head is reversed during writing, the direction of magnetizing flux is reversed and the direction of bit magnetization is also reversed. Thus, during reading, the polarity of induced voltage at all points is the reverse of that shown in Figure 4-17.

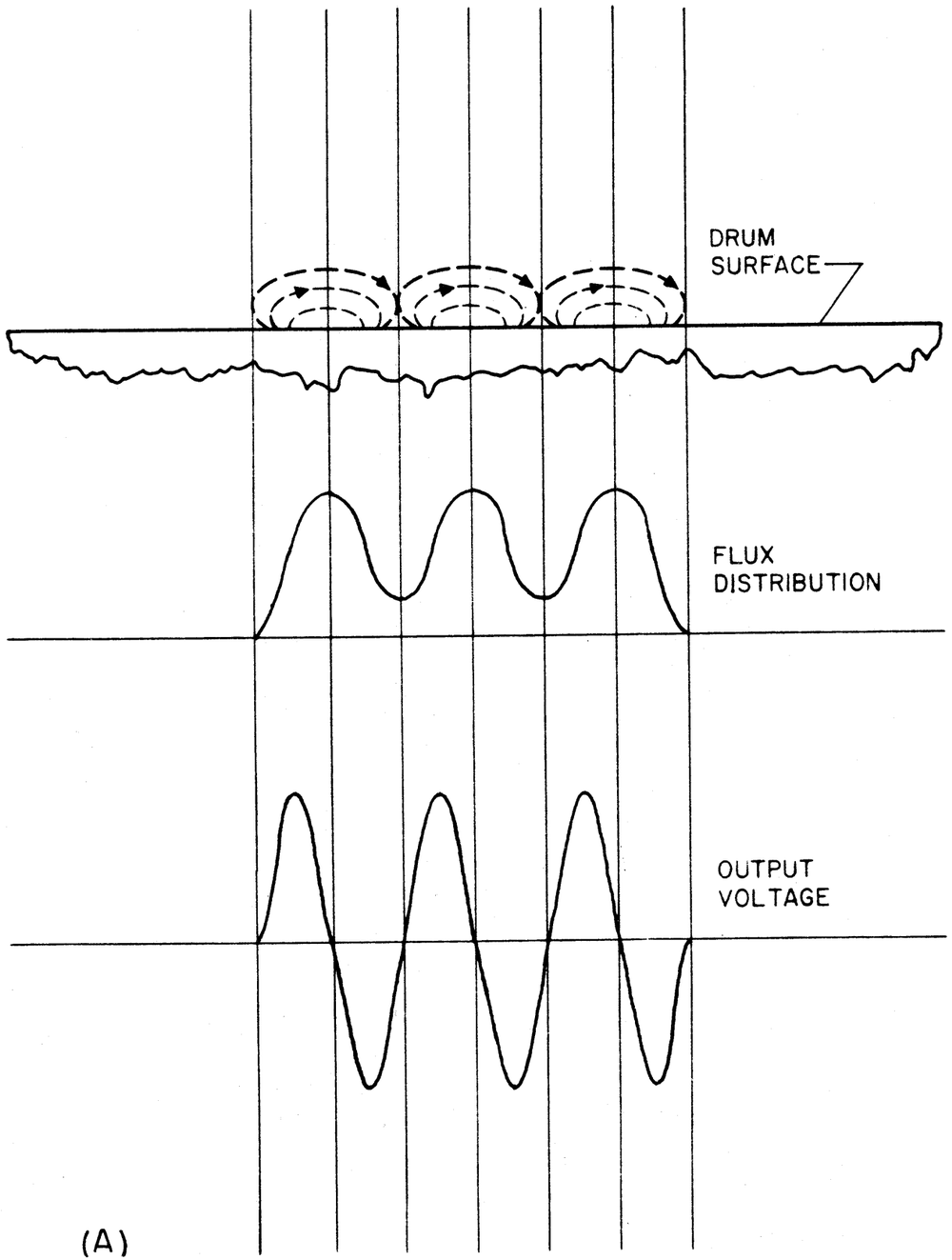
Both reading and writing are accomplished while the drum rotates at constant speed. However, with direct current flowing through the drum head while the drum rotates, a complete strip (channel) passing around the drum circumference under the drum head is magnetized by fringing flux. Use of brief current pulses (which produce magnetization of only a small area) instead of direct current prevents this total magnetization of a channel.

Figure 4-18 illustrates that the current pulse (shown as a square wave) produces a flux distribution pattern on the rotating drum surface that starts at time 1 and ends at time 4. Flux distribution from time 2 time 3 is of uniform amplitude. This portion of the flux pattern produces no output voltage from the drum head as it passes under the air gap, and the length of the channel between the read head output voltage pulse

is effectively wasted. If the width of the pulse applied to the drum head is reduced during writing, the flux distribution of the magnetized area on the drum approaches the flux distribution produced by writing on a stationary drum. Maximum information storage in a channel is thus made possible. A pulse width of 1.5 microseconds is used.

In addition to pulse width and drum speed, bit length is also determined by the width of the air gap in the head and by the distance between the drum head core and the magnetized drum surface. The gap width is usually on the order of 0.001 inch. Core-to-drum spacing is also set at 0.001 inch to permit optimum flux density and to maintain a safety factor that prevents the drum head from touching the rotating drum. The voltage produced by reading a bit is a function of core-to-drum spacing, write current magnitude, and the number of turns in the coil. Small core-to-drum spacings permit more fringing flux on the drum to be cut by the head. The 0.001-inch spacing represents a compromise between close and safe spacing. The magnitude of the write current pulse determines the amount of flux induced on the drum surface. This magnitude can be increased until the magnetized bit is completely saturated (maximum flux density).

Since bits are written with sufficient magnitude to produce maximum flux density in the magnetized bit area, a bit of opposite polarity can be written over an existing bit by sending a current


DRUM
SURFACE
FLUX
DISTRIBUTION
OUTPUT
VOLTAGE
(A)

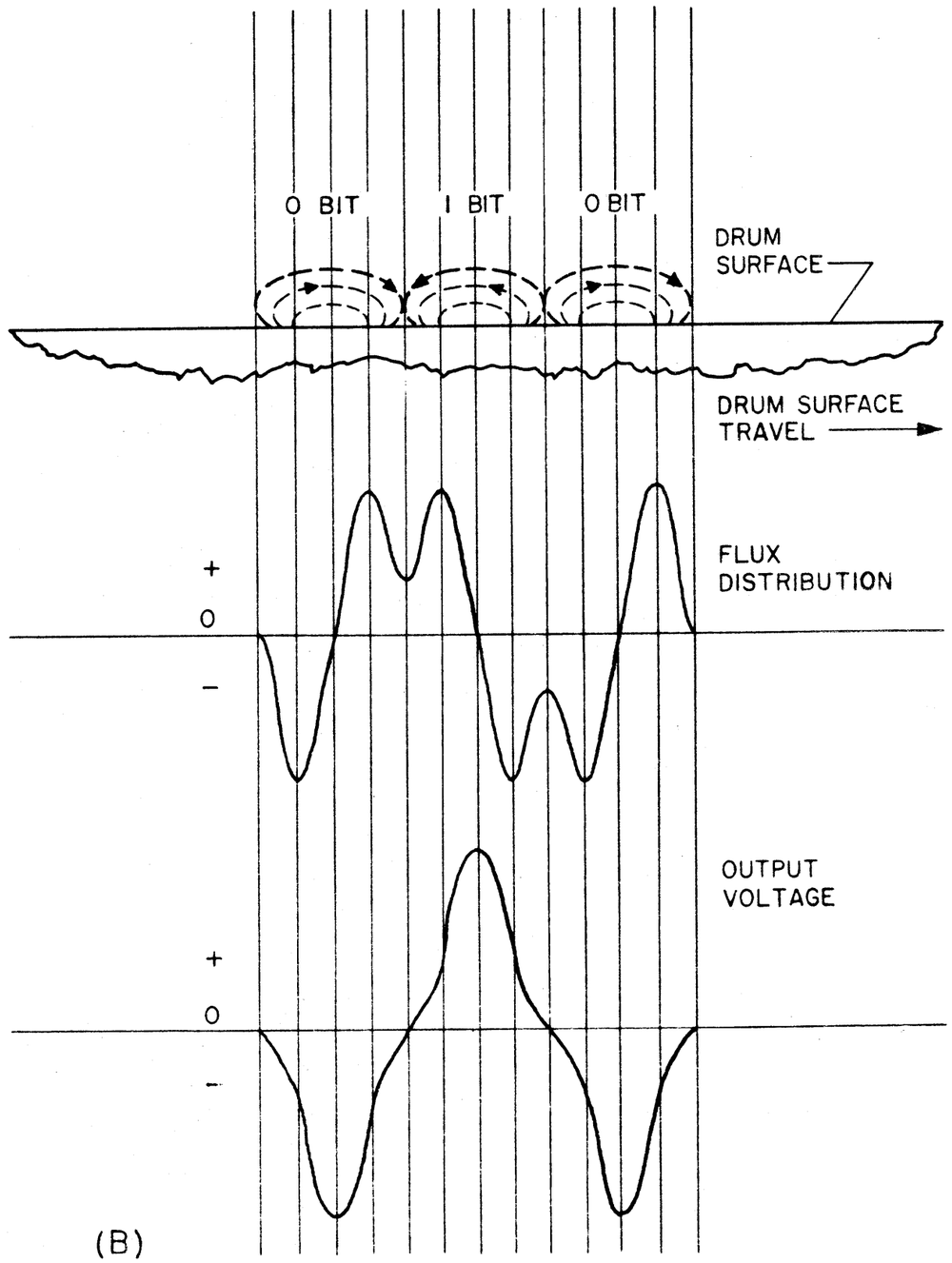

0 BIT
1 BIT
0 BIT
DRUM
SURFACE
DRUM SURFACE
TRAVEL
+
0
−
FLUX
DISTRIBUTION
OUTPUT
VOLTAGE
+
0
−
(B)

Figure 4-19

pulse of the same magnitude but opposite polarity through the drum head. It has been found that a write current on the order of 110 milliamperes is sufficient to produce bit staturation but prevent spread of flux into adjacent signal areas. As the number of turns on the read coil is increased, the additional turns are cut by the flux induced in the head. The result is an increased output voltage.

2.2.5.1 Reading and Writing Waveforms

When 0 bits are written on a drum, each bit produces a flux pattern similar to that shown in Figure 4-17. Writing 1 bits produces a flux pattern that is similar but of opposite polarity to the 0 bit pattern. The read head output voltage produced by both bits starts a zero at the beginning of the bit and falls to zero at the end of the bit.

As the density of recorded information increases on a drum (spacing between adjacent bits decreases) the spread of the fringing flux from one bit may interfere with the flux pattern of an adjacent bit area. This interference changes the waveform of read head output voltages. The amount of change depends upon the sequence of bits. When 1 or 0 bits are written in close sequence, flux lines of each successive bit oppose the flux lines of the preceding bit. As a result, flux at the end of one bit does not fall to zero, but rises again in the same direction as the next bit passes under the read head. This sequence produces the flux distribution pattern shown in Figure 4-19-(a). A sine wave output voltage is

produced when the varying flux pattern of these bits passes under the read head.

The flux pattern illustrated in Figure 4-19-(b) represents a 0-1-0 bit sequence. The read head output voltage of the first 0 bit, as it passes under the read head, rises to a maximum negative value, decreases to zero, rises to a maximum positive value, and starts to fall to zero. For an isolated bit, the voltage drops all the way to zero, but when a 1 bit follows, the flux lines of the adjacent 1 bit add to and interfere with the flux lines of the 0 bit. As a result, the rate of flux decrease accelerates, giving rise to an increase in voltage. The first part of the 1 bit then passes under the drum read head. Since the flux of the 1 bit is opposite in phase to that of the 0 bit, the rate of flux increase accelerates and the corresponding read head output voltage rises to the same point that was reached during the decreasing half of the 0 bit.

2.2.5.2 Theory of Erasing

The entire drum surface is erased by applying an a-c field to the drum surface. This field initially saturates the drum surface and then is reduced to zero. At the beginning of the erasing procedure the a-c field produces flux densities at the drum surface that are stronger than the flux densities produced by the write pulses. The varying polarity of the a-c field causes a varying polarity of the flux induced on the drum surface. As the strength of the a-c field is reduced, each reversal of the a-c polarity produces an induced flux density that is lower than that of the previous

cycle. When the a-c field is finally reduced to zero, the residual magnetism in the drum produced during writhg is also zero, resulting in a completely erased drum. Erasing may be employed to rid the drum surface of noise. It is not necessary to erase the drum to change the information in its registers; a register may be changed by writing the desired word over the old one.

2.2.6 Magnetic Cores

2.2.6.1 General

The requirement for greater speed of access to stored information in a computer memory has led to the development of magnetic cores. Two types of magnetic cores are in use today. These are ferrite cores which are used solely to perform the storage function, and tape cores which are used as drivers (for writing on the ferrite cores) and to form shift and counting registers. Although the tape cores are not primarily storage elements they are covered here because of their functional similarity to the ferrite cores and because of their use as drivers in connection with the ferrite cores.

A ferrite core is a single piece of magnetic material of toroidal shape. A tape core, on the other hand, is formed by wrapping a thin magnetic tape around a small bobbin a sufficient number of times to build up the required volume of magnetic material. Each turn of tape may be thought of as a single lamination, and by virtue of this essentially laminated structure, eddy currents are held to a minimum. Thus, the tape cores are suitable for operation at higher

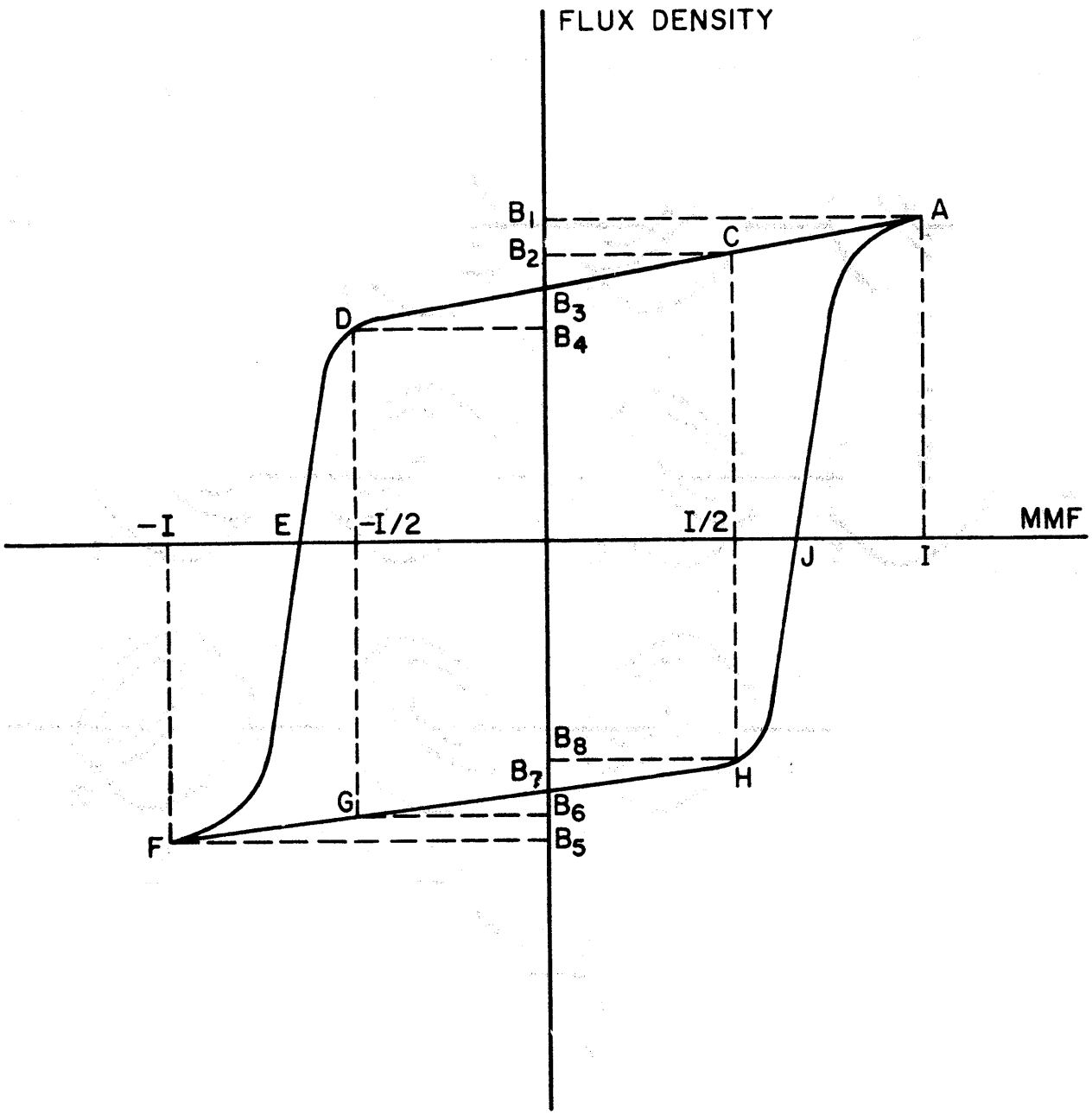


Figure 4-20

power levels than the ferrite cores in which eddy current losses would be much greater.

2.2.6.2 Ferrite Cores

Ferrites are ceramic materials possessing relatively square hysteresis loop characteristics which make them eminently suitable for use in binary storage systems. A ferrite core containing enough material to afford reliable binary storage can be made in the form of a toroid about an eighth of an inch in diameter. Such a core requires a magnetomotive force on the order of 1 ampere-turn to switch it from one magnetic state to the other.

The hysteresis loop for a typical ferrite core is shown in Figure 4-20. As indicated by point A on the loop, the mmf produced by the application of a current of magnitude, I, to a single turn winding passing through the core causes a flux density of magnitude B_1. When the current is reduced from I to I/2, the flux density decreases from B_1 to B_2 as indicated by the portion of the loop between points A and C. When the current is further reduced from I/2 to 0, the flux density is reduced from B_2 to B_3 as indicated by the portion of the loop between point C and the flux density axis. A current of I/2 applied to the winding further reduces the flux density from B_3 to B_4 as indicated by the portion of the loop between the flux density axis and point D. When the current is increased from - I/2 to -I, the flux density decreases rapidly, passes through 0 and rises in the opposite direction to a value of B_5 as indicated by the portion of the loop between D and F.

When the current is reduced from -I to - I/2, the flux density decreases from B_5 to B_6 as indicated by the portion of the loop between points F and G. When the current is further reduced from - I/2 to 0, the flux density decreases from B_6 to B_7 as indicated by the portion of the loop between point G and the flux density axis. A current of I/2 applied to the winding decreases the flux density from B_7 to B_8 as indicated by the portion of the loop between the flux density axis and point H. When the current is increased from I/2 to I, the flux density decreases rapidly, passes through 0 and rises in the opposite direction to B_1 as indicated by the portion of the loop between points H and A. Notice that the loop is symmetrical about the axis of zero flux density; that is: $B_1 = -B_5$, $B_2 = -B_6$, $B_3 = -B_7$, $B_4 = -B_8$.

If remanent flux, in the direction associated with the positive direction of the flux density axis, is defined to represent 1 and remanent flux in the opposite direction is defined to represent 0 and if a flux density of magnitude $B_4 = -B_8$ is sufficient to afford reliable operation, then the following statements can be made:

a. A 1 can be written on the core by the momentary application of a current, I. Moreover, the core will continue to store this 1 even if a reverse current of - I/2 is passed through its winding. This latter fact is extremely important since most schemes for selecting particular locations in a core storage device depend upon it.

b. A 0 can be written on the core by the momentary application

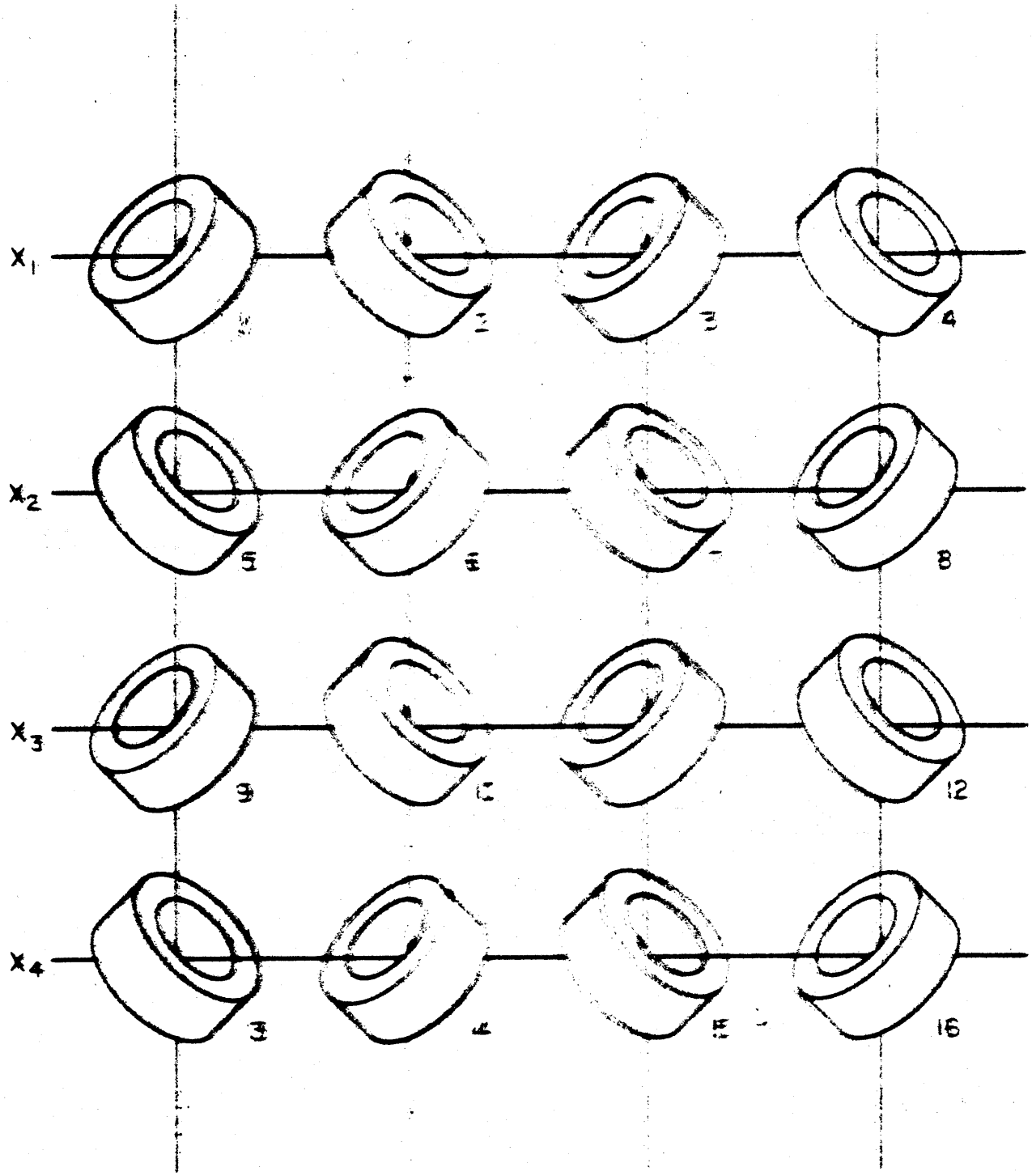


Figure 4-21

of a current, - I. Moreover, the core will continue to store this 0 even if a reverse current of I is passed through its winding.

When the direction of the remanent flux in a core is reversed between the state representing a 1 and the state representing a 0, the core is said to be switched. The driving current, I, required to switch a core is not actually supplied to a single winding. Instead, I/2 is supplied to both an X and a Y winding which pass through the core. The mmf's produced by these two so-called half-currents add, providing an mmf sufficient to switch the core. This configuration provides the means for selecting particular cores in an array. For example, a two-dimensional array of cores can be organized on the basis of rows of cores having their X windings connected in series and columns of cores having their Y windings connected in series. Such a two-dimensional core array or memory plane as it is sometimes called is shown schematically in Figure 4-21. Referring to the figure, assume that all sixteen cores are storing 0's. Assume further that a current pulse of I/2 is applied to the X_1 line and the Y_3 line. Then, both the X and Y windings of core #3 will pass currents of I/2. The mmf's produced by the two current will add, producing a total mmf sufficient to switch the core to the state representing a 1. In terms of the hysteresis loop of Figure 4-20, the remanent flux density of core #3 will be driven through the portion of the loop from the flux density axis through H, through J to A. When the current pulses die out, the value of the remanent flux density of core #3 will

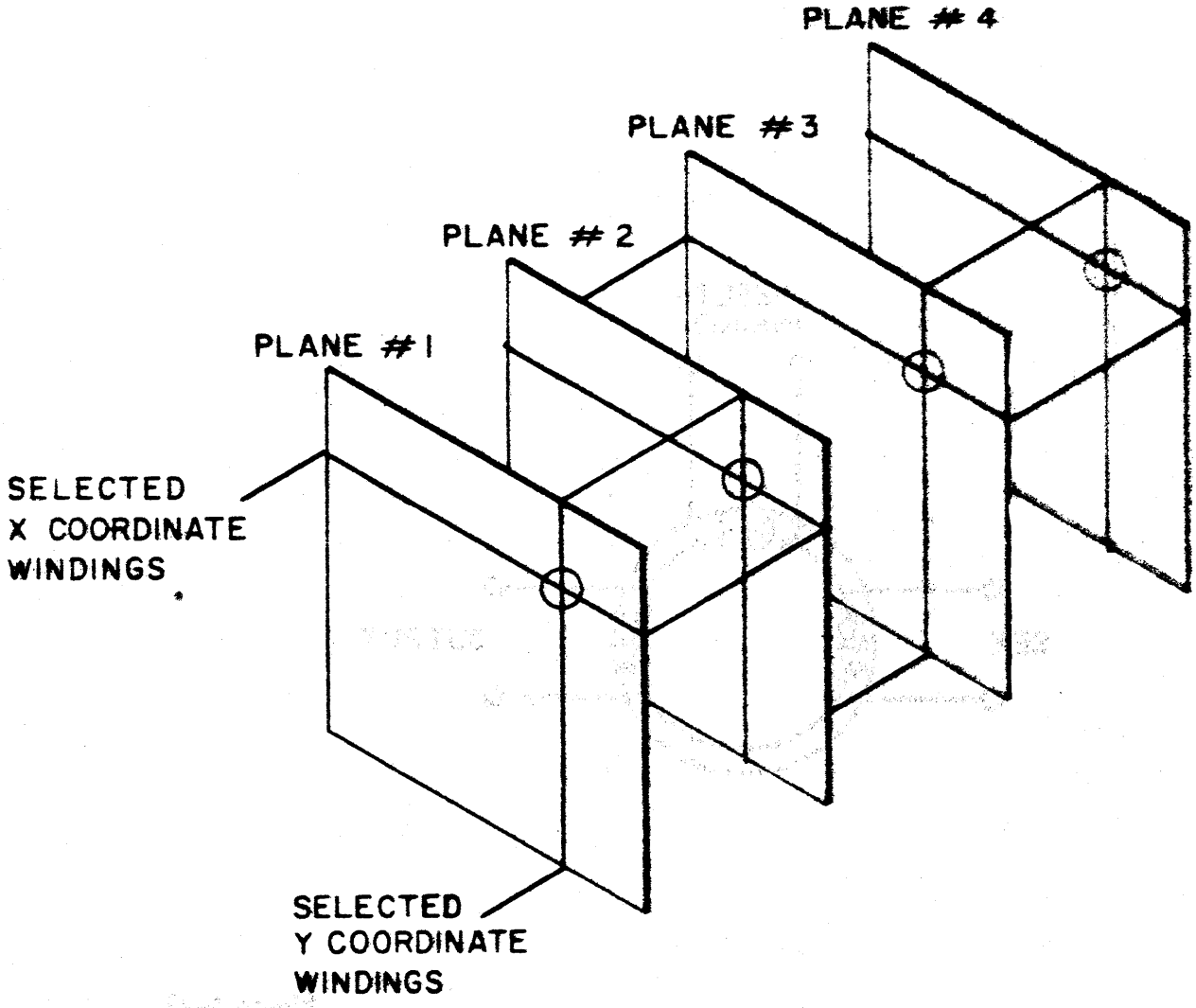

PLANE # 4
PLANE # 3
PLANE # 2
PLANE # 1
SELECTED
X COORDINATE
WINDINGS
SELECTED
Y COORDINATE
WINDINGS

Figure 4-22

fall to B_3 which is more than sufficient to represent a 1. Under these circumstances, core #3 is said to have been fully selected. At the same time, the X windings of cores #1, #2 and #4 and the Y windings of cores #7, #11 and #15 pass currents of magnitude, I/2. In each of these cores the remanent flux density decreases from a magnitude of B_6 to a magnitude of B_7 (as indicated by Figure 4-20) and then rises again when the I/2 current pulses die out. Thus, these cores, which are said to be half selected, continue to store 0's.

Memory planes are stacked to form three-dimensional arrays such as the one shown schematically in Figure 4-22. In the three-dimensional array, each plane is associated with a different bit position. Thus the array of the figure is capable of storing four bit words, where the first bit of any word is stored on a core which is a member of memory plane I, the second bit is stored on a core which is a member of memory plane II, the third bit is stored on a core which is a member of memory plane III and the fourth bit is stored on a core which is a member of memory plane IV. A storage address in a three-dimensional array comprises a set of cores occupying the same junction of row and column in each of the memory planes of the array. Thus for example, the set of cores which are in row 5 and column 7 of their respective planes constitute a location or register. The windings of corresponding X rows in all the planes of an array are connected in series; similarly the windings of corresponding Y columns are connected in series. The result of these connections is that the

application of a pulse to the X windings of one row and to the Y windings of one column fully selects all the cores of a register. If the direction of the current pulses is such as to set up a magnetic field in the direction which is associated with a 1, then a 1 is written in each bit position of the register. Since a word, in general, contains both 1's and 0's, it is obvious that some method must be provided for writing 0's as well as 1's into a selected storage register. The method used to accomplish this is to inhibit the writing of 1's into those bit positions where 0's are to be stored. An inhibit winding associated with each core performs this function. When a word is written into the core array, the inhibit windings of those orders which are to store 0's are supplied with inhibit current pulses at the same time that the half-write pulses are supplied to the X and Y windings. The inhibit pulses being of half-write (I/2) magnitude and being opposite in polarity to the half-write currents, cancel half the effect of the full selection. Thus the cores receiving inhibit pulses are not switched. Thus, if all the cores of a register are set to 0 prior to the writing of a word into the register, then the selection of a register by means of X and Y half-write pulses will cause 1's to be written into those bit positions which do not receive simultaneous inhibit pulses, while those bit positions which do receive inhibit pulses will continue to store 0's. Since each plane is associated with a particular bit position, all the inhibit coils of the plane may be connected in series. An inhibit pulse will, then, half-select all the

cores of a particular plane. However, this will affect only the single core which is simultaneously being fully selected by pulses supplied to the X and Y windings.

A fourth or sense winding is associated with each core. This winding is used in the process of reading information out of the core array. In the read operation, a core location is selected by applying half-read currents to the appropriate X and Y windings. Half-read currents are equal in magnitude but opposite in polarity to half-write currents. Thus, when a register is selected by half-read currents all its cores are driven to the magnetic state representing 0. For those cores representing 0 this implies no change of magnetic state. However, for those cores which are storing a 1, it does imply a reversal of magnetic state. In the course of this magnetic reversal, magnetic flux lines cut across the sense winding inducing a voltage pulse in it. Thus a word is read out of a register in a core array by selecting the register with X and Y half-read pulses and sampling the outputs from each of its sense windings. Since only one register is selected at any one time, all the sense windings of a particular plane can be connected in series so that a single output line serves each plane.

Notice that reading a word out of a core memory register clears that register. For this reason, core memory readout is said to be destructive. Since it is usually desirable to retain in a register a copy of the word that is read out of it, means must be provided to rewrite each word in a core memory

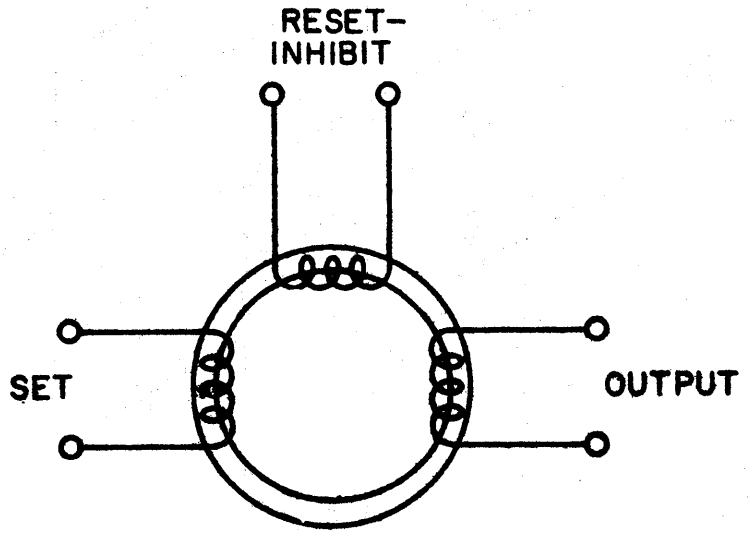


Figure 4-23

immediately after it has been read out.

2.2.6.3 Metallic Tape Cores

2.2.6.3.1 General

There are two principal types of tape materials used in metallic tape cores. One of these is a 50 percent nickel-iron alloy. A second one is molybdenum permalloy which is used to a greater degree than the other. The hysteresis loop of this material is not so rectangular as that of the nickel-iron alloys, but its pulsed characteristics are much better. The permalloy is a faster switching material than the other alloys, because it lacks certain time delays which occur in the magnetization of the other materials. The permalloy also requires a lower coercive force and is available in thinner tapes.

The metallic tape cores have magnetic properties which allow them to maintain one of two states of remanent flux as is in the case of the ferrite cores. These states of remanence can be assigned the significane of 1 and 0.

2.2.6.3.2 Core Current Drivers

Tape cores can be used to generate the half-write (I/2) currents required to set ferrite cores. When serving this function they are referred to as core current drivers. A tape core used as a core current driver has three windings; a Set winding, Reset-Inhibit winding and an output winding as shown in Figure 4-23. If the remanent flux in the core is in the direction associated with 0 and a current of I is applied to the Set winding, then the core is switched to the state repre-

senting a 1. Similarly, if the remanent flux in the core is in the direction associated with a 1 and a current of - I is applied to the Reset-Inhibit winding, then the core is switched to the state representing a 0. If current of I and - I are simultaneously applied to the Set and Reset-inhibit windings of a core when the resultant MMFs produced cancel each other and the core is not switched. When the core is switched, a voltage is induced in the output winding. It is this voltage which is used to drive a half-write current through a set of X or Y windings associated with a group of ferrite cores.

The Set and Reset-Inhibit windings of the tape cores provide the means for writing a word onto a selected set of such cores. Each of the cores in the set which is to receive the word is selected by applying a current of magnitude, I, to its Set winding. Simultaneously, those cores which are to store 0's of the word are inhibited from switching to the 1 state by the application of current pulses of magnitude - 1 to their Reset-inhibit windings. This implies that all the cores of the set must first be cleared by an application of current pulses of - I to their Reset-Inhibit windings. When tape cores are used as ferrite core current drivers, a word to be written into the ferrite core array is first written onto two sets of tape cores, one which drives a group of X windings and the other of which drives a set of Y windings. The word is then transferred from the tape cores to the ferrite cores by pulsing the Reset-Inhibit windings of the tape cores. Thus, the tape cores serve as power amplifiers and at the same time function as components in the selection matrix by means of which a word is transferred to a particular core

memory register. They also perform an incidental storage function; that is, a word is first written on the tape cores and then read from the tape cores onto the ferrite cores.

2.2.6.3.3 Tape Core Shift Registers

Tape core shift registers can be designed for either parallel or serial entry of information and for either parallel or serial output of information. Parallel-entry, serial-output shift registers are used to convert from parallel to serial operation while serial-entry, parallel output shift registers are used to convert from serial to parallel operations. Serial-entry, serial-output shift registers are used as counters or as delay devices.

The serial-entry, serial-output registers are used for counting, as follows: A 1 is entered at the left-hand end of the register and is shifted to the right one place for each shift pulse received. The number of shift pulses required for the 1 to travel through the register corresponds to the number of orders (i.e. cores) in the register. Thus, the register counts some predetermined number of shift pulses. It is equally true to say that the shift register delays the 1 pulse for n shift pulse repetition intervals, where n is the number of orders in the register. Thus the shift register acts as a delay device.

The parallel-entry core shift register is used to convert a word from parallel form to serial from. The bits of the word are read into the core register simultaneously and

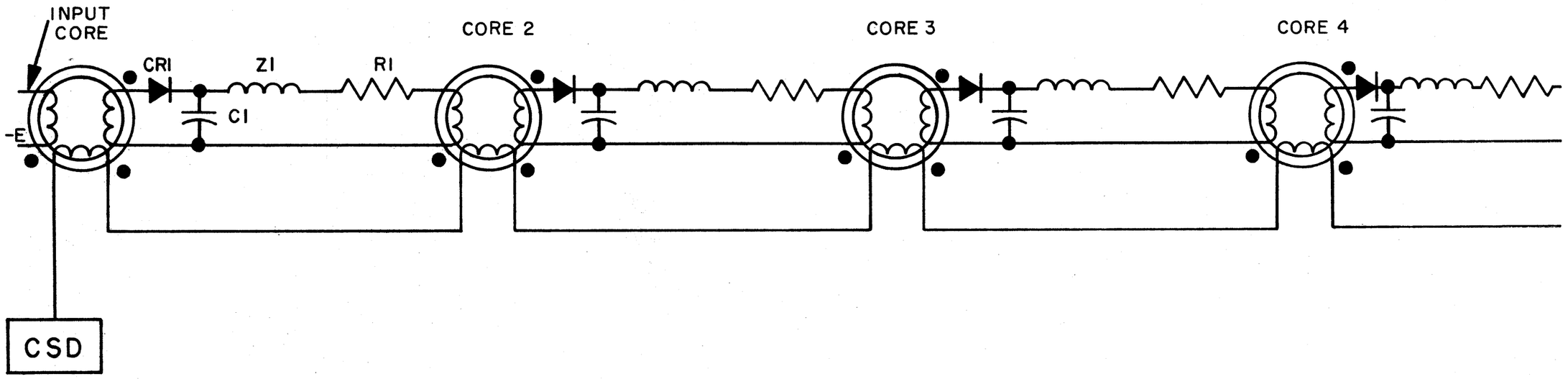


Figure 4-24

are then shifted out of the right-hand end of the register.

The serial-entry, serial output type register, which is shown schematically in Figure 4-24, is comprised of a set of tape cores each of which has three windings; an input winding, a shift drive winding, and an output winding. The output winding of each core in the register (with the exception of the core at the right-hand end of the register) has its output winding coupled to the input winding of the core on its right. The phase relations between windings are such that as a given core reverses its magnetic state from 1 to 0, it induces a pulse on its output winding which writes a 1 into the core on its right. Information is read into the circuit by application to the input winding of the left-hand core of the register. It is shifted to the right in the register by means of a shift pulse which is applied to the series connected shift drive windings of all the cores of the register. The shift pulse drives all the cores to 0. Those cores which are storing 1's produce pulses on their output windings by virtue of the reversals of magnetic state which they undergo. On the other hand, no such pulses are produced by those cores which are storing 0's, since their magnetic states do not change. The coupling networks between cores provide time delays, so that the pulses transferred from one core to the next, do not arrive until the shift drive pulse has died out. Thus these pulses are able to switch the cores at which they appear to the magnetic state representing 1. The time delay is provided by an

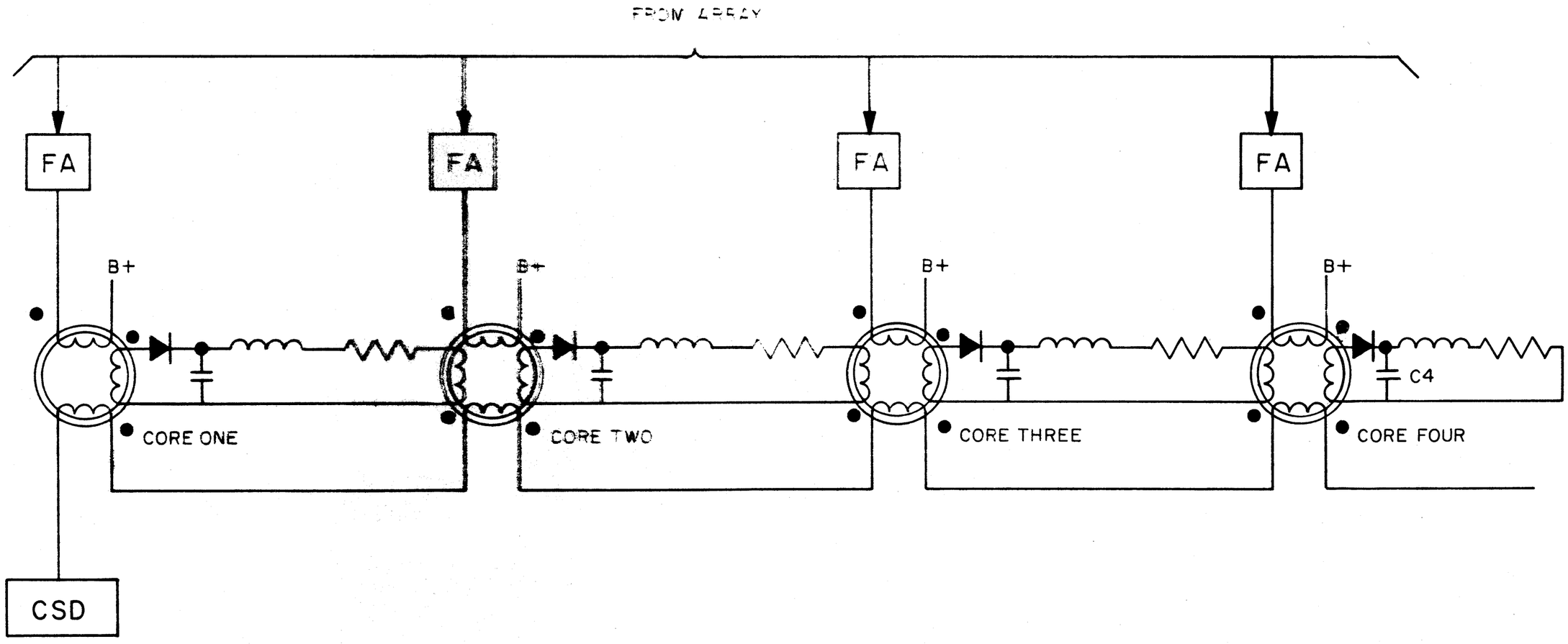

FROM ARRAY
FA
FA
FA
FA
B+
B+
B+
B+
C4
CORE ONE
CORE TWO
CORE THREE
CORE FOUR
CSD

Figure 4-25

identical network in each coupling circuit. With reference to the element numbers of the circuit between the first two cores, the time delay is obtained as follows: During the interval when the shift drive pulse is occurring and core #1 is reversing its state (here it is assumed that the core is initially storing a 1) C1 is charging, in response to the induced output pulse, through the low impedance path comprising the output winding and the low forward resistance of diode CR1. When the pulse dies out, C1 cannot discharge through the high back resistance of CR1; thus it discharges through Z1, R1 and the input winding of core #2, producing a current pulse through core #2 sufficient to switch that core to the state representing a 1. Outputs from the series tape core shift register are taken off the output winding of the right-hand core of the register.

In order to provide for parallel entry of information into a core shift register, a fourth coil is provided for each of the cores of the register as shown in Figure 4-25. By simultaneously pulsing any combination of these input coils any desired pattern of 1's can be read into the register in parallel. The shift function is then performed in the same way as before.

The serial-entry, parallel-output type shift register used for conversions from serial to parallel operation is very similar to the register of Figure 4-24, except that facilities are provided for sensing the output of each of the cores simultaneously as these outputs appear across the

capacitors in the individual output networks in response to a shift pulse.

2.3 ELECTROSTATIC OR CATHODE RAY TUBE STORAGE

Electrostatic storage involves the storage of 1's as positive charges and 0's as negative charges on specified areas of a dielectric plate.

The device used to provide electrostatic storage is essentially a cathode ray tube. By application of various combinations of discrete X and Y voltages to the horizontal and vertical deflection plates of the tube, the electron beam can be aimed at a number of discrete areas on the dielectric, where charges are to be stored. Between the dielectric plate and the cathode of the tube is a screen grid through which the electrons of the beam pass before they impinge upon the dielectric plate. Beyond the dielectric plate and adjacent to it is a plate made of conducting material. Regardless of the charge existing on any area of the dielectric, that area can be made temporarily positive or negative with respect to the screen grid by applying a sufficiently positive or a sufficiently negative potential to the conductor plate. Assume that the stream of electrons is impinging on some particular area of the dielectric and that simultaneously that area has been made positive with respect to the screen. Under these circumstances there is virtually no secondary emission from the dielectric to the screen so that many more electrons strike the area on the dielectric than leave it. Thus, this area of the dielectric becomes negatively charged and, upon the removal of the positive potential from the adjacent conductor plate, it assumes essentially the

potential of the cathode. On the other hand, if the area on the dielectric on which the electron beam is impinging is made negative with respect to the screen, a high rate of secondary emission from the dielectric to the screen is established. Under these circumstances, many more electrons leave the target area of the dielectric than strike it, so that the area is charged positively and upon the removal of the negative potential from the adjacent conductor plate, it assumes essentially the potential of the screen. To summarize: by a proper selection of the potential applied to the conductor plate adjacent to the dielectric storage plate, an area of the dielectric can be caused to assume either screen potential or cathode potential. Since the plate is a nonconductor, regions of potential persist after the impinging beam has been removed. Thus, a mechanism exists for storing either 1's (areas of screen potential) or 0's (areas of cathode potential) on the plate. Since, by choosing a particular pair of discrete deflection voltages, the beam can be aimed at any selected discrete area on the plate, the 1's or 0's can be written at selected locations.

Unfortunately, the charged areas representing 1's and 0's have a tendency to discharge through existing high resistance paths; thus stored information must be regenerated periodically. This constitutes a disadvantage since the time required for regeneration of information becomes unavailable for transfer of information between the storage device and other parts of the computer.

Information can be read out of an electrostatic storage tube by a technique which involves the secondary emission characteristics of the tube. Assume that the electron beam is impinged on a particular area of the dielectric plate which is storing a 1 (that is, an area which is essentially at screen potential) and that the potential of the area is decreased slightly, by the application of a slightly negative potential to the adjacent conductor plate. This drops the potential of the target area from a state of equality with the screen potential to a slightly negative potential with respect to the screen, producing a substantial momentary increase in secondary emission which can be sensed as an increase in the screen current to the tube. On the other hand, if the electron beam is impinged upon an area of the dielectric plate which is storing a 0 (that is, an area which is essentially at cathode potential) and the potential of the area is slightly decreased by the application of a negative potential to the adjacent conductor plate, the rate of secondary emission (which is high because the area is at cathode potential) is not perceptably affected. Thus, there is no transient screen current effect as there is in the case of stored 1.

Electrostatic storage allows very rapid access to stored information; however, access speed for electrostatic storage is not quite as high as it is for magnetic core storage. This is because a significant amount of time is required for the development of sufficiently reliable deflection voltages to insure aiming the electron beam at the selected area with the required

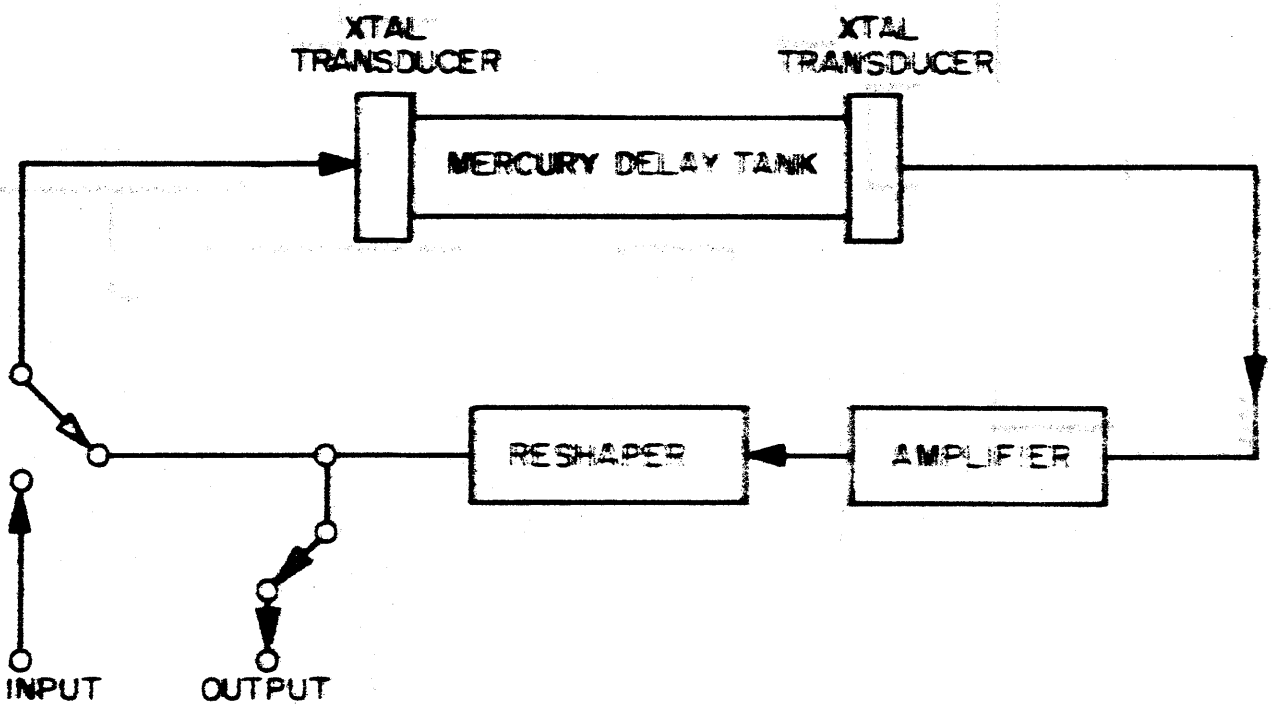

XTAL
TRANSDUCER
XTAL
TRANSDUCER
MERCURY DELAY TANK
RESHAPER
AMPLIFIER
INPUT
OUTPUT

Figure 4-26

precision. In addition to providing slightly slower access to stored information than does core storage, electrostatic storage has the disadvantage already mentioned, that it requires regeneration of stored information. This implies that all information is lost in the event of a temporary power failure.

2.4 SONIC STORAGE

A sonic delay line consists principally of a material which transmits pulses as a series of physical vibrations. The transmitting material may be solid, liquid or gaseous in form, and the proper selection of the material to be used as a medium determines the principle of operation of the sonic delay line system.

Although a gaseous medium may be used it is not particularly suited for this application due to high attenuations and other difficulties. Likewise, solid media are seldom used because of their tendency to transmit waves in many directions and with different velocities. For these reasons, liquid is nearly always used in a delay line system.

The type of liquid delay line most commonly used in electronic computers is the mercury delay line or mercury tank as shown in Figure 4-26. Mercury is chosen as the medium for sound transmission in liquid delay lines, because its acoustical impedance almost precisely matches that of the crystal

transducer to which it is coupled at each end. Improper impedance matching causes too much energy to be wasted, which must be avoided since the efficiency of transfer of energy is important if strong signals are to be transmitted through the medium. Any energy not absorbed by the receiving crystal, due to improper impedance matching, is reflected back and forth from the receiving crystal to the transmitting crystal as echoes. Proper impedance matching, therefore, is necessary in order to avoid these reflections which cause a serious modification of the original transmitted signal.

When a pulse is impressed on the quartz crystal in contact with the mercury tank shown in Figure 4-26, its shape changes due to the piezo-electric effect. This causes the quartz crystal to vibrate resulting in a wave-like or rippling effect in the mercury contained in the tank. The quartz crystal at the opposite end of the tank starts to vibrate under the influence of this ripple resulting in a regeneration of the original pulse.

Since there is some loss of energy (due to friction) and loss of shape due to the presence of secondary waves developed in the tank, the regenerated pulse must be amplified and reshaped. Both the amplifier and reshaper are shown, to complete the circuit. In actual practice, some additions would have to be made to the mercury delay line circuit shown in Figure 4-26 if control is to be exercised over its function.

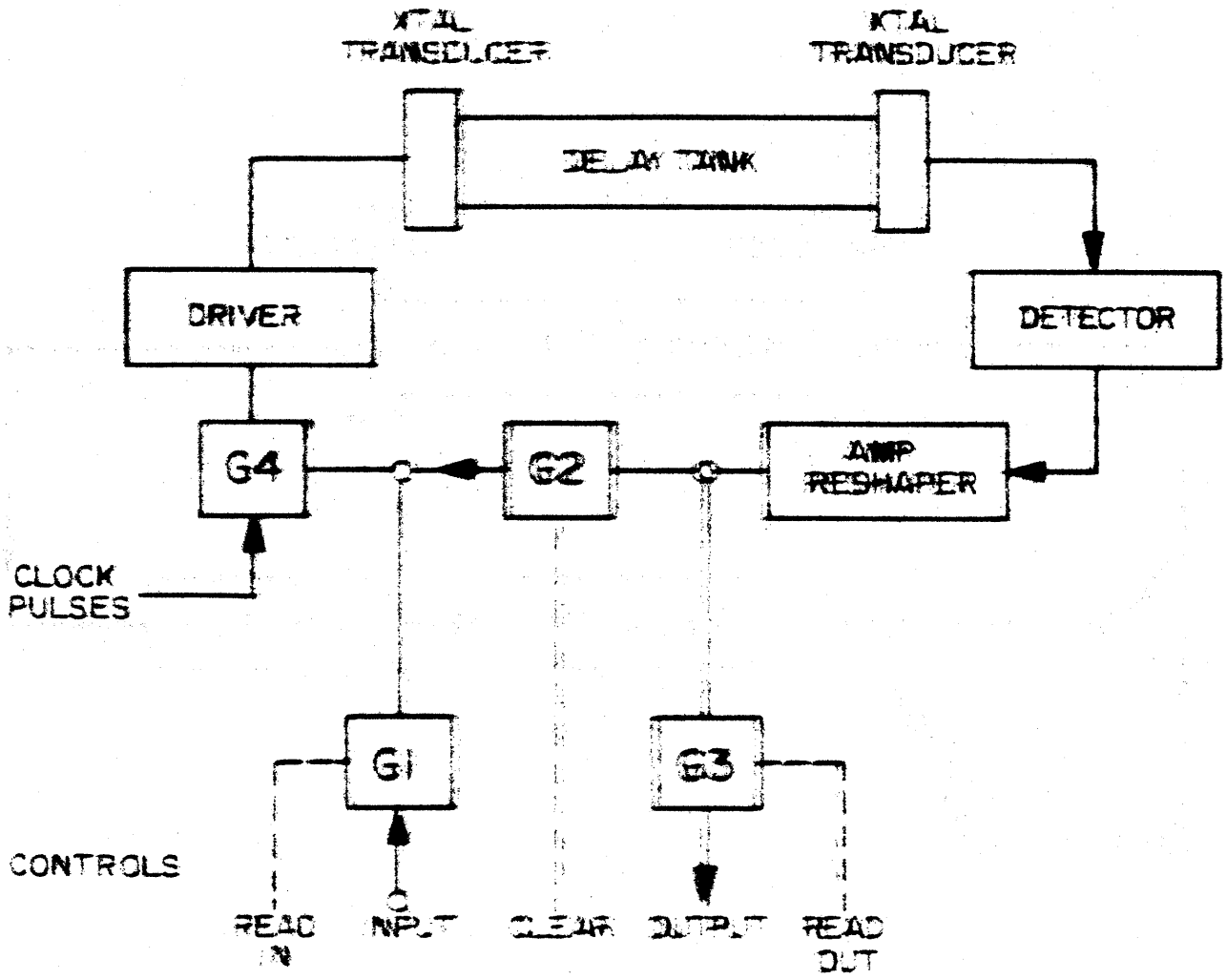

XTAL TRANSDUCER
XTAL TRANSDUCER
DELAY LINE
DRIVER
DETECTOR
G4
G2
AMP RESHAPER
CLOCK PULSES
G1
G3
CONTROLS
READ IN
INPUT
CLEAR
OUTPUT
READ OUT

Figure 4-27

As can be seen from Figure 4-27, a driver is added to excite the input crystal, also a set of gates is provided with which to control the course of the stream of pulses. Coupling the output crystal with the amplifier reshaper is a detector whose function is to demodulate the carrier.

Pulses are introduced into the tank through G1 and continue to circulate until gate G2 is opened. Pulses are read out through G3. The synchronizing or clock pulses are introduced through G4 and require that the temperature of the mercury in the tank be closely controlled so that the actual delay time remains nearly constant.

The velocity of sound in mercury like its velocity in air, varies with temperature. At frequencies of a few megacycles, the velocity of sound in mercury varies one part in 3,000 for each degree centigrade of temperature change. Thus if 3,000 binary digits are stored in a delay line, and if the temperature in the line is not known to within $1^{o}C$, it is not possible to tell which digit is being read at a given time, unless a record is kept of the digits as they emerge. This imposes certain limitations on the length of mercury delay lines.

As already noted, any signal passing through the mercury tank is somewhat attenuated. Specifically, attenuation is about 5 decibels per millisecond at 10 megacycles. At this frequency, therefore, this is not a serious consideration. The chief objection to the acoustical delay memory, as the

sonic delay is also called, is its rather long access time. The signals that are traveling through the mercury as sound pulses are not immediately available to the computer. Access to stored information entails waiting until this information reaches the crystal transducer and is demodulated to electrical pulses. On the average this entails a delay of one half the total delay introduced by the mercury tank, or about 200 microseconds in a typical case. Compared to the speed of arithmetic operations in a modern computer, this is a long time. Furthermore, it is impractical to store a large number of digits in a mercury system because of the prohibitive number of vacuum tubes required by the associated circuitry.

2.5. MECHANICAL STORAGE

2.5.1 Punched Tape

One of the oldest and most common methods for storing of information is punched tape. Its physical strength, which enables it to be read by means of mechanical feelers, without significant deterioration and the compact manner in which it stores data render it useful as a permanent medium.

Information is stored on the tapes by punching in an array of holes in adjacent columns or data channels along a length of tape. Two possibilities exist, a particular location on the tape may be punched or not punched. Therefore, the binary notations 1 (punch) and 0 (no punch) may be utilized. Factors such as the tendency of the tape to shrink or stretch, and the minimum spacing at which holes can be reliably

sensed by the reading equipment determine the minimum spacing between holes.

Electrical and photoelectric sensing are the two methods used to read information stored on the punched tapes. Electrical sensing of holes is accomplished by passing the tape over an electrically charged metallic platen, over which are placed metal wipers in position to make contact with the platen through holes in the tape or by arranging fingers which slip through the holes in the tape and operate switches. The information is thus read out in the form of electrical pulses, which is convenient for computer use. With electrical sensing the top speed that is obtainable is approximately 100 digits per column per second.

Photoelectric sensing of holes provides an alternative and much faster method of reading information than the electrical sensing of holes by means of feelers. When a hole in the punched tape is driven past an aperture in the presence of a constant source of light, the light passes through the holes and the apertures and is gathered in an optical system which focus the light on the photocells. Each time a hole passes over an aperture an electrical pulse is generated in the photocell. With this system of writing-out information a speed of 5,000 digits per second per column is obtainable.

A characteristic of the punched tape storage is that the information stored is nonerasable, therefore it is most suitable as a medium for permanent storage. However, there is a practical limit to the number of digits which may be stored by

this method due to the mechanical difficulty of handling a large loop of tape. Rapid access of information is impossible because the nature of this system is a cyclic one. However, if a shorter access time is desirable it is quite possible to use independent read and write heads at various locations around the loop. Another disadvantage of this method is the wearing characteristic. The tapes tend to wear out with use especially when the electrical sensing method is used. However, to some degree this is overcome by using the photoelectric method whereby the use of metal wipers or fingers are not used thereby eliminating one source of wear.

When data punches on the tape is to be corrected or changed, it is only necessary to cut the tape, add or remove sections and splice it together again. Since the recorded data can be inspected visually and directly this process of altering the tape is made much simpler.

2.5.2 Punched Cards

Punched cards like punched tape is one of the oldest methods of storing information. However, due to the physical size of the cards the amount of information that can be stored is limited. These cards are mainly used as the primary program input medium because of their great flexibility and because of the availability of associated key-punching, verifying, and duplicating equipment.

Information is stored on the cards by punching the various rows and columns. On a binary-punched card, a punch is read

as a 1 and no punch as a 0, which is similar to the method used with tape punch.

Electrical sensing is the method that is used to write-out the information stored on the punched cards, which works on the same principles as the electrical sensing performed on the punched tapes.

The punched cards provide a permanent, easy to file, record of data. Since the recorded data can be inspected visually and directly, errors can be discerned rapidly and rectified.

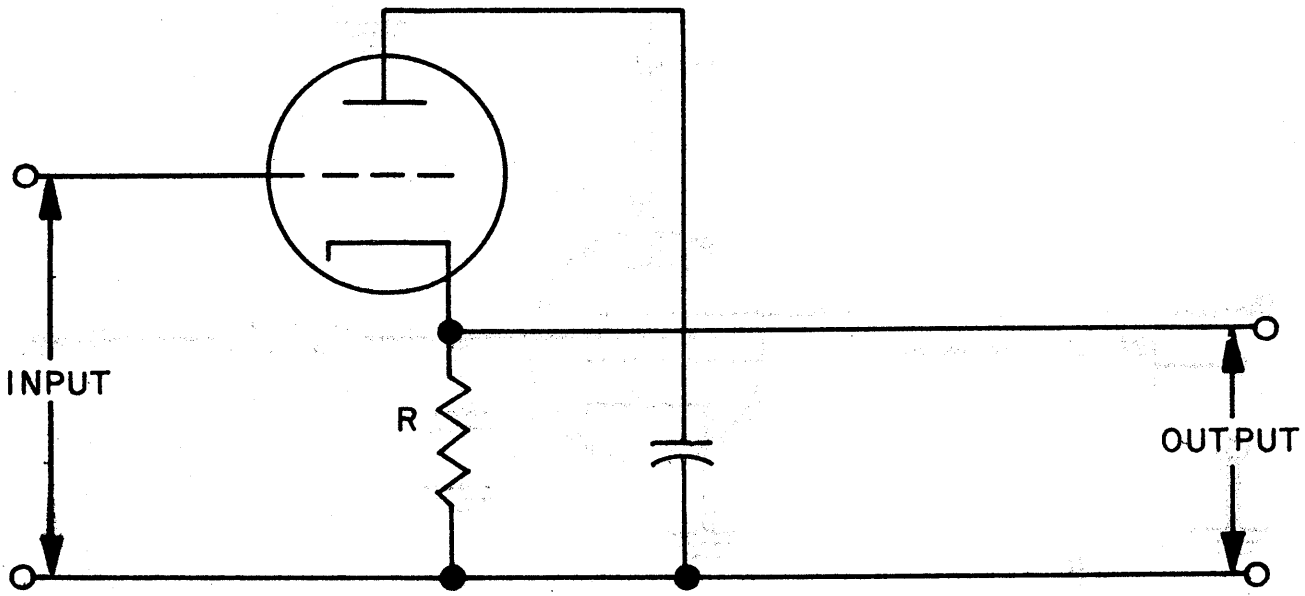


Figure 4-28

PART 4

CHAPTER 3

MISCELLANEOUS CIRCUITS

3.1 GENERAL

The miscellaneous circuits that are present in most digital computers may be grouped into three main categories:

1. Power amplification and isolation circuits.
2. Voltage amplification circuits.
3. Pulse generating and wave shaping circuits.

3.2 POWER AMPLIFICATION AND ISOLATION CIRCUITS

The power output of a circuit may not be sufficient to drive a mechanical device or a following electronic circuit. Thus power amplifiers must be employed. Where it is necessary to isolate a circuit from a preceding stage in order to prevent excessive loading or to obtain a signal of proper polarity, isolation circuits are employed.

3.2.1 Cathode Follower

The basic cathode-follower shown in Figure 4-28 is essentially a single stage inverse feedback amplifier in which the output voltage is taken from across the cathode resistor.

Some of the valuable characteristics of a cathode follower are its low input capacitance, high input impedance, and low output impedance. The voltage gain of a cathode-follower is less than unity, and because it is a degenerative amplifier it can handle high input voltages without distortion. Since the output voltage is taken from the cathode, the input signal is

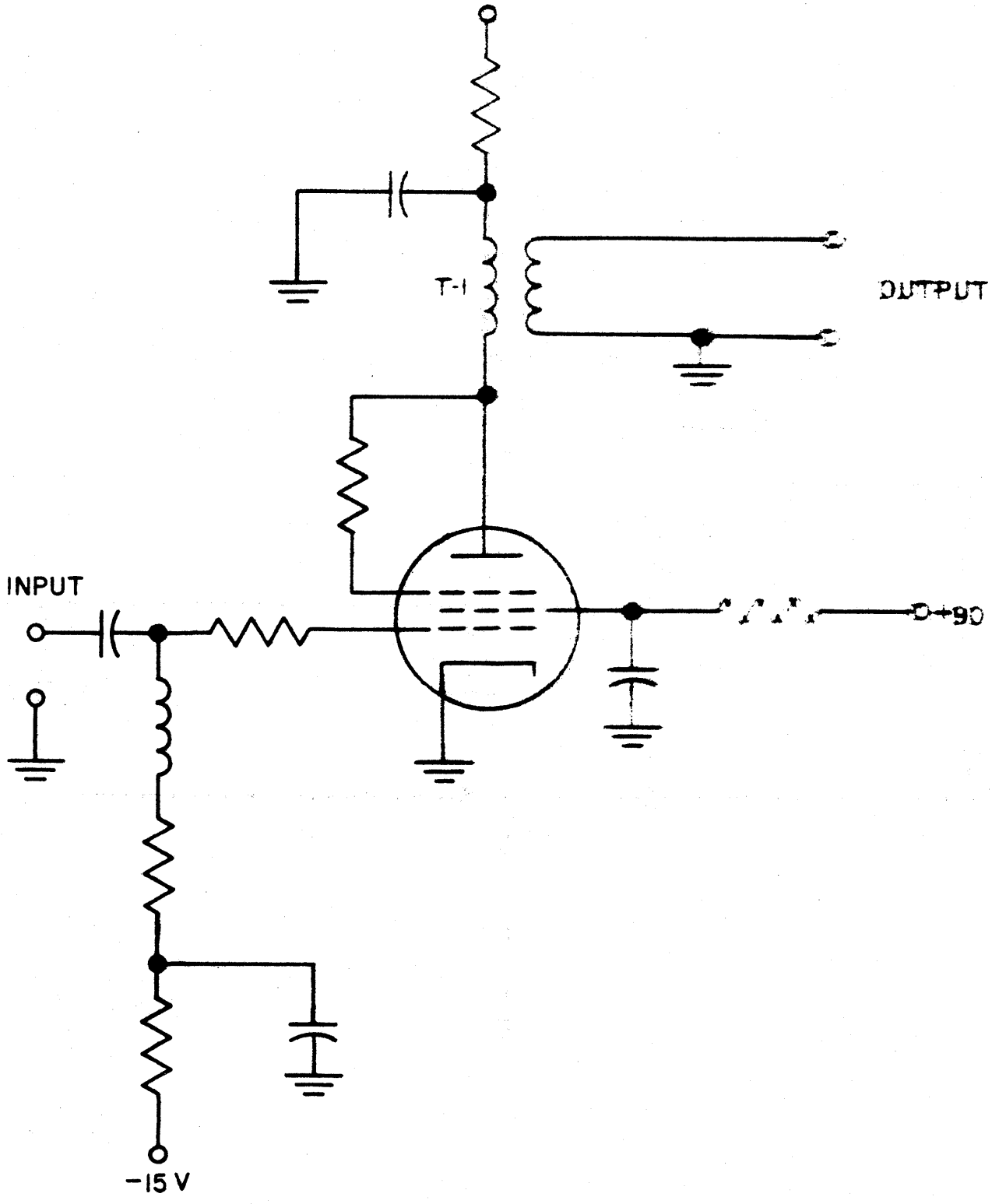


Figure 4-29

in phase with the output signal.

Cathode followers are used as Power Amplifiers in digital computers to couple circuits having high output impedance to low impedance loads, i.e. isolate circuits with a low current output from circuits requiring a large current input. They are also used to give an output voltage that is in phase with an input signal.

3.2.2 Pulse Amplifier

Pulse amplifiers in computers are usually used as power amplifiers of standard pulses which without power amplification would be unable to drive a given stage. The development of standard pulses is covered later in the section dealing with pulse generating and wave shaping circuits.

Figure 4-29 illustrates a representative pulse amplifier circuit. The control grid voltage is kept at a steady value which is negative enough to prevent the pulse amplifier tube from conducting. A positive going pulse of sufficient magnitude when impressed upon the control grid causes sudden conduction. Cut-off occurs when the trailing edge of the input pulse dies out, allowing the grid to return to the cut-off bias level. The amplified pulse is taken off the pulse transformer, T1, in the plate circuit, which provides a second phase inversion so that the output signal is in phase with the input signal.

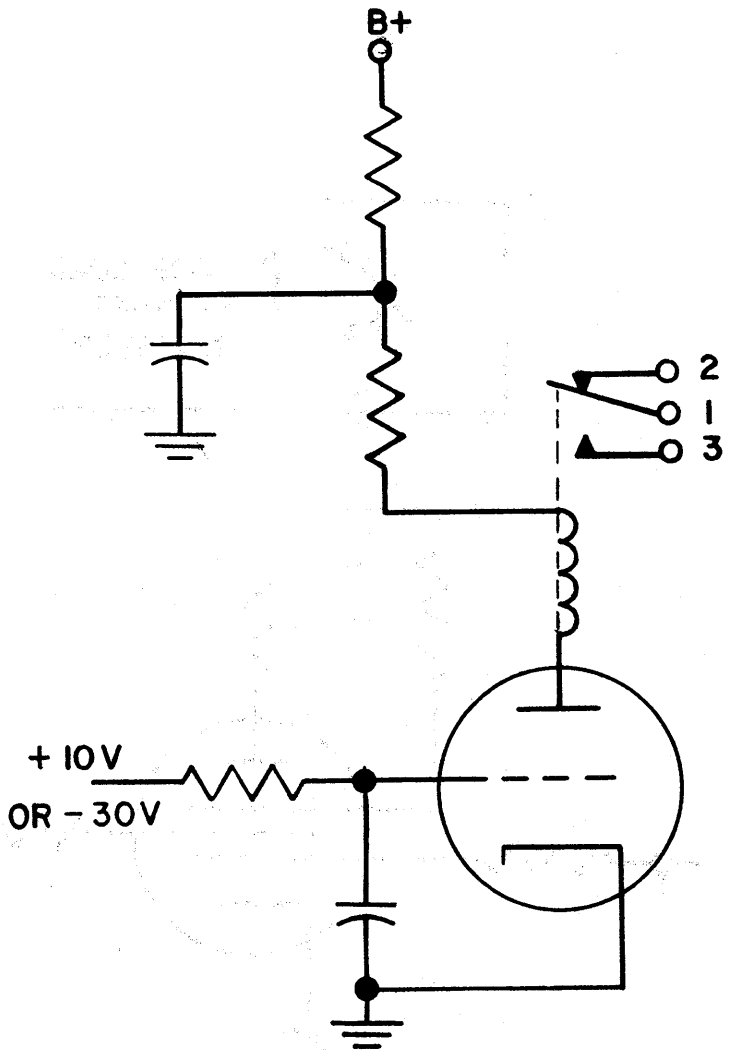


Figure 4-30

3.2.3 Relay Driver

The relay driver circuit, Figure 4-30 is used to interpret the state of a flip-flop circuit and to actuate a sensitive relay depending upon the state of the flip-flop output. A flip-flop, as previously noted, has but two output states. Therefore, the output signal level of a flip-flop may be used to cause conduction or cut-off in a circuit to which the output of the flip-flop is fed.

In Figure 4-30 the grid is fed a d-c voltage of either +10 or -30 volts which represents one or the other state of conduction of a flip-flop. At minus 30 volts the relay driver tube is not conducting and the relay contact 1 is in one position. At plus 10 volts the relay driver tube conducts causing current to flow through the relay coil winding. The relay is thus activated and contact 1 changes position. Thus, the relative position of relay contact 1 reflects the relative state of an associated flip-flop circuit.

3.2.4 Tetrode Drivers

Tetrode amplifiers are essentially power amplifiers that are used when the rise and fall time of the output pulse must follow as closely as possible that of the input pulse. The usual form of a tetrode driver is derived from a pentode tube in which the suppressor and plate are tied together and are at the same potential level. This arrangement offsets to some extent undesirable characteristics of a pentode type circuit where pulse distortion would occur when the plate voltage

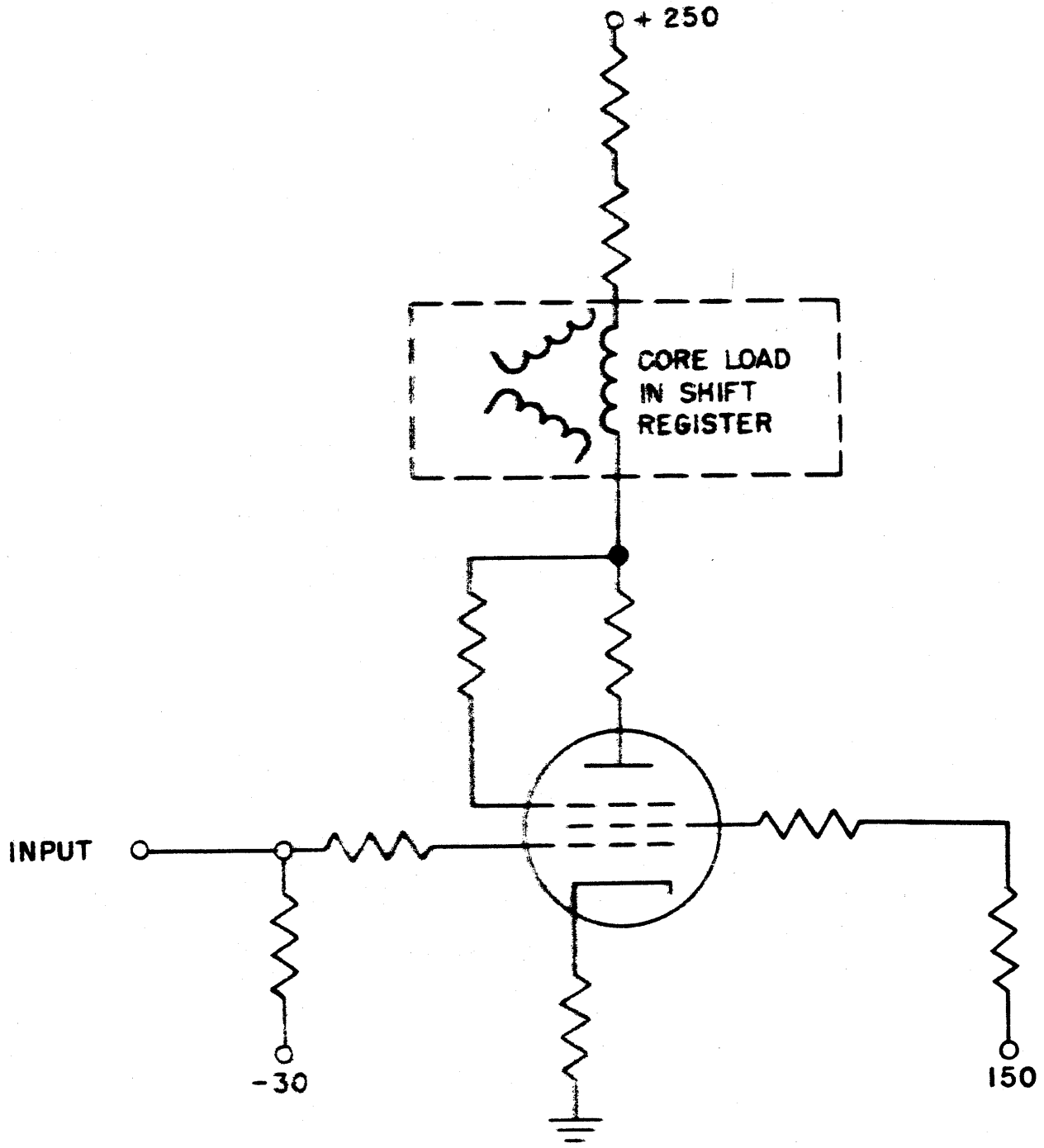


Figure 4-31

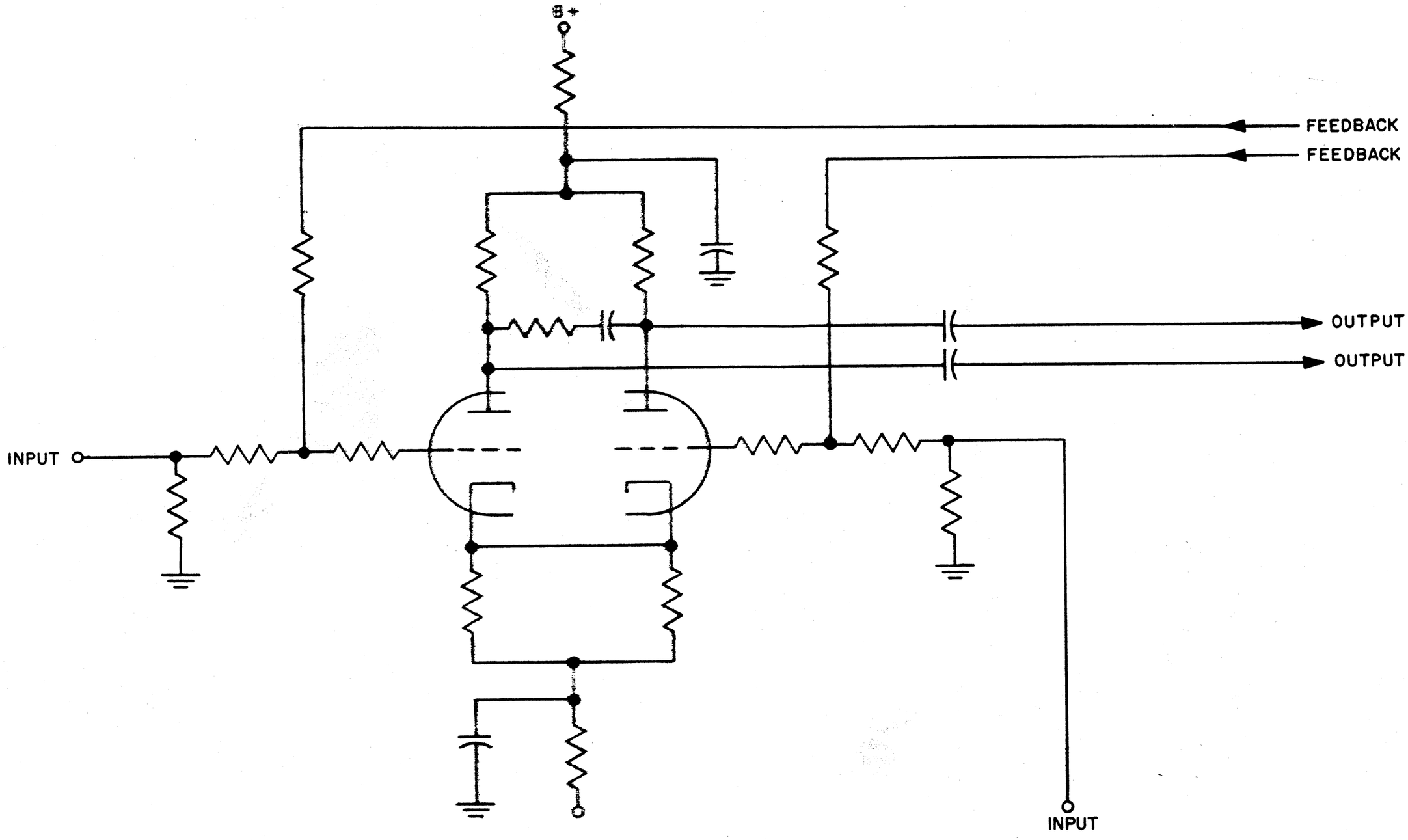


Figure 4-32

falls below the screen voltage.

Tetrode drivers are used in computers to provide current for switching tape cores.

Figure 4-31 illustrates the application of a tetrode circuit as a core shift driver.

3.2.5 Differential Amplifier Drivers

A differential amplifier comprises two tubes or two sections of the same tube sharing a common cathode circuit but having separate plate and grid circuits. The stage amplifies the difference between the signals applied to the separate grid circuits. An output is available from either plate circuit, the one output being 180° out of phase with the other. Thus, with respect to a signal applied to either one of the grids, an in-phase output or an out-of-phase output is available. Alternatively, the circuit may be used as a phase splitter by taking outputs off both plates.

A differential amplifier circuit is shown in Figure 4-32. Referring to the figure, notice that the application of in-phase signals to the two grids results in the appearance of a degenerative signal across the unbypassed portion of the common cathode resistance. For example, if both grids receive positive signals, the cathode potential is increased, minimizing the increase in plate current through both sections of the tube. Thus, the gain of the circuit in response to in-phase grid signals is very low. Suppose, on the other hand, that a positive signal is applied to one of the grids while the potential on the other is

held fixed. Current will increase through the section of the tube receiving the positive grid signal. This will tend to raise the potential of the common cathode. However, any increase of the cathode potential will cause a decrease in the plate current drawn by the other section of the tube (since the grid of that section is held fixed). Thus, the cathode potential will remain relatively constant; that is the total current drawn through both sections of the tube will remain relatively constant. This implies that the increase of current through the section of the tube whose grid is driven positive will be matched by a decrease of current through the section of the tube whose grid is held fixed. Since, in this situation, there is no signal loss across the common cathode resistance, the gain of the circuit is high. Thus, the amplifier responds essentially to difference signals between the two grids.

One advantage that the differential amplifier affords, in addition to the flexibility of connections that it allows, is that, because of the degeneration through the cathode resistance in response to signals which are common to both sides of the circuit, it is relatively insensitive to variations in heater voltage.

3.2.6 Flux Amplifier

The flux amplifier receives signals from the sense windings of the ferrite core array and delivers an output to a core shift unit in a shift register. In the flux amplifier, signals from the core array output are discriminated, amplified, and shaped.

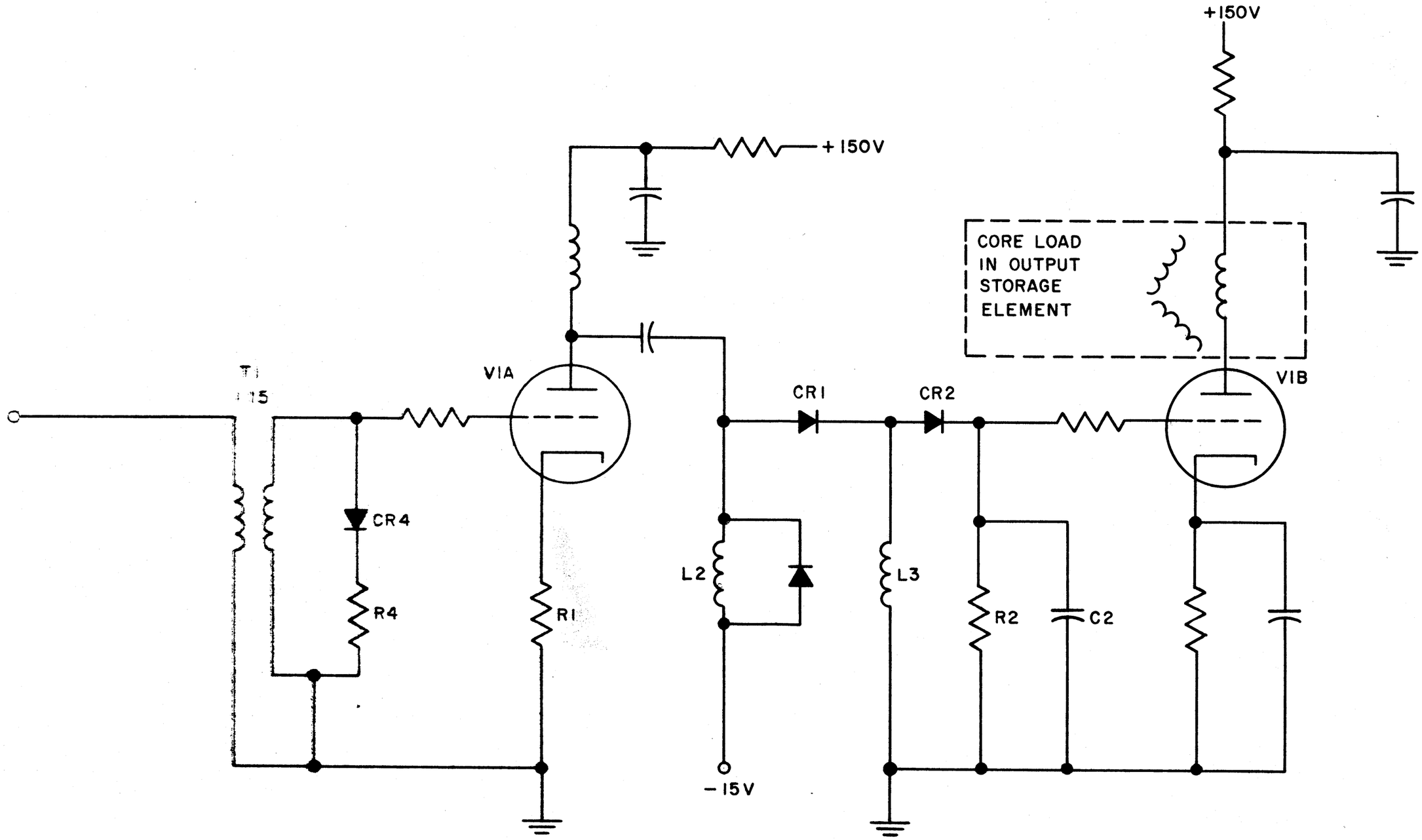

+150V
+150V
CORE LOAD
IN OUTPUT
STORAGE
ELEMENT
T1
115
VIA
VIB
CR1
CR2
CR4
R4
R1
L2
L3
R2
C2
-15V

Figure 4-33

Figure 4-33 illustrates a flux amplifier circuit. T1, in the grid circuit of V1A, is a step-up transformer which amplifies incoming signals. Crystal CR4 prevents ringing. R4 is used to limit current through CR4. The unbypassed cathode resistor, R1, causes degeneration and limits amplification.

The network that produces amplitude discrimination in the flux amplifier circuit consists of chokes L2, L3 and diode CR1. Since CR1 is biased negatively by the minus fifteen volt supply fed through choke L2, an output signal greater than plus fifteen volts is necessary from the output of V1A in order to drive V1B.

The output of the discriminator network is fed through crystal diode CR2. Since a positive signal is fed through CR2, the forward resistance of the crystal is low, and as a consequence, C2 will charge rapidly. The grid of V1B will simultaneously go rapidly positive. V1B will then conduct with a relatively fast rising plate current. The positive pulse on the grid of V1B will decrease as C2 discharges through R2. Since the time constant of this circuit is designed to be longer than the width of the incoming signal to V1A, the output signal of V1B will reflect the effect of time constant R2C2.

3.3 VOLTAGE AMPLIFICATION CIRCUITS

Voltage amplification circuits are used to restore the d-c level and wave form of pulses that have been attenuated by passing through several levels of switching and cathode followers. Where the same phase on the output signal is required, a level setter is used. If an inverted pulse is required, attenuated

to produce standard voltage levels in response to attenuated level inputs.

3.3.2 D-C Inverter

The d-c inverter is similar to the d-c level setter described in the preceding section. However, in addition to restoring the magnitude of input signals, the inverter provides a phase inversion. Such a circuit is shown in Figure 4-35. Comparing the circuit of Figure 4-35 with the d-c level setter circuit of Figure 4-34, notice that the input stage of the inverter is a triode while the input stage of the level setter is a differential amplifier. The cathode follower stage is the same in both circuits. Either circuit restores incoming signals to standard voltage levels. However, the triode input stage of the inverter provides a phase inversion where the differential amplifier input stage of the level setter does not.

3.4 PULSE GENERATING AND WAVE SHAPING CIRCUITS

Data from one unit of a Computer must be transferred to another unit while a problem is being solved. In electronic digital Computers data is transformed into pulses which may be initiated, controlled, or altered by various pulse-generating and wave-shaping circuits. The first example of a pulse generating circuit is the thyratron pulse generator.

3.4.1 Thyratron Pulse Generator

The thyratron pulse generator is used to produce a standard pulse when a contact is closed or when an input signal triggers the thyratron pulse circuit. However, the repetition rate of a thyratron pulse generator is severely limited by the de-ionization time of the gas tubes used in thyratron circuits. Consequently, a thyratron pulse generator is used where the frequency of the pulses desired is comparatively low.

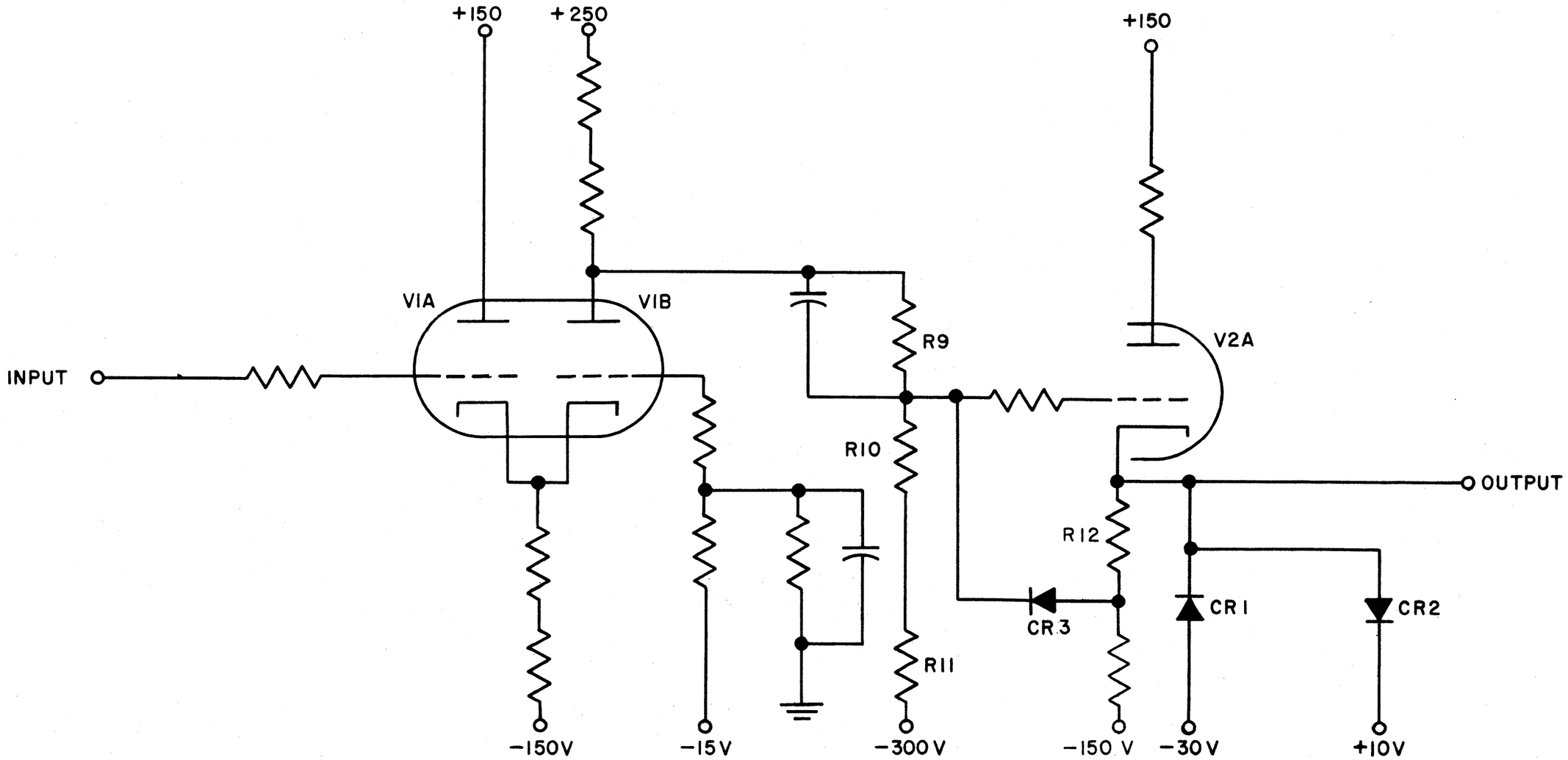

+150
+250
+150
VIA
VIB
INPUT
R9
V2A
R10
OUTPUT
R12
CR1
CR2
CR3
R11
-150V
-15V
-300V
-150 V
-30V
+10V

Figure 4-34

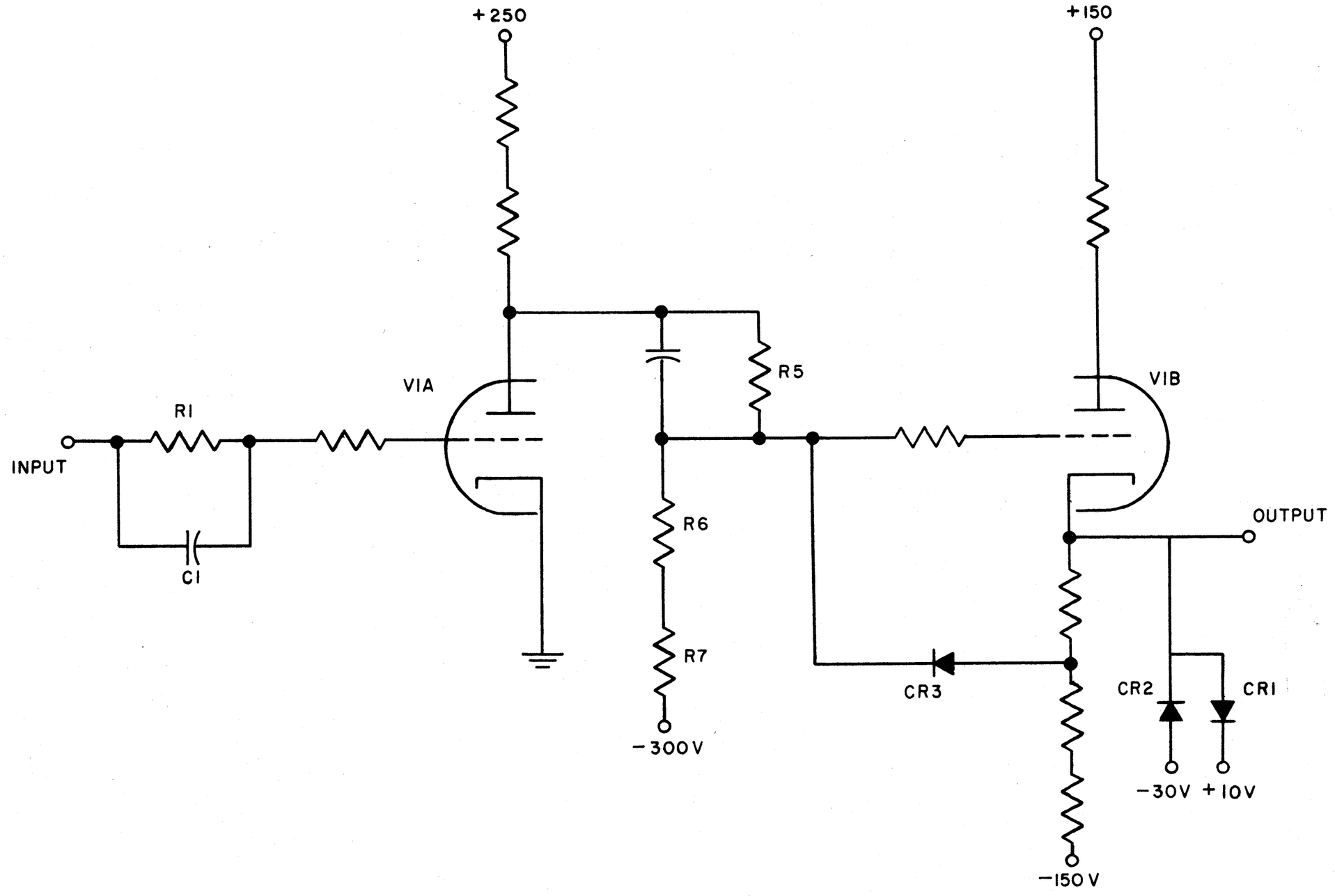

+250
+150
VIA
VIB
R1
R5
INPUT
C1
R6
R7
-300V
CR3
OUTPUT
CR2
CR1
-30V
+10V
-150V

Figure 4-35

pulses are fed to an inverter. In either case the result of passing attenuated pulses through a d-c voltage amplifier circuit is the formation of pulses that have a standard d-c level.

3.3.1 D-C Level Setter

The d-c level setter illustrated in Figure 4-34 consists of overdriven differential amplifier V1 and cathode follower V2A. Attenuated pulses are fed directly to the grid of V1A. The circuit parameters are such that positive incoming voltage levels drive V1B below cut-off while negative incoming voltage levels drive V1B to saturation. Hence levels of either polarity are standardized and the output pulses at the plate V1B are in phase with the attenuated pulses fed to the grid of V1A.

A portion of the output of the differential amplifier, which is taken off the plate of V1B, appears at the junction of resistors R9 and R10 which, together with resistor R11, comprise a voltage divider between the plate of V1B and - 300 volts. The signal appearing at the junction of R9 and R10 is coupled to the grid of cathode follower V2A which provides power amplification. The output of the cathode follower is clamped between - 30 volts and + 10 volts by diodes CR1 and CR2 respectively A negative output from the differential amplifier is sufficient to drive the cathode follower output to the negative clamping level, while a positive outpput from the differential amplifier is sufficient to drive the cathode follower output to the positive clamping level. Thus, the level setter functions

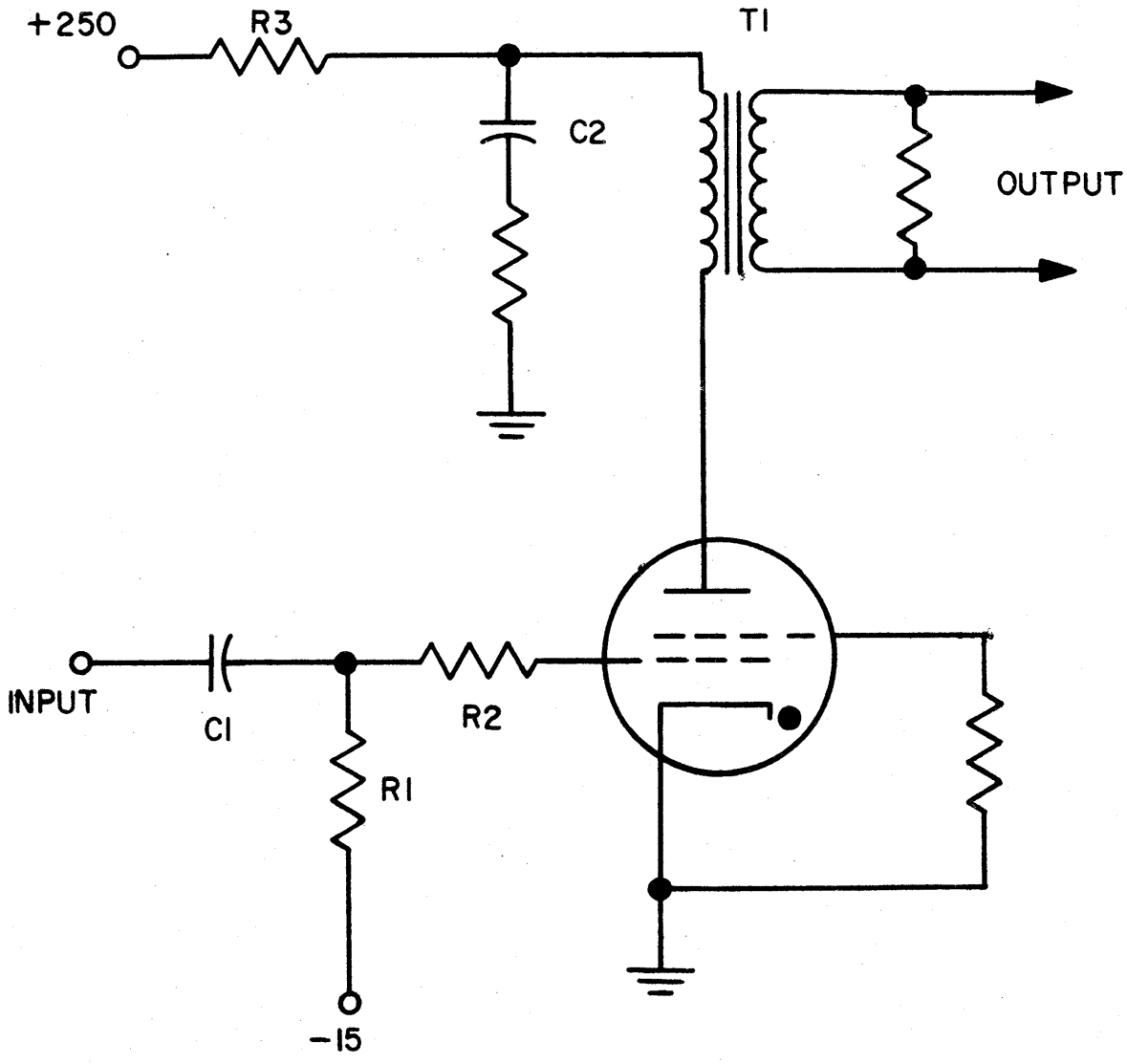


Figure 4-36

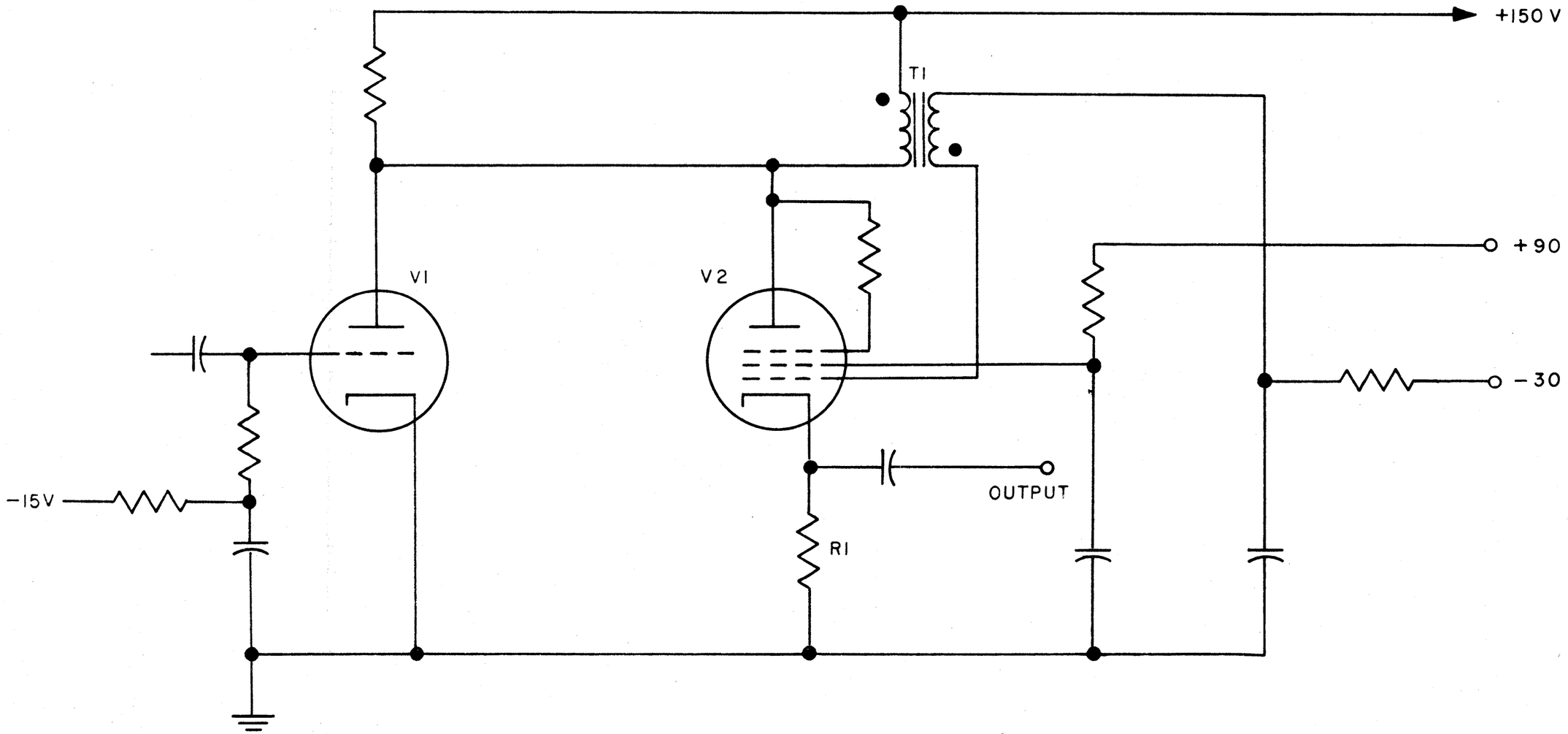


Figure 4-37

The representative thyratron pulse generator tube, in the circuit of Figure 4-36, is essentially a grid controlled gas rectifier tube. The control grid is only able to initiate the flow of current in the thyratron tube. Conduction is stopped by reducing the plate potential sufficiently.

The thyratron pulse generator shown in Figure 4-36 is kept at cut-off by the negative 15 volts impressed upon the control grid through R_1 and R_2. R_2 is used to prevent excessive grid current. If a pulse of sufficiently positive amplitude is fed through C1, the thyratron tube will suddenly conduct with a subsequent rapid drop in plate potential. When the plate voltage is of a sufficiently low potential, conduction will stop. The plate voltage will then rise with a rate that depends upon the values of R_3 and C_2. The resultant pulse is taken from pulse transformer T_1.

3.4.2 Blocking Oscillator Pulse Generator

Another type of pulse generator is the blocking oscillator pulse generator. This type of circuit is used to generate standard pulses at a more rapid frequency than those formed in the thyratron pulse generator. However, a relatively large input pulse is required to trigger the blocking oscillator pulse circuit. Therefore, the output of a pulse amplifier previously described is usually used as an input to a blocking oscillator circuit.

The blocking oscillator pulse generator, Figure 4-37 consists of a pulse amplifier, V_1, and a blocking oscillator, V_2. A sudden rising D-C pulse fed to the grid of V_1 will

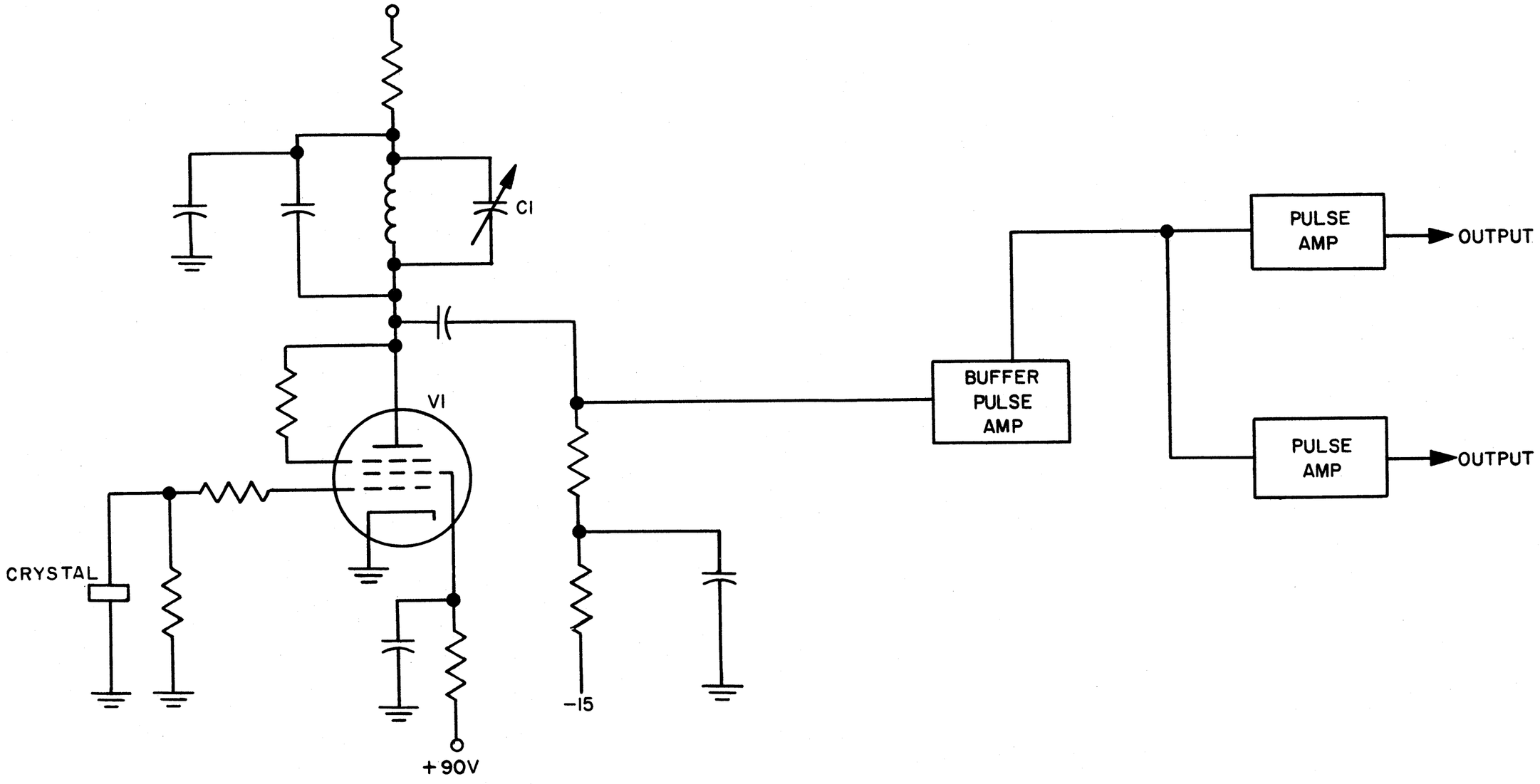

C1
V1
CRYSTAL
+90V
-15
BUFFER
PULSE
AMP
PULSE
AMP
OUTPUT
PULSE
AMP
OUTPUT

Figure 4-38

cause V_1 to conduct dropping the voltage on the plate of V_1, but raising the voltage on the grid of V_2. The polarity of the windings of pulse transformer T_1 are such that a drop in plate voltage V_1 raises the grid voltage of V_2. As the control grid of V_2 goes positive, V_2 conducts and further reduces the voltage of V_1 and V_2. The control grid thus becomes more positive. If V_2 were a free running blocking oscillator circuit, the control grid would eventually draw enough current and go negative so that a reverse cumulative effect would take place. But since the voltage on V_1 suddenly drops, as the trailing edge of the input pulse is impressed on the control grid of V_1, the grid of V_1 is returned to its normal cut-off condition. The plate voltage on V_1 rises sharply as a result, while the control grid of voltage V_2 drops suddenly. The resulting output pulse taken from across R_1 varies with the state of conduction of V_2. Thus, an output pulse is formed across R_1 for every input pulse at the grid of V_1.

3.4.3 Crystal Oscillators

Crystal oscillators are used as master clocks in the instruction control element of the central computer and in the same manner in the tape inputs. The output of a crystal oscillator tube itself cannot be used, but must first be passed through a buffer pulse amplifier and then to two pulse amplifiers (Figure 4-38). The resultant outputs are then of sufficient amplitude and of proper pulse form to be effective in the stages to which the pulse outputs are fed. Thus, the definition

of a crystal oscillator in a computer not only includes the oscillator itself, but buffer and amplifier stages.

As pulse amplifiers have already been described, only the action of the crystal oscillator tube circuit V_1 will be given. The crystal oscillator circuit illustrated (Figure 4-38) is essentially a tuned grid, tuned plate oscillator with the crystal supplying the tuned grid circuit. The plate tuned circuit, on the other hand, must be tuned so that its resonant frequency is slightly higher than that of the crystal. This may be obtained by variable condenser C_1.

The value of the crystal in controlling frequency is due to the extreme sharpness of its resonance curve. Thus, a crystal oscillates over a limited frequency range, resulting in a very stable oscillator.

3.4.4 Sawtooth Generator

The sawtooth generator is employed to develop linear sweep to be used for deflection purposes in oscilloscopes. However, since a very linear sweep is desired, special circuits are incorporated. The most common type of linear sawtooth generator found in computers is the bootstrap sweep circuit.

The exponential output of a sweep generator is not of sufficient linearity to be used where close linearity tolerance is required. In order to obtain a linear sweep it is necessary to recharge the output condenser of a sweep generator from a constant-current source. The most commonly used device that provides a constant-current source is a pentode tube,

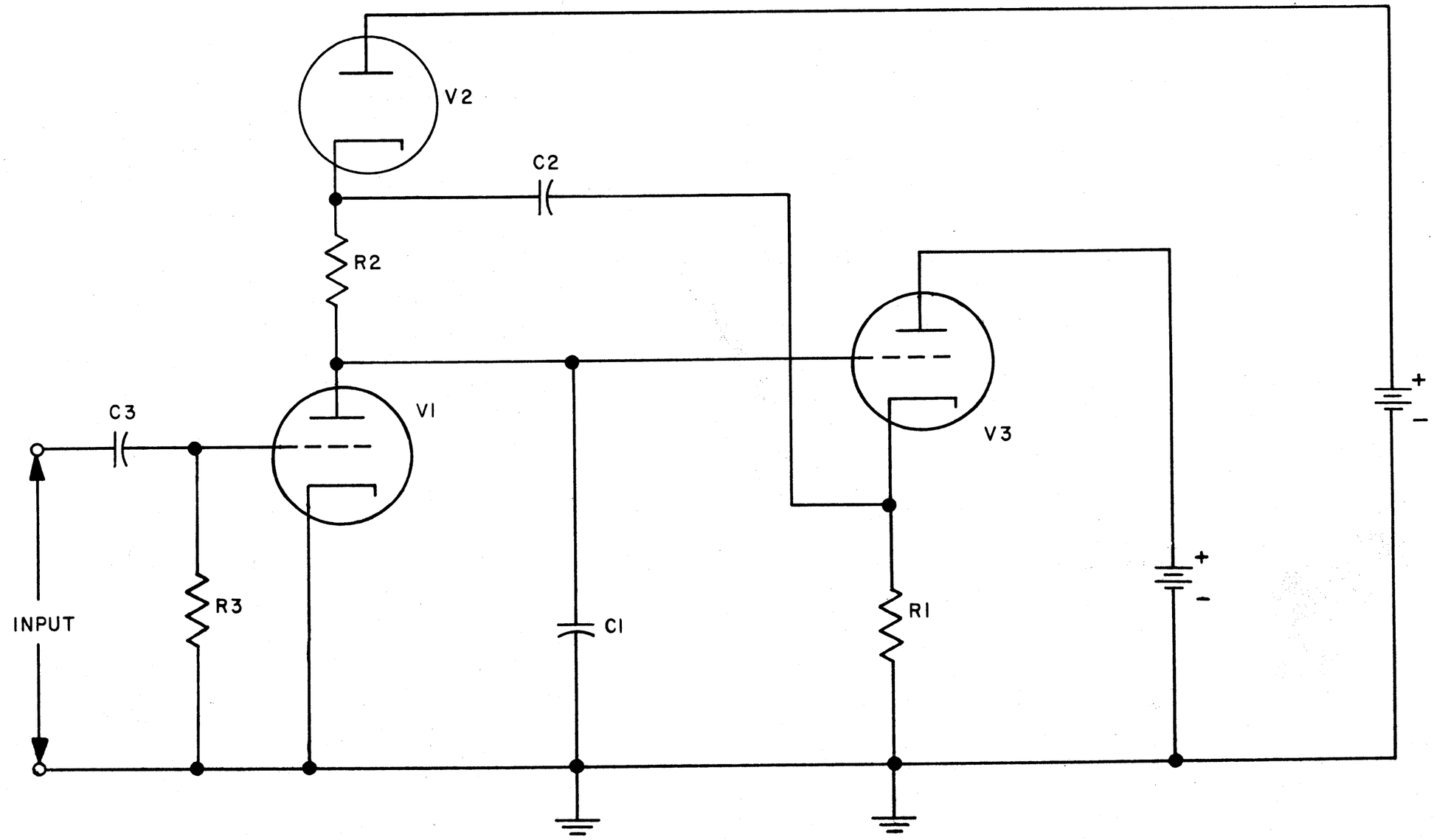


Figure 4-39

since the plate current for a varying plate voltage is quite constant as long as the tube bias is constant. However, the difficulty in maintaining various element potentials limits the usefulness of a pentode.

Satisfactory linearity correction, however, may be obtained by the use of a feedback amplifier as shown in Figure 4-39. This type of circuit is called a "bootstrap" circuit since the change in sweep capacitor voltage is added to the charging voltage at each point along the sweep.

The cathode-follower V_3, Figure 4-39 is used as a feedback amplifier. C_2 is used to couple the clamping diode V_2 to the cathode-follower. The value of capacitor C_2 is relatively large so that there will be little change in the voltage across it during the time of the sweep. V_1 has its grid clamped at zero voltage by the large time constant of R_3C_3. Consequently, with tube V_1 conducting, the voltage across capacitor C_1 is at a minimum. The plate voltage of V_1 is also effectively coupled to C_2 if the static conducting resistance of V_2 is small.

When the grid of V_1 is driven below cut-off, V_1 is effectively removed from the circuit. The voltages across C_1 and C_2 cannot change instantaneously, neither can the operating point of the cathode follower V_3. The current through R_2 remains the same value as before the cut-off and can only flow through C_1. If the potential across C_1 increases, the same potential increase occurs across R_1, since the gain of cathode follower V_3 is practically unity. The increase in potential

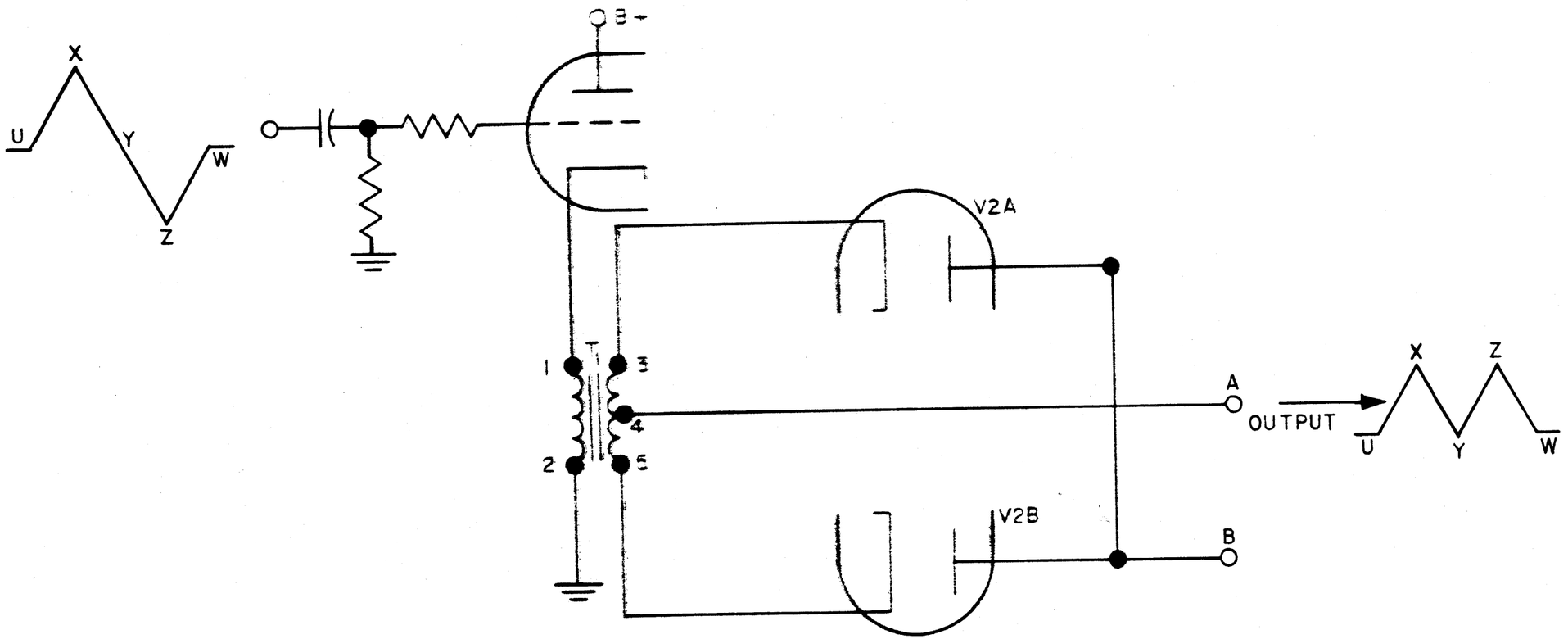


Figure 4-40

R_1 is reflected across C_2 and cuts off diode V_2. Consequently, C_2 will then act as a charging source, and if C_2 is very much larger than C_1, it can supply the necessary current to C_1 without changing its own potential. As the increase in voltage across C_1 is added to the charging source, the current through R_2 will be essentially constant. Thus the sweep voltage across C_1 will be nearly linear.

In computer circuits where relatively low frequency repetition pulses are fed to the sweep generator tube, a thyratron tube is usually used instead of the vacuum type, illustrated in Figure 4-39.

3.4.5 Frequency Doubler

A frequency doubler is used when one component of a system is to be sychronized with another that has twice the frequency of operation of the first component. Often the use of a frequency doubler permits the use of smaller reactive components in circuits that follow the doubler.

The frequency doubler, illustrated in Figure 4-40 is composed of a triode, V_1, and one duo-diode, V2A and V2B.

The input to V_1 is a sawtooth pulse. The input of this stage is taken from across V_1 in the cathode circuit of V_1 and fed to the cathodes of V2A and V2B.

A positive going sawtooth starting at point U, Figure 4-40 fed to the control grid of V_1 would cause conduction in the duo-diode circuit. If the positive going sawtooth is assumed to cause terminal 3 of T_1 to go relatively positive and terminal 5 relatively negative with respect to centertap, terminal

4, then tube V2B will conduct raising the relative potential of output terminal A with respect to output terminal B. As the input sawtooth passes its positive peak, point X, conduction in V2B lessens and the relative output potential at point A decreases. When input sawtooth sweep passes point Y, then terminal 3 goes negative and as a consequence V2A conducts and again output potential A rises with respect to B. Finally, when Z of incoming sawtooth is passed, the relative potential of output A drops. Thus, for every input sawtooth, there are two positive pulses in the output line and therefore the output positive pulsing frequency is twice that of the effective positive pulsing frequency of the input. The action of this circuit is similar to that of a full-wave rectifier with a consequent drop in peak to peak voltage between the input and output signals.

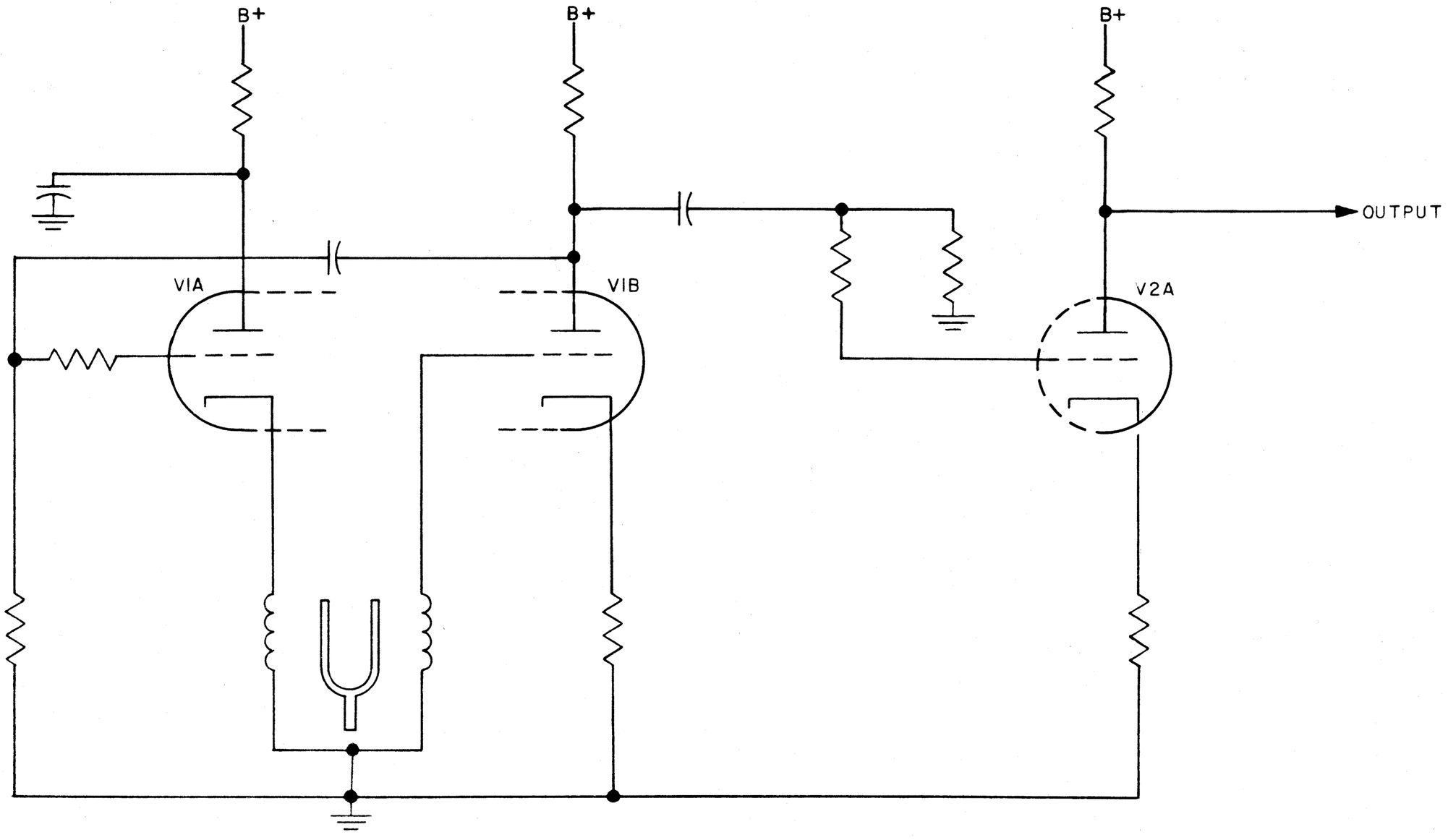


Figure 4-41

3.4.6 Tuning Fork Oscillator

The tuning fork oscillator circuit illustrated in Figure 4-41, is used to generate a sine wave at a low frequency. The output is usually used to drive a trigger circuit, but may be used for other applications.

V1A and V1B, Figure 4-41 form the actual oscillator circuit. V2A is a buffer amplifier. The tuning fork oscillator circuit is essentially that of a magnetostriction oscillator which depends upon the relation between mechanical stresses in a metallic device and the magnetic field in a surrounding coil.

If current flows through the cathode circuit of V1A, stresses in the tuning fork due to magnetic forces will cause the fork to oscillate at its natural frequency. As a result, the voltage impressed on the grid of V1B will vary causing in turn, an alternating voltage on the plate of V1B. The varying plate voltage is impressed upon the grid of V1A. As a result the cathode current of V1A varies with a subsequent variation in magnetic forces around the tuning fork. This results in further stresses within the tuning fork, so that the fork will continue to oscillate.

3.4.7 Single-Shot Multivibrator

Single-shot multivibrators are used to change an input pulse to a desired polarity and wave shape. That is, an input pulse to a single-shot multivibrator may have its polarity inverted and at the same time its width may also be altered.

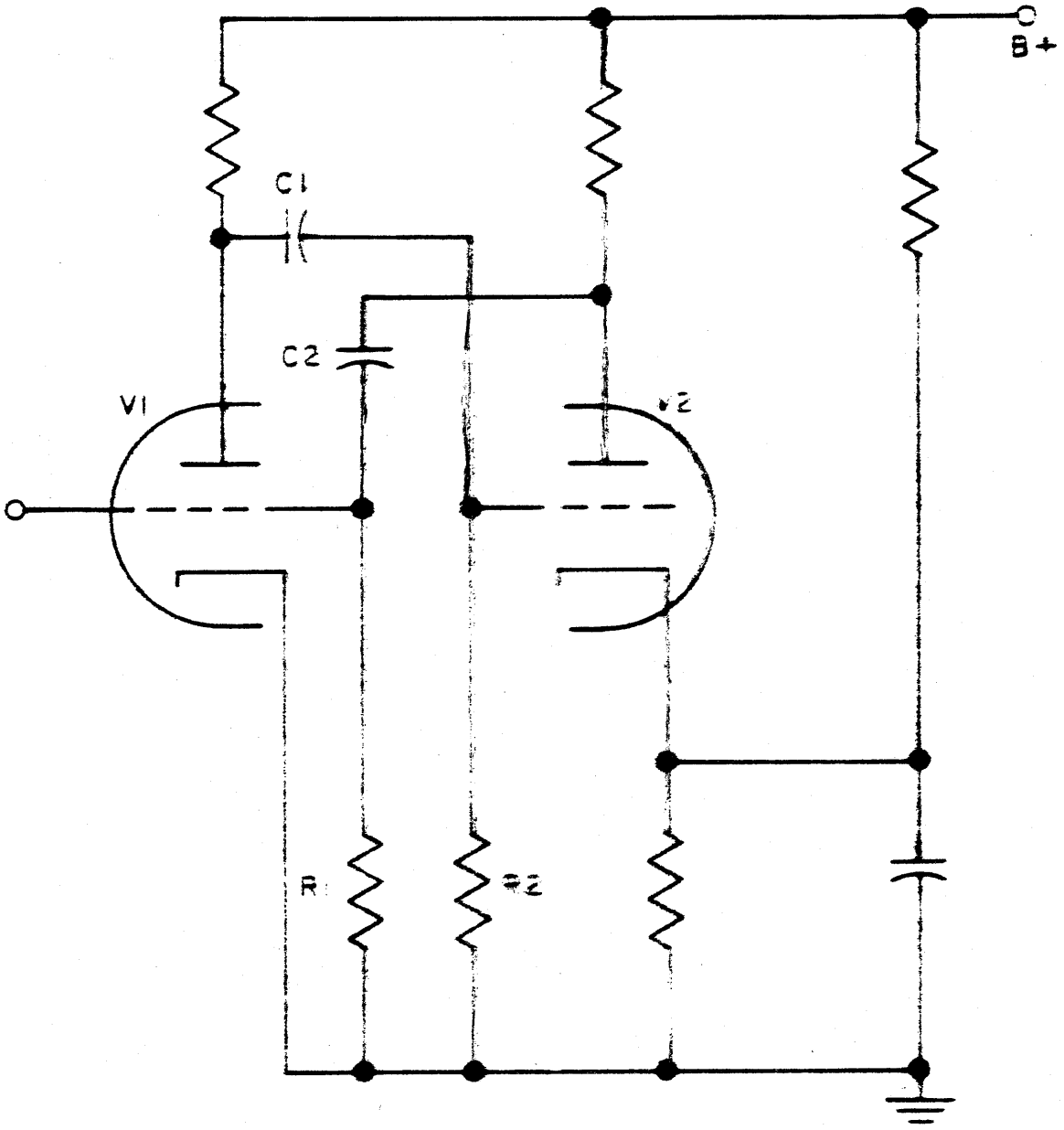


Figure 4-42

The single-shot multivibrator, unlike the flip-flop previously described, has but one stable and one unstable condition. The change from the stable condition is initiated by an input pulse which drives the circuit to the unstable condition. After a period, which is determined by the time constant of the circuits, the circuit returns to its stable condition.

Figure 4-42 illustrates a basic single-shot multivibrator. If V2 is biased to cut-off, a negative signal applied to the grid of V1 causes conduction in V2 by raising the plate voltage of V1 and hence the grid potential on the grid of V2. The conducting time of V2, which represents the unstable condition of the single-shot multivibrator depends upon the time constants C1R2 and C2R1. Thus it is possible to have the output pulse width determined by the RC values in the circuit. However, positive pulses may be fed to the grid of V2 if operation of the circuit is desired when only positive trigger pulses are available. Since the output may be taken from the plate of V1 or V2, and since the circuit may be operated with positive or negative input pulses, the polarity of the output may be the inverse of the incoming signal. In computer circuits the output of a single-shot multivibrator is usually fed through a cathode follower circuit.

3.5 MATRICES

A matrix is essentially a distribution system. It is used in a computer circuit as a central point into which various elements of data emanating from different sources are combined to carry out an instruction.

Matrices are used in a number of different places in a digital computer to perform widely different functions. They are sometimes used in a circuit as a means of converting binary to digital notations. Other applications include their use to set flip-flop circuits arranged in such a way as to cause a biasing voltage to be applied to all the cores of an array, except one. The voltage is such that the cores will be biased into the saturation region. The excepted core is then the only one which is switched when the current pulse from the driver is applied. To reset the switch so that it may be ready again for the next operation, it is only necessary to apply the driving pulse in the reverse direction.

One common type of matrix is known as the Diode Matrix. The diodes are arranged in the circuit in such a way that only pulses of one polarity will be accepted for comparison. Depending on the application, a given number of pulses must arrive at a junction from different points at the same time to represent the binary digit 1. Failure of any pulse to arrive would indicate the digit 0.

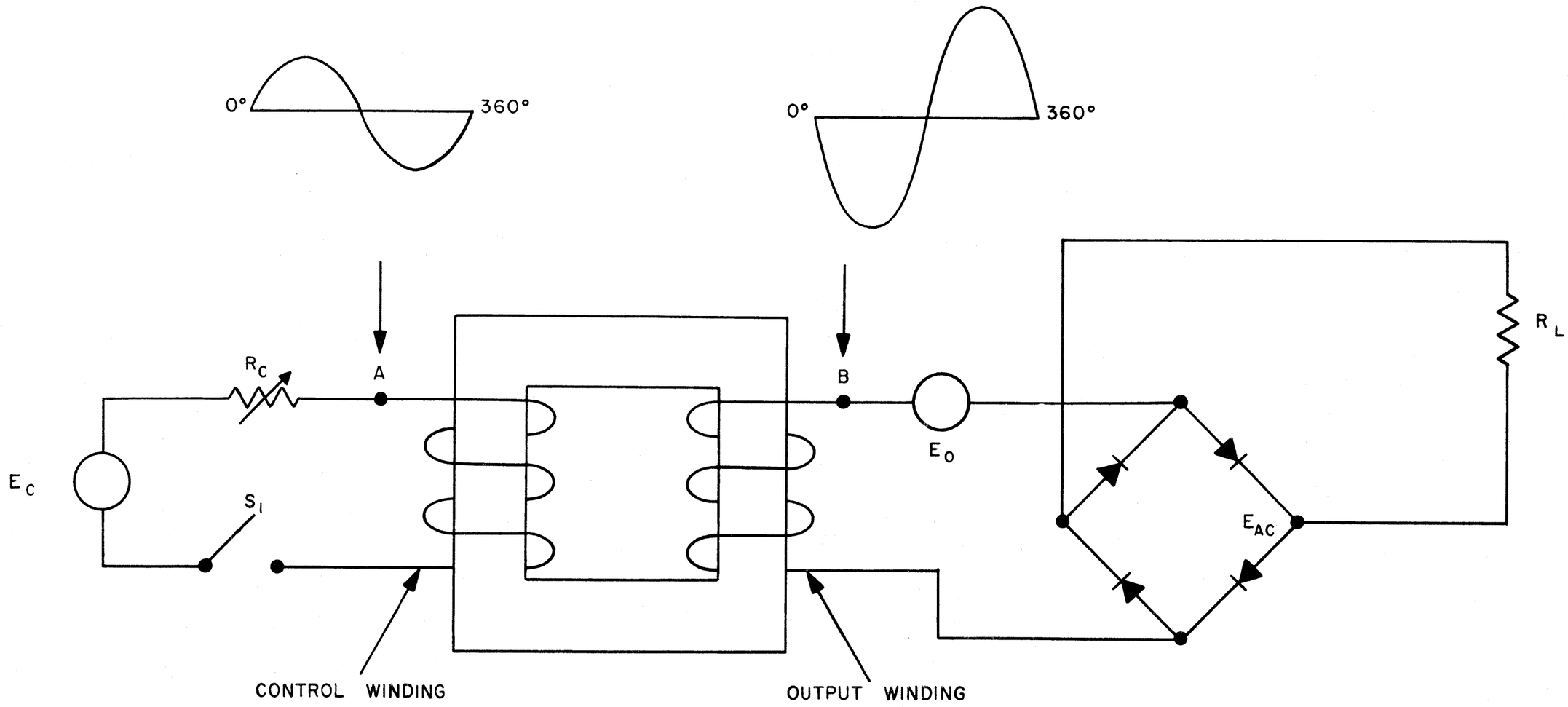

0°
360°
0°
360°
R_C
A
B
E_O
R_L
E_C
S_1
E_{AC}
CONTROL WINDING
OUTPUT WINDING

Figure 4-43

3.6 MAGNETIC AMPLIFIERS

A basic magnetic amplifier comprises a control winding and an output winding wound on opposite legs of a rectangular saturable core. The reactance of the output winding can be varied by varying the degree of saturation of core. This, in turn, is accomplished by employing the control winding to set up a magnetic field in the core which opposes the field created by the output circuit current.

Referring to Figure 4-43, which shows a basic magnetic amplifier driving a bridge rectifier, assume that switch S1 is opened as shown. Then the current driven through the output winding by output generator E_o saturates the core so that the output winding offers a low impedance to current in the output circuit. Suppose, on the other hand, that S1 is closed. The phase relationship between the control generator E_c and the output generator E_o (which is illustrated in the figure in terms of the signal at point A of the control circuit and the signal at point B of the output circuit) is such that at every instant the control circuit current produces a field which opposes the field produced by the output circuit current. Thus, the field produced by the control circuit acts to desaturate the core in direct proportion to the amplitude of the signal which appears across the control winding. This, in turn, can be varied by varying R_c. In this way, the impedance of the output winding can be varied so that a selected proportion of E_o appears across the bridge rectifier. Thus, the magnitude of the d-c voltage which appears across the load R_L can be controlled by adjustment of the R_c.

The control circuit of the magnetic amplifier is comparable to the grid circuit of a vacuum tube, while the output circuit is comparable to the plate circuit of a vacuum tube. However, in a vacuum tube, current in the plate circuit can be entirely cut off by making the grid sufficiently negative. In the case of the magnetic amplifier, on the other hand, the control current can never be made large enough to completely desaturate the core. Thus, a complete cut off of output circuit current can never be obtained. The condition of minimum output current is therefore defined to be the cut-off condition of a magnetic amplifier.

As already noted, the two-winding magnetic amplifier is a basic unit. Practical magnetic amplifiers may have, in addition to the two basic windings, bias windings are feedback windings.

BASIC THEORY OF DIGITAL COMPUTERS

DC1

PART 5

COMPUTER ORGANIZATION

Draft No. 2

INTERNATIONAL BUSINESS MACHINES CORPORATION

KINGSTON, NEW YORK

PART 5

COMPUTER ORGANIZATION

CHAPTER 1

SYSTEM CONSIDERATIONS

1.1 GENERAL

In Part 1 of this book the functions that must be performed as a part of any computation were classified as arithmetic, storage, control and input-output. In Part 4 various components for implementing these functions were introduced. The purpose of Part 5 is to demonstrate how such components are organized to form a computing system. However, since computing systems operate upon representations of information, it is convenient to introduce two terms which define units of information before beginning a discussion of systems. This is done in the following paragraph.

1.2 BITS AND WORDS

A number is a specification of quantity or order. For example, the number seventeen may indicate that there are seventeen individuals in a group or it may indicate that a particular individual is seventeenth in a group. A number may be represented by a word or by a set of symbols. The ten symbols 0 through 9 of the decimal system are called digits. The two symbols 0, 1 of the binary system are called bits.

Much of the data represented in a digital computer is numeric in content, that is it specifies quantity or order.

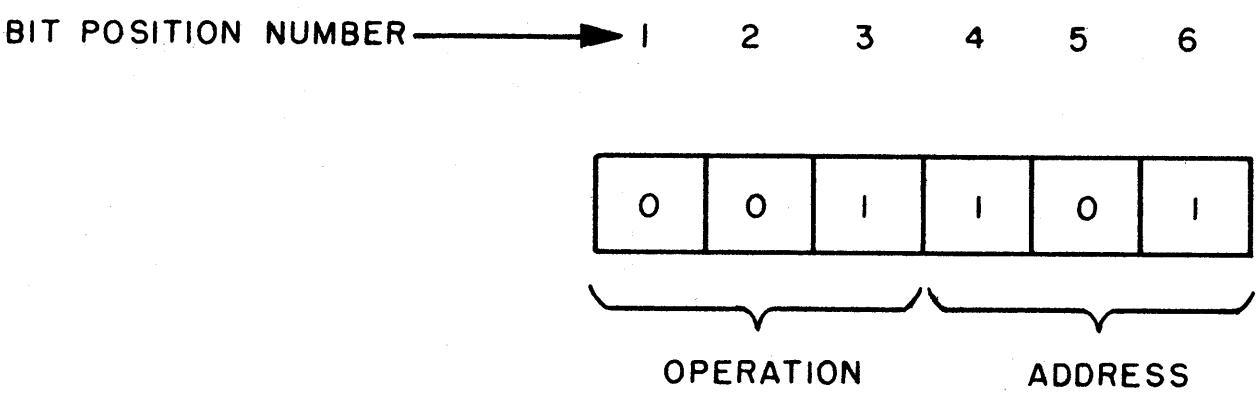


Figure 5-1

Moreover, the data that is non-numeric in content may still be considered numeric in form. The command add, for example, is specified by a pattern of states of flip-flops or other such devices which can be interpreted as a number. It is, therefore, impossible to specify a non-numeric item of information without specifying, at the same time, a number; just as it is impossible to dial an exchange letter on a telephone without, by the same motion, dialing a digit. Telephone circuits do not distinguish the letter from the number. The alphabetic coding on the dial is entirely for the convenience of the human user. The same thing is true of digital computers. The concept add may, in code form, be identical to the number eleven, for example.

Because computers in general represent information in the form of patterns defined by the states of binary devices, the state of one device is said to represent one bit of information. Since patterns of bits may specify information other than quantity or order, the term number is not much used. Instead, a set of bits which is interpreted on the basis of the pattern formed by its individual members is called a word. A single word may contain both non-numeric and numeric information. For example, an instruction word has, in terms of content, a non-numeric part which defines the operation to be performed and a numeric part which specifies the address or addresses of the operand or operands. Such a word is illustrated in Figure 5-1. Notice that in form it is numeric.

However, the first three bits are interpreted as non-numeric information. This should not be taken to mean that the first three bits of any word in a particular computer will always be given a non-numeric interpretation. Instead, the interpretation of a word will depend upon where it is stored, since this, in turn, will define the circumstances under which it is brought forward to be operated upon.

1.3 TYPES OF COMPUTING SYSTEMS

1.3.1 Special Purpose Versus General Purpose Computers

The choice of components and the manner of their organization into a particular system will depend upon the type of task for which that system is designed. The most general classification of digital computers is that which describes them as either special purpose or general purpose machines. The special purpose computer is one that is designed to solve a single problem. For example, a five control computer solves the problem of aiming a gun so as to hit a target. The input information it receives is always the same, i.e. target position or rate information from a radar set, wind direction and speed and ballistic data, and the outputs it generates are always the same, i.e. gun positioning commands.

The general purpose computer, by contrast, can be used in the solution of many different problems. For this reason its capabilities are much more difficult to describe. A good measure of its flexibility is provided by the number and types of explicit instructions it is able to execute, the capacity and access speed of its storage element and the speed with

which it can perform the arithmetic operations.

In a sense there really is no such thing as a general purpose computer; that is, there is no such thing as a computer which is equally efficient in solving all types of problems. Thus any computer is designed with some general range of problems in mind and its design characteristics will reflect the demands of this range.

1.3.2 Types of Computer Programming

A second way of classifying digital computers is in accordance with the manner of setting up the program of instructions required for a problem solution.

A computer which executes individual instructions immediately as they are received from an input device such as a punch card machine is called an externally programmed computer. The speed of such a machine is limited by the speed of the input device and in addition its versatility is limited by the fact that it cannot choose between alternative routines on the basis of some contingency which occurs during the solution.

When the set of instructions performed during a solution is predetermined by the connections made on a plugboard (similar to a telephone switchboard), the computer is said to be plugboard programmed. This method of programming does not impose the limitation on speed of operation that external programming does, but, on the other hand, it does not afford any increase in versatility.

The most versatile type of computer is the stored program machine. Here, instructions are loaded into a high-access-speed storage device prior to the start of computation. They are then normally referred to in an order determined by the sequence of numbers defining their addresses in storage. This order can be modified as necessary by a branch instruction which "jumps" the computer to an examination of any arbitrary storage address in the event that some particular condition is fulfilled, or allows it to continue its examination of successive locations in the event that that condition is not fulfilled. For example, the computer program might branch to the arbitrary location if some number were positive and continue its successive operation if that number were negative.

Of course, the stored program computer functions as an externally programmed computer during the period when a program is being loaded into its high-access-speed storage device, but this is not too serious a limitation.

In the case of a real-time problem, i.e., one where the outputs are used to control events which are in progress during the solution, a stored program computer executing a cyclic program is capable of providing a continuous solution for an indefinite period of time. The cyclic program is one in which the last instruction in the program is a branch instruction which returns the computer to an examination of the first instruction.

1.3.3 Scientific Versus Data Processing Computers

Another method of classifying computers labels them as either scientific or data processing machines. The division between the two classes is by no means clear cut. However, the general characteristics of each class are as follows:

The scientific computer is capable of handling efficiently problems involving complex mathematical operations, such as integration and differentiation. This requires the ability to perform long routines and to store extensive intermediate results. However, it does not require facilities for handling any very large amount of input or output data.

In the case of the data processing machine, the emphasis is on an ability to handle large amounts of input data and produce large amounts of output data.

1.3.4 Single Address Versus Multiple Address Computer

Items of data are associated with most computer instructions. A command to add, for example, is meaningless unless the numbers to be added are specified. In general an item of data is specified by the address of the location in storage which it occupies. A single address machine is one in which each instruction word specifies the address of just one item of data. By contrast, a multiple address machine is one in which each instruction word can specify more than one address.

In a single address machine the addition of a number, a, to a second number, b, and the storage of the sum, a + b,

might be indicated as follows:

```
ca    13
ad    14
ts    15
```

where:

a is stored at address 13
b is stored at address 14
and address 15 is assigned
as the storage location for a/b.

In this case, the clear and add instruction (ca 13) would cause a to be transferred from storage location 13 to the arithmetic element. The add instruction (ad) would then cause b to be transferred from storage location 14 to the arithmetic element and finally the store instruction (ts 15) would cause the sum, a/b, to be transferred from the arithmetic element to storage location 15.

The same addition could be specified in a multiple address machine by the simple instruction

```
ad 13 14 15
```

where it is the convention that the first two numbers following the instruction code are the addresses of the operands and the last number is the address assigned to the result.

It would appear from the example that exactly three times as many instructions are required to execute any routine using the single address system as are required using the three address system. However, this is not the case.

Suppose, for example, that the addition a+b+c+.....+n is to be performed. The single address machine requires one instruction for each number involved and a final instruction to store the total. The three address machine requires one instruction to add a to b, a second to add (a+b) to c, a third to add (a+b+c) to d and so on. Since the last of these instructions can also specify a storage location for the total, two instructions are saved in adding a sequence regardless of its length. In forming the sum of a long sequence of numbers, this saving is hardly substantial enough to justify the longer word which is required to specify three address instead of one.

To summarize: The single address machine requires the use of more instructions to perform any given routine involving more than one item of data, while the multiple address machine requires longer instruction words in order to specify additional addresses. The number of bits required to specify an address depends upon the total number of storage locations to be identified. Where the number of storage locations is large, the single address type of operation is likely to be chosen in order to minimize the length of instruction words.

1.3.5 Parallel Computers Versus Serial Computers

In a parallel computer each bit of a word is represented by a separate flip-flop or other such device. When the word is transferred from one part of the computer to another, each bit is transferred simultaneously along a separate path.

In a serial computer each bit of a word is represented by the state of some device at a particular instant of time. When the word is transferred from one part of a computer to another, the bits are transferred in succession along a single path.

Parallel operation may be compared to a line of men marching side by side, whereas serial operation may be compared to a line of men marching in single file. The two types of operation may both be used in the same computer. The conversion from one to the other is then analogous to a flank movement by a line of marching men.

Serial operation requires less equipment; however it is slower. The choice between the two types of operation is, therefore, largely a choice between economy and speed.

1.3.6 Decimal Versus Binary Computers

Since most of the components available for use in computers are essentially binary in nature it would seem reasonable to represent and operate upon numbers in the binary system. The only disadvantage of the binary system is that it cannot easily be interpreted or manipulated by human operators. Use of the binary system, therefore, implies conversion of the input and output between the number language of the human operators and that of the machine. For some commercial applications it may be more efficient to represent numbers by decimal codes than to perform the conversions between decimal and binary systems. However, large scale

high-speed computers generally operate upon binary representations of numbers.

1.4 TYPE OF COMPUTING SYSTEM TO BE STUDIED

As stated in Paragraph 1.1 of this chapter, the purpose of Part 5 is to demonstrate how components are organized to form a computing system. As can be seen from a reading of paragraphs 1.3.1 through 1.3.5 there are a number of basic decisions that must be made prior to the design of a particular computing system. These decisions will be made on the basis of the application for which the computer is intended. The result of these decisions will be a computer which can be classified in terms of the types of operation discussed in Paragraph 1.3.1 through 1.3.5.

Before discussing the organization of components to form a computing system, the type of system to be formed should be defined.

In this book a computing system intended for an application similar to that of the AN/FSQ-7 Combat Direction Central will be discussed. This will be a general purpose, stored-program, data processing, single address, parallel, binary computer. Its primary task will be the continuous solution of a real-time problem involving an enormous amount of input and output data.

PART 5

CHAPTER 2

ARITHMETIC ELEMENT

2.1 GENERAL

As explained in Part 4, the components available for performing binary arithmetic are of two types; i.e. registers and logical networks. Either type of component can be used without the other to perform the four arithmetic operations. However, when they are used together, the inherently high speed of logical circuits can be exploited and, at the same time, the versatility of the flip-flop registers can be employed to minimize the quantity of circuitry required.

High speed is a primary requirement for the arithmetic element of a computer that is to be used in a real-time situation. This indicates that the element must be designed for parallel rather than serial operation. An arithmetic element capable of performing all four of the arithmetic operations and satisfying the need for high-speed operation can be built using three flip-flop registers together with a logical network which is essentially a set of full adders.

2.2 ADDITION

Addition can be performed by an element consisting of two flip-flop registers and the set of full adders. The routine is as follows: the augend is entered into one of the flip-flop registers (called the accumulator) and the addend is entered into the other flip-flop register (called the A-register). The

outputs of the two flip-flop registers are gated to the full adders by means of an add command pulse. The sum developed by the adders is gated to the accumulator where it replaces the augend. Incidentally, the addend remains in the A-register, unchanged by the operation. However, this latter feature is more important to the multiplication routine than it is to the addition routine.

As noted in Part 2, the addition process can result in a sum which exceeds the capacity of the machine. This would appear as a carry from the full adder associated with the most significant bit position. Such a carry could be used to initiate an alarm. In general, of course, the programmer has the responsibility of scaling the problem variables in such a way as to avoid exceeding the capacity of the machine.

2.3 SUBTRACTION

Subtraction can be performed by almost the same routine as addition, using the same components. The only extra requirement is that the A-register be able to form the complement of the number it holds. As was shown in Part 4, Section 1.1, 1's complements can be obtained readily in a flip-flop register. The subtraction routine, then, consists of entering the minuend in the accumulator, entering the subtrahend in the A-register, complementing the subtrahend and performing the addition operation. That this is equivalent to subtraction by the usual pencil and paper method is shown in Part 2.

Subtraction, like addition, can produce a result which exceeds the capacity of the computer. For example, the subtraction

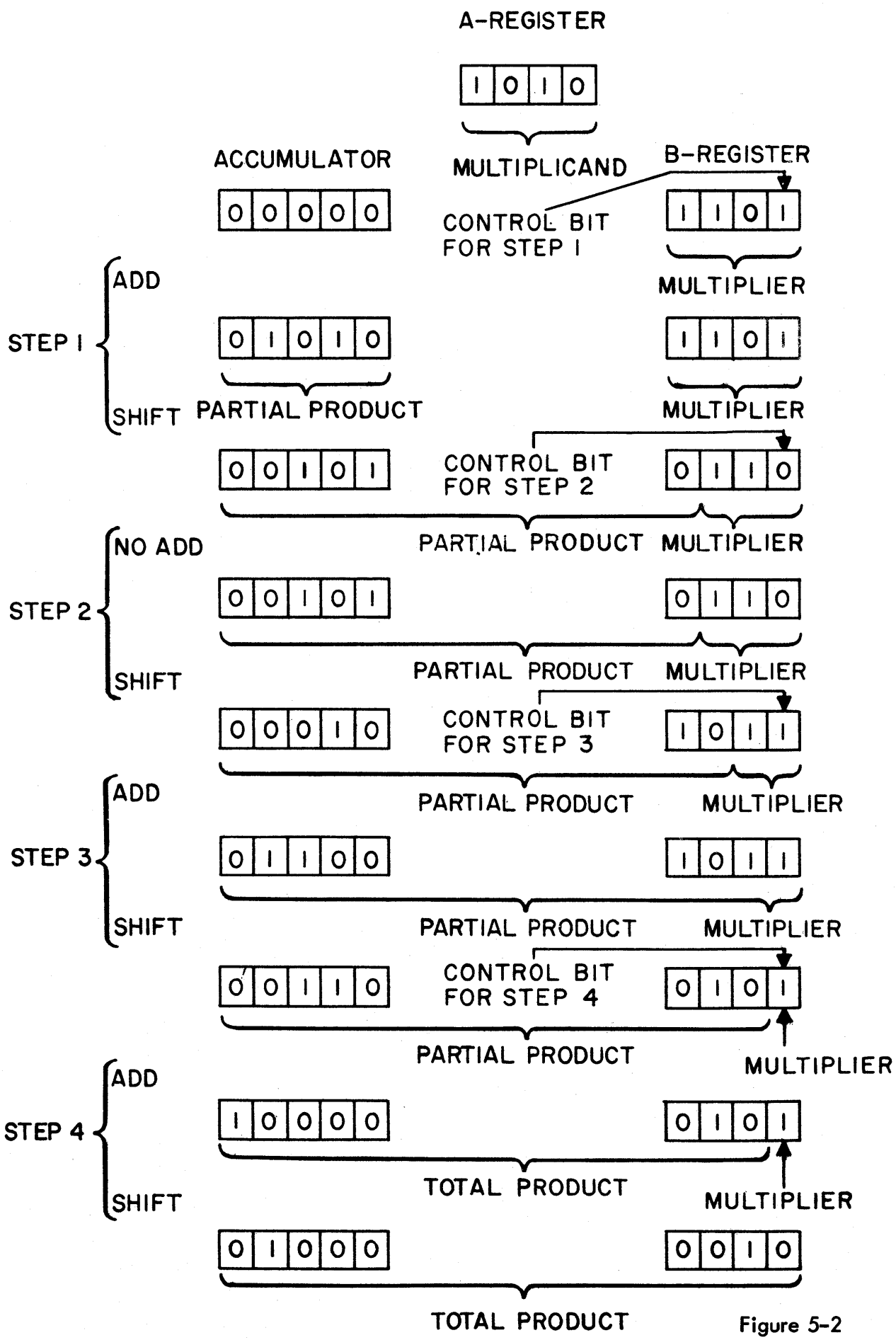

A-REGISTER
1 0 1 0
MULTIPLICAND
ACCUMULATOR
0 0 0 0 0
B-REGISTER
1 1 0 1
CONTROL BIT FOR STEP 1
MULTIPLIER
STEP 1
ADD
0 1 0 1 0
1 1 0 1
SHIFT
PARTIAL PRODUCT
MULTIPLIER
0 0 1 0 1
CONTROL BIT FOR STEP 2
0 1 1 0
PARTIAL PRODUCT
MULTIPLIER
STEP 2
NO ADD
0 0 1 0 1
0 1 1 0
SHIFT
PARTIAL PRODUCT
MULTIPLIER
0 0 0 1 0
CONTROL BIT FOR STEP 3
1 0 1 1
PARTIAL PRODUCT
MULTIPLIER
STEP 3
ADD
0 1 1 0 0
1 0 1 1
SHIFT
PARTIAL PRODUCT
MULTIPLIER
0 0 1 1 0
CONTROL BIT FOR STEP 4
0 1 0 1
PARTIAL PRODUCT
MULTIPLIER
STEP 4
ADD
1 0 0 0 0
0 1 0 1
SHIFT
TOTAL PRODUCT
MULTIPLIER
0 1 0 0 0
0 0 1 0
TOTAL PRODUCT

Figure 5-2

of a negative number from a positive number would result in a difference greater in magnitude than either the minuend and the subtrahend. Again, it is the responsibility of the programmer to scale the problem variables in such a manner as to avoid exceeding the capacity of the machine.

2.3 MULTIPLICATION

High-speed multiplication can be performed by an add and shift routine which employs a third flip-flop register (called the B-register) as well as the accumulator, the A-register and the logical full adders. The numbers held in each of the three registers after each step of such a routine are shown in Figure 5-2. For the purposes of this example, the multiplicand is assumed to be .1010 (decimal .625) and the multiplier is assumed to be .1101 (decimal .8125). This particular multiplication performed by the same add and shift routine is illustrated in Figure 2-34 and is justified from the point of view of arithmetic theory in the accompanying text, Part 2, Section 3.3. The details of the routine are as follows:

a. The multiplicand (.1010) is placed in the A-register, the multiplier (.1101) in the B-register and the accumulator is cleared.

b. The least significant bit of the multiplier is examined. If this bit is 1, an add command pulse is generated, placing the multiplicand in the accumulator (without erasing it from the A-register). This is the case in the example of Figure 5-2. If the least significant bit is 0, no add command is generated and the accumulator remains cleared. In either case, multiplication

by the least significant bit of the multiplier has been carried out and the accumulator contains the first partial product.

c. The contents of the accumulator-B-register, treated as a single register, is now shifted one place to the right. This moves the least significant bit of the first partial product from the accumulator to the B-register. At the same time, the least significant bit of the multiplier is lost. However, this is of no consequence, since the bit has already completed its part in the multiplication routine. (In the example of Figure 5-2, the product bit moved from the accumulator to the B-register is 0 and the multiplier bit dropped is 1.)

d. The second least significant bit of the multiplier is now examined. If this bit is 1, an add command is generated, causing the multiplicand to be added to the shifted first partial product in the accumulator. If the bit is 0, no add command is generated. (This is the case in the example of Figure 5-2.) In either case the number occupying the accumulator and the first bit position of the B-register is now the sum of the first and second partial products. In pencil and paper multiplication, the second partial product is shifted left one place with respect to the first partial product. It should be understood that the right shift of the first partial product prior to the addition of the second partial product is a completely equivalent operation.

e. The contents of the accumulator-B-register, treated as a single register, is now shifted right a second time, preparatory to the addition of the third partial product. After this shift,

two bits of the sum of the partial products are occupying flip-flops of the B-register and the second least significant bit of the multiplier has been dropped.

f. The routine continues in the same way, providing a step for each multiplier bit. At each step, the multiplier bit is examined. If it is 1, an add command is generated. On the other hand, if the multiplier bit is 0, no add command is generated. For each step, there is a shift right of the partial product in the accumulator-B-register combination until, after the final step, the product has completely replaced the multiplier and occupies the entire combined register. Assuming the three registers to have the same capacity, the combined register can hold a product having twice as many bits as the multiplier or the multiplicand. This corresponds to the longest possible product of such a multiplication.

The three registers shown in Figure 5-2 do not have equal capacity. Instead, an extra bit position is provided in the accumulator to accommodate carries from the fourth order arising out of an addition. Such a carry occurs in Step 4 of the multiplication shown. There are reasons why such an inequality in storage capacity is not apt to be found in an actual machine. However, this point will be considered after the division routine has been discussed.

Each of the add operations performed during multiplication is identical to the basic add routine discussed in Section 2.1.2 of this Part, i.e. the number in the A-register is added to the number in the accumulator and the sum is stored in the accumulator.

This is an important consideration, since it means that exactly the same circuitry which is employed to carry out the addition routine can be used to perform the additions required by a multiplication routine. In this way, the complexity of the arithmetic element is minimized.

The multiplication routine, then, employs the components and control circuitry already required for the addition routine. However, it requires a new component, the B-register, and it also requires two new operations, examination of the multiplier bits and shift right. The performance of shift operations in flip-flop registers has been demonstrated in Part 4, Section 1.1. The examination of a multiplier bit is merely a matter of sampling the output of a flip-flop by means of a gate circuit. The control circuitry required for this sampling operation is simplified by the fact that the bit to be examined is always found in the least significant bit position of the B-register. This follows as a result of the shift right operation performed at each step on the combined accumulator-B-register.

2.5 DIVISION

No new components are required to perform division by a subtract and shift routine. The subtractions can be performed by the complement and add method of Section 2.1.3. This implies that the divisor, which plays the part of the subtrahend, will be entered in the A-register. As in multiplication, the accumulator and B-register is used to form a single register of double capacity. At the outset of the routine, the dividend is placed in this combination register. The shift used in this routine

is to the left, making room for quotient bits in the B-register and dropping remainder bits as one step follows another. At the end of the routine the entire quotient is found in the B-register while the final remainder occupies the accumulator.

As is shown in Part 2, the division operation offers several difficulties not encountered in the other arithmetic operations. For this reason, a computer is usually designed so that it will perform division only under certain favorable circumstances. Thus, after the entry of the operands in the proper registers, the first step of a division routine may be a test to discover whether the division operation is permissible. If it is not, an alarm may be actuated. It should be understood that any limitations placed upon division imply a responsibility on the part of the programmer to scale the problem variables in such a way that no illegal division operation situations arise. Division must receive special attention from the programmer in any event, since the operation has the inherent limitation that division by 0 is meaningless.

The test performed to discover whether division is permissible will depend upon the specific nature of the restrictions placed upon the performance of the operation in any particular computer. The computer being developed in this discussion is assumed to operate upon fractional numbers only. Such an assumption implies a very straightforward limitation upon the division; i.e. the absolute value of the divisor must be greater than or equal to the absolute value of the dividend.

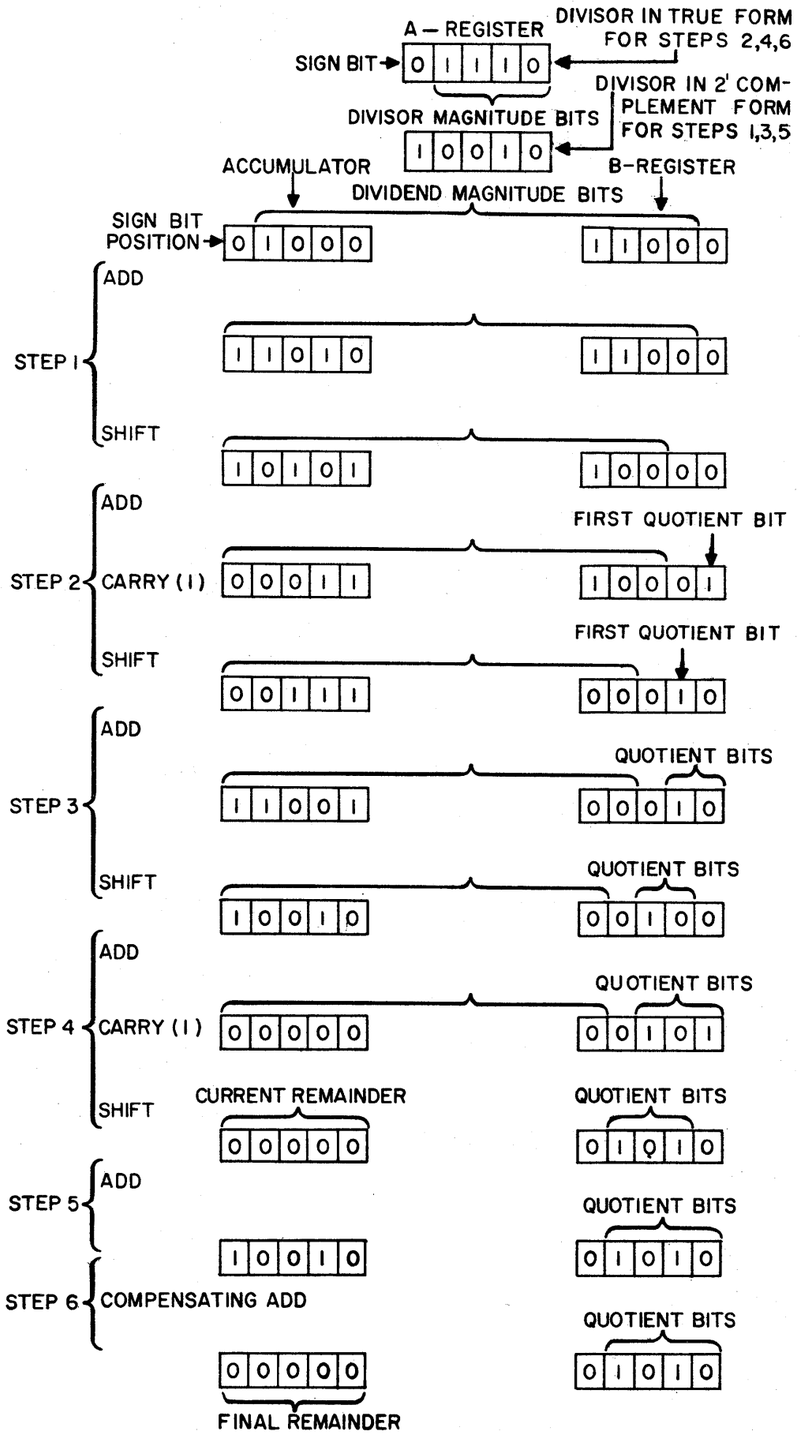

A – REGISTER
DIVISOR IN TRUE FORM FOR STEPS 2,4,6
SIGN BIT
0 1 1 1 0
DIVISOR MAGNITUDE BITS
DIVISOR IN 2' COMPLEMENT FORM FOR STEPS 1,3,5
1 0 0 1 0
ACCUMULATOR
B-REGISTER
DIVIDEND MAGNITUDE BITS
SIGN BIT POSITION
0 1 0 0 0
1 1 0 0 0
STEP 1
ADD
1 1 0 1 0
1 1 0 0 0
SHIFT
1 0 1 0 1
1 0 0 0 0
STEP 2
ADD
FIRST QUOTIENT BIT
CARRY (1)
0 0 0 1 1
1 0 0 0 1
SHIFT
FIRST QUOTIENT BIT
0 0 1 1 1
0 0 0 1 0
STEP 3
ADD
QUOTIENT BITS
1 1 0 0 1
0 0 0 1 0
SHIFT
QUOTIENT BITS
1 0 0 1 0
0 0 1 0 0
STEP 4
ADD
QUOTIENT BITS
CARRY (1)
0 0 0 0 0
0 0 1 0 1
SHIFT
CURRENT REMAINDER
QUOTIENT BITS
0 0 0 0 0
0 1 0 1 0
STEP 5
ADD
QUOTIENT BITS
1 0 0 1 0
0 1 0 1 0
STEP 6
COMPENSATING ADD
QUOTIENT BITS
0 0 0 0 0
0 1 0 1 0
FINAL REMAINDER

Figure 5-3

Otherwise, the quotient will be greater than 1 and the capacity of the computer will be exceeded. The test to discover whether the absolute value of the divisor exceeds that of the dividend is as follows:

The signs of the divisor and dividend are examined. If either is not a positive number, it is converted to a positive number. In pencil and paper notation, this would merely be a case of changing the sign. However, in the computer of this discussion, negative numbers are represented in complement form. Thus to convert a negative number to its absolute value, involves converting it from its complement form to its true form. When both divisor and dividend are in true form, the former is subtracted from the latter. If the difference is negative, the division is permissible.

Figure 5-3 illustrates the states of the three registers throughout a non-restoring, subtract and shift division routine. In this example, the divisor is plus .1110 (decimal .875) and the dividend is plus .10001100 (decimal .5468750). This particular division, performed by the same routine, is illustrated in Figure 2-39 and is justified from the point of view of arithmetic theory in the accompanying text, Part 2, Section 3.4. In this example, the divisor is a four-bit number and four quotient bits are to be generated. Nevertheless, each of the registers is assumed to have five orders. The extra (most significant) bit position is used, in general, to hold a sign bit. A 0 in this position indicates that the number is positive and is in true form while a 1 indicates that the number is negative and

is in complement form. That sign indication is vital in the division routine can be seen from the test step that has just been described.

The details of the division routine are as follows:

a. The divisor is entered in the A-register and the dividend in the combined accumulator-B-register. The dividend is allowed twice as many magnitude bits as the divisor. (In the example, the last bit of the dividend is a 0 and is not, therefore, significant.) With its sign and magnitude bits, then, the dividend occupies all the places of the combined register except the least significant place.

b. The signs of the dividend and the divisor are examined as noted above and if either is negative it is converted to its true form. (In the example, both numbers are positive as indicated by 0's in the sign bit positions of the A-register and accumulator. Thus no conversion is necessary.) The number of conversions necessary is counted and the result of the count, which indicates the sign of the quotient, is stored.

c. The divisor is now complemented and added to the portion of the dividend in the accumulator; i.e. the divisor is lined up left with the dividend and is subtracted from it. If the remainder obtained is negative, then the division is permissible and may proceed.

A negative remainder is indicated by a 1 in the sign bit position of the accumulator (as in the example). A positive remainder is indicated by a 0 in the sign bit position and also causes a carry from the sign bit position. Thus, it is not

necessary to examine the contents of the bit position after the arithmetic operation in order to determine the sign of the remainder. Instead, the sign can be determined from the carry lines of the sign bit full adder.

In the performance of the division routine, 2's complements are used rather than 1's complements. This eliminates the need for corrective end around carry operations which would be difficult to perform in connection with the division routine, since the remainder is shifted left after each operation so that its least significant bit does not remain in the same flip-flop. The need for end around carry corrections in operations involving 1's complements is discussed in Part 2.

d. If the test is successful, the current remainder is a negative number (in complement form). The next step in the non-restoring division routine is, therefore, to shift the remainder to the left one place and then add the divisor to it. (The shift left of the dividend is equivalent to the shift right of the divisor which occurs in the pencil and paper version of the operation.) When the dividend is shifted left, by shifting the contents of the combined accumulator-B-register, the sign bit is lost and the most significant magnitude bit of the remainder occupies the sign bit position of the accumulator. The divisor, in true form, is now added to the contents of the accumulator. Notice that the sign bit of the divisor (which is 0) is lined up with the most significant magnitude bit of the remainder as a result of the shift operation. If the result of the addition is a positive current

remainder is 0 corresponding to the fact that the dividend is an integral multiple of the divisor.)

As has been noted, the division routine begins by converting both divisor and dividend into positive numbers. If both are initially positive numbers, no conversion operation is necessary. If neither is initially positive two such operations are necessary. If one is initially positive and the other initially negative, one such operation is necessary. This can be re-stated as follows: if the numbers are of like sign an even number of

remainder, then half the divisor has been successfully subtracted from the dividend. That is, in the first step, the full divisor has been subtracted from the dividend leaving a negative remainder, while, in the second step, half of the divisor has been replaced. Thus, the net amount removed is one-half the divisor. Since the positive remainder (i.e. the successful division) is characterized by a carry from the sign bit adder, this carry is used to enter a 1 in the least significant bit position of the B-register. If no carry occurs (that is, if the net removal of one-half the divisor leaves a negative current remainder) the least significant bit position of the B-register remains cleared. In either case it now contains the most significant bit of the quotient (i.e. the bit which indicates whether one-half the divisor is smaller than or equal to the dividend).

e. The routine continues in the same way providing one step for each quotient bit that is to be generated. As the current remainder moves left out of the B-register, the vacated places in the B-register are occupied by quotient bits. At the end of the routine, the quotient occupies the entire B-register (except for the sign bit position which is vacant) and the remainder occupies the accumulator. If the last quotient bit is 0, then an extra addition must be performed so that the final remainder will be positive. (This is the case in the example. However, in this case, after the final addition, the

operations is required while if they are of opposite sign an odd number of operations is required. This can be compared to the rule for division of signed numbers which states that if the divisor and dividend are of like sign then the quotient is positive while if they are of unlike sign then the quotient is negative. Thus conversion operations at the start of the routine can be counted by a single flip-flop. If an even number are performed, the flip-flop will hold a 0; if an odd number are performed it will hold a 1. At the end of the operation, the contents of this sign counter flip-flop can be transferred to the sign bit position of the B-register. At the same time, if the stored sign bit is 1, the number in the quotient and remainder must be complemented, since negative numbers are represented in complement form in this computer.

In summary, it should be noted that while division, employs the same number of registers as multiplication, it requires the performance of several new operations. These are as follows:

a. Left shift of accumulator-B-register.

b. Connection of sign bit adder carry output to least significant bit position of B-register.

c. Complementation of contents of the combined accumulator-B-register (in order to convert the dividend to a positive number in the case when it is initially negative and in order to convert the quotient and remainder to complement form when the divisor and dividend are of unlike sign.

d. Initial examination of the sign bits held in the A-register and the accumulator.

e. Counting of complementation operations by sign flip-flop counter.

2.6 THE FOUR ARITHMETIC OPERATIONS CONSIDERED TOGETHER

A comparison of Figures 5-2 and 5-3 reveals that larger capacity A- and B-registers were assumed for the division routine than for the multiplication routine. This was done to provide sign bit positions, which were absolutely essential to the performance of the division operation. The comparison, then, calls attention to the fact that no facilities were provided for handling signed numbers in the multiplication routine. However, there is no reason why the extra facilities developed for division cannot be used to extend the multiplication routine to cover the case of signed numbers. If this is done, then the first step in the multiplication routine will be an examination of the signs of the multiplicand and multiplier. If either is negative, it can be converted, by a complementation operation, into its absolute value. The number of complementation operations can be counted by the sign flip-flop and stored. The actual multiplication can be performed upon the absolute values. At the end of the routine, the product can be complemented if the contents of the sign flip-flop is 1 indicating a negative product.

In the case of the addition and subtraction operations, negative operands remain in complement form throughout the routines. Therefore, there is no need to employ the sign

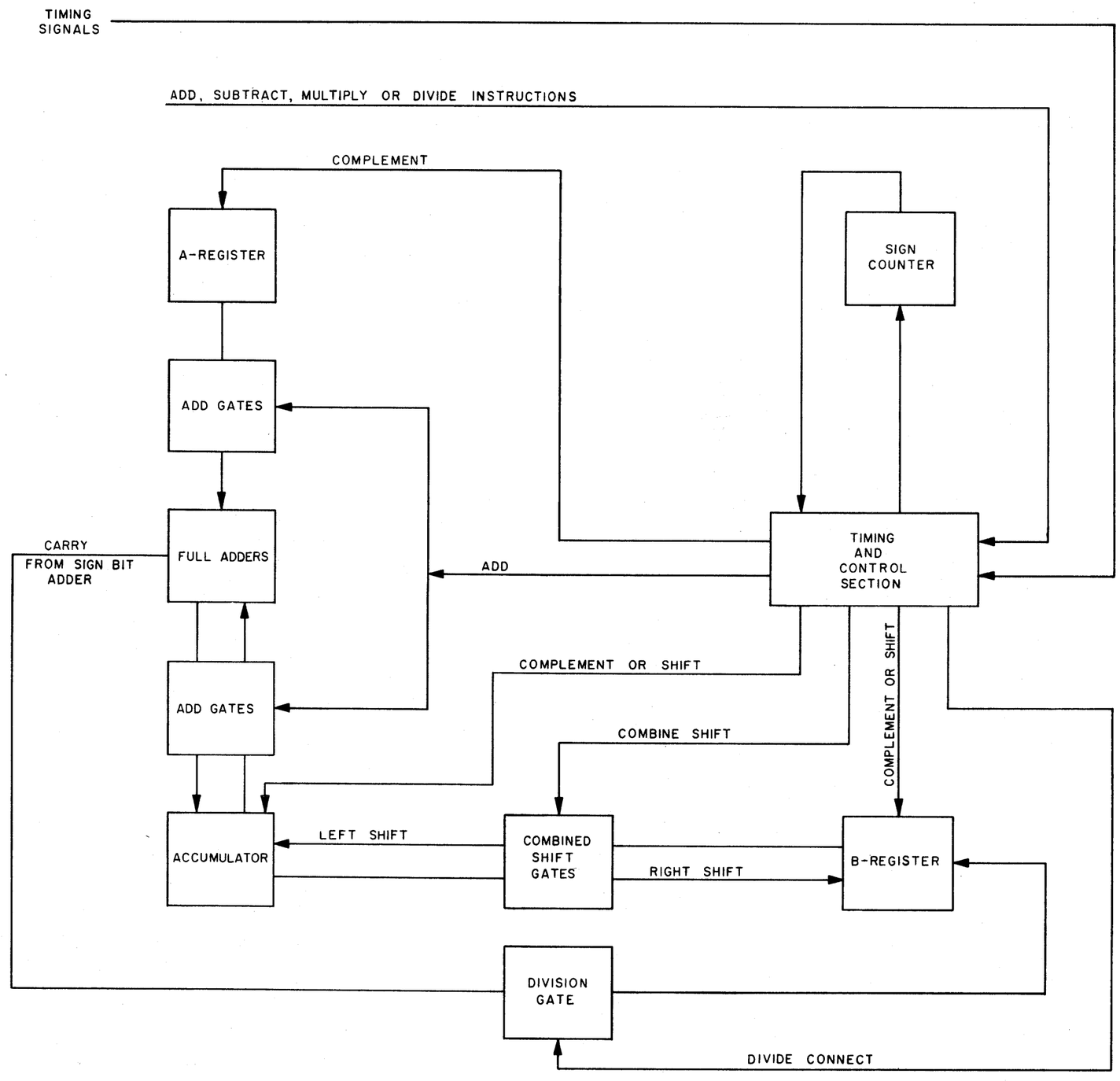

TIMING SIGNALS
ADD, SUBTRACT, MULTIPLY OR DIVIDE INSTRUCTIONS
COMPLEMENT
A-REGISTER
SIGN COUNTER
ADD GATES
CARRY FROM SIGN BIT ADDER
FULL ADDERS
ADD
TIMING AND CONTROL SECTION
ADD GATES
COMPLEMENT OR SHIFT
COMBINE SHIFT
COMPLEMENT OR SHIFT
ACCUMULATOR
LEFT SHIFT
COMBINED SHIFT GATES
RIGHT SHIFT
B-REGISTER
DIVISION GATE
DIVIDE CONNECT

Figure 5-4

counter flip-flop. However, the sign bit positions provided by the larger capacity registers are required in order to indicate whether the magnitude bits represent a positive number in true form or a negative number in complement form.

The internal flow of commands and data in an arithmetic element which performs the four arithmetic operations in the manner discussed is shown in Figure 5-4. It is worthwhile, at this point, to list the set of facilities upon which the performance of the four operations depends.

a. An A-register in which numbers can be complemented.

b. An accumulator in which numbers can be complemented and shifted either to the right or left.

c. A B-register in which numbers can be complemented and shifted either to the right or left.

d. A set of full adders with a capacity equal to that of the registers.

e. The gate networks necessary to implement all the commands that are shown in Figure 5-4.

f. Sequencing and timing elements which decode the add subtract multiply and divide instructions; i.e. convert those instructions into the sequence of commands necessary to implement the operations called for.

In addition to the provisions for internal data flow which are shown in Figure 5-4, provision must be made for entering and withdrawing operands from the arithmetic element. This activity calls for synchronization of the arithmetic element operation with operation of the memory element in which operands are stored.

Thus master timing signals, presumably from some control element exterior to the arithmetic element are shown entering the arithmetic element timing section.

Before the control of computer operations can be discussed, it is necessary to develop a specific storage element. This will be done in Chapter 3. Then in Chapter 4, computer control can be discussed in terms of the arithmetic element developed in this chapter and the storage element developed in Chapter 3.

PART 5

CHAPTER 3

STORAGE

3.1 TYPES OF STORAGE

The types of storage required by a computer may be compared to three storage devices used by the individual; namely a wallet, a desk and a box in the attic. In the wallet are kept those items of data which must be produced upon demand. For example, it is here that the individual keeps his driver's license to which he may require access at any moment. In the desk drawer are kept those items of data for which the need can be anticipated. For example, a man may store his fishing license in the desk drawer during the week and transfer it to his wallet before starting out on a weekend fishing expedition. The box up in the attic is used for the storage of items of data which may not be needed for a long time. Here, for example, a man may keep his cancelled checks.

The function of the primary storage device in a computer may be compared to that of the wallet. Here are stored those items of data which must be produced without advance notice when called for by an instruction. Here also, in a stored program computer, are stored those instructions comprising the program in progress.

A digital computer operates in step-by-step fashion. It brings forward an instruction from the storage element to the arithmetic element. If the instruction calls for an item of

data, the computer then brings forward that item of data from the storage element. Finally, it performs an operation upon the data. The operating speed of the computer depends, therefore, as much upon the time required to transfer instructions and data between the primary storage element and the arithmetic element as it does upon the operating speed of the arithmetic element.

The data processing computer operating in a real-time situation must have as short a memory cycle as possible, where the memory cycle is defined as the time required to transfer a single word into or out of the primary storage. Since machine failure during a real-time solution may be disasterous, the reliability of the storage element is also extremely important. Both these requirements suggest the magnetic core type of primary storage discussed in Part 4. This storage medium allows memory cycles on the order of a few microseconds. In addition, it does not lose information if power is temporarily removed and is relatively insensitive to spurious electrical disturbances. Moreover, the reliability of magnetic core storage is increased by the relatively high level of its output signals which minimizes the amount of associated circuitry required.

The secondary or auxiliary storage of a computer is comparable to the desk drawer mentioned above. Here are kept those items of data for which the need can be anticipated. An example of this type of data is that of tables of functional values which are required during a particular part of the solution only. In the case of this example, a set of values over some range of the independent variable which is of immediate

interest can be transferred from auxiliary to primary storage. As the range of interest changes, the set of values in primary storage can be constantly modified by dropping those values no longer within this range back into auxiliary storage and bringing forward other values which are entering this range.

Another example of information that may be held in auxiliary storage is that of a program which, although not currently in use, may be required more quickly than it can be loaded into the computer by means of an input device. For example, in a military application, a computer may be required to execute a particular program in response to some tactical situation which may develop without much advance warning. Such a tactical program is normally held in auxiliary storage during execution of other programs.

It is apparent that the access speed requirement of the auxiliary storage is not as demanding as in the case of primary storage. Here, the consideration of capacity versus size may be more significant. For the auxiliary storage of the real-time computer, reliability is still a vital concern.

Magnetic drums are a reasonable choice as the auxiliary storage for a computer using magnetic cores for primary storage. Access time for a drum memory is on the order of milliseconds rather than microseconds as in the case of the core memory. However, a drum is considerably more compact a storage medium than a core array. Therefore, where access time is not critical, its space-saving characteristics should be taken advantage of.

In many applications, of course, the access speed of the drum memory may be sufficient to allow its use as the primary storage device. Where this is true, auxiliary storage may be provided by magnetic tape or punched card devices.

In addition to the primary and auxiliary storage devices, there is another level of storage associated with a computer. This is external storage which may be compared to the box in the attic mentioned in the first paragraph of this chapter. Here is the dead storage department for computer programs and data. Storage media such as punched cards and magnetic tape can hold an almost unlimited amount of information without becoming so bulky as to be impractical. They are, therefore, ideal for keeping permanent records of phenomena observed during computer solutions. Such data may not be used for long periods of time but it is important to have it recorded so that it is available for subsequent analysis. Various programs that may be executed by the computer at infrequent intervals may also be stored externally on tape or punched cards.

In addition to primary, auxiliary and external storage, a computer may require buffer storage devices. The requirement for buffer storage between elements A and B implies that for some reason information cannot be transferred directly between the two elements. The most common reason for this is incompatibility of timing between operations of one element and the other. It may, for example, be necessary to have buffer storage between the slow-speed input devices or output devices and the high-speed computer proper. In the case of the data

processing machine operating in a real-time situation the capacity of such buffer storage may have to be considerable.

The computer employing magnetic core arrays for primary storage and magnetic drums for auxiliary storage and having the requirement for a large capacity buffer storage between the input system or output system and the central computer will probably employ magnetic drums for that buffer storage. The speed of the drum system is high compared to that of the input devices but low compared to that of the internal primary storage. Thus, it provides a transitional stage in matching the speed of input operations to the speed of central computer operations.

3.2 MAGNETIC CORE STORAGE

The choice of a magnetic core array as the primary storage device for a computer has important effects upon the organization of that computer. In the first place, it affects the design of the arithmetic element which must now provide operational speeds matching the high access-speed provided by the core storage. In the second place, it requires particular logical combinations of circuitry in order to select specified storage locations and perform write and read functions.

In order to understand the requirements for logical circuitry to implement the transfer of information into and out of the core memory, it is necessary to specify the following:

a. the form in which information is stored on the cores,

b. the mechanism by which information is written onto the cores,

c. the mechanism by which information stored on the cores is sensed during the read-out process, and

d. the organization of the core array into individual storage locations.

In the discussion which follows, a core array such as is introduced in Part 4, Section 2.2.6.2 is assumed. A review of the specifications of this array follows.

Information is sorted in the cores in binary form; that is, a single bit of information takes the form of either one of two possible states of magnetization. Magnetization in one direction is interpreted as a 1 while magnetization in the other direction is interpreted as a 0.

In order to write 1's into all the cores of a particular memory location, so-called half write pulses are applied to two coils (designated as x and y) associated with each of the cores of that location. The effect of the simultaneous appearance of these two pulses is to drive each core into the magnetic state representing a 1. In general, of course, an item of information contains 0's as well as 1's. If all the cores of a storage location are cleared, i.e. driven to 0, prior to writing an item of data into that location, then the 0's of the item may be produced simply by inhibiting the writing of 1's on those cores where 0's are to be stored.

This can be done by applying an inhibit current pulse to a third coil associated with each of the cores where a 0 is to be stored, just at the instant when the half-write currents are applied. The polarity of the inhibit current must be opposite to that of the write currents so that it cancels out a part of their effect.

Notice that in the scheme for writing information in the core array which has just been described, the half-write currents perform a selection function; that is a word is written into that set of cores which receives two half-write currents. The form of the word which is written onto this set of cores is determined by the pattern of inhibit pulses which is applied. For example, if every core of the set receives an inhibit pulse then each of the cores stores a 0. On the other hand, if none of the cores receives an inhibit pulse, then each of the cores stores a 1.

In order to sense a word stored in a particular storage location, so-called half-read current pulses are applied to the x and y coils associated with the cores of that location. The half-read currents are equal in magnitude but opposite in polarity to the half-write currents mentioned earlier. The effect of the simultaneous reception of two half-read pulses is therefore, to drive a core into the magnetic state representing 0. If a 0 is stored in the core, this results in no change of magnetic state. If, on the other hand, a 1 is stored in the core, it reverses its magnetic state. This polarity reversal induces a voltage pulse in a sense winding. The read-out process therefore causes pulses to appear on the output lines associated with each of the cores in the selected location that contains a 1.

The half-read pulses, like the half-write-pulses, perform the selection function. The form of the word read-out is, on the other hand, established by the signals appearing on the sense windings.

In summary, four coils are associated with each core; i.e. the x and y or selection windings, the inhibit winding and the sense winding. Two other facts should be noted at this point; the write-read scheme outlined requires that the cores of a location be cleared (i.e. driven to 0) prior to writing, and the scheme for reading out information held by the cores results in the information being erased from the cores as it is sensed. This latter effect is called destructive read-out.

The selection scheme involving x and y windings is based upon the organization of the core array into a number of x groups and a number of y groups. The coils of any one x group or of any one y group are connected in series. A set of cores which belongs to the same x group and the same y group comprises a storage location. Thus to select a storage location for reading or writing it is merely necessary to apply half-read or half-write current pulses to two input terminals, namely the input terminals of the particular x and y groups which specify the particular storage location. It should be understood that cores of many locations belong to each x group and each y group. However, when a core is supplied with only one half-write or half-read current its magnetic state is not affected. For this reason it is said to be only half selected.

Since information is transferred into or out of only one location in the array at any one time, groups of sense windings or groups of inhibit windings can be series connected on

a bit basis. For example, the cores which store the first or most significant bit in each of the locations of the array can have their sense windings connected in series. They can also have their inhibit windings connected in series. The same thing is true of the cores which store the second bit, and the third, and so on. This means that the same set of output lines can be used to sense the bits of every location in the array and also that the same set of input lines may be used to perform the inhibit function for every location in the array.

Now that the organization of the array has been reviewed, it is worthwhile to return to two characteristics of the read-write scheme which were mentioned earlier:

a. A memory location must be cleared of 1's prior to the writing of information on it.

b. Read out from the core memory is destructive.

Since read out is destructive, a read operation can be used to clear a location prior to the performance of a write operation. Also, since it is usually desirable to retain in a memory location a "copy" of the word read out of it, a write operation will normally follow the destructive read operation. These two facts can be re-stated as follows: regardless of whether the purpose of an operation is the transfer of information out of the core memory or the transfer of information into the core memory, the same basic cycle must be performed, i.e. a read operation followed by a write operation.

When the purpose of the cycle is to transfer a word out of the core memory, the process is as follows:

Half-read current pulses are supplied to the x and y coils of the specified storage location causing all those cores containing 1's to be driven to the 0 state. The polarity reversal of these cores is sensed and caused to condition a previously cleared buffer register which stores the information temporarily. Notice that this buffer register must be cleared prior to receiving the sense pulses for, otherwise, it might contain some 1's which did not originate in the core location. The read part of the cycle is followed by a re-write part during which the 0's of the number just read out of the storage location are used to provide inhibit pulses as that location is again fully selected, this time by x and y half-write pulses. The result is that the word removed from storage is re-written in the same location. However, a "copy" of it remains in the buffer register ready for transfer to some other part of the computer.

When the purpose of the cycle is to transfer a word into the core memory, the process is as follows:

The storage location is first fully selected by x and y half-read currents just as in the previous case. However, in this operation, the voltage pulses induced in the sense windings by the cores containing 1's are of no interest, since the operation is performed merely to clear the location for the receipt of the new word. Moreover, it is reasonable to employ the same buffer register used in the previous case to store the word read out of memory, for the purpose of storing

3.3 MAGNETIC DRUM STORAGE

Considered from the point of view of computer organization, the requirements of drum storage are less complex than those of core storage. This is because addresses can be selected and read and write functions performed in a more straightforward manner.

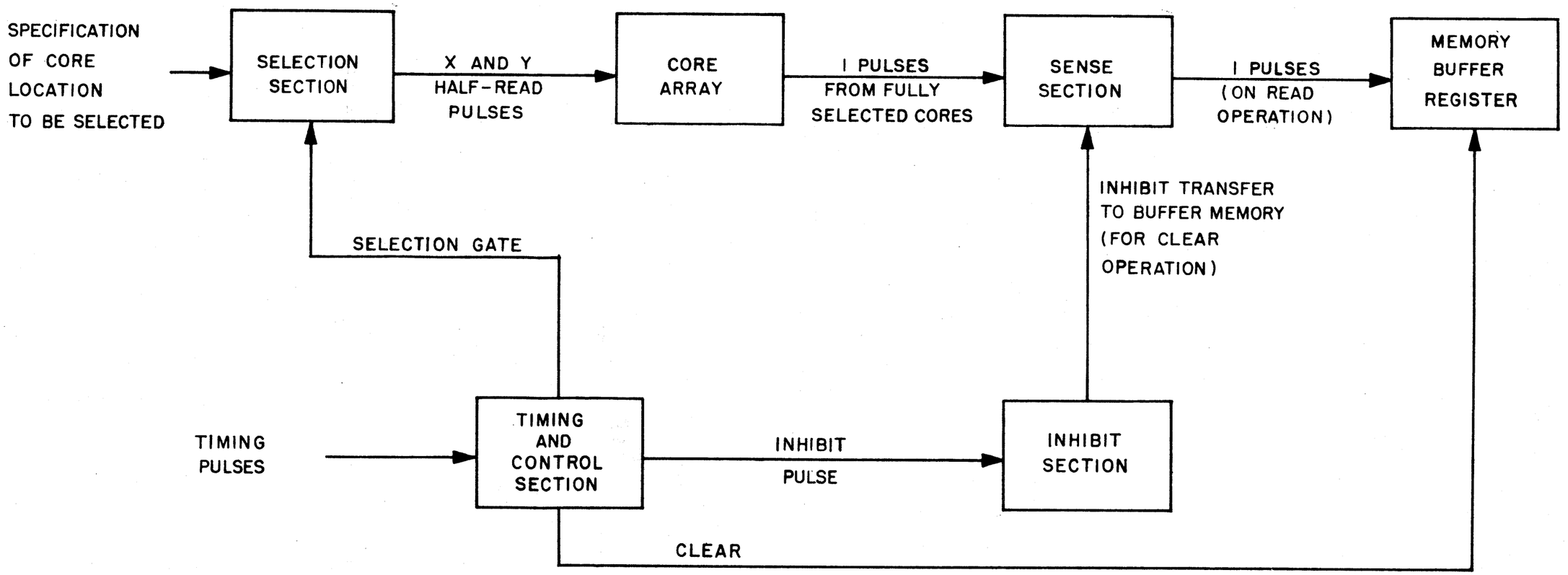

SPECIFICATION OF CORE LOCATION TO BE SELECTED
SELECTION SECTION
X AND Y HALF-READ PULSES
CORE ARRAY
1 PULSES FROM FULLY SELECTED CORES
SENSE SECTION
1 PULSES (ON READ OPERATION)
MEMORY BUFFER REGISTER
SELECTION GATE
INHIBIT TRANSFER TO BUFFER MEMORY (FOR CLEAR OPERATION)
TIMING PULSES
TIMING AND CONTROL SECTION
INHIBIT PULSE
INHIBIT SECTION
CLEAR

Figure 5-5

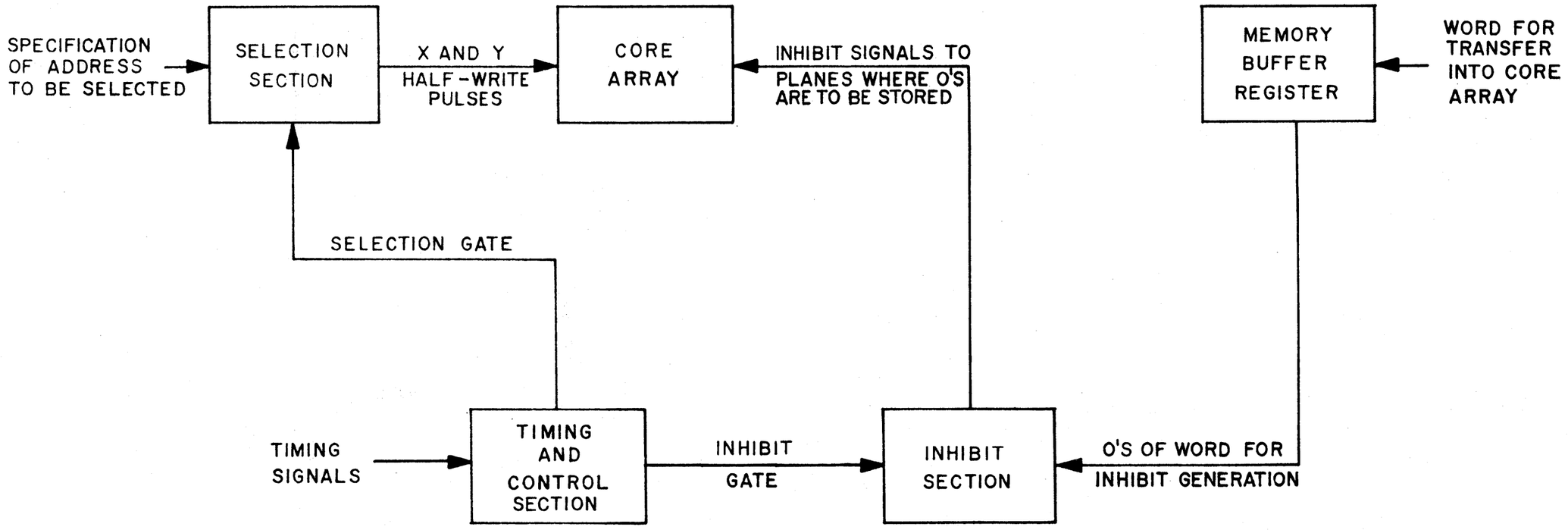

SPECIFICATION OF ADDRESS TO BE SELECTED
SELECTION SECTION
X AND Y HALF-WRITE PULSES
CORE ARRAY
INHIBIT SIGNALS TO PLANES WHERE O'S ARE TO BE STORED
MEMORY BUFFER REGISTER
WORD FOR TRANSFER INTO CORE ARRAY
SELECTION GATE
TIMING SIGNALS
TIMING AND CONTROL SECTION
INHIBIT GATE
INHIBIT SECTION
O'S OF WORD FOR INHIBIT GENERATION

Figure 5-6

the new word to be read into memory. This is convenient, since the write part of this operation is identical to the re-write part of the previous operation. Therefore, the output of the sense windings must be prevented, in this case, from conditioning the buffer register. After the storage location has been cleared by the half-read pulses, it is fully selected again, this time by means of x and y half-write pulses. Simultantaneously, the 0's of the word in the memory buffer which is to be entered into storage are used to generate inhibit pulses. Thus, the new word is written into the selected location.

The various functional blocks necessary to implement the transfer out (read) and transfer in (write) operations just discussed are illustrated in Figures 5-5 and 5-6. Figure 5-5 shows the read function. It also shows the clear function (i.e. reading with transfer to memory buffer register inhibited) that must precede the transfer of information into the core memory. Figure 5-6 shows the write function. Notice that in addition to the functions explicitly referred to in the discussion above, there has been added a timing and control function. The idea of timing is implicit in the basic read-write cycle which is used to implement both read and write operations. Since this cycle must be synchronized with operations in other parts of the computer, timing signals are shown entering the timing and control section. Presumably these arise in timing circuits in other parts of the computer.

Read-out from drum storage is not destructive. Moreover, it is not necessary to clear a drum register (storage location) prior to writing on it.

A magnetic drum is driven at a nearly constant speed. Data is written onto it from a set of stationary heads and read from it by the same set of heads or in some cases by a second set of such heads. As the drum rotates through 360 degrees a complete circle on the perimeter of the drum passes under each head. Each such circle is called a track or channel. In parallel operation, each bit of an individual word is recorded on a separate channel. This is accomplished by supplying the pulse representing each bit to a separate write head at the same instant. The location where the word is written depends upon the angular position of the drum and, since the drum rotates at a constant speed, this is a linear function of time. A set of channels which can accommodate all the bits of a word plus some associated control bits is called a field. A single drum may accommodate several fields. Each field may contain thousands of individual registers. It should be understood that there are no physical divisions on the surface of the drum. Fields and channels are defined by the positioning of the heads while registers are defined by the timing of the write impulses.

Selection of a drum register for reading or writing is merely a question of sensing the pulses induced on the read heads of a particular field at a particular time or applying write pulses to the write heads of a particular field at a particular time. Since drum systems, in general, comprise

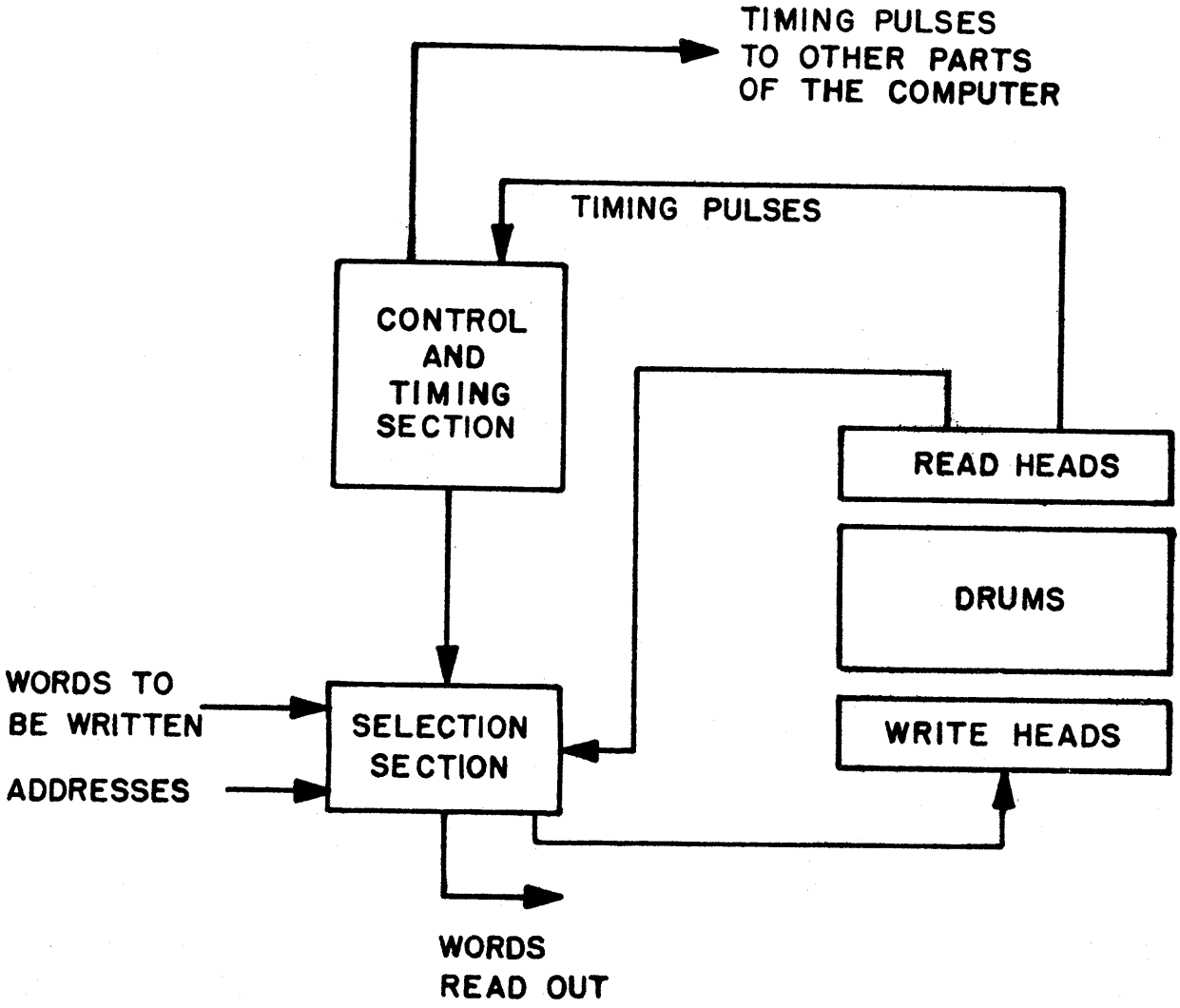

TIMING PULSES
TO OTHER PARTS
OF THE COMPUTER
TIMING PULSES
CONTROL
AND
TIMING
SECTION
READ HEADS
DRUMS
WRITE HEADS
WORDS TO
BE WRITTEN
ADDRESSES
SELECTION
SECTION
WORDS
READ OUT

Figure 5-7

more than one drum, some bits of the word specifying an address in a drum system will be assigned the significance of identifying the particular drum. Other bits will indicate the particular field on that drum. Still other bits will define the register, by indicating the time with respect to the drum revolution cycle when the write pulses are to be applied to the heads. For example, assume that a storage system is comprised of four drums and that each drum is divided into four fields. Assume further that each field comprises 64 registers. Then six bits are required to specify the register (since 64 is 2^6), two bits are required to specify the field and two bits are required to specify the drum. Thus, the storage address 1001111110 would specify register 111110 (sixty-two) of field 01 (one) of drum 10 (two).

Magnetic drum storage functions are illustrated in block form in Figure 5-7. Notice that, as in the case of the core storage, provision has been made for a control and timing section. The timing section differs from that of core storage in that it accepts no timing commands from other parts of the computer. Instead, drum timing is a function only of the angular position of the drums. Thus, all timing pulses originate on the drums themselves. This must be the case, since the positions of storage registers are defined by the instantaneous angular position of the drum with respect to the read and write heads.

The scheme for reading and writing on core storage that

was outlined in Section 3-2, is economical of input and output equipment since all inhibit pulses for an array are applied to the same groups of input lines and all sense pulses appear on the same groups of output lines. However, it has the limitation that only one word can be transferred into or out of the array at any one time. In the case of the drum system, on the other hand, each field can have its own driving and sensing circuits. This increases the number of vacuum tubes that must be used, but it does allow simultaneous transfer of information on to and off a number of fields. This is of particular importance when the drums are used as a time buffer between input or output equipment and the central computer.

The scheme for writing on and reading from a drum system discussed above assumes programmed operation. However, there is no reason why the buffer drum system write operation cannot be automatic. One way of accomplishing this is by what is called status writing. When this scheme is used, each storage location has an associated status bit. When the register is written on, a 1 is recorded on the status channel. When the contents of the storage register are read, a 0 is written in place of the 1 on the status channel. As input information becomes available, it is written on the first storage register whose status bit is 0.

PART 5

CHAPTER 4

CONTROL

4.1 GENERAL

A digital computer functions on a step-by-step basis. It generates problem solutions by executing ordered sequences of instructions called programs. In a stored program computer, the program instructions as well as the operands are loaded into the primary storage element prior to the performance of computations.

A program is a plan for the solution of a particular problem. It is developed by a human programmer in terms of concepts which have meaning to him. The programmer, for example, knows why to specify addition in a particular case rather than subtraction or division. His understanding enables him to arrange a set of instructions in an order such that their execution generates a problem solution.

A digital computer is a piece of machinery that responds to physical stimuli. Thus the meanings of the programmer must be transformed into the physical stimuli to which the machine is designed to respond. The first step in this transformation is to represent the instructions in numerical form. The instruction, divide, for example, may be written as 0101100. But numbers are still abstractions, and a machine cannot operate upon abstractions. Thus a further transformation is necessary. An operator types the numbers into a punch card machine. The

keys of the machine look much like those of an ordinary typewriter. However, depression of a key causes a hole to be punched in a card. Thus numbers are caused to assume the form of patterns of holes on a card. The pattern of holes can then be used to establish an analogous pattern of voltages at the output of a flip-flop register. These outputs in turn can be used to establish an analogous pattern of magnetism on the surface of a magnetic drum or on the set of torroidal cores comprising a primary storage register.

The operation upon items of data, which are as represented by patterns of voltage or magnetism, takes the form of switching. The addition operation, for example, is defined by one set of connections, the subtraction operation by another. The switching operations defining an instruction are actuated by the voltage levels at the outputs of a flip-flop register which holds the instruction. In the case of the addition operation, for example, the voltage levels are used to gate the outputs of the A-register and accumulator to the full adders and to gate the full adder outputs to the inputs of the accumulator. The matrices which distribute the voltage levels from a flip-flop holding an instruction are said to decode the instruction.

As noted above, instructions and operands are loaded into primary storage prior to computation. In order to obtain an instruction or operand from storage its address must be specified. An address, like any other item of data, is represented by a pattern of voltages or of magnetism.

The transfer of information, like any other operation upon information, takes the form of switching. Thus addresses must be decoded in the same manner as instructions.

An instruction word, in general, specifies an operation and the address of an operand. Thus part of the pattern of voltage levels representing the word is used to carry out the operation and another part of the pattern of voltage levels is used to carry out a transfer of another word from storage to the arithmetic element. This implies that the two parts of the word must be decoded by separate matrices. In order to facilitate this separate handling, an instruction word is split into two parts when it is transferred from storage to the control element. One part is entered into an operation register and the other part into an address register.

4.2 PROGRAM-OPERATE CYCLE

As noted above, a particular instruction must be specified (by specifying its address in storage) and transferred to operation and address registers of the control element before it can be decoded for execution. In general, the address part of the instruction specifies an operand to be transferred to the arithmetic element. Since the operation part of the instruction is to be associated with the operand specified by the address part of the instruction, execution of the operation must be delayed until the transfer of the operand has been completed. Thus execution of a single instruction is an ordered sequence of steps. Moreover, the same classes of

steps must be performed for each instruction of the program; i.e. the instruction must be transferred from storage, the specified operand must be transferred from storage and finally the operation part of the instruction must be executed. The two transfers are part of the program function, that is they set up the specified operation. The entire cycle of transfer and operation is called the program-operate cycle.

The requirement for ordering individual steps within the program-operate cycle implies the existence of some timing device which can supply "go ahead now" pulses to trigger each of the steps of the cycle. An oscillator generating pulses at a constant repetition rate satisfies this requirement for a timer.

Previously, the operational speed of the computer has been discussed in terms of time required to complete an arithmetic operation or time required to gain access to an item of stored information. However, these in turn are related to the repetition rate of the basic timing pulses. A computer using core arrays as its primary storage device and flip-flop registers as its operational units requires a basic timing pulse repetition rate on the order of two megacycles in order to take advantage of the inherently high access speed and operating speed of those storage and operational devices.

4.3 PROGRAM ELEMENT

The basic principle that underlies the execution of a stored program is that, in general, a set of instructions is executed in an order that is predetermined by the numbers

associated with the addresses into which individual instructions are loaded. This implies the existence of some mechanism for examining storage addresses in order. The simplest such mechanism is a counter. When the program is initially loaded into the computer, this program counter is set to 0. Pushing a START button initiates the first program-operate cycle. Since the program counter reads 0, the instruction stored at address number 0 is executed. At the same time, the program counter is stepped to 1. Thus the second instruction to be executed is the instruction stored at address number 1. In general, the nth instruction to be executed is the instruction at address number n - 1. There are, however, important exceptions to this.

It has already been noted (in Part 1) that a computer makes a much more powerful computational tool if it is able to modify the order in which it executes instructions as a result of various contingencies which may arise during the solution of a problem. Such modifications of order can be initiated by conditional branch instructions. An instruction of this type specifies that if some particular contingency is satisfied, then the next instruction to be executed is the one stored at arbitrary address X. If, on the other hand, the contingency is not satisfied, then the computer can continue to execute instructions in consecutive order. The selection of arbitrary address X is straightforward. If the contingency is satisfied, then address

X is caused to replace the contents of the program counter. For example: after the instruction at address X has been executed, the instruction at address X + 1 will be executed and that, in general, n cycles after the branch, the instruction at address X + n will be executed, unless another branch instruction has intervened.

The operation or instruction to branch may be used to make the entire program cyclic. Thus the last instruction of the program may be a branch instruction specifying address 0; i.e. resetting the program counter so that the first instruction of the program is executed again immediately after the last instruction.

It should be clear that each program-operate cycle starts with the decoding of the contents of the address register. It is worthwhile to compare the transfer paths set up by the program counter matrix with those set up by the control element address register matrix (discussed in Section 4.2 above). In both cases a word is transferred from primary storage. In the case of selection by the program counter, the word is placed in the operation and address registers of the control element. However, in the case of selection by the address register, the word is transferred to the arithmetic element.

4.4 TIMING PROBLEMS

It should be clear from the discussion of the arithmetic element in Chapter 2 of this Part that the four arithmetic operations require different amounts of time. Addition, which is a part of the other three operations, is obviously the

fastest. Subtraction, which requires only the initial step of complementing the A-register, is not much more time consuming. Multiplication requires one add and one shift operation for each multiplier bit, and is, therefore, substantially more time consuming. Division, with all the special steps it requires, is the longest arithmetic operation of all.

In addition to arithmetic operations, the computer must perform transfer operations which require a varying amount of time. For example, an instruction which merely calls for the transfer of a word from the arithmetic element to a register of the primary core storage requires a minimum of time. On the other hand, a transfer between primary core storage and an auxiliary or buffer drum storage register calls for synchronization between the primary storage and the slower drum storage and therefore requires more time.

It has been indicated in Section 4.2 above that the individual steps with a program-operate cycle are to be ordered by timing pulses. The most straightforward way of using such pulses depends upon having all program-operate cycles equal in time duration. In this case, the same number of pulses occur in each cycle so that the pulses comprising a cycle can be numbered and supplied on separate lines. An action can then be initiated at a particular time in any cycle by connecting a particular pulse line to an initiating gate.

If all the program-operate cycles are the same length, then each must be long enough to allow the completion of the most time consuming instruction. Such an arrangement is intolerable, since it slows the operation of the computer needlessly. However, there are at least two ways in which the principle of numbered pulses on separate lines can be retained without insisting that all cycles be the same length.

The first such method is to allow cycles of different lengths, but to impose the restriction that all cycles must be integral multiples of some unit interval. One interval may be used to perform the programming function of transferring and decoding the instruction, a second interval may be used for transfer of the specified operand and a third interval may be used for the execution of the instruction. Under this system, the number of time intervals required is specified when the instruction is decoded thus defining the time when the next cycle will begin. This is important since much time can be saved by allowing cycles to overlap.

A second method for providing cycles of different lengths, which incidentally can be used in conjunction with the first method, is to stop the generation of the regular time pulses during the more time-consuming operations. Such operations can then be performed under the control of a special set of timing pulses or else can be performed asynchronously. When the asynchronous method is adopted, each step in the operation is used to trigger the following step so that no timing pulses are required. Asynchronous operation is fast

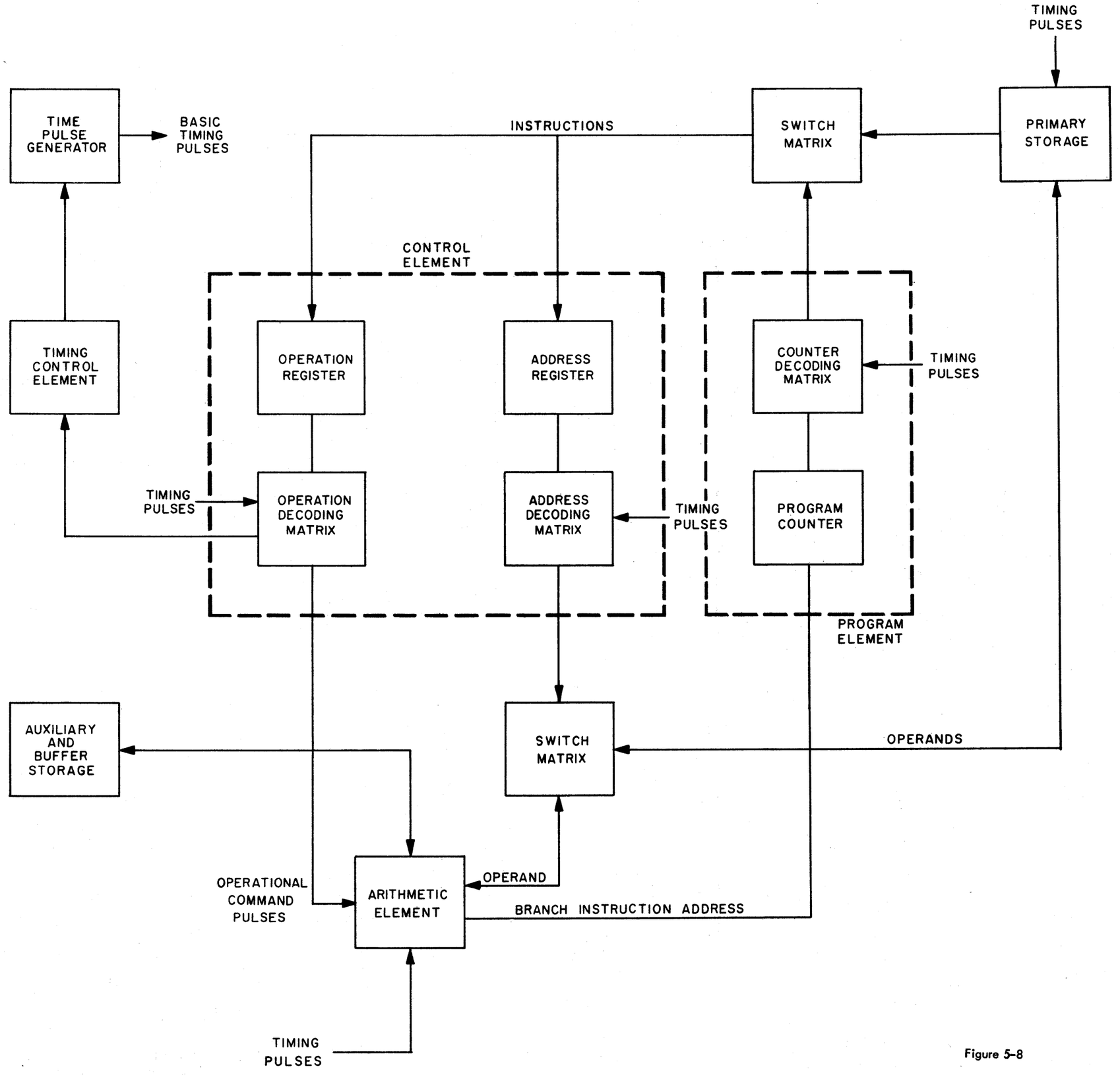

TIMING PULSES
TIME PULSE GENERATOR
BASIC TIMING PULSES
INSTRUCTIONS
SWITCH MATRIX
PRIMARY STORAGE
CONTROL ELEMENT
TIMING CONTROL ELEMENT
OPERATION REGISTER
ADDRESS REGISTER
COUNTER DECODING MATRIX
TIMING PULSES
TIMING PULSES
OPERATION DECODING MATRIX
ADDRESS DECODING MATRIX
TIMING PULSES
PROGRAM COUNTER
PROGRAM ELEMENT
AUXILIARY AND BUFFER STORAGE
SWITCH MATRIX
OPERANDS
OPERATIONAL COMMAND PULSES
ARITHMETIC ELEMENT
OPERAND
BRANCH INSTRUCTION ADDRESS
TIMING PULSES

Figure 5-8

but it tends to increase circuit complexity and decrease flexibility.

One of the operations for which the generation of regular timing pulses is likely to be stopped, is the operation of transferring blocks of information between primary core storage and the slower operating auxiliary or buffer drum storage.

4.5 SUMMARY OF CONTROL FUNCTIONS

The control functions discussed in this chapter are illustrated in block form in Figure 5-8.

Control provides problem solutions by means of automatically sequenced steps. The sequences of steps exist on two levels. On one level, there is the set of instruction steps which comprises the program. The execution of this sequence is ordered under the control of the program element which causes instructions to be drawn from consecutive storage addresses or from arbitrary addresses specified by branch instructions. Sequences of steps exist on a second level within each instruction step. These sequences are ordered by timing pulses and by commands decoded from the contents of the operation and address registers of the control element.

PART 5

CHAPTER 5

INPUT-OUTPUT

5.1 GENERAL

The input-output facilities of a computer provide the means of communication between man and machine. This communication is characterized by conversions between the symbols that have meaning to man and the patterns of physical states upon which the machine is designed to operate.

Sometimes a digital computer is but one element in a system which includes other devices. This is usually the case in a real-time solution. Here the computer receives data directly from other machines such as radar sets, and may supply command signals to still other devices. In this case, the input-output facilities of the computer must also provide the means for communication between machine and machine. This communication may require conversions of the form of data between an analog form and a digital form. For example a radar set may represent the range of a target in terms of the continuously variable magnitude of a voltage applied to a particular output terminal. The digital computer, is capable of operating only upon physical representations of discrete values. Thus the continuously variable output of the radar set must be quantized, that is converted into a set of discrete values, before it can be presented to the computer.

5.2 COMMUNICATION BETWEEN MAN AND MACHINE

Communication between man and machine may be limited to a presentation of the problem by the human operator and a presentation of the solution by the machine. The first step in the presentation of the problem to the machine would be to transfer the set of instructions comprising the program and the set of operands onto a deck of punched cards where the information would appear in the form of patterns of holes. This operation would be performed by the use of a keyboard much like a typewriter keyboard. The next step would be to read the information on the cards into the computer by means of a device which would convert the pattern of holes into a pattern of voltage pulses that would condition the flip-flops of a storage register which, in turn could be used to establish patterns of magnetism on a buffer storage drum. From buffer storage, the program and operands might pass through another flip-flop register to addresses in the primary core memory of the computer where they would again take the forms of patterns of magnetism. With the instructions loaded into consecutive addresses in primary storage and the operands loaded into the addresses specified for them in associated instructions, a START button could be depressed initiating the solution, which could then continue to completion without any further human intervention. The data comprising the solution could be transferred from primary storage through buffer storage to some output device which might convert it into typewritten form, all under the control of the program.

The case of a data-processing machine providing a continuous solution to a real-time problem is not likely to be that simple. In such an application, the computer may be required to furnish coordinated items of data to its human operators on a more or less continuous basis. The operators may then command the computer to modify the program it is performing as a result of some item of data that it has presented. This amounts to a kind of conversation between the operators and the machine which cannot be carried on fast enough to satisfy most real-time situations through the types of input and output channels outlined above.

A visual display system satisfies the requirement for rapid presentation of substantial amounts of information to human operators. Cathode-ray tubes can be used to generate special digital displays or more conventional analog displays.

There are several methods which can be used to allow the operators to communicate quickly with the computer. One such method involves the use of a special command keyboard which allows instructions to be applied directly to the internal computer circuits, bypassing the usual input channels including the buffer drum system. It should be understood that instructions introduced in this manner can only be used to initiate actions that have been provided for in the program which the computer is executing at the time, since even a very large number of human operators could not supply the computer with

commands fast enough to keep it operating with any efficiency if it were stopped from following its stored program.

One of the methods for allowing an operator to communicate with the computer involves the use of a display device and a photo pickup device as elements in the path through which a particular kind of data reaches the computer. By establishing or breaking the light path from the display device to the photo pick up element, the human operator can either allow an item of data to pass to the computer or can reject it.

5.3 COMMUNICATION BETWEEN MACHINE AND MACHINE

A computer which is used to provide a continuous solution to a real-time problem must be equipped to accept input data continuously. This data may arrive from a number of different devices in an intermittent and unpredictable fashion. If the arrival of such data were allowed to interfere with computations, the effectiveness of the computer would be greatly reduced. However, this is unnecessary since the new data can be accepted and stored by the buffer drum system without involving the control facilities of the computer proper. This can be done by an automatic system which causes the drum registers to be written on in an order determined by their status. The data can then be read off the buffer drums for use in the computer, under the control of the program and at times which cause the least disruption of computational operations.

5.4 EXTERNAL STORAGE

As has already been noted, programs and data are initially read into the machine from punched cards. Since reading from the cards does not destroy the information contained on them, they can be kept as permanent records and also for subsequent re-reading into the computer. Decks of cards can also be punched under the control of the computer program in order to make records of results obtained during solutions.

Tape recorders provide another means for storage quantities of data. Tape is not only a very compact storage medium but is also one allowing a moderately rapid read-in of stored information.

5.5 SUMMARY OF INPUT-OUTPUT FUNCTIONS

Input and output devices provide the means for communication between a computer and its environment. The environment includes other devices and human operators with which the computer must be in contact. The input-output requirements of a computer vary tremendously according to the type of mission it is required to perform. The data processing machine providing a continuous solution to a real-time problem has the heaviest input-output requirements. A computer operating in such a situation must be equipped with an adequate buffer memory system which can accept input information without taxing the control facilities of the computer proper and hold it until the program calls for it.