A PULSE-DENSITY COMPUTER USING TRANSFLUXERS

by

Marvin Carl Woodfill

A Dissertation Submitted to the

Graduate Faculty in Partial Fulfillment of

The Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: Electrical Engineering

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

Head of Major Department

Signature was redacted for privacy.

Dean of Graduate College

Iowa State University
Of Science and Technology
Ames, Iowa

1964

TABLE OF CONTENTS

I. INTRODUCTION

The purpose of this thesis was to investigate the possibility of designing a computer utilizing the density of pulses as the basic medium of computation and transportation of information within the computer. The availability of new magnetic core devices, primarily transfluxers, created the possibility of the efficient use of current pulses in such an application. The proposed computer will represent analog magnitudes by the density of equal-amplitude and equal-width current pulses in a pulse train. The sign of the analog value will not be represented within the pulse train. This type of variable representation facilitates the arithmetic and computational operations within the computer. The computer will contain a clock (pulse generator) which will precisely control and synchronize all of the operations within the computer. Because of the nature of magnetic devices, i.e. an output can only be obtained during an input pulse, synchronization of the operations with some source of pulses is mandatory.

Much of the preliminary developmental work accomplished on the computer was done in the laboratory of the Cedar Rapids Division of the Collins Radio Company. Some of the Collins' staff had been investigating the use of transfluxers as memory elements and other applications when the author joined the Collins' staff. The author's supervisor, who also has an intense interest in pulse-density computational techniques, suggested the possibility of utilizing transfluxers in pulse-density computing devices. The intended application for such a computer was in the design of aircraft navigational-aid and autopilot equipment.

It is believed that such a computer would prove to be more accurate, more reliable, less costly and less bulky than the devices currently in use. Because of the intended applications, attempts were made to develop a computer which would be, as nearly as possible, compatible with other electronic equipment involved in such applications.

This computer was designed for application as a special purpose computer and would not be applicable to general purpose problem solution. Although an accuracy advantage is claimed, the proposed computer has an accuracy of only one part in sixty-four. This would probably not be sufficient for many applications, but improving the accuracy could be accomplished by increasing the number of bits associated with the computational components within the computer. It was believed that one part in sixty-four was sufficient accuracy to demonstrate the validity of the method of computation involved.

II. REVIEW OF CURRENT PRACTICE

The proposed computer cannot be directly compared with any of the three major types of computers currently in commercial use, i.e. analog, digital and digital differential analyzer computers, because of the special purpose nature of the proposed computer. Conceptual comparision, however may aid in establishing a basis of understanding of the proposed computer.

The proposed computer is conceptually similar to analog computers and digital differential analyzers whose main function is to integrate. Problems that cannot be reduced to an integration process do not lend themselves to solution by any of these methods. The proposed computer, similar to the digital differential analyzer, has the advantage over voltage analog computers of stability and precisely repeatable results. Conventional voltage analog devices have the advantage over all digital devices in that the representation of values is a continuous process. In digital devices variable representation is a discrete process and must in general be interpolated if the exact value is to be determined at any arbitrary time. Then since the problem is one of discretely representing a continuous function, the increments of representation must be chosen such that the value of the digital approximation never differs from the exact value more than a tolerable magnitude of error.

The proposed computer is similar in concept to digital differential analyzers. The major difference between the two devices is the method of component employment in problem solution. Conventional digital differential analyzers use incremental multiplication and integration together

with digital function generators based on table-look-up and interpolation routines, as given by Korn (12). Gschwind (9) states that commercial digital differential analyzers have not received the general recognition that they undoubtedly deserve and will probably eventually receive primarily because of their cost and limited application. A conventional digital computer of comparable cost can do anything that the differential analyzer can do and much more. The differential analyzer, like the proposed computer, provides parallel integration, i.e. integration of several variables simultaneously, which cannot be done with conventional digital computers. Furthermore, with equivalent circuitry, the proposed type of computer or the differential analyzer can integrate faster than a digital computer because integration is "wired in" in the former computers whereas the digital computer must be programmed to integrate.

As previously stated, direct comparison of the proposed pulse-density computer with any of the three major types of computers is not a reasonable comparison because of the special purpose nature of the proposed device. It is believed that the proposed computer does take advantage of some of the attributes of each type and is believed to be the optimum combination of the basic types for the purpose for which it was intended.

III. REVIEW OF PREVIOUS EXPERIENCE

The basis for the feasibility justification of pulse-density computation lies in the fact that if the magnitude of a variable is represented by the pulse density of a pulse train, integration of a variable is simply a matter of counting the number of pulses in the pulse train.

The author's masters thesis (18) was concerned with the development of a pulse-density computer employing voltage pulses and conventional electronic circuitry. Many problems were encountered with the circuitry concerned with that computer, but because of improvements in semiconductors since then, the circuitry could certainly be improved if redesign were attempted. The use of current circuitry techniques to produce a revised version of the previous computer would not solve all of the problems involved in its application. Most current applications of this type of computer would require a clock frequency greater than the two-kilocycle rate that was used in the previous computer. Increasing the clock frequency, however would result in a serious loss in reliability of computer results because of the random nature of the pulses contained within the internal variable representations. The clock within the computer produced most of the pulses in the pulse trains within the computer, however because of the "adder" unit other pulses were added to the pulse trains which were not coincident with clock pulses. The purpose of the adder unit was to add two pulse trains to obtain one pulse train containing the sum of the number of pulses in the two input trains. The random pulses were generated in the addition process because when coincidence occurs between input pulses, the output of the adder composed of the

logical addition (OR) of the two inputs would contain only one pulse but the correct sum is two. Since only one pulse can exist at one time, this error was compensated for by the addition of another pulse an arbitrary fixed time after the coincidence occurred. This method of adder compensation has three disadvantages: first, it prohibits synchronization; second, if coincidence happens to occur again before the extra pulse is added this coincidence is ignored; third, there is no assurance that the added pulse is not coincident with another pulse in the sum, thus not actually affecting the value of the sum.

In the proposed computer, because of the nature of magnetic core circuitry, the pulses must be synchronized with some clock. This is because a magnetic core can produce an output only during an input pulse. Consequently, the method of pulse-train addition in the proposed computer requires more components, but the output is assured to be the correct sum and it will be synchronized. The most important component of a pulse-density computer is the flip-flop. The flip-flop used in the proposed computer contains two cores, four transistors and two resistors. The flip-flop in the previous computer, however contained two transistors, ten resistors, four capacitors and six diodes. Furthermore, in the scaler application each flip-flop in the previous computer had to be connected to the remainder of the circuitry by a differentiating circuit called a shaper which contained one capacitor, two resistors, one diode and one transistor. In the proposed computer the flip-flop is connected directly to the remainder of the circuitry. Even with circuitry improvements, the electronic flip-flop and its associated circuitry would contain many more parts than the core flip-flop developed herein. This fact alone would

indicate that the proposed computer would possess a reliability and size advantage over its predecessor.

The previous computer contained conceptually the same components and functional units as the proposed computer except that the former contained a unit called a pulse train subtractor. The purpose of the subtractor unit was to subtract two pulse-trains and produce two outputs, one representing the positive difference and the other representing the negative difference. Conceptually, the only way two pulse density variables can be subtracted is to integrate them over a period of representation and then generate a pulse train containing a number of pulses equal to the difference in the two values. Since integration always follows a subtraction operation, this would mean that the integration would be duplicated later and furthermore that the representation of variable would no longer be a continuous process within the computer, an undesirable effect. In practicality the subtractor unit in the previous computer was supposed to eliminate coincident pulses in the two pulse trains and to eliminate alternate pairs of pulses in the two trains, however the implementation of the latter caused pulses to be incorrectly eliminated. The proposed computer will not attempt pulse-train subtraction except in the integration process where coincident pulses will be eliminated.

Representation of signs of analog quantities in the pulse-density analogies can be accomplished in several ways. In the previous computer the pulse trains (one for each variable) conveyed no sign information and the sign representation was conveyed by a bus voltage for each variable. This would not be feasible in the proposed computer because

magnetic core circuitry cannot easily produce a constant bus voltage or even utilize such a voltage.

Originally it was decided that the pulse train, if practicable, would convey its own sign information. To facilitate this the analog zero value would be represented by a pulse rate equal to half of the clock rate, thus the maximum analog value would be represented by the full clock rate and the minimum (maximum negative) analog value would be represented by the absence of pulses. It was found that this method of representation facilitated the integration process but the process of pulse train addition is very difficult using this type of representation. The sum of two pulse trains using this representation would be in error by half of the clock rate. Subtraction of half of the clock rate would involve exact subtraction which has already been shown to be a difficult process resulting in discontinuous variable representation.

The method of sign representation finally used in the proposed computer was to represent a particular variable by two pulse trains, one representing the positive value of the variable and the other representing its negative value. This method of representation of sign information is believed to be the optimum method for magnetic core circuitry of this type.

IV. MAGNETIC DEVICE APPLICATION

The transfluxer is one of a number of magnetic core devices developed in recent years. The transfluxer is a two aperture magnetic core device with some rather unique properties. An explanation of the internal operations of the transfluxer is given by Hershberg (5), however a working knowledge of some of the switching properties of the device can be obtained through the use of a simplified picture of the flux paths within the device. Considering the geometry of the system as shown in Figure 1, the cross section of the device is composed of three areas labeled A_1, A_2 and A_3. The device is designed so that $A_2 = A_3 = 1/2\ A_1$, which, might permit one to imagine that the device, after being magnetized, could be thought of as possessing two separate magnetic paths, as shown in Figure 2. This condition is called the "blocked" condition and could have been created by a current flowing in a winding as shown in Figure 2. This winding will be called the "block" winding. If another winding is added as shown in Figure 3 and a current of sufficient magnitude is passed through it in the direction shown then the flux in the inner path would be reversed. Appealing to the imagination again, the flux patterns of Figure 3 might be thought of as the flux patterns shown in Figure 4. This condition is called the "set" condition and it is important to note that a magnetic circuit, isolated from the rest of the device, has been created around the minor aperture. This circuit can now be magnetized in either the clockwise or counterclockwise sense without affecting the pattern around the major aperture. The winding shown in Figure 3 is known as the "set" winding. With the transfluxer in the set condition,

the addition of the windings shown in Figure 5 allows the use of the magnetic circuit around the minor aperture as a conventional magnetic core. The "read" winding forces flux around the minor aperture in the counterclockwise sense and produces an output from the transistor. The "write" winding reverses the flux around the minor aperture and does not produce an output. In this manner, alternate read and write pulses will cause output pulses to be generated coincident with each read pulse, until the transfluxer is once again blocked, i.e. current is passed through the block winding. Note that when the transfluxer is blocked flux is forced downward on both sides of the minor aperture which means that regardless of the previous state of the flux around the minor aperture, no output is obtained. Furthermore, regardless of subsequent read and write inputs no output will be obtained until after the transfluxer has once again been set (a pulse applied to the set winding). Note also that when a set pulse occurs the flux on the right-hand side of the minor aperture remains unchanged and no output is produced until the next read pulse is applied.

The transfluxer in switching applications operates exactly as the foregoing simplification would predict, however there are a few precautions that must be observed in the design of transfluxer circuits that may not be obvious at first glance. Set and read pulses must not be coincident if an output is desired. The reason for this is apparent from the wiring geometry. Note that the two windings attempt to saturate the same portion of the device in different directions. Another precaution that must be observed is that read and block pulses must never be coincident. If they are the output will be a pulse of short duration

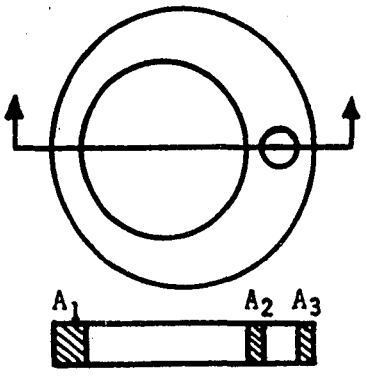


Figure 1. Transfluxer geometry

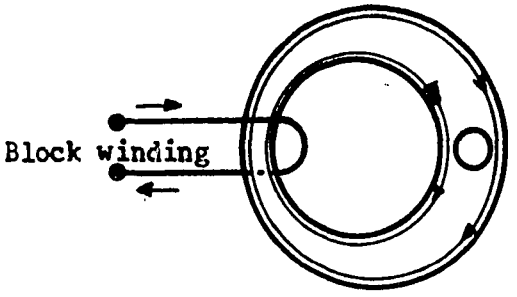


Figure 2. Flux paths in a blocked transfluxer

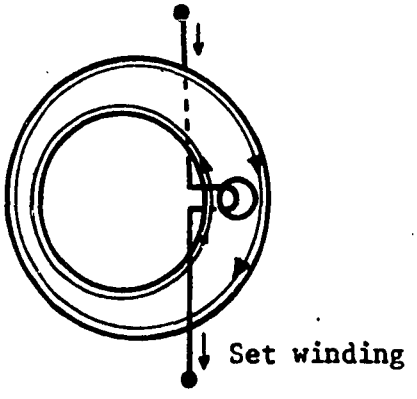


Figure 3. Flux paths in a set transfluxer, view a

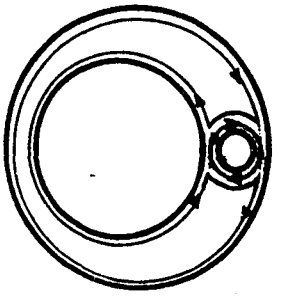


Figure 4. Flux paths in a set transfluxer, view b

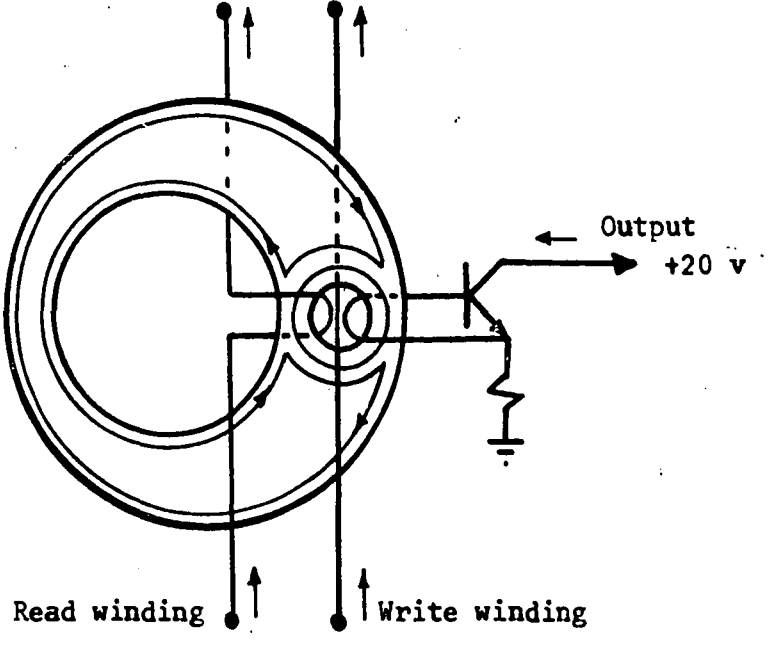


Figure 5. Transfluxer read and write windings

and small amplitude which may or may not be ignored by the circuitry to which it is applied.

Perhaps the most noteworthy attribute of the transfluxer, aside from its unique switching characteristics, is the fact that although all of its windings have minimum values or threshold values of current (different for each winding) that must be exceeded in order to produce the desired effect only one of the windings has a maximum value of current which must not be exceeded. The current in the write winding must not exceed a maximum value, which is a function of temperature. If the maximum is exceeded, under blocked conditions enough flux can be coupled to the major aperture to set the transfluxer. Under normal operating temperatures, the maximum is large enough, with respect to normal write current, that it imposes no serious limitation.

The transfluxers used in this computer were manufactured by the General Ceramics Division of Indiana General Corporation. They were made of a ferramic (ferromagnetic core material) S-6 with General Ceramics die F1023. The switching time for S-6 is approximately one microsecond, which means that a pulse of sufficient magnitude must possess a minimum duration of about one microsecond to change the state of the material. The nominal dimensions for the F1023 are: Outside diameter-205 mils. Major aperture diameter-118 mils. Minor aperture diameter-30 mils. Thickness-42 mils. The optimum values for operating currents, determined experimentally for one microsecond pulses are: Block winding, 1230 milliamp-turns. Set winding, 820 milliamp-turns. Read and write windings, 840 milliamp-turns. Under these conditions,

when the transfluxer is set, a two-turn sense winding yields an output with a maximum value of one volt.

The transistors used in this computer were 2N697 NPN double-diffused silicon mesa transistors manufactured by The Fairchild Semiconductor Corporation. The 2N697 is a high speed, medium power transistor housed in a TO-5 case.

A single 2N697 connected in the common emitter configuration to the sense winding of a transfluxer produces an output of sufficient magnitude to drive another transfluxer, but the transistor output pulse, being approximately one-fifth microsecond in duration, is not sufficient to switch the core material of another transfluxer. It was found after several combinations were tried that a "Darlington" amplifier with the sense winding connected between the emitter and base of the first transistor would provide an output pulse of sufficient duration to switch another transfluxer (approximately 1.2 microsecond).

After attempting to use several different sizes and types of copper wire for the fabrication of transfluxer circuits, size 38 with polyurethane ("solder-eze") insulation was selected. Size 38 is large enough to facilitate the wiring of the devices and small enough to permit a sufficient number of turns to be wound in the minor aperture of the transfluxer.

The d.c. bias voltage used within this computer was selected to be +20 volts. The conventional aircraft bias voltage is 28 volts, however extremely good regulation of the bias supply used within this computer will be required and regulation of the lower bias voltage within the computer should be easy to maintain with the eight volt difference.

The clock frequency selected for this computer was twenty kilocycles. The main reason for this choice was the recent development by Collins Radio Company of a 20kc magnetic amplifier and an extremely stable and well-regulated 20kc square-wave source with which to drive it. It was believed that in an aircraft application of the proposed computer, the 20kc magnetic amplifier would also be needed and the clock pulses for the computer could be derived with very little additional hardware from the output of the 20kc source. The 20 kc source requires an input of 20 volts d.c. which was another reason for the choice of a 20 volt bias voltage within the computer.

The actual clock pulses within the computer were composed of two trains of one microsecond pulses. One of these trains, referred to as the clock (C), was obtained from the differentiation of the positive going portion of the 20kc square wave. The other train, referred to as the complement of the clock ($\overline{C}$), was derived from the differentiation of the negative going portion of the square wave. These two pulse trains are complementary, i.e. their pulses are alternately spaced in time.

Considering one microsecond pulses with a repetition rate of 20kc, the apparent duty cycle (ratio of pulse duration to pulse separation) is 2.0 per cent, however the actual duty cycle (the ratio of the total time that a transistor is neither cutoff nor saturated to the time that it is in one or the other of these states) is much less than 2 per cent. Tests conducted on the units within this computer indicated that a clock rate in the order of ten times the rate used could be employed without design change.

Determination of the pulse amplitude to be used within the computer is an arbitrary choice within the limits of the components involved. The upper limit is set by the tolerable power dissipation within the transistor. The lower limit is determined by the number of turns that can be wound through the minor aperture of the transfluxers. With a collector load of 56 ohms, the saturation collector-to-emitter saturation voltage of the 2N697 is about two volts at normal temperatures. This combination results in current pulses of about 320 milliamps. With this value of current, the following wiring data were used for the transfluxers: Block windings - four turns. Set windings - three turns. Read and write windings - three turns. Sense windings - three turns. These data and the wiring geometry are shown in Figure 6. Any particular transfluxer may have as many of any of these windings as are necessary for the desired application. Figure 7 shows the symbol used herein to represent a transfluxer and its sense amplifier in its normal application. This type of symbol allows the representation of transfluxer circuitry with single-line signal diagrams which are much easier to interpret than wiring diagrams with current directions and wiring geometry depicted.

There are some operations within the proposed computer that can be efficiently performed by single aperture cores. In these applications, General Ceramics' MC - 137 cores were used. These cores are also made of ferramic S-6. These cores have an inside diameter of 125 mils, an outside diameter of 230 mils, and a thickness of 60 mils. This magnetic device, because of its size, will produce a much greater output than the circuit involved with the minor aperture of the transfluxer. If this magnetic device were driven as proportionately hard as the minor aperture

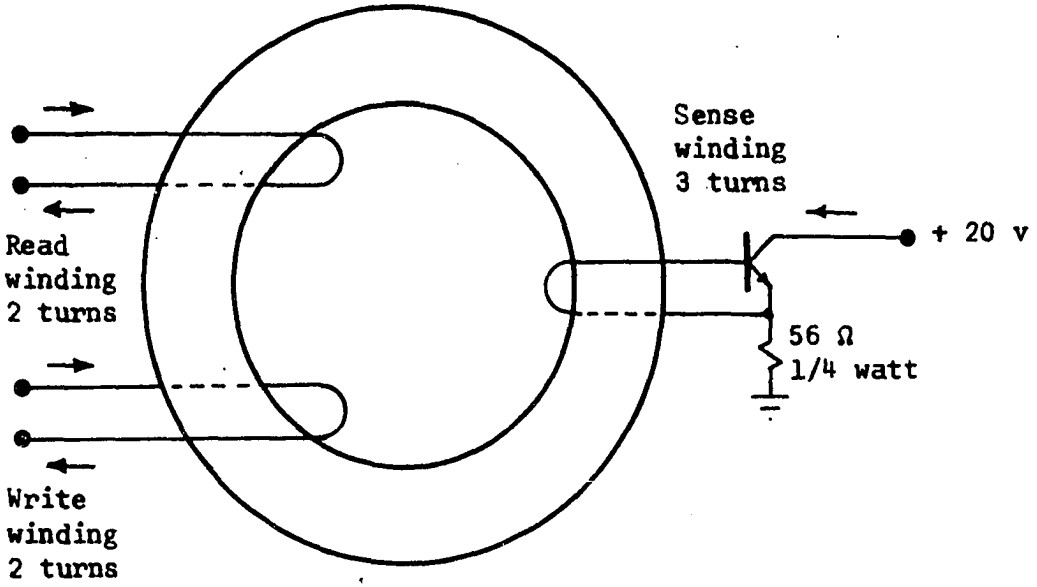


Figure 8. Circuit diagram for the single aperture core

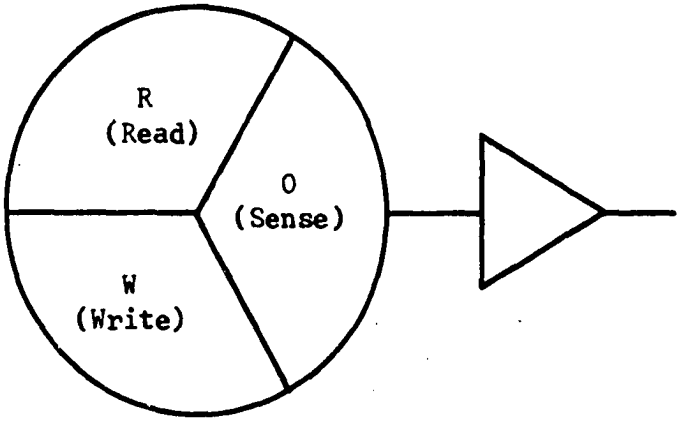


Figure 9. Representation for the single aperture core and its sense amplifier

V. DESIGN OF PROPOSED COMPUTER

The simulation of a system using the methods of the proposed computer usually involves the use of various numbers and configurations of the major functional units. Before discussion of the functional units, it is necessary that the design and operation of the computer components of which the functional units will be constructed be investigated.

A. Basic Operational Components

1. Flip-flop

The most important component of any pulse-density computer is the flip-flop (bistable multivibrator). Its importance stems from the use of the device within the computer. The flip-flop is used for pulse counting, for pulse-density scaling and as a logical element.

The philosophy of a core flip-flop differs substantially from that of an electronic flip-flop. The state of an electronic flip-flop is represented at all times by a d.c. voltage, thus at any time the state of the flip-flop can be determined externally. In a magnetic core flip-flop the state is represented by the flux within a core and cannot be determined at any arbitrary time except by changing it. The magnetic core flip-flop emits a pulse when its state is changed, thus only by placing the flip-flop in a known initial condition and keeping track of subsequent changes in state can the state always be known. The device has two outputs usually called Q and $\overline{Q}$. The Q output emits pulses when the device is placed in the "one" state and $\overline{Q}$ emits pulses when the device is placed in the "zero" state. Taking advantage of the memory capabilities of the transfluxer, the state of the flip-flop

can be interrogated at any time without changing the state of the device. This is accomplished by using the Q output of a flip-flop to set a transfluxer and using the $\overline{Q}$ output to block the device.

The first flip-flop developed using transfluxers is shown in Figure 10. This flip-flop, similar to previously developed core flip-flops, operates on a round-robin scheme. At all times two of the cores are blocked and the other two are set. The purpose of the two transfluxers on the left-hand side was to provide the necessary delay so that the two cores on the right-hand side would possess alternate outputs. This device operated as predicted but reliability proved to be a problem. The device lost synchronization in an erratic manner (more than two transfluxers ending up in the blocked or set condition at the same time), and when this happened the operation of the device would cease and it would have to be reset before further output could be obtained. The cause of this condition is that pulses have unintentionally been created or lost by imperfect regulation of the bus voltage. An extremely well-regulated power supply was used in this project, but it was found that whenever any other piece of equipment was turned on or off in the laboratory, the transients in the power system were sufficient to cause a loss of synchronization. Furthermore, this flip-flop did not take advantage of the capabilities of the transfluxers (conventional magnetic cores could have been used) and there were too many components to attain good reliability.

Since only two outputs are required from the flip-flop it would seem that such a device could be built with only two transfluxers. Using the conventional transfluxer configuration, the transfluxer output, which is

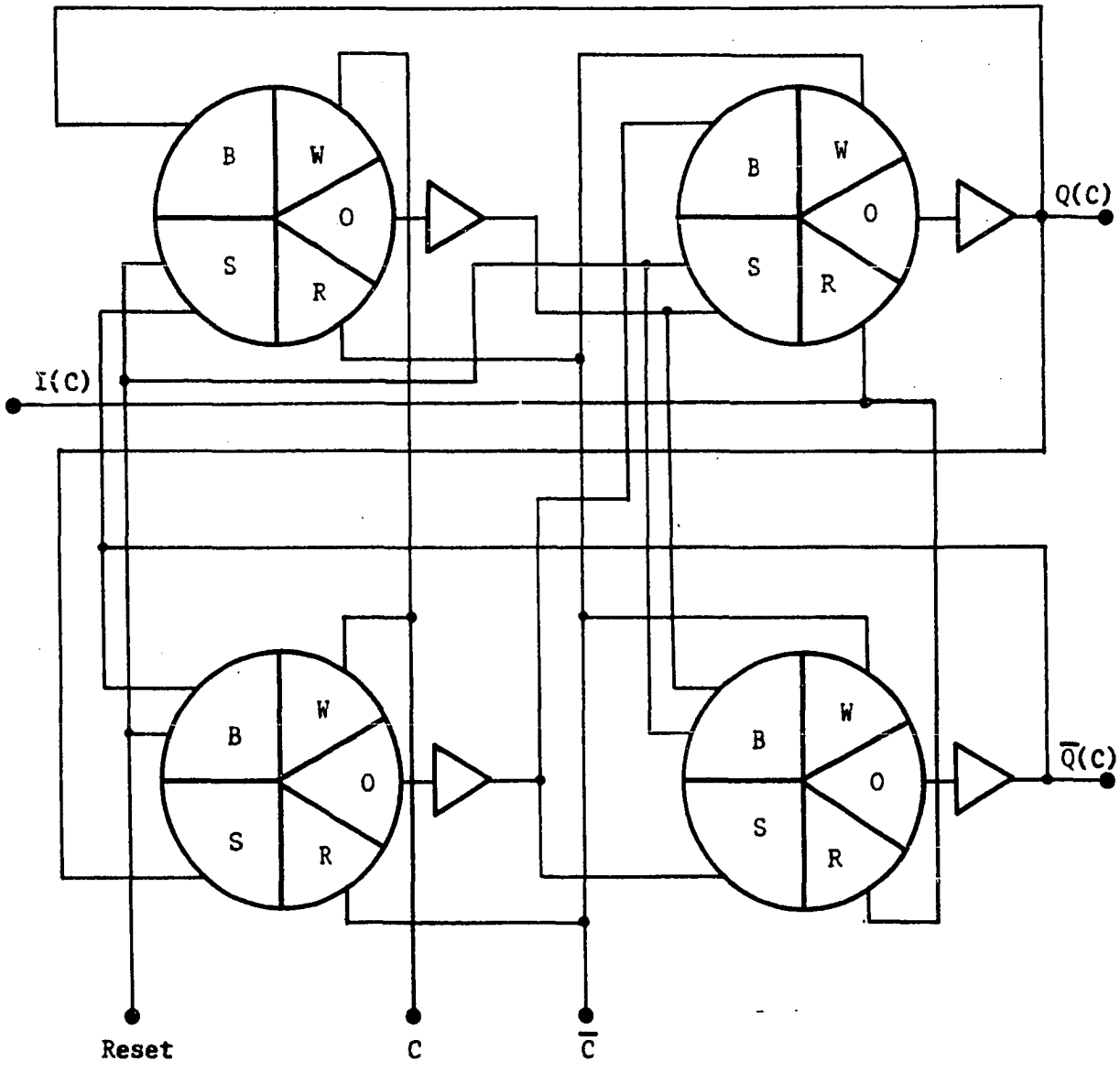


Figure 10. Signal diagram for the first attempt to produce a transfluxer flip-flop

coincident with the read input, must be used to block the transfluxer. This creates a condition in which the block and read pulses are coincident which, as discussed in the previous chapter, results in an output of small amplitude and short duration. This means that the transfluxer may or may not revert to the blocked condition and the other transfluxer will not be set.

The design of an operating flip-flop composed of two transfluxers was finally accomplished by simple rearrangement of the wiring geometry. Inversion of the sense winding provides an essential feedback which yields the solution. With the sense winding inverted, the output becomes coincident with write inputs but more importantly when the output, which is connected to the block winding, starts to block the transfluxer, the effect of the write winding which is to force flux downward on the right-hand side of the minor aperture, is reinforced by the effect of the block winding. Under these conditions the transfluxer ends up in the blocked condition and in the process the normal transfluxer output has been created which will set the other transfluxer. This winding arrangement will be represented by semicircular enclosure of the right-hand side of the transfluxer symbol. Note that although the functions of the read and write winding have been reversed, they will retain the labels indicating their geometrical significance as in the normal transfluxer application.

The signal diagram for the flip-flop is shown in Figure 11. The reset function is necessary to set the initial state of the flip-flop. Reset 1 indicates that the flip-flop will initially be set in the one state, i.e. the Q transfluxer blocked and the $\overline{Q}$ transfluxer set. Reset

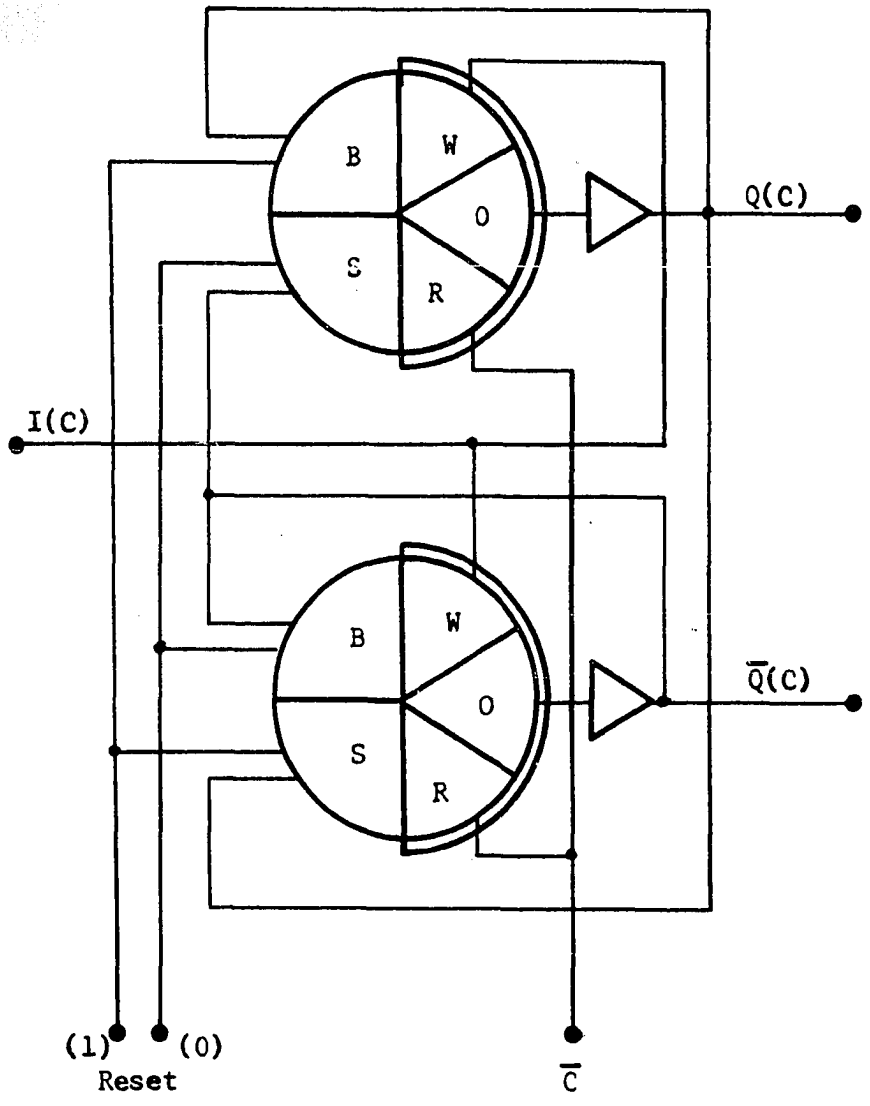


Figure 11. Signal diagram for the final flip-flop design

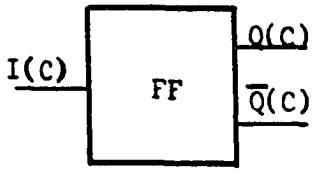


Figure 12. Representation for the flip-flop

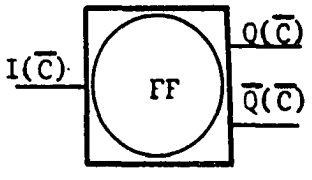


Figure 13. Representation for the complementary flip-flop

0 indicates an initial zero state, i.e. the Q transfluxer set and the $\overline{Q}$ transfluxer blocked. Figure 12 shows the signal notation used to represent the flip-flop. Note that the notation (C) or ($\overline{C}$) over a signal line indicates that pulses on that line will be coincident with the clock or the complement of the clock respectively. The notation of (0) or (1) within the symbol indicates the initial state set by the reset input.

The inputs and outputs of this device are coincident with the clock, however it is sometimes necessary to use flip-flops that have inputs and outputs coincident with the complement of the clock. Such a device would be identical in construction to the flip-flop shown in Figure 11 except that the read inputs of the transfluxers (which perform the logical write function in this configuration) would be connected to C instead of $\overline{C}$. This device will be called the complementary flip-flop and the signal notation for the device is shown in Figure 13.

2. Unit delay

The unit delay is a device which can be used to delay a pulse train one unit. One unit is defined as the time difference between adjacent C and $\overline{C}$ pulses. For a clock rate of 20 kc the unit delay amounts to a time delay of 25 microseconds. The unit delay accomplishes no logical function, and as will subsequently be demonstrated can be built into most of the logical function devices. The unit delay is necessitated by the fact that all of the logical operations for some of the necessary applications cannot be accomplished simultaneously. Thus, by the use of the unit delay a finite delay is introduced which can be used to facilitate this type of operation.

Accomplishing the unit delay function is relatively simple. The use of a single aperture core is sufficient. If it is desired to convert a pulse train whose pulses are coincident with C to one whose pulses are coincident with $\overline{C}$, the input is applied to the write input of the core and $\overline{C}$ is used to read the core. This function is called the unit delay and the signal diagram for the device is shown in Figure 14. Figure 15 shows the notation used to indicate the unit delay. If the read winding is read with C instead of $\overline{C}$, the input will have to be coincident with $\overline{C}$ and the output will be coincident with C. This function is called the complementary unit delay and the symbol used to represent it is shown in Figure 16.

3. Logical OR

The logical OR function can be accomplished in several ways using magnetic core circuitry. Serial connection of the sense windings to a sense amplifier provides what might be called an output OR function. This type of facilitation of the OR function will work only if the outputs to be combined are each represented by the state of the output of a readily accessible core. The major difficulty is that if the pulses in two of the cores are coincident at any time, the loading effect of one core on the other through the common-sense winding will cause the pulse output to be distorted. The OR function can also be facilitated by the duplication of the input windings in other computer components. This is what might be called an input OR function. An example of this is found in the flip-flop. One or the other of the transfluxers in the flip-flop has two block windings, one of which is connected to the output

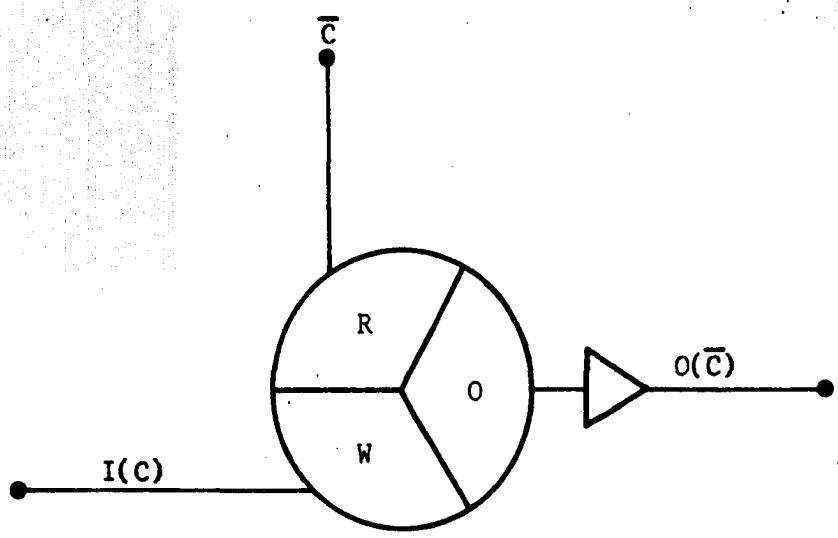


Figure 14. Signal diagram for the unit delay

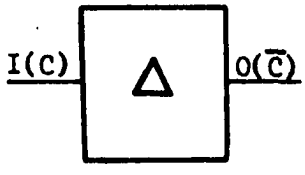


Figure 15. Representation for the unit delay

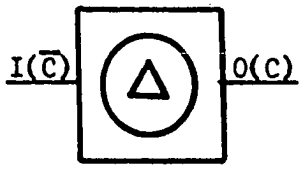


Figure 16. Representation for the complementary unit delay

of the other transfluxer and the other connected to the reset input. Thus the blocking state of this transfluxer would be represented by the logical addition (known as the OR function) of the two inputs.

It is sometimes necessary to perform a simple OR operation when there is not another component in which it can be readily accomplished. This can be accomplished by the use of a single aperture core with two read windings. Thus whenever either read winding is pulsed the output will contain a pulse. The core uses $\overline{C}$ as the write input. The signal diagram for the logical OR is shown in Figure 17. Figure 18 shows the signal notation used for the logical OR. If the inputs to the device contain pulses coincident with $\overline{C}$ then C would have to be used for the write input and the output pulses would be coincident with $\overline{C}$. This device will be called the complementary OR function and the symbol shown as Figure 19 will be used to represent it. If the write winding instead of the read winding is duplicated the resulting output will be the logical OR delayed one unit. Figure 20 shows the symbol that will be used for the delayed logical OR and Figure 21 shows the symbol that will be used for the delayed complementary logical OR.

4. Logical AND

The logical AND function is actually performed by the transfluxer in its normal configuration. The transfluxer must be set in order for read pulses to appear at the output, however with reference to the two sets of inputs, i.e. block-set and read-write, simultaneous occurrence of these inputs is not permitted. The transfluxer then does perform a logical AND function, but in some applications a coincidence detection

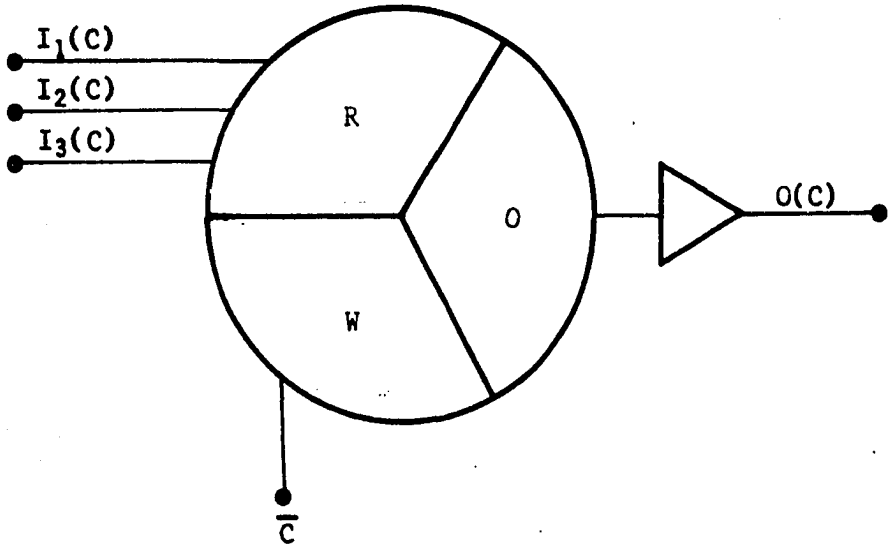


Figure 17. Signal diagram for the logical OR

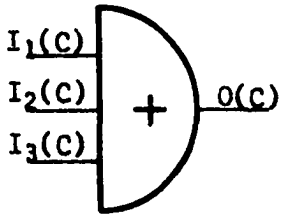


Figure 18. Representation for the logical OR

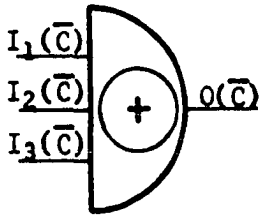


Figure 19. Representation for the complementary logical OR

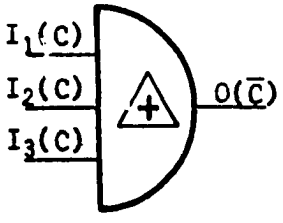


Figure 20. Representation for the delayed logical OR

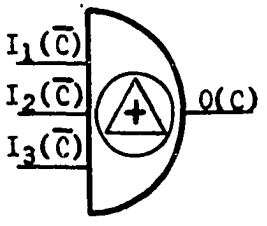


Figure 21. Representation for the delayed complementary logical OR

scheme is necessary. In the applications within this computer only two level AND gates are necessary and it should be noted that the device that will be subsequently described cannot be readily extended to more than two inputs. Recall that for the single aperture cores two-turn drive windings are required for saturation. If the read winding is composed of two single-turn windings instead of one two-turn winding, coincident pulses on the two windings would be required to obtain an output from the core. If the inputs are coincident with C, then $\overline{C}$ would have to be used as the write input to the core. This device is called the logical AND and the signal diagram for it is shown as Figure 22. Figure 23 shows the signal notation used to represent the device. If the inputs to the device contain pulses coincident with $\overline{C}$ then C would be used as the write input and the output would be coincident with $\overline{C}$. This device will be called the complementary logical AND and the symbol used to represent it is shown in Figure 24. If the two one-turn input windings are used for the write function instead of the read function, the logical AND function will be combined with a unit delay. Figure 25 shows the symbol used to represent the delayed logical AND and Figure 26 shows the symbol used to represent the delayed complementary logical AND.

5. Inhibitor

The purpose of the inhibitor is to remove selected pulses from a pulse train. The inhibitor has a signal input, an inhibit input and an output. If the signal input is labeled A and the inhibit input is labeled B the logical expression for the output would be $A\overline{B}$. Expressed in words this means that the output is composed of the pulses entering

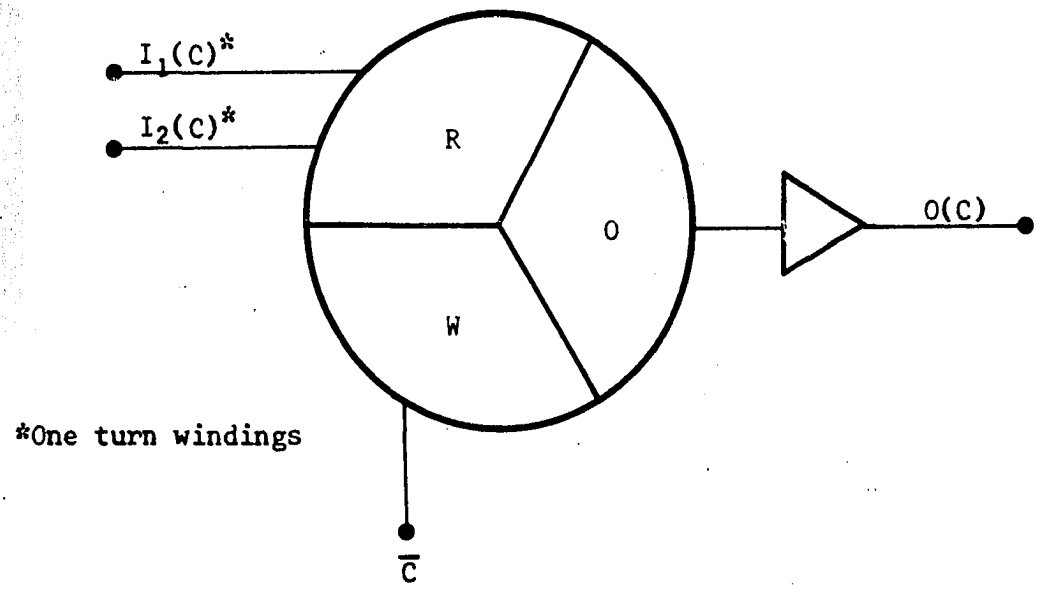


Figure 22. Signal diagram for the logical AND

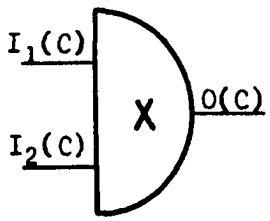


Figure 23. Representation for the logical AND

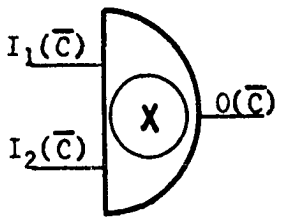


Figure 24. Representation for the complementary logical AND

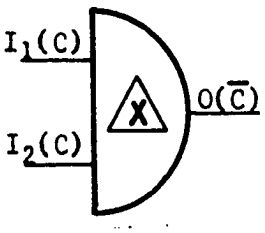


Figure 25. Representation for the delayed logical AND

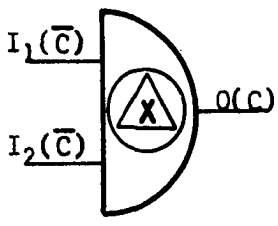


Figure 26. Representation for the delayed complementary logical AND

the signal input except that any of the signal input pulses that are coincident with inhibit inputs are eliminated from the output. The only problem in the fabrication of the inhibitor is the fact that a one unit delay is necessary in the device to insure correct operation. This means that the inputs are both coincident with C but the output pulses emitted are delayed, with respect to the input, one unit and are coincident with $\overline{C}$. The inhibitor is composed of two transfluxers and two sense amplifiers. The signal diagram for the inhibitor is shown in Figure 27. The top transfluxer is initially blocked and the bottom is initially set, this condition is created by a reset input. The read input to the bottom transfluxer is the signal input and when the first signal pulse enters the device (assuming no coincident inhibit pulse) an output pulse from the bottom transfluxer is emitted which sets the top transfluxer. The next $\overline{C}$ pulse causes an output from the top transfluxer which is the first output from the device and blocks the bottom transfluxer. The next signal pulse will produce a subsequent output from the top transfluxer unless there is a coincident inhibit input pulse. The inhibit pulse will block the top transfluxer and set the bottom one. The next signal pulse will cause an output from the bottom core unless there is a coincident inhibit pulse, if there is then the affect of the signal pulse, reading the bottom transfluxer, will be cancelled by the affect of the inhibit pulse, setting the transfluxer, hence no output will be emitted from the bottom transfluxer and it remains set and the top one remains blocked. The symbol used to represent the inhibitor is shown in Figure 28. Note that either of the input functions can be composed of more than one pulse train by identical duplication of the appropriate windings. As previously

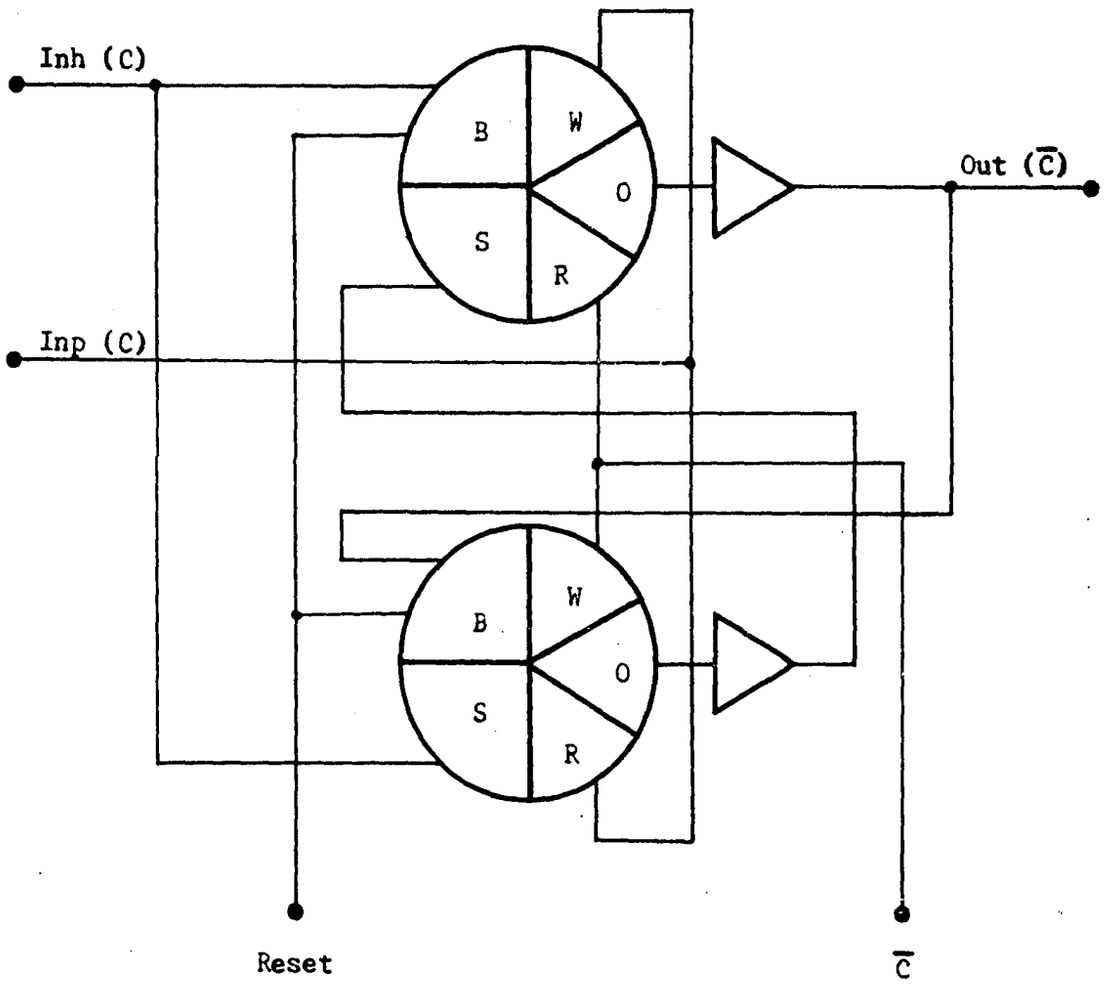


Figure 27. Signal diagram for the inhibitor

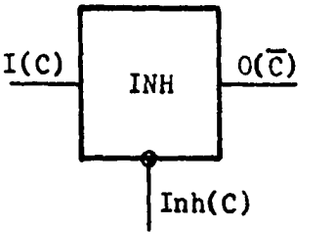


Figure 28. Representation for the inhibitor

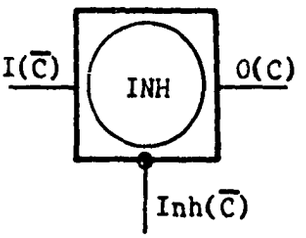


Figure 29. Representation for the complementary inhibitor

mentioned this is referred to as an input OR function and will be shown in the signal diagram representation by multiple input connections on the appropriate edge of the symbol.

The inhibitor performs one of the most important logical functions within the computer. It not only performs the inhibit function but is the only unit within the computer which is capable of logical inversion. If a variable B is used to inhibit the clock the output will possess the logical expression $\overline{B}$ because the clock has a logical value of one.

A variation of the inhibitor that may be useful is one in which the input pulses are coincident with $\overline{C}$ which would yield an output coincident with C. This type of inhibitor will be called the complementary inhibitor and the signal diagram will be identical to the inhibitor except that $\overline{C}$ used to read the top transfluxer and write the bottom one will be replaced by C. The signal diagram representation for this device is shown in Figure 29.

6. Director

Directors are devices which are used to direct pulses. Two types of directors are of interest in the proposed computer. The first type, referred to as type I, is used to generate one pulse train from the selected pulses of two input pulse trains. For each desired output pulse space, a control pulse will be provided one unit in advance that will select which of the input pulse trains will produce the possible output pulse. The second type of director, referred to as the type II, is used to generate two pulse train outputs from one input. For each input pulse, a control pulse will be provided one unit in advance that will select

which of the two outputs will contain the particular pulse. The directors provide the function of single-pole double-throw switches whose position is selected in advance of the appearance of the pulse to be directed.

The operation of the type I director is facilitated through the use of two transfluxers and a common sense amplifier. The signal diagram for the type I director is shown in Figure 30. The control inputs K_1 and K_2, whose inputs lead the signal inputs by one unit and are thus coincident with $\overline{C}$, are used to block and set the appropriate transfluxers. External circuitry will insure that only one of the control inputs will be pulsed before the occurrence of a signal input, thus only one of the transfluxers will be set during any input pulse. Note that a preceeding pulse on K_1 will cause the subsequent output to come from the input I_1. Similarily a K_2 input causes I_2 to be selected. Since at any particular time, output can only be produced from one of the transfluxers the logical OR function required at the output of the director can be obtained by serial connection of the sense windings of the transfluxers. The signal inputs are used to read the transfluxers and the write input for each is accomplished by $\overline{C}$. Figure 31 shows the notation used to represent the type I director.

A variation of the type I director that will prove to be of interest is a type I director which can be used to direct pulses that are coincident with $\overline{C}$. Such a device will require control pulses coincident with C that lead the signal pulses by one unit. The only change required in the signal diagram is that the write input to the transfluxers will have to be C instead of $\overline{C}$. Figure 32 shows the notation used to represent this device which is referred to as the complementary type I director.

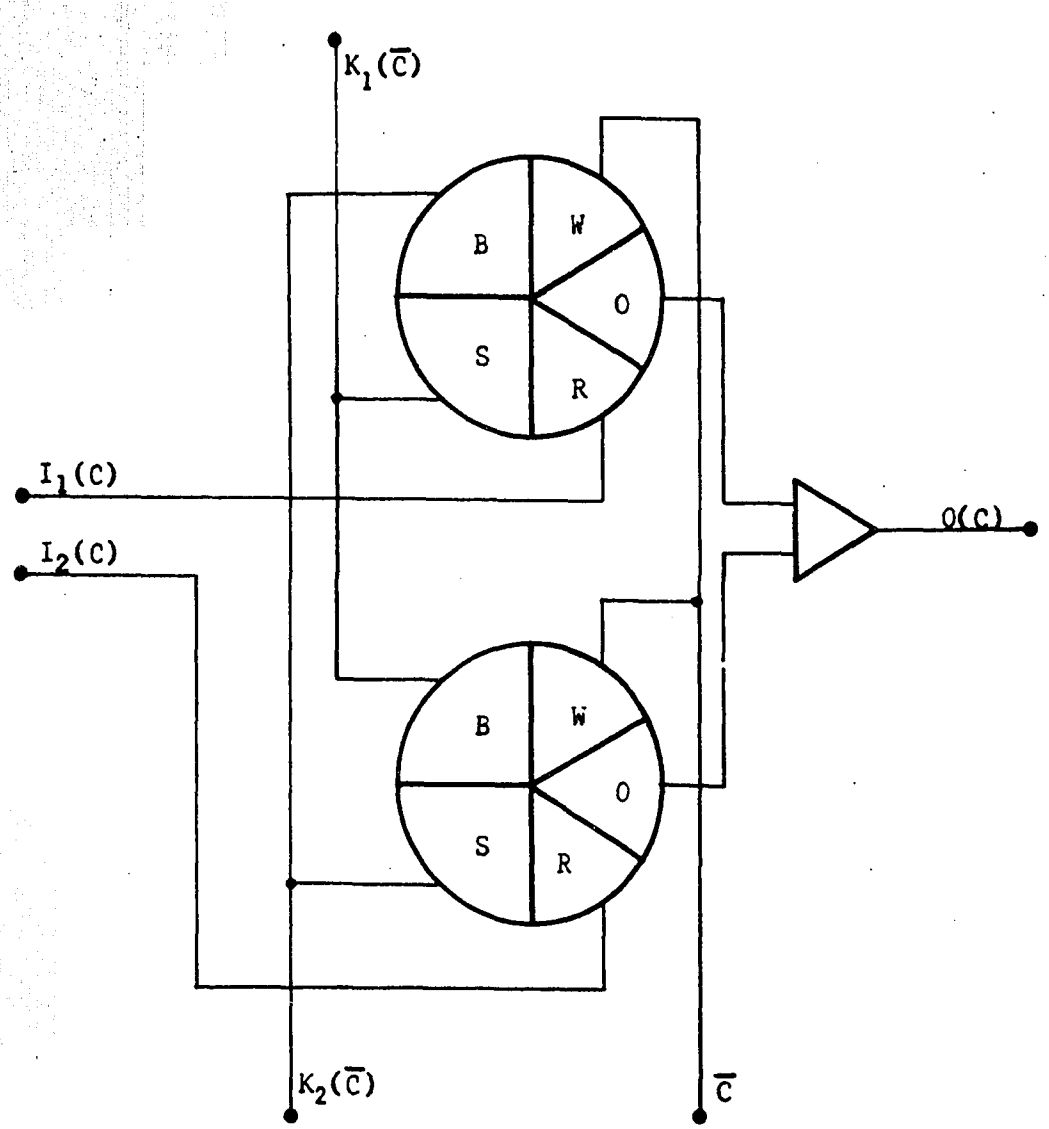


Figure 30. Signal diagram for the type I director

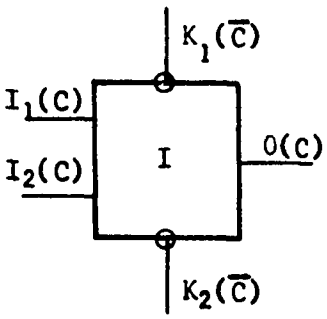


Figure 31. Representation for the type I director

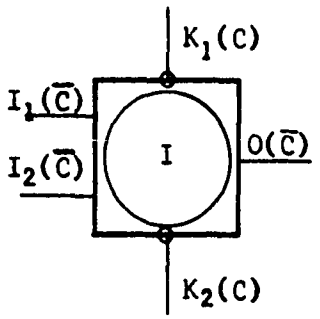


Figure 32. Representation for the complementary type I director

The operation of the type II director is facilitated through the use of two transfluxers and two sense amplifiers. Figure 33 shows the signal diagram for this device. The control inputs K_1 and K_2, whose pulses lead the signal pulses by one unit and are thus coincident with $\overline{C}$, are used to block and set the transfluxers. If K_1 is pulsed, the ensuing input pulse will appear at the output O_1. Similarily a K_2 input will cause O_2 to be selected for the next output. The input pulses are used to read the transfluxers and the write function is accomplished by $\overline{C}$. The notation used for the type II director is shown in Figure 34.

A useful variation of the type II director is a device that can be used to direct pulses that are coincident with $\overline{C}$. Such a device will be called a complementary type II director and the only change necessary in the signal diagram is that C be used to write the transfluxers in place of $\overline{C}$. Figure 35 shows the notation that will be used to represent the complementary type II director.

B. Major Functional Units

1. Integrator

The integration of a pulse-density-coded analog variable is a matter of counting the number of pulses contained in the representation. This is a major advantage of pulse-density coding. The integration process will usually be an integration from time zero to time t, however integration between definite limits is also possible and in certain applications may be necessary.

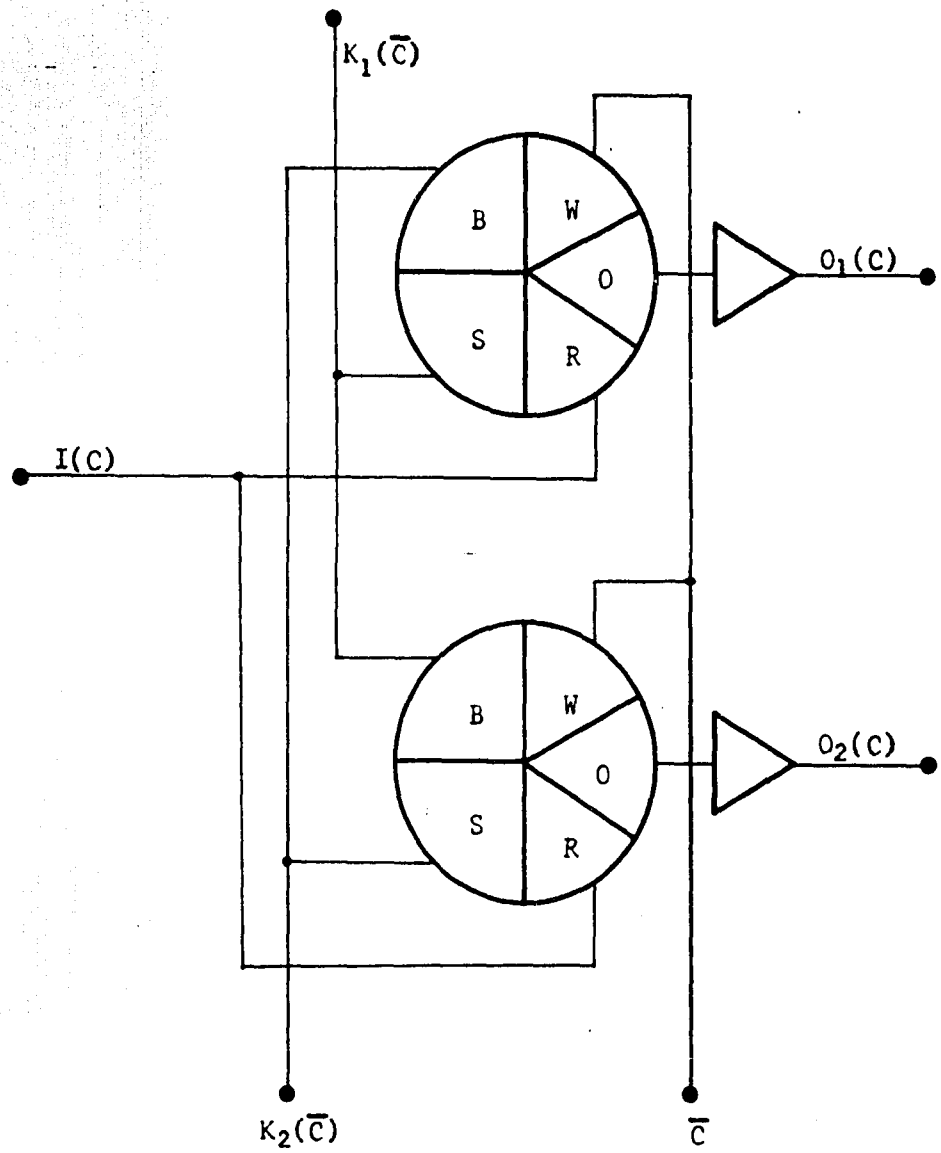


Figure 33. Signal diagram for the type II director

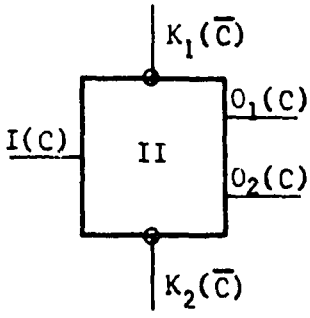


Figure 34. Representation for the type II director

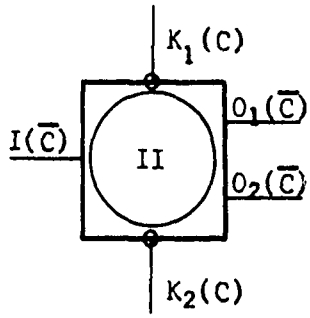


Figure 35. Representation for the complementary type II director

The use of flip-flops as the basic counting medium provides a binary count of the number of pulses. Since the variable will probably not maintain the same sign over the entire desired period of integration, the counter within the integrator must be capable of forward or backward counting. During forward counting each successive pulse results in an increase in the binary value represented by the states of the counter, whereas in the backward counting mode each pulse decreases the value.

Recall that a Q output from a flip-flop indicates a transition into the "one" state and that a $\overline{Q}$ output indicates a transition into the "zero" state. If several flip-flops, three for instance, are cascaded by connection of the Q outputs to the subsequent inputs, backward counting is achieved. To illustrate this, assume that the flip-flops are all initially set in the one state. The first flip-flop represents the least significant bit of the number represented by the counter. The first pulse into the counter will cause the state of the first flip-flop to change to the zero state, indicated by a $\overline{Q}$ output, but since there is no Q output from the first flip-flop the states of the remainder of the flip-flops will remain unchanged. As a result of this pulse, the binary number represented by the counter has changed from 111_2 or 7_{10} to 110_2 or 6_{10}. Subsequent pulses into the counter will cause subsequent reduction of the number represented at the rate of one unit per pulse. Cascading of the flip-flops with the use of their $\overline{Q}$ outputs will result in forward counting. To illustrate this, assume that the flip-flops are all initially in the zero state. When the first pulse enters the counter the state of the first flip-flop will be changed to the one state, resulting in a Q output, and the states of the remainder of the flip-flops will remain unchanged. The binary number

represented by the states of the flip-flops in the counter has thus been changed from 000_2 or 0_{10} to 001_2 or 1_{10}. Each subsequent pulse will increase the value of the number by one unit.

For integrator applications, a counter is required which can be made to count either forward or backward on the basis of the desired effect of each pulse. This can be accomplished by the addition of a type I director between adjacent flip-flops in the counter. Using this type of arrangement, since a control pulse supplied to the directors must be supplied one unit in advance of each ensuing counter input pulse, the desired direction of the count (forward or backward) will be established prior to each signal input thereby insuring the desired effect. This means that the signal to be integrated must actually be delayed at least one unit (with respect to the integrator input), although this is not an ideal arrangement, it imposes no serious limitation on the computer for the purposes intended.

Sign representation of the integral may be accomplished in several ways. The previously constructed voltage pulse device represented the sign of the integral by the state of a flip-flop external to the counter. This method simplifies the circuitry external to the integrator, however the circuitry within the integrator becomes greatly complicated. The problem lies in the fact that with external sign representation, the decision as to whether the counter should count forward or backward for a particular input pulse is a function not only of the sign of the input at that time but also the sign of the integral. Furthermore, when the value of the integral is zero the device must be capable of changing the rules by which the decision will subsequently be made, i.e. the

integral sign must be allowed to change. It was possible to facilitate these decision making capabilities in the voltage device because the computer involved was not synchronous. In a synchronous computer, these decisions represent too many operations that must be performed simultaneously and the circuitry required to facilitate this complicates the design beyond reality.

The proposed computer will use complementary representation for negative integrated values. This method is used in most digital computers to represent negative binary numbers. To facilitate this the integrators will contain one more flip-flop in the counter than would be necessary to represent the actual binary number. The last flip-flop in the counter, which represents the most significant bit, will represent the sign of the integral. The actual method of representation used is what is known as "two's complement" representation. In this system, 011•••11 represents the maximum positive value, 000•••01 represents the smallest positive value, 000•••00 represents zero, 111•••11 represents the smallest negative value, and 100•••00 represents the maximum negative value.

Using the selected method of integration, the construction of the integrator is relatively simple. Figure 36 shows the signal diagram for the integrator. The two signal inputs are coincident with C and are used to set the forward or backward direction of the counter and to create the signal input for the counter which is delayed one unit in the inhibitor. The logical AND and the inhibitor are necessary to prevent a counter input pulse from being created if there is a simultaneous input on both integrator signal inputs. The length of the counter

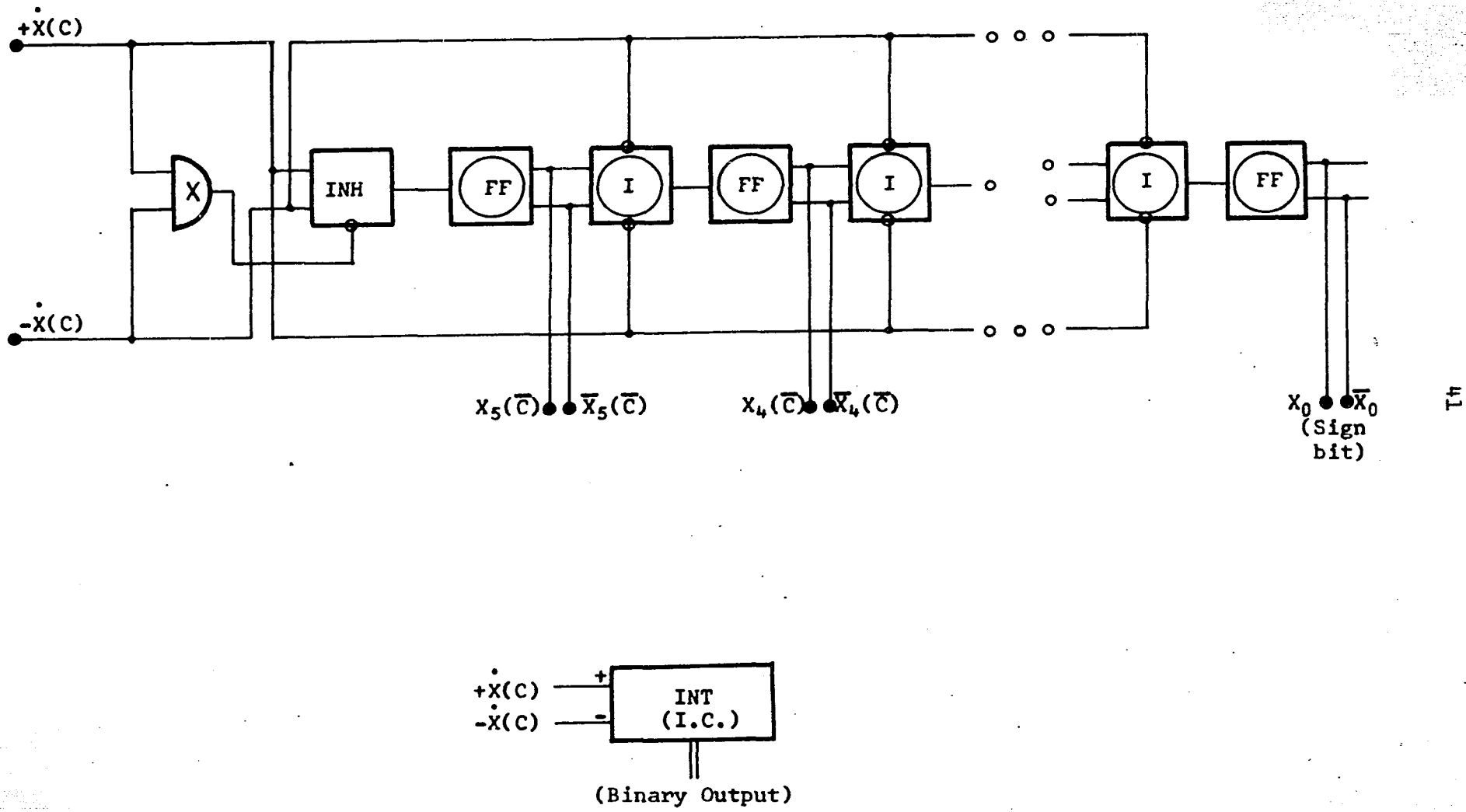


Figure 36. Signal diagram and representation for the integrator

in the integrator is equal to the number of significant bits plus one. Complementary flip-flops were used in counter not only to eliminate an additional unit delay in the device but for reasons that will become more apparent after the multiplier unit is considered.

The symbol used to represent the integrator unit is shown in Figure 36. Initial conditions must always be set on each flip-flop. These initial conditions are stated in parenthesis within the integrator symbol as shown and if none are stated the initial conditions are assumed to be zero. The manner in which these initial conditions are imposed on the integrator is perhaps the most important factor that limits the proposed computer to special purpose applications. These conditions are wired into the integrator flip-flops in a continuous circuit. Since there is a finite number of block and set windings that can be wound on any one flip-flop the number of different sets of initial conditions that can be imposed on a particular integrator is limited. For special purpose applications this scheme would be satisfactory in most cases, but for general purpose computation some other scheme would have to be used.

It should be noted that the integrator unit has no "built in" provisions to prevent "integration over the top" or "overflow", i.e. counting forward when the counter is full. In the normal application such an error would be easily detected from the results since the binary number one unit greater than the largest positive number is the largest negative number and visa-versa. Such a discontinuity in an integrated variable in the physical case could only be accounted for by this type of error, however if it is necessary to detect such an error detection

could be most easily accomplished by the addition of another director and flip-flop after the sign bit. Such an error would then be indicated by a change in state of this flip-flop whose output could be used to inhibit further input into the integrator in the incorrect direction and/or to signal the operator that such an error has occurred.

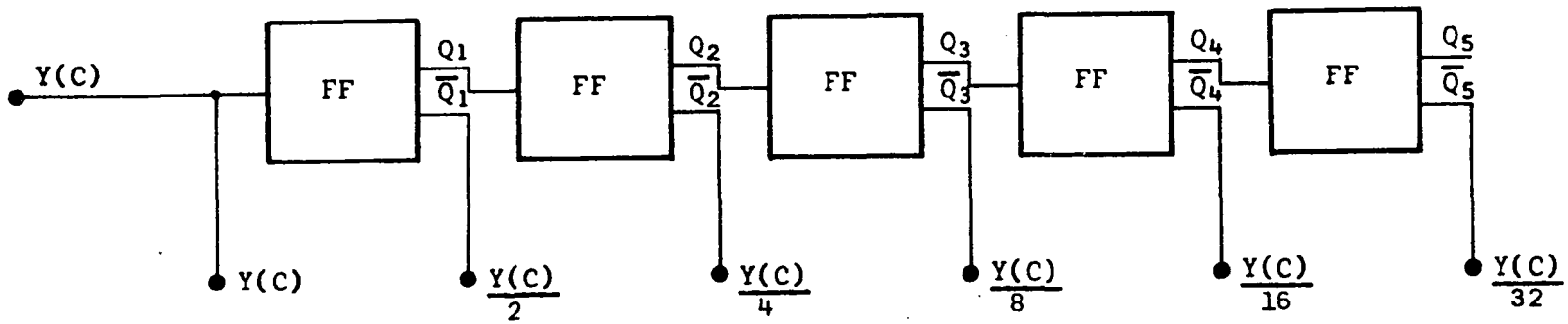
2. Multiplier

The multiplication of pulse-density variables is a process of logically combining a pulse train and a binary number. The multiplier is also used as a converter to convert a binary number (the output of an integrator) into the pulse-density representation of its value. This is accomplished by the multiplication of the binary number by the clock. The multiplier can be used to multiply any two quantities within the computer if one of them is or can be represented by the output of an integrator. The multiplication process is facilitated by the use of two minor assemblies, the binary scaler and the matrix buffer, which will be discussed separately in some detail.

The binary scaler is a frequency dividing unit. It is composed of an arbitrary number of cascaded flip-flops. The Q output of a flip-flop will contain half as many pulses as are contained in its input signal, since the input pulses are alternately emitted on the Q and $\overline{Q}$ outputs. This means, in terms of frequency, that the output frequency of the flip-flop is half that of its input. Similarily if this output is used to drive a second flip-flop, the output of the second will have a frequency one-fourth that of the original input. Thus, the output frequency of the N'th flip-flop is equal to the input frequency divided by 2^N.

The signal diagram and the representation for the scaler are shown in Figure 37. Note that the Q outputs of each flip-flop is connected to the input of the next flip-flop and also that the actual outputs of the scaler are taken from the $\overline{Q}$ outputs of the flip-flops. All of the flip-flops of the scaler are initally reset to the zero state, this means that the first pulse to enter the scaler will cause a Q output from all of the flip-flops in the scaler but will not cause any $\overline{Q}$ outputs. The next pulse causes a $\overline{Q}$ output from the first flip-flop, the next pulse causes a $\overline{Q}$ output from the second, and so on. Each successive pulse will cause one and only one $\overline{Q}$ output until all of the flip-flops eventually end up in the zero state again and then the next pulse will not cause a $\overline{Q}$ output from any of the flip-flops. This action is most easily understood by examination of the outputs for a particular input. Figure 38 shows the outputs of the scaler for an input of C, note that the components of the scaler output are underlined. The length of time between the adjacent times when the states of all of the scaler flip-flops are in the zero state is referred to as the period of the scaler and is what determines the period of variable representation within the computer. It is important to note that during this period the superposition of all of the flip-flop outputs contains one less pulse than the input. For a five-bit computer at 20 kc clock frequency, the length of this period is 1.6 milliseconds.

The purpose of the matrix buffer is to combine the several outputs of the binary scaler logically with the binary number output of an integrator to form the numerical product of the two. The process of



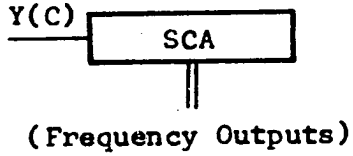


Figure 37. Signal diagram and representation for the scaler

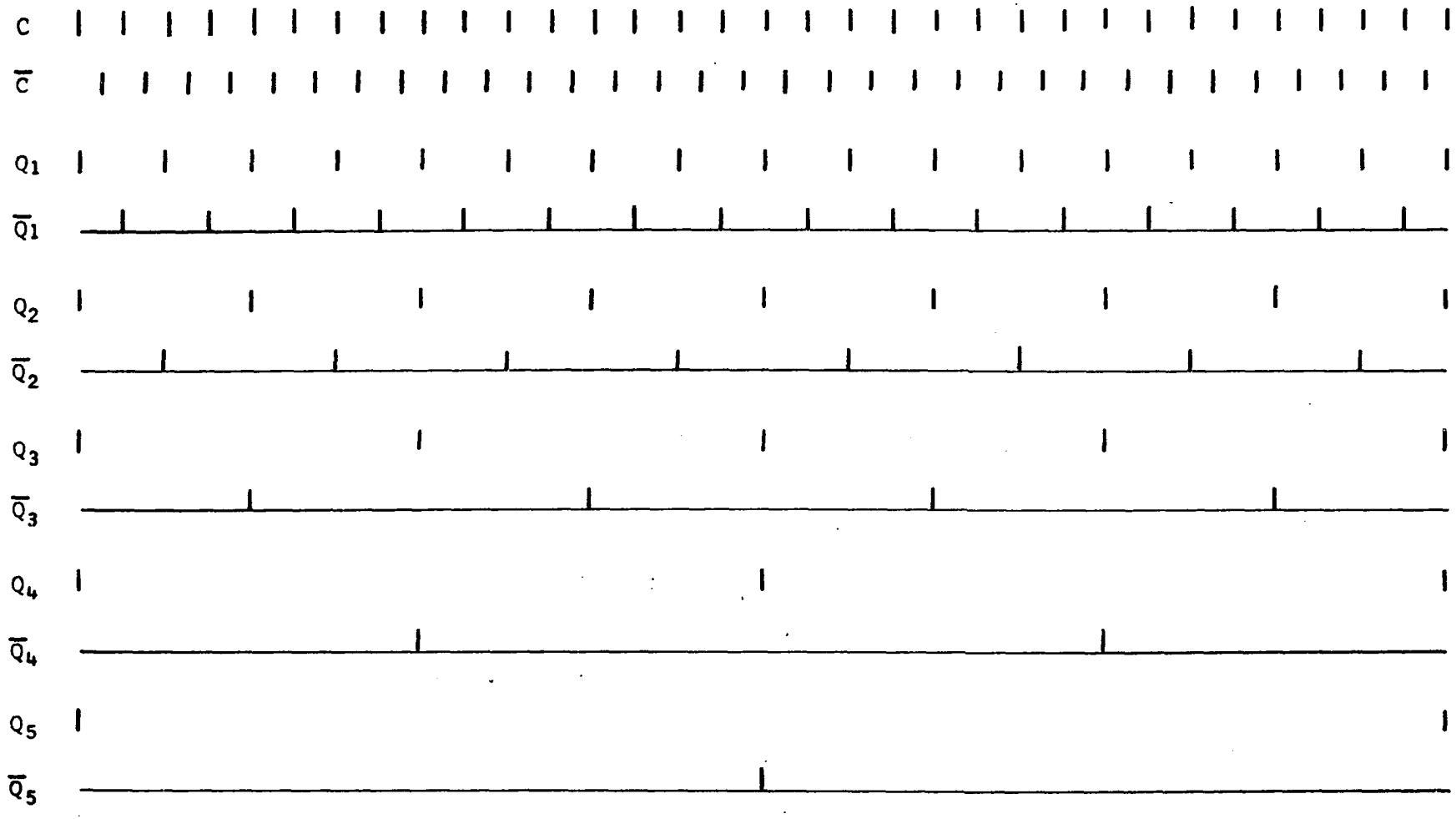


Figure 38. An illustration of the scaler output

multiplication is conceptually simple in this type of computer. It involves forming the logical product (AND) of the binary frequency divided outputs of the scaler respectively with the bits of the binary number represented by the integrator outputs and determining the logical sum (OR) of these products. The logical products are made starting with the combination of the most significant bit of the binary number (excluding the sign bit) which represents a value of one-half, with the output of the first scaler flip-flop which represents half of the frequency or number of pulses in the input, and so forth down to the least significant bit and the output of the corresponding stage in the scaler.

Figure 39 shows the signal diagram for the basic matrix buffer and the notation used to represent it. The multiplication process is facilitated by the use of transfluxers that are blocked and set by the $\overline{Q}$ and Q outputs of the integrator stages respectively, recall these outputs are coincident with $\overline{C}$. The transfluxers use the corresponding scaler outputs, coincident with C, to read the transfluxers and the write function for all of the transfluxers is performed by $\overline{C}$. The required output OR function can be accomplished by serial connection of the sense windings to a sense amplifier. This is possible because the outputs of the scaler stages ($\overline{Q}$'s) are never coincident, see Figure 38.

In addition to the logical function of forming the product, the matrix buffer used in the actual multiplication process must also compensate for the two's complement representation used for negative numbers within the integrator. Recall that the final output must be composed of two pulse trains, one representing the positive value of the product and the other representing the negative value. For the present, assume that

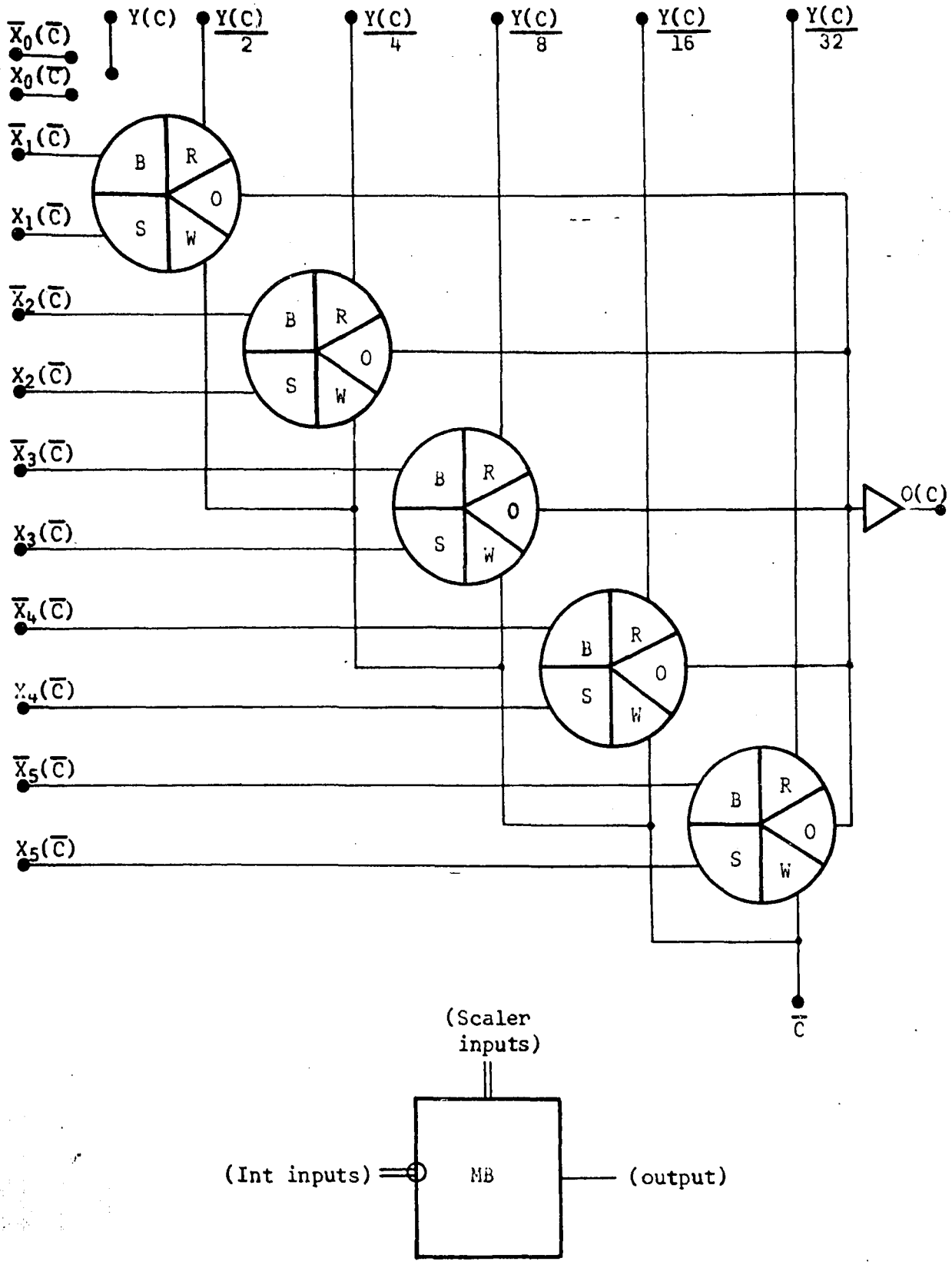
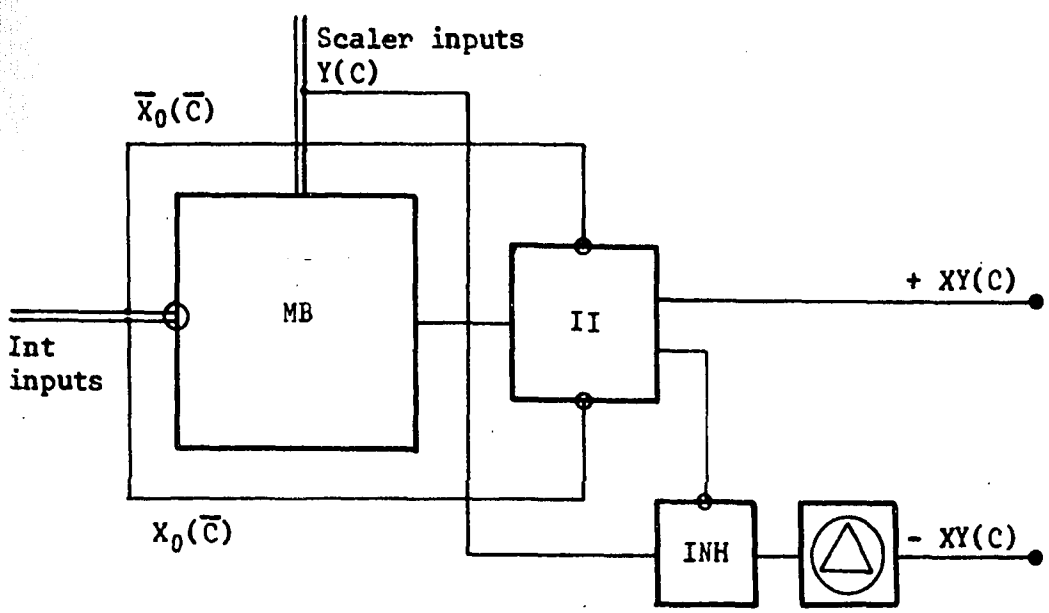


Figure 39. Signal diagram and representation for the basic matrix buffer

the pulse-train input to the scaler represents a variable that either possesses no sign or is always positive. An example of this assumption is the use of the multiplier to convert a binary number to its equivalent pulse-train representation by multiplication by the clock. When the binary number is positive the output of the buffer amplifier is actually the pulse-density representation for the product and can be routed directly to the positive signal output. When the binary number is negative however, the output of the sense amplifier is actually the complement of the pulse density that it should be. To illustrate this, consider the case of the smallest negative number, i.e. 111•••11. All of the bits of this number except the sign bit are connected to the binary inputs of the buffer. The output of the buffer in this case then will contain the sum of all of the pulses from the scaler which recall for one period of the scaler contains all of the scaler input pulses but one. The correct negative output for this binary input would contain one pulse per period. The correct output can be formed by the use of the inhibitor. The signal diagram and the symbol used to represent the matrix buffer with multiplier logic are shown in Figure 40. If the amplifier output is used to inhibit the same pulse train that is being scaled, the complement of the amplifier output is formed. Thus in this case the amplifier output would inhibit all but one of the pulses per cycle thus forming the correct result.

A type II director provides the proper control function to determine whether the amplifier output pulses will be directly routed to the positive output or will be routed through the complementing circuitry and subsequently to the negative output. The control inputs to the director



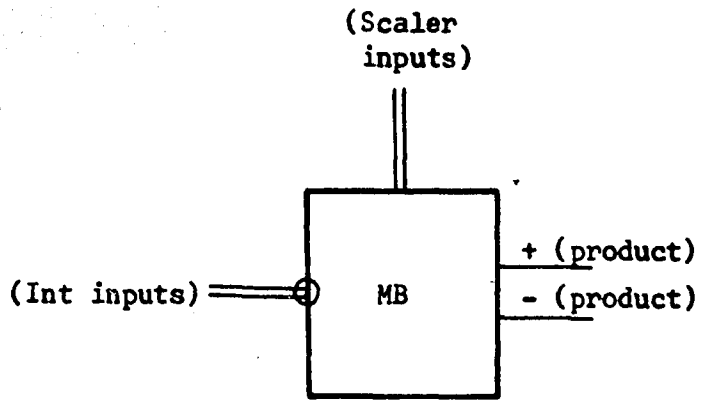


Figure 40. Signal diagram and representation for the matrix buffer with multiplier logic

will be the output of the sign bit of the integrator. The only difficulty encountered in this process is that the inhibitor involves a mandatory one unit delay and since the input is coincident with C and the output is required to be coincident with C, an additional unit delay is required after the inhibitor. It is believed that this does not represent a serious problem because with a clock frequency of 20 kc this entails a signal delay of only 50 microseconds.

The signal diagram for the matrix buffer doesn't show any initial conditions on the transfluxers, however, the same initial conditions that are imposed on the integrator bits will be imposed on these transfluxers since the transfluxer that is blocked in each flip-flop will produce an output that will reset the corresponding matrix element if the state is different than the previous state of the flip-flop.

The multiplication of a binary number times a pulse train representing a variable (conveying a sign) will be considered in the chapter concerning applications.

3. Adder

The addition of two pulse-density-coded analog variables could be accomplished by their logical addition (OR) if the pulses within the two pulse trains were never coincident. In the case of coincident pulses in the two trains, the output of the logical OR would contain only one pulse and the correct sum would be two. Since two pulses cannot occur in the same pulse space of one pulse train, the extra pulse needed to correct the sum must be inserted into the pulse train representing the sum at some time later when a "hole" (a pulse space in the

sum that does not contain a pulse) exists. The problem here is that there is no way to predict when the next hole will exist, or how many more input coincidences may exist before the appearance of such a hole. The adder then must have the capability of remembering how many times coincidence has occurred, sensing the holes in the output and inserting pulses in the holes to compensate for past coincidences.

The signal diagram and representation used for the adder unit are shown in Figure 41. The pulse trains are checked for coincidence by a logical AND whose output (indicating the occurrence of such a coincidence) increases the binary number (initially zero) represented by the integrator. The input trains are then logically added in the delayed logical OR whose output is used in conjunction with an inhibitor to detect the occurrence of holes in the pulse train. When such a hole is detected the inhibitor produces an output, coincident with C, which is used to read all of the cores in the matrix buffer. If any of the bits of the binary number are non-zero a buffer output is created which is added to the pulse-train output of the adder in the final logical OR and is also used to reduce the count in the integrator by one, unless at the same time another coincidence in the adder inputs is detected.

The only difficulty in the design of the adder is that there exists a two-unit delay between its input and output. This is an unfortunate occurrence that cannot be avoided in a synchronous computer because the addition process requires that too many decisions be made simultaneously to allow zero delay in the device. The length of the integrator, in binary bits, is not necessarily the same for this integrator as the other integrators within the computer. The

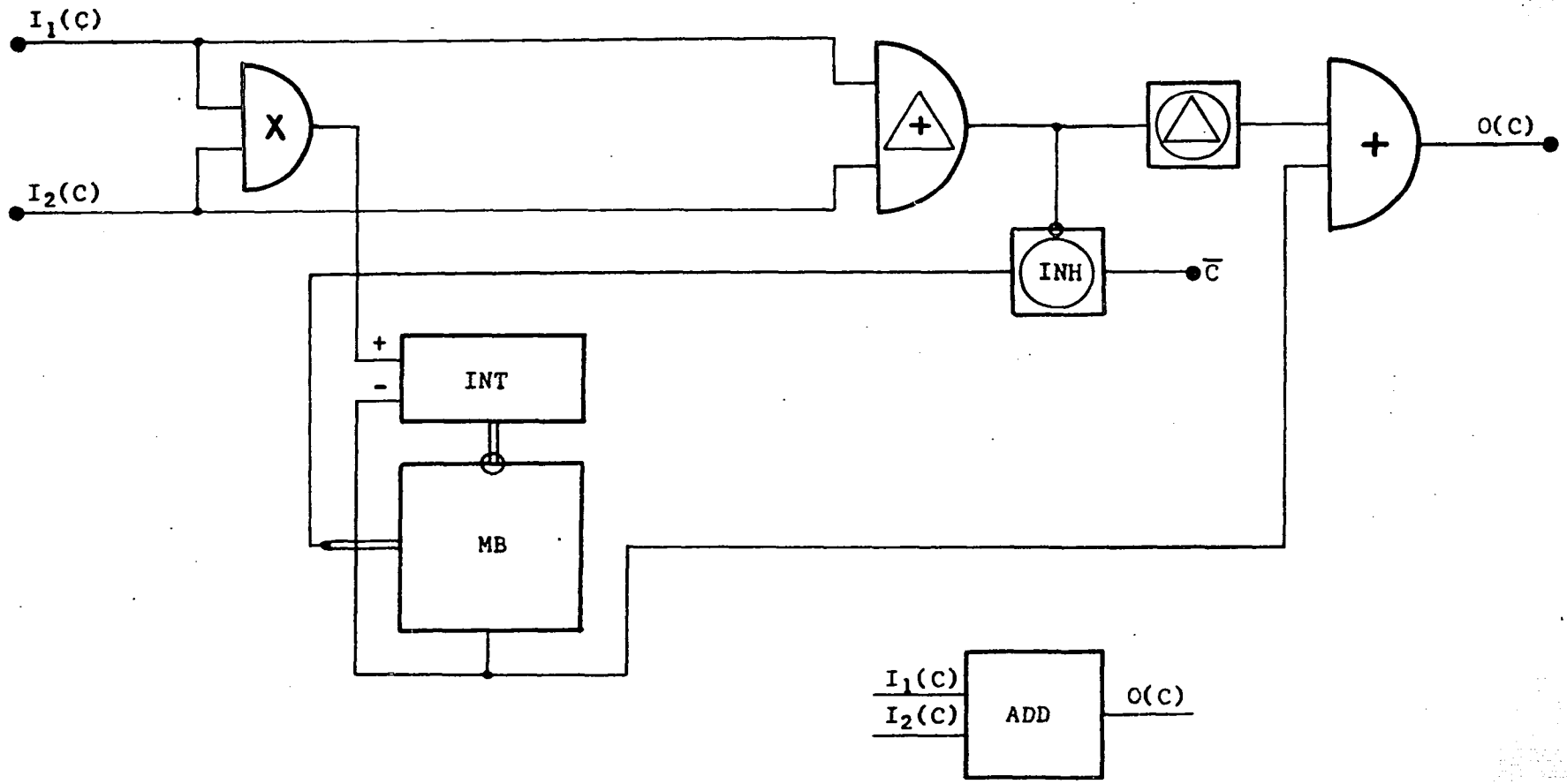


Figure 41. Signal diagram and representation for the adder

required length of this integrator is dependent upon how many input-pulse coincidences can occur before a hole appears in the output pulse train. This is a function of the magnitude of the expected inputs and may be rather difficult to predict for a particular application.

It should be noted that since no overflow prevention or correction scheme has been built into the integrator unit itself, in this particular application, if an overflow occurs, indicating that too many coincidences have occurred before a sufficient number of holes have appeared, these coincidences will all be lost. These coincidences will be lost because, since this particular integrator possesses no sign bit, the binary number one unit in excess of the largest positive number is zero. This type of error would represent a rather serious loss in accuracy but unlike the situation that occurs when the limit of the integrator (with sign bit) is exceeded, this overflow is not immediately externally apparent. Such an error could be detected by the addition of another stage to the integrator, i.e. one more than the number believed to be necessary in the particular problem, which similiar to the sign bit of the normal integrator is not connected to a control input of the matrix buffer. The output of this stage would then be used to signal the operator that such an error had occurred. When such an error is indicated, the operator may circumvent the difficulty by either re-scaling the variables involved or increasing the number of stages in the integrator and matrix buffer.

VI. APPLICATIONS

A. Solution of Differential Equations

Since the proposed computer solves differential equations by the same techniques as used in conventional analog devices, i.e. by integration, equations are simulated in the same manner. This method involves rearrangement of the differential equation(s) so that the highest order derivative of the dependent variable appears alone on one side of the equality and has a coefficient of one. This derivative forms the input to an integrator whose output will be the next lowest derivative of the dependent variable. Subsequent integrations may be necessary to form the dependent variable. Using the proper combinations of the outputs of the various integrators and the appropriate driving function(s), the input to the first integrator is forced to satisfy the form of the differential equation being simulated.

In the proposed computer, variables are represented by two pulse trains. One of the pulse trains represents the positive magnitude of the variable, i.e. contains a number of pulses proportional to the magnitude of the variable when the variable is positive, and the other pulse train represents the negative magnitude of the variable. The simulation of an algebraic equation with this type of representation is accomplished by the parallel addition of the appropriate magnitudes of the variables. For example, the simulation of the algebraic equation:

$$A = X - Y,$$

would be accomplished as:

$$(+A) = (+X) + (-Y)$$

and $$(-A) = (-X) + (+Y).$$

To illustrate the method of differential equation solution used with the proposed computer, consider the linear first order differential equation:

$$\dot{x}(t) + K\, x(t) = f(t)$$

in which K is a positive constant, f(t) is an arbitrary function of time and x(t) is initially zero. Since f(t) is an arbitrary function of time it must be represented by two pulse trains within the computer, however since K is known to be positive, it may be represented by one pulse train. Rewriting the differential equation in the form:

$$\dot{x}\,(t) = f(t) - K\, x(t)$$

yields the simulation equation. This equation states that $\dot{x}(t)$, which will be used as an integrator input, is composed of the sum of the pulse trains f(t) and minus K x(t). The output of the integrator will be a binary number representing the value of x(t), the dependent variable in this case, and subsequent integrations will not be necessary. This binary number must, however be multiplied by the clock to form the pulse train representation of x(t). This number must also be multiplied by the pulse train K to form the product K x(t) which is necessary to satisfy the simulation equation. Figure 42 shows the computer diagram for this simulation.

To analyze the actual mechanics of the solution of this equation, for the present let f(t) be some fixed positive value (F). Under these

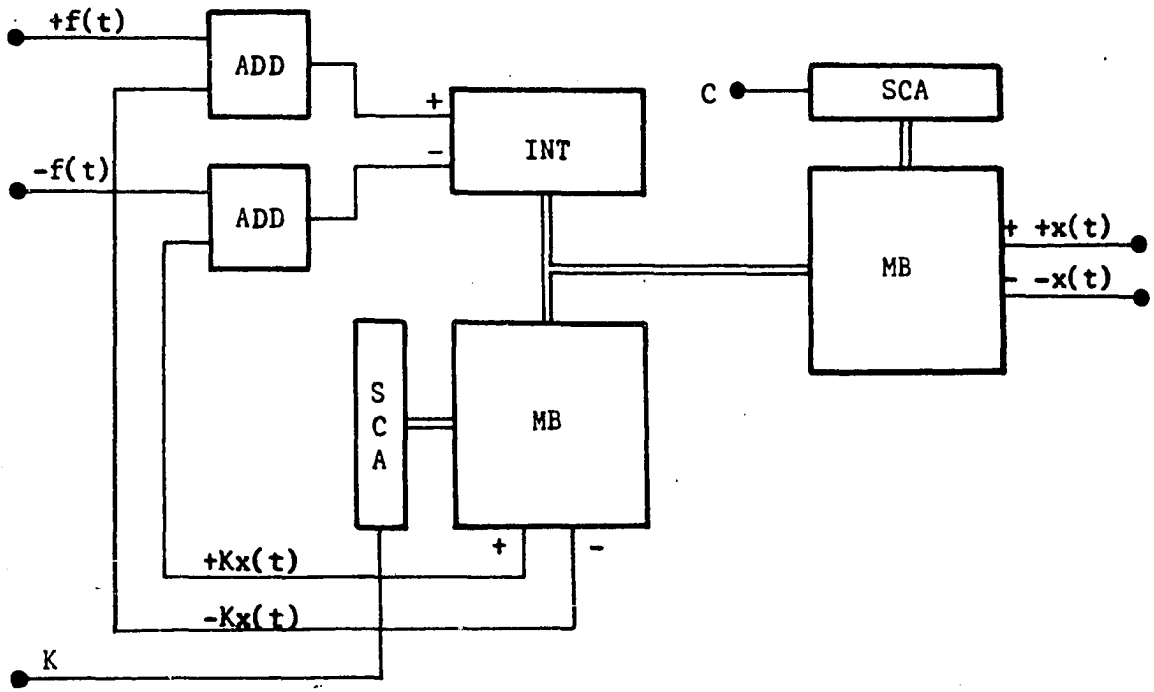


Figure 42. Computer diagram for the simulation of a linear differential equation

conditions, the equation corresponds to a first order system with a step input, which has a solution of:

$$x(t) = \frac{F}{K} (1 - e^{-Kt}).$$

Expressing this solution in words, x(t) will start at a value of zero and increase exponentially with a time constant of 1/K to its final value of F/K.

In the pulse-density simulation of the differential equation shown in Figure 42, the binary number (x(t)) is initially zero thus at time zero plus (immediately after time zero) the K x(t) product will contain no pulses regardless of the value of K because of its multiplication by zero. The pulses occurring in +f(t) at time zero plus will cause x(t) to attain a small positive value which will result in a +K x(t) product which contains a small number of pulses (compared with the total number of pulses in one cycle of representation of K). These pulses will appear at the negative integrator input which would decrease the value of x(t) if it were not for the presence of the f(t) pulses at the positive integrator input. The net result is that the pulses in +K x(t) will cancel some of the pulses in +f(t), thereby decreasing the rate of increase of x(t) as x(t) increases. Finally, when x(t) attains a large enough value so that the number of pulses in the +K x(t) input is the same as the number of pulses in the +f(t) input, x(t) will cease to increase and will have attained its final value. At this time, K x(t) will equal F, or x(t) will be F/K, its predicted final value. The length of time required for the value of x(t) to reach its final value will be determined by the value of K and the number of significant bits of the binary

number x(t). It is important to note that although external to the computer the number of binary bits in the integrators determines only the accuracy of variable representation, within the computer the number of significant bits defines explicitly the relationship between the binary values and the corresponding pulse-density representations. This relationship is the determining factor concerned with the period of variable representation. This period is an inverse function of the clock rate and a direct function of the number of significant bits. For a five-bit system with a 20 kc clock rate this period is 1.6 milliseconds. Note that each additional bit will, for the same clock rate, double this value. The significance of this period is that to determine the actual value of a variable within the computer, the number of pulses occurring in the pulse-density representation for the variable must be counted for this length of time to insure a correct result. This is not a particularly serious problem within the computer, because each pulse is treated individually, however when the computer is interfaced with a non-digital system this time becomes an important consideration.

The differential equation previously solved can be easily simulated using conventional analog equipment, however some variations of this problem that can be solved with the same simulation shown in Figure 42 are not so easily solved using conventional voltage devices. One such variation is obtained if K instead of being a constant is set equal to K_1t. For the purposes of this example let f(t) equal F_1t. Note that the new K function is positive (for positive values of time) and thus satisfies the constraint imposed on the previous problem. The resulting equation is:

$$\dot{x}(t) + K_1 t\, x(t) = F_1 t.$$

This equation is more difficult to solve on conventional voltage devices because the formation of the $K_1 t x(t)$ product involves variable multiplication which is a relatively expensive process in conventional equipment. This equation is solved by the multiplication of both sides of the expression by an integrating factor of e raised to the integral of $K_1 t\, dt$ power. The resultant equation possesses an exact differential on the right-hand side which can be easily integrated and solved for x(t). Assuming, as before, that x(0) is zero, the solution becomes:

$$x(t) = \frac{F_1}{K_1}\left(1 - e^{\frac{-K_1 t^2}{2}}\right).$$

This solution is similiar to the previous solution except that x(t) achieves its final value exponentially with t^2.

Many other variations of the first order system shown are possible, however most problems encountered in practice are higher order systems. Higher order systems are solved in exactly the same manner as the preceeding equations except, of course, that most units are required.

Another type of equations that can be readily solved with the proposed computer are nonlinear differential equations. To facilitate this, however the ability to multiply a binary number times a pulse train with a sign is often required. Accomplishing this multiplication is not difficult but the method that appears to be the most obvious does not yield the correct result.

It would seem that since a pulse-density variable is represented by wo pulse trains, one representing the positive magnitude and one representing the negative magnitude, and since the actual magnitude of a

product is not dependent upon the signs of the factors being multiplied, that the absolute magnitude of a pulse-density variable could be used in the multiplication process. It would seem that the absolute magnitude could be formed by the same type of arrangement used at the input to the integrator unit which would add the pulses in the two trains (eliminating any coincidence) and produce a single output train representing the absolute magnitude of the input variable. The fallacy in this approach is that the output of such a combination is not the absolute magnitude because the absolute magnitude of such a variable can only be determined by its integration over one cycle of its representation.

To illustrate this fallacy, assume that the pulse-train variable is to be multiplied by a binary number with the constant value of 0.010••0. This value indicates that every fourth scaler input pulse will create a matrix buffer output. Assume that the pulse-train inputs to the device will be composed of repeated series of three pulses at the positive input followed by one pulse at the negative input. During the three positive input pulses, the multiplier-output sign logic (logic added to compensate for the pulse train sign) would be set so that any resulting output would cause a positive-product output. Since the matrix buffer is set so that only every fourth input pulse causes an output, no output will occur during this time. When the negative input pulse arrives, the multiplier-output sign logic is switched so that any output caused by this pulse would be emitted on the negative-product output. Since this pulse is the fourth pulse into the scaler, it creates an output. When the series is repeated the same thing happens, and over a period of time, although

there have been three times as many positive inputs as negative inputs, all of the product pulses show a negative sign.

The valid solution to this problem involves more components, but does yield the correct result. The correct results are obtained by using separate scalers and separate matrix buffers for the two pulse trains. The outputs of the two buffers are combined with two logical ORs into the two appropriate output pulse trains. The signal diagram for this configuration is shown in Figure 43. A standard symbol is not used for this configuration because there are a number of variations of this idea that may be used to fit the particular situation.

Considering the previous illustration using this scheme, the first three pulses at the positive input will cause no output from the positive output, since the matrix buffers are set so that every fourth pulse causes an output. The negative pulse (the first pulse to enter the other scaler) also does not cause an output. The first pulse of the next series of three pulses at the positive input does however cause a positive-product output. As the series is repeated again and again, every fourth series will produce a negative-product output, but between adjacent negative outputs there will be three positive-product outputs. Over a period of time there will be three times as many positive as negative outputs, and since there were three times as many positive as negative inputs, the correct solution has been obtained.

It should be noted that this solution to the pulse-rate variable multiplication problem possesses one weakness. Recall that the negative outputs of the matrix buffers lag the scaler inputs by two units whereas the positive outputs are simultaneous with the inputs. This could result in the

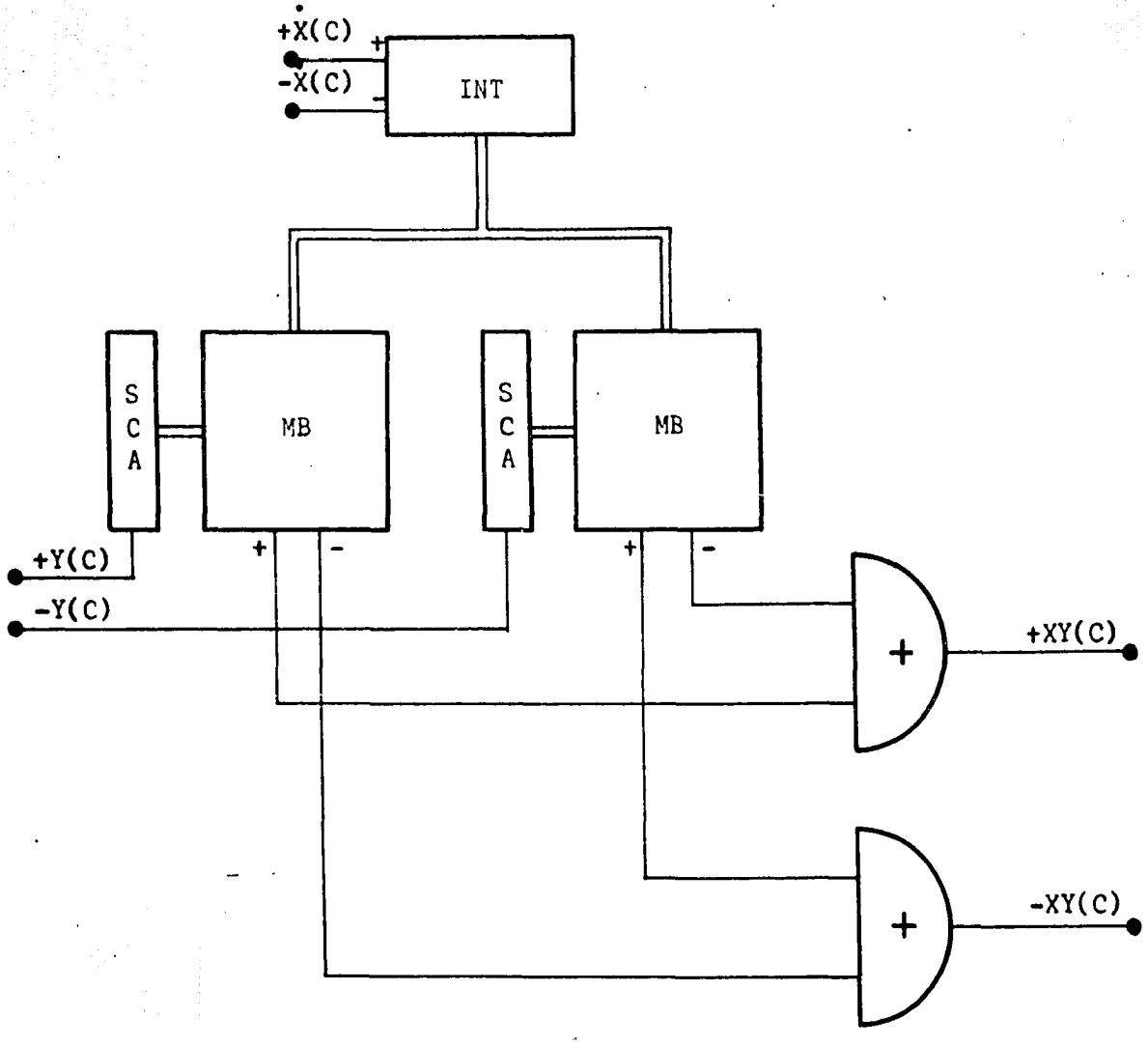


Figure 43. Signal diagram for the pulse train multiplication scheme

loss of one pulse when the binary number changes sign because it could result in a negative product output from one of the buffers occurring simultaneously with a positive output from the other buffer. The probability of this seems rather slight since sign changes can occur only when the value of the binary number is zero which means no buffer output regardless of the pulse-train input. If in a particular application this proves to be a problem, two solutions seem apparent and although not believed to be necessary in general would eliminate the problem. The first is to add two units of delay (a unit delay followed by a complementary unit delay) in each buffer's positive output. The second solution would be to use adders instead of logical ORs to form the multiplier output.

The ability to multiply binary numbers by pulse-train analogies conveying signs allows the extension of the two previous examples to somewhat more general differential equations. This ability also facilitates the solution of a class of nonlinear differential equations that are often found in engineering problems. This class of equations is composed of equations whose terms are composed of products of the several derivatives of the dependent variable. The major limitation on equations of this type imposed by the capabilities of the computer is that when the simulation equations are written, they must contain no quotients. Pulse-train division will be subsequently discussed but at best such division is accomplished by an approximation process.

To illustrate the mechanics of the simulation of a differential equation of this type, consider the equation:

$$\ddot{x}(t) + \dot{x}(t)x(t) + x^2(t) = f(t).$$

This equation results in the simulation equation:

$$\ddot{x}(t) = f(t) - \dot{x}(t)x(t) - x^2(t).$$

Assuming that x(t) and its derivatives are initially zero, the simulation of this equation is accomplished as usual by assuming that the input to an integrator is $\ddot{x}(t)$. The output of this integrator will be the binary representation of $\dot{x}(t)$ which is multiplied by the clock to obtain the equivalent pulse-train analogy. When this analogy is integrated the binary equivalent of x(t) is obtained. Figure 44 shows a computer simulation for this equation. Manufacturing the terms necessary to satisfy the simulation equation can be accomplished several ways. The $\dot{x}x$ product can be formed by either multiplying the binary representation for $\dot{x}$ times the pulse-rate analogy of x or multiplying the binary representation for x times the pulse-rate analogy of $\dot{x}$. The former approach was used in Figure 44 because it permits more efficient use of components. Note that since the right-hand side of the simulation equation contains three terms the addition process could require as many as four adders since the adder is only a two input device. In this particular case, if x is assumed to be real, x^2 will be a positive quantity and thus can be represented by one pulse train.

In summary the types of differential equations that can be readily simulated by the proposed computer include linear differential equations with constant and non-constant coefficients and nonlinear differential equations composed of product terms. Equations are simulated by conventional analog means and the major limitations imposed on equations to

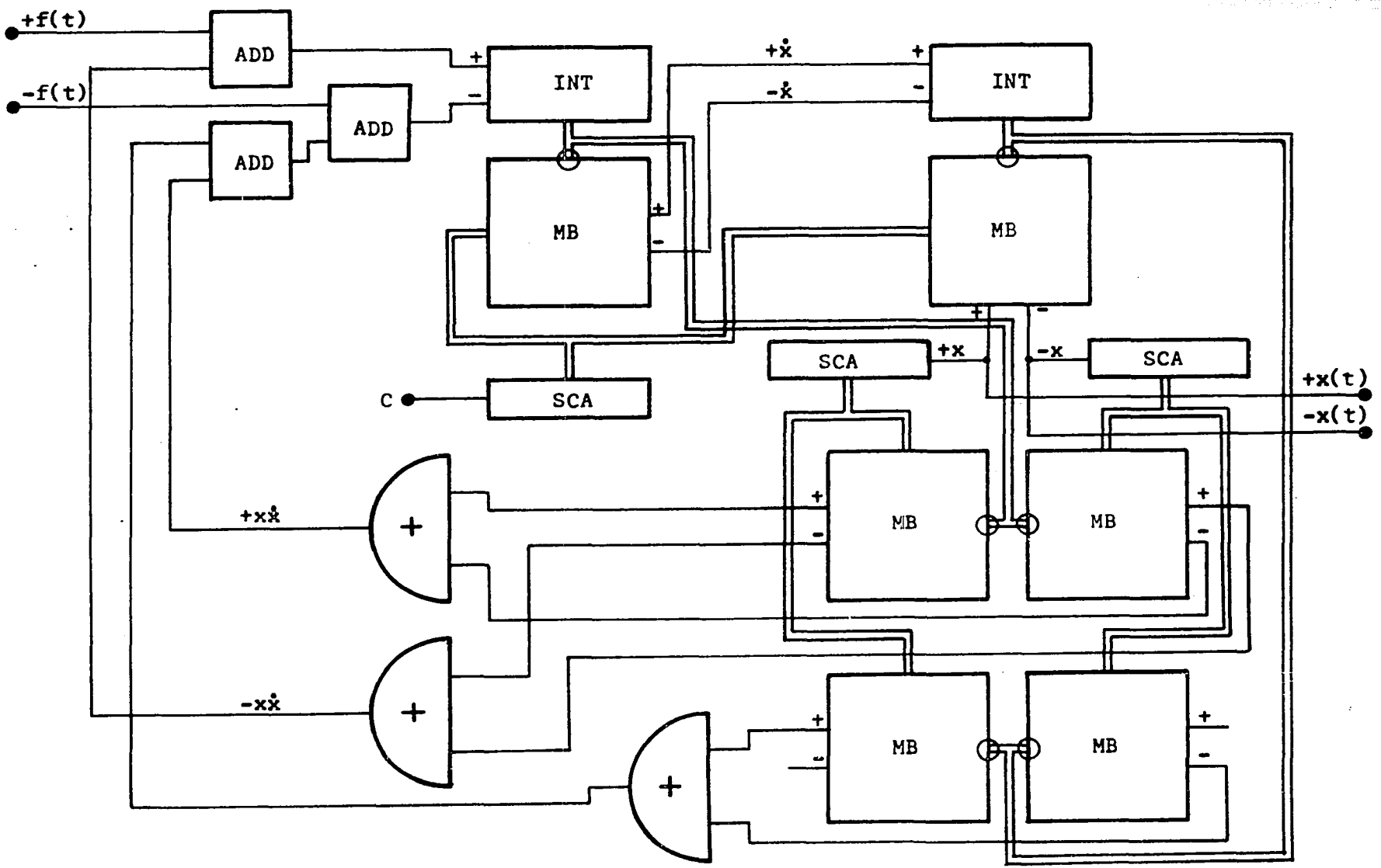


Figure 44. Computer diagram for the simulation of a nonlinear differential equation

be solved are that the terms of the simulation equation must be composed of products of the derivatives of the dependent variable and appropriate driving functions and must contain no quotients.

B. Computer Subroutines

Several computer subroutines can be used to extend the scope of the differential equations that can be simulated with the proposed computer. Most of these routines are of the seeking variety, i.e. routines in which there is a finite delay between the application of the input signal and the correct corresponding output. These routines then provide a means of simulating some of the functions that cannot normally be simulated, however because of their seeking nature, they are applicable only under a rather rigid set of limitations. Seeking routines are used in many conventional analog computing systems and within their limitations perform the desired functions satisfactorily. Examples of these devices are d-c servomechanical multipliers and mechanical resolvers. Three such subroutines will be considered for the proposed computer. There are many others that could be considered however the ones proposed are believed to be the most useful in the solution of flight control problems and in other engineering applications, and demonstrate the concept of such routines.

The first seeking subroutine considered performs two functions that might be necessary in the solution of particular types of problems. This routine determines the binary equivalent for a particular pulse-train analogy and also can be used to produce a pulse-rate representation for the time derivative of the pulse-rate analogy. This routine is based

on the solution of the simple linear differential equations with constant coefficients:

$$\dot{x}(t) + x(t) = x'(t)$$

in which x' is some arbitrary driving function. Figure 45 shows the computer diagram for this subroutine. The solution of this equation is exponential however if x' is essentially constant then in the steady state x(t) = x'(t). This means that if x' changes slowly with respect to the period of variable representation that x and x' will never differ by more than a fixed quantity (determined by how fast x' is allowed to change in one period of representation) and if x' assumes a constant value this difference will be approximately zero within a finite time after the constant value is attained. Furthermore under the specified assumptions, that x and x' are approximately equal, the input to the integrator which represents the time derivative of x will thus represent the time derivative of the input x'. In the simulation of certain types of equations, it is necessary to form the derivative of a function whose analogy was not obtained directly by an integration process. This subroutine could be used to form the derivative of such a quantity. In the implementation of certain types of equations it is necessary to form the product of two functions, neither of which are represented by a binary number. In these cases, this subroutine could be used to obtain the binary representation for one of the quantities which could be used as the binary input to the multiplication process.

A similiar type of seeking subroutine can be used to obtain a quotient also under a rather rigid set of limitations. The equation used

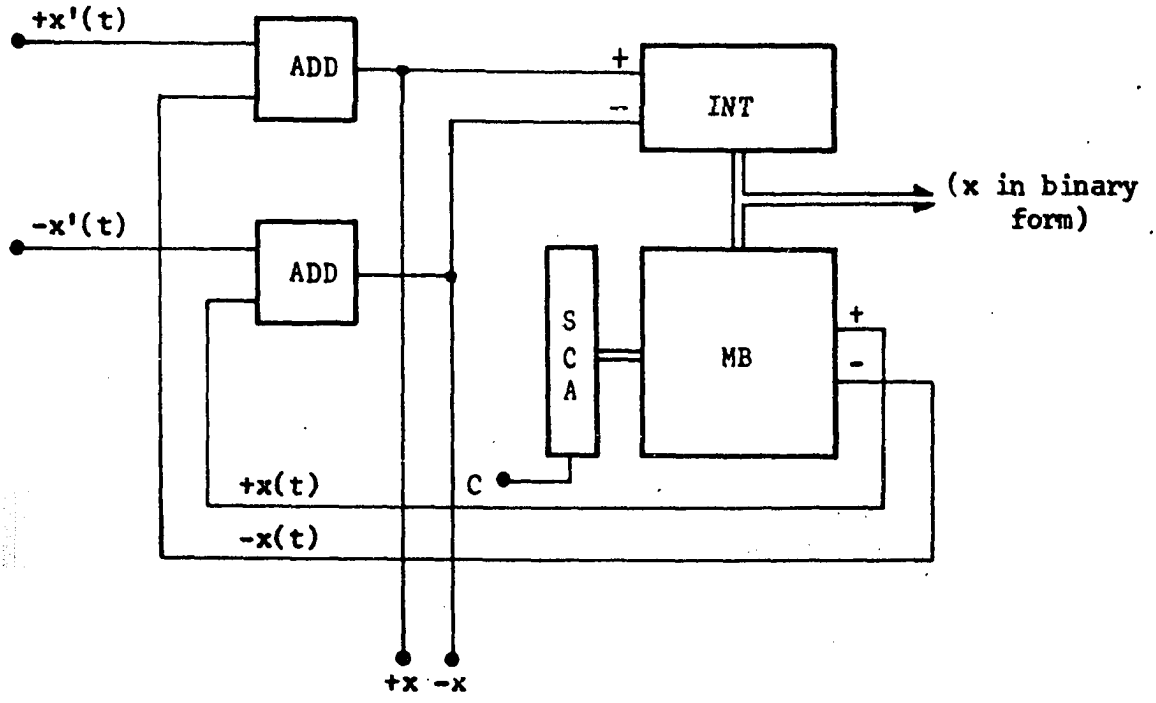


Figure 45. Computer diagram for the differentiation subroutine

in the simulation is quite similiar to the equation used in the previous subroutine. The equation is:

$$x(t) + f_2(t)\, x(t) = f_1(t).$$

The solution of this equation is also of an exponential nature, but if f_1 and f_2 are essentially constant (equivalently if f_1 and f_2 change very slowly with respect to the period of variable representation) then the final value of x is their ratio. The computer diagram for this routine is shown in Figure 46. In this simulation, for f_1 and f_2 essentially constant, the final value of $x(t)$ will be $f_1(t)/f_2(t)$ and the length of time required for x to achieve its final value is determined by how fast f_1 and f_2 are allowed to change. This subroutine does provide a means by which a quotient can be obtained under very strict limitations on the functions to be divided.

The third subroutine considered is not, strictly speaking, a seeking type of routine. In a seeking routine the output or some function of the output is compared with the input to obtain an error signal which is used to change the output. This is similar to the operation of a closed-loop control system in which the input is continually compared to some function of the output and the output is corrected if an error exists. An open-loop system is one in which the output is not compared with the input and aside from the predictability of the system connecting the input to the output there is no assurance that the output obtained possesses the desired relationship to the input. This subroutine is of the open-loop variety however the fact that there is a finite delay between the application of the input signal and the corresponding

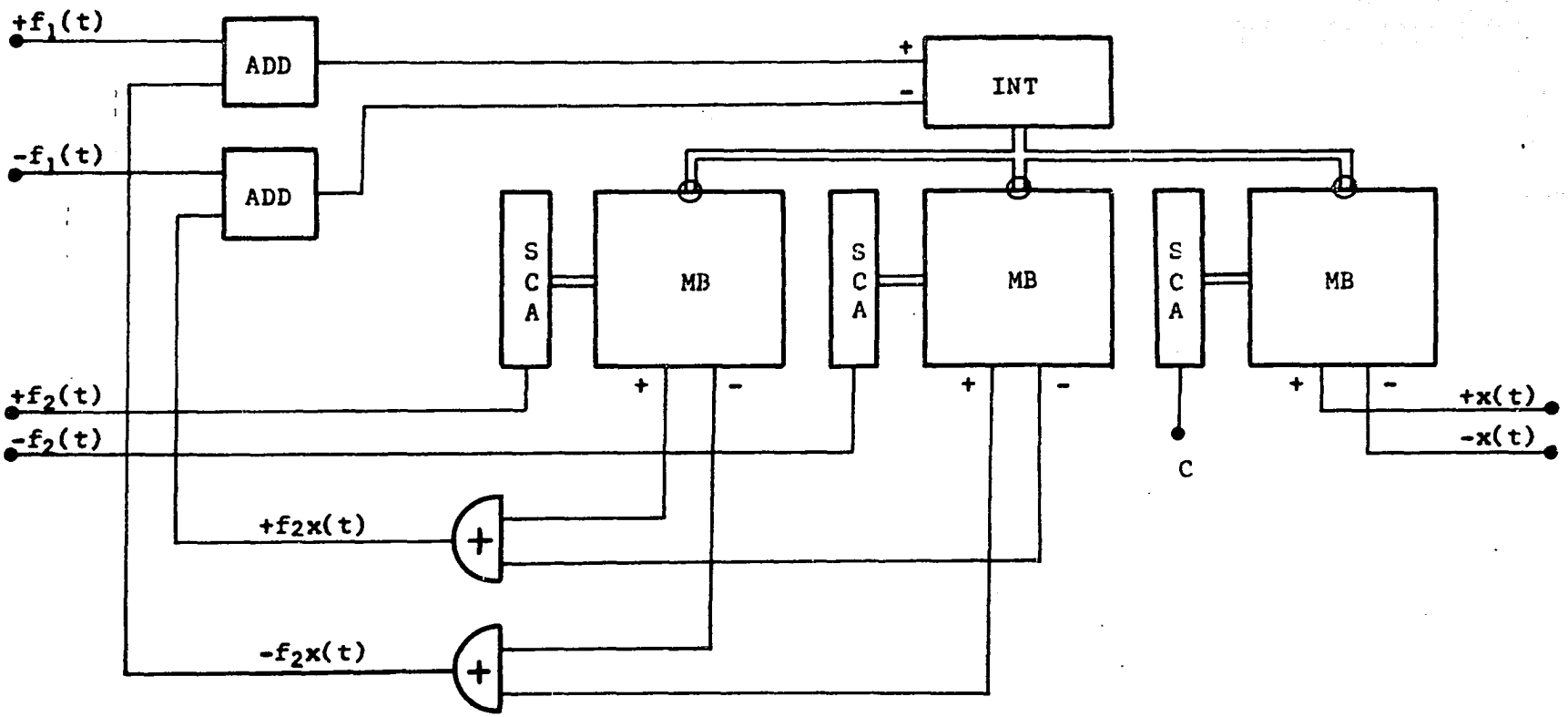


Figure 46. Computer diagram for the division subroutine

output causes it to possess many of the characteristics found in the seeking routines previously considered.

The purpose of this subroutine is to determine the sine and the cosine of a pulse-density variable. The function performed is similiar to that performed by angular resolvers in conventional analog devices. The facilitation of this function within the proposed computer is accomplished in essence by the solution of the differential equation:

$$\frac{d^2x(\theta)}{d\theta^2} + x(\theta) = 0,$$

with initial conditions:

$$x(0) = +1$$

and

$$\frac{dx(0)}{d\theta} = 0.$$

The solution of this equation is:

$$x(\theta) = \text{Cos } \theta$$

thus

$$\frac{dx(\theta)}{d\theta} = \text{Sin } \theta.$$

The implementation of this subroutine is more easily understood by recalling that:

$$\int_0^{\theta_1} \text{Cos } \theta \frac{d\theta}{dt} dt = \text{Sin } \theta_1,$$

and that

$$1 + \int_0^{\theta_1} -\text{Sin } \theta \frac{d\theta}{dt} dt = \text{Cos } \theta_1.$$

These equations are actually the equations used to implement the subroutine. The computer diagram for this routine is shown in Figure 47. This routine

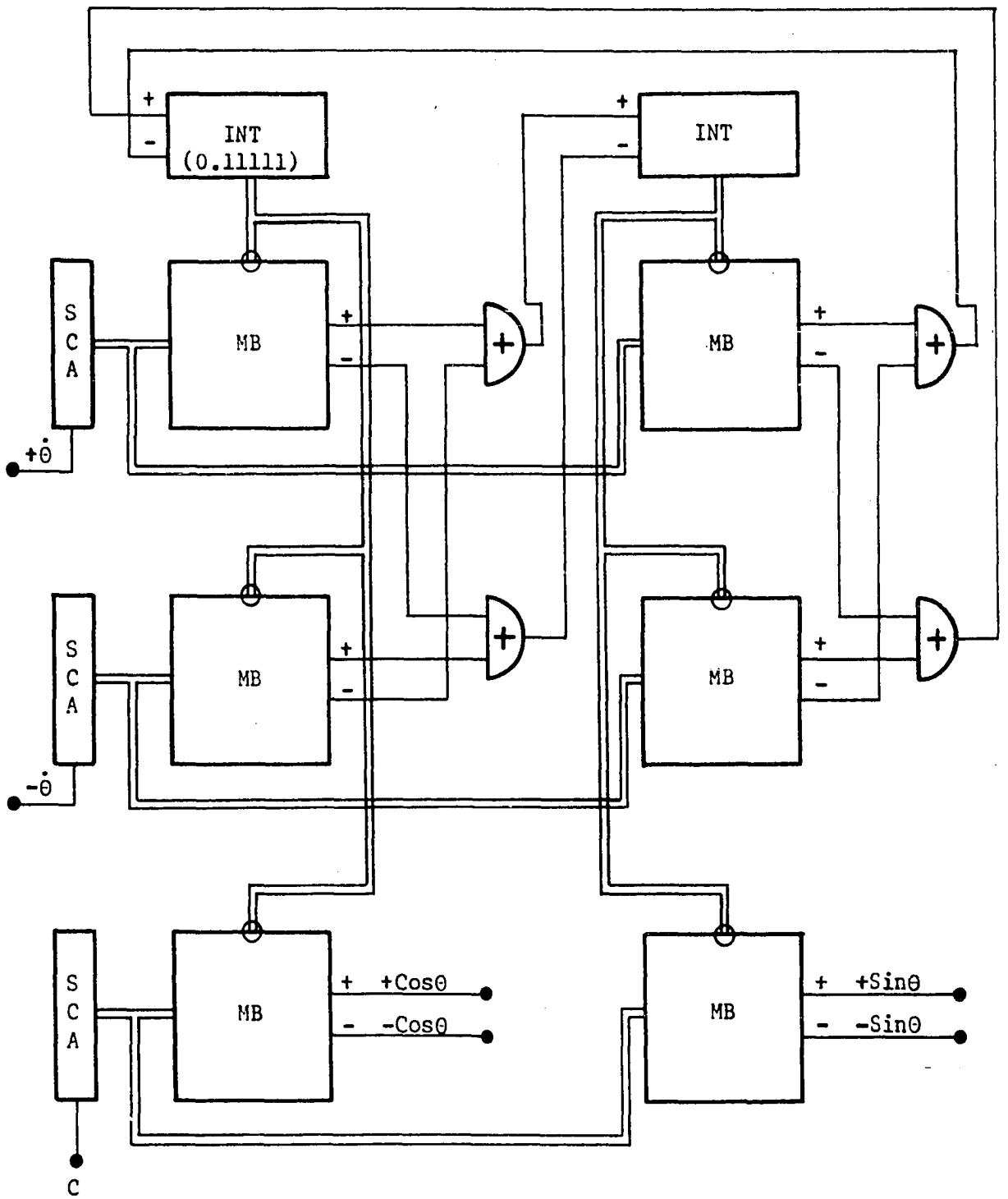


Figure 47. Computer diagram for the sine-cosine subroutine

can be used to compute the sine and cosine of some angle θ_1, assuming that θ is initially zero. Other initial values of θ could be simulated by a different set of initial conditions on the integrators in this subroutine. Note that the determination of sine and cosine of θ need not be a process starting only at the initial value and going to some fixed θ_1. The process can be continuous because if the values of the sine and cosine are correct for some value of θ, and θ changes the sine and cosine outputs of this subroutine will change in the correct relationship. Note that this change cannot be instantaneous and thus if θ changes from one fixed value to another, it will be a finite time later before the outputs of this subroutine attain their correct values. The amount of time required is dependent on how much and how fast θ is allowed to change, or in other words $\dot{\theta}$. Essentially, then the limitation on this routine is the maximum time rate of change of θ (or the amount θ can change during one period of variable representation). The $\dot{\theta}$ pulse-rate variable representation necessary as the input to this routine may be available from the other parts of the simulation in which the subroutine is used, if not the first subroutine proposed could be used to obtain the derivative. It is important to note that in either case, θ is assumed (by the initial conditions imposed in Figure 47) to have an initial value of zero.

In summary, three subroutines for the proposed computer have been described. Many other types of subroutines are possible but the three described can be used to perform several functions that are necessary in the solution of certain types of engineering problems. These subroutines all impose strict limitations on the character of the input

functions involved and thus provide only a partial extension of the types of systems that can be simulated with the proposed computer, however in many cases the solution of problems even under severe limitations possesses a significant advantage over the use of rough approximations to obtain a solution.

C. Interface Schemes

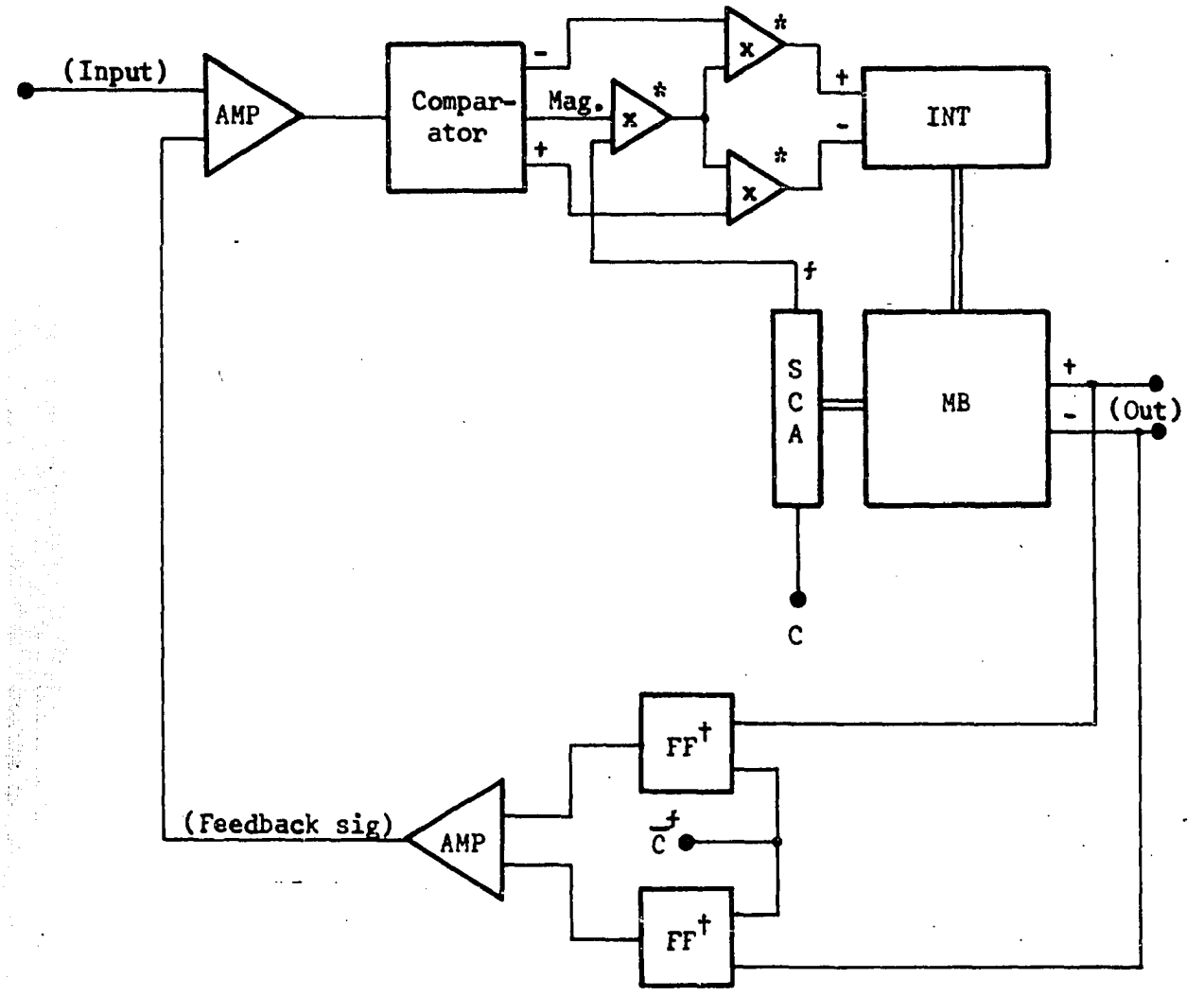
One of the major obstacles involved in the utilization of the proposed computer or any other digital device in the solution of the problems of the physical world is that of interfacing the computer with the problem. The basic problem is one of representing the continuous with the discrete. Regardless of the accuracy attained within the computer, the computer cannot compensate for erratic inconsistencies between the computer language and the problem language occurring at the interfaces between the two systems. One solution to this problem would be to make the external system completely digital. Such an approach would eliminate the interface problems but cannot always be facilitated.

One of the major advantages of pulse-density coding is the ease of analog-digital conversion. Korn (12) states that this type of analog-digital conversion requires no exotic coding equipment because it is simply a matter of comparing the effective value of the pulse train with the value of the analog variable. Conversion from analog values to pulse-density representation is trival compared with the conversion scheme necessary to convert analog information into Grey code binary representation, for example.

There are basicly two different philosophies of analog to pulse-density conversion. The first philosophy involves comparison of analog values with the time average of the pulse-train value. This method will be referred to as "average-value conversion". The other philosophy involves comparison of the analog value with the exact integral of the pulse-density representation over one period of the variable representation. This method will be referred to as "exact-value conversion". Which of these two methods would be used in a particular problem would depend upon a number of factors. Average-value conversion possesses a simplicity advantage, however exact-value conversion is more predictable and is the faster of the two methods.

The average-value method of conversion treats the pulse-density variable as a d.c. level. Since the d.c. level of a pulse train (assuming equal width pulses) is the time average of the pulses, as the number of pulses is increased, the d.c. level will increase proportionately. Two types of interfaces are of concern. The first of these involves converting an analog level (assumed to be represented by or representable as a voltage) into an equivalent pulse-rate representation. This conversion is called encoding. The other type of interface occurrs at the output of the computer and involves the representation of the pulse-rate variables as analog levels (voltages). This conversion is called decoding.

The average-value encoding scheme is a closed-loop scheme that consists of comparing the analog input with the analog equivalent of the pulse-rate output and correcting the pulse-rate output until the two are the same. Figure 48 shows the block diagram of the average-value



*Electronic AND gates
†Electronic flip-flops
ƒSee text

Figure 48. Block diagram for the average-value encoder

encoder. The analog input and the feedback signal obtained from decoding the output pulse train form the inputs to a conventional operational amplifier. An operational amplifier is an extremely stable voltage amplifier with predictable negative gain. It is believed that the previously mentioned 20 kc magnetic amplifier developed by Collins Radio could be used for this and all other operational amplifiers used in the conversion schemes. The output of the amplifier is fed into a comparator that determines the sign of this difference voltage and its magnitude. The comparator outputs are used to drive electronic logic gates which in turn are used to change the binary number within the integrator which will subsequently modify the pulse-rate of the matrix buffer output.

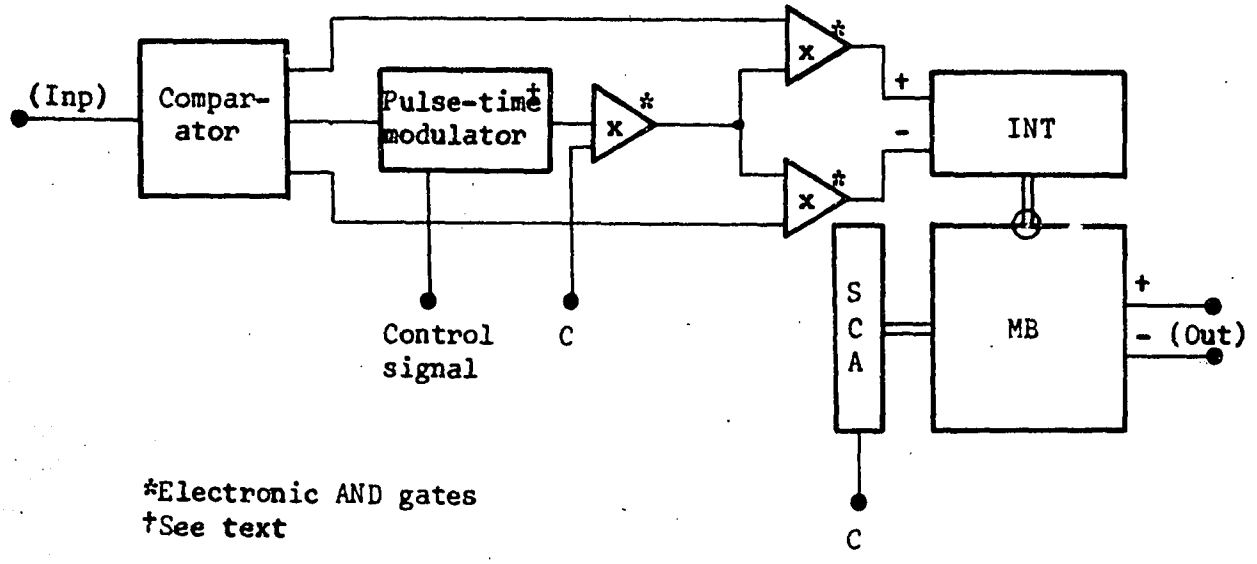
The maximum rate at which the binary number in the integrator can be meaningfully changed is a function of the electronic components involved however from conceptual considerations it would seem that it would not be feasible to try to change this number more than one unit per period of representation, because only by the consideration of a pulse-density variable over one period of representation can its value be accurately determined. Note that the output of the last flip-flop of the scaler produces one pulse per period of representation, i.e. the length of time between consecutive pulses is equal to this period. In this encoder then, the output of the last stage of the scaler provides the input to the integrator. When the pulse rate of the buffer output is approximately equal to the correct representation for the analog input, this rate would certainly represent the maximum rate at which the binary number could be changed even if the electronic circuitry were

impeccable, however if the binary number is initially zero, and the analog input represents the maximum positive value then since 32 pulses are required to produce the desired binary number for a five bit system, with a clock rate of 20 kc, it would take about one half a second for the pulse rate to achieve its desired value. As the number of significant bits is increased this time also increases, thus this scheme would probably not be satisfactory for many applications. This problem could be eliminated if some type of variable pulse-rate integrator input were used. Thus when the difference between the analog value and pulse-rate equivalent is large the pulse-rate into the integrator will be almost equal to C and as this difference becomes smaller the rate into the integrator will be decreased proportionately. This type of scheme would improve the efficiency of the encoder, however at a sacrifice in system simplicity.

The feedback signal used in this encoder (the signal that is compared with the analog input) is formed by the same device that could be used for a decoder for the average-value method. This device uses two conventional electronic flip-flops that produce output pulses of half the duration of the clock frequency's period. One is arranged so that the signal pulses turn the output on creating e.g. positive output pulses and the other is arranged so that the signal pulses turn the device off creating opposite polarity pulses. Using this arrangement, the outputs of the two flip-flops can be added algebraically with an operational amplifier to obtain an analog signal of the appropriate magnitude and sign to properly represent the pulse-rate input to the device. The required output of the flip-flops is obtained by reseting the flip-flops

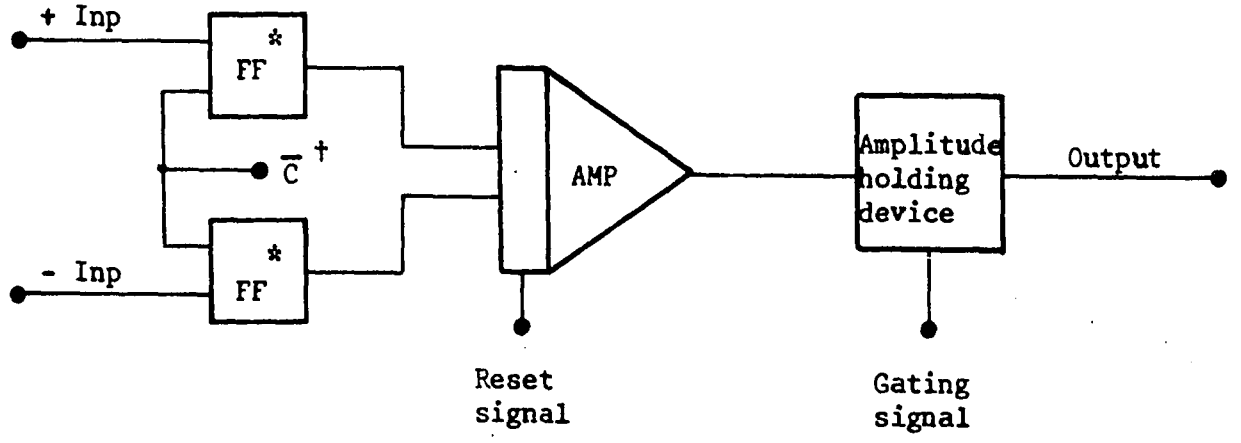
with $\overline{C}$. This creates a maximum average value of the flip-flop outputs equal to half of the maximum flip-flop output. The ratio between these two quantities could be made to be one if C instead of $\overline{C}$ were used for the reset function. Unbalancing the flip-flop circuits so that in case of coincident inputs each flip-flop would be left in its appropriate state. The major philosopical disadvantage of the average-value method of decoding is the question of how soon the output can respond to a slight change in the pulse-density value. As before this time lag is a function of the electronic components involved.

The exact-value method of conversion used controlled integration over one period of representation in the decoding sequence. The encoding process is conceptually the same as that used in the average-value method except that a more accurate conversion scheme is employed which eliminates the necessity for closed loop encoding. Figure 49 shows a block diagram of the exact-value encoder. Most of the components in the encoder are identical to those used in the average-value method encoder. The major difference between the two is the use of a pulse-time modulator between the comparator and the integrator-input controlling gate. The purpose of the pulse-time modulator is to generate, on command, a pulse whose duration is proportional to the magnitude of the analog voltage input from the comparator. Such a pulse-time modulator is described in Seegniller (16) using a magnetic circuitry. Note that if the binary number is initially zero it's maximum value can be attained in 1.6 ms for a five-bit system using a 20 kc clock, and thus this method of encoding is much faster than the previously described method. Note that this same method of encoding could be used in the average-value approach



*Electronic AND gates
†See text

Figure 49. Block diagram for the exact-value encoder



*Electronic flip-flops
†See text

Figure 50. Block diagram for the exact-value decoder

because the basic difference between the two methods stems from the method of forming an analog signal from a pulse-train representation. Figure 50 shows the block diagram for an exact-value decoder. This decoder uses the same flip-flop generation technique as the average-value decoder but the outputs of the flip-flops are now integrated instead of added. This integration is accomplished by using an operational amplifier arranged for integrating, a magnetic amplifier could also be used in this application. This amplifier must be capable of being reset (zeroed) in a very short time and before each subsequent integration it will be reset but just before reseting it, its output is transfered to an analog holding device that will hold this value until the end of the next period of integration when the value will again be transferred to the analog holding device. The analog output will be composed of discrete analog values, maintained over a period of representation. This represents the most accurate analog representation possible for the pulse-density variables within the computer. The same technique discussed for the average-value encoder to increase the efficiency of the flip-flop operation could also be used with this scheme.

The exact-value method of conversion could be operated in closed loop by the same type of closed loop scheme as shown for the average-value method, if closed loop operation were necessary. The major advantage of the exact-value method of encoding is that it can be operated open-loop which would probably be manditory if the device were to be time shared i.e. used to encode several variables by periodically encoding each of them.

It should be noted that although consideration of a pulse-density variable over a complete period of its representation is necessary to determine the variable's exact value, consideration of such a variable over a shorter period will yield approximately the same results. The maximum possible error involved in such an approach is an inverse function of the period of consideration of the variable, however for function's whose pulse rate is reasonably constant, the value obtained from consideration of less than a complete period of representation will not differ substantially from the actual value of the function, as long as the period of consideration is not excessively short.

VII. DISCUSSION

The purpose of this thesis was to investigate the possibility of designing a pulse-density computer using magnetic devices, primarily transfluxers. The proposed computer was to be synchronous (clock controlled) and be capable of the solution of certain types of differential equations. The proposed computer is composed of various combinations and configurations of the six basic operational components. These components are used for the most part in the formation of the three major functional units, however, in some applications some of the basic components find application external to the functional units.

The basic operational components have all been built and tested and have been found to perform their respective functions in a reliable manner. One of the major hardware problems involved is the sense amplifier used for the transfluxer. This amplifier could be built with one transistor instead of two if transistors with a gain higher than that of the 2N697 were used. The development of this computer used 2N697's because of their relative low cost ($1.20) and their availability. The major functional units which are composed primarily of combinations of the operational components have all been at least partially constructed and found to perform their required function reliably. Time and available hardware limited the number of major functional units that could be constructed. This necessitates the verification of the feasibility of the computer be based on theoretical considerations, physical intuition and the results of previous experience with pulse-density computational schemes.

The major advantages of the proposed computer over conventional voltage analog devices include accuracy, stability, relibility, cost and bulk. The only way to prove these advantages would be to build a prototype of a particular flight control system and compare it with the voltage device in current use. The proposed five-bit computer using a 20 kc clock rate would probably not possess sufficient accuracy for most flight control applications, however increasing the number of bits within the computer is simply a matter of adding components. This would increase the accuracy within the computer but using the proposed interface schemes could create a problem of time delay at the interfaces. This problem could be circumvented by using digital sensers, e.g. digital shaft encoders, which could obtain the input information directly in digital form.

There are several problems that still exist within the proposed computer. One of these is the finite delay (one or two units) that must be built into some of the major functional units. These delays were necessary to allow synchronous operation of the computer and are inescapable to some degree in any magnetic circuitry application. Since the actual time delay involved is quite small these delays are not believed to pose any serious limitation on the computer's ability to solve the type of problems for which it is intended. Another aspect of the proposed computer that might warrent improvement is the method used to put initial conditions on the integrators. Since the initial conditions must be "wired" into the integrator and there is a limited number of windings that can be wound on a transfluxer at any one time, the number of different initial conditions that can be wound into a particular

integrator is limited (using the present scheme) to perhaps three or four. There are two possible ways that this problem can be circumvented. The first is to wind an independent zero reset and one reset on each flip-flop in the integrator and electronically control which of the flip-flops are initially in which state. This would allow all possible initial conditions to be set, however it would involve quite a bit of additional electronic circuitry. The other method of setting any set of initial conditions is what might be called the "dynamic" approach, which consists of zeroing all of the integrator flip-flops and before the computation process starts (time zero) input into each integrator the appropriate number of pulses to yield a final binary value of the desired initial condition. For a five-bit system with a 20kc clock this process would take a maximum of 1.6 ms which would probably be satisfactory, however this scheme would mean that some additional components would have to be added, but for a system requiring any arbitrary set of initial conditions, this would probably provide the most feasible solution. Another improvement that could be made within the computer would be to extend the adder unit so that more than two pulse trains could be added with the same adder. The logic necessary in such an extension soon becomes inordinate in the general case, but in specific problems such an extension might prove to be practical.

The proposed computer uses conceptually the same type of functional units as does a conventional analog computer and is used to simulate differential equations using the same techniques. The proposed computer would possess an accuracy comparable with a digital computer with the same word length but can integrate at a higher speed because integration

is built into the proposed computer. Thus it is believed that the proposed computer combines the advantages of analog computers, that of operational simplicity or logical correlation with the system being simulated, with the speed and accuracy obtainable with a digital device, and is believed to be the optimum combination of the two approaches to problem solution for the type of applications for which the computer was conceived.

VIII. SELECTED REFERENCES

1. Aron, N. and Granger, C. Analog-to-digital converter using transfluxers. Electronics 35, No. 20: 62-66. 1962.

2. Brain, A. E. The simulation of neutral elements by electrical networks based on multi-aperture magnetic cores. Institute of Radio Engineers Proceedings 49: 49-52. 1961.

3. Burkstein, E. Digital counters and computers. New York, N. Y., Rinehart and Co. 1960.

4. Cushman, N. and Park, D. Reversal of a loaded ferromagnetic core. Institute of Radio Engineers group on Component Parts Transactions CP-7, No. 4: 117-124. 1960.

5. Hershberg, P. I. Ferromagnetic domains. Electrical Technology 39: 71-82. 1962.

6. Feigleson, C. O. Experimental digital doppler computer using pulse density methods. Unpublished project summary report. Cedar Rapids, Iowa, Collins Radio Co. 1962.

7. Gianola, U. F. Design problems in all-magnetic logic. Electronics 34, No. 18: 61-65. 1961.

8. Grimm, D. K. The low-capacity magnetic memory. Unpublished project interim report. Cedar Rapids, Iowa, Collins Radio Co. 1962.

9. Gschwind, H. W. Digital differential analyzers. In Von Handel, P., ed. Electronic computers. pp. 139-210. Englewood Cliffs, N. J., Prentice-Hall, Inc. 1961.

10. Johnson, C. L. Analog computer techniques. New York, N. Y., McGraw-Hill Book Co. 1962.

11. Karnaugh, M. Pulse-switching circuits using magnetic cores. Institute of Radio Engineers Proceedings 43: 570-583. 1955.

12. Korn, G. A. The impact of hybrid analog-digital techniques on the analog computer art. Institute of Radio Engineers Proceedings 50: 1077-1086. 1962.

13. Meier, D. E. Design of a 20 kc type W magnetic amplifier. Unpublished technical report. Cedar Rapids, Iowa, Collins Radio Co. 1962.

14. Rubenfield, N. Computation with pulse analogs. Institute of Radio Engineers National Convention Record 1957, Part 4: 150-155. 1957.

15. Scott, N. R. Analog and digital computer technology. New York, N. Y., McGraw-Hill Book Co. 1960.

16. Seegniller, W. R. Accurate analog computation with pulse-time modulation. Electronics 35, No. 13: 54-57. 1962.

17. Soroka, W. W. Analog methods in computation and simulation. New York, N. Y., McGraw-Hill Book Co. 1954.

18. Woodfill, M. C. An analog computer using digital techniques. Unpublished M. S. thesis. Ames, Iowa, Library, Iowa State University of Science and Technology. 1961.

IX. ACKNOWLEDGEMENTS

The author acknowledges with gratitude the encouragement and assistance of his major professor, Dr. J. E. Lagerstrom. The author also wishes to extend his thanks to Collins Radio Company who partially financed this project and especially to D. C. Sather, group leader at Collins Radio's Cedar Rapids Division, for his invaliable assistance and advice during the initial phases of this project.