NOTICE: When government or other drawings, specifications or other data are used for any purpose other than in connection with a definitely related government procurement operation, the U. S. Government thereby incurs no responsibility, nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.

INSTITUTE OF TECHNOLOGY

AIR UNIVERSITY

UNITED STATES AIR FORCE

SCHOOL OF ENGINEERING

THESIS

WRIGHT-PATTERSON AIR FORCE BASE, OHIO

NEURISTOR CIRCUITS TO ACCOMPLISH SOME DIGITAL COMPUTER OPERATIONS

THESIS

Presented to the Faculty of the School of Engineering of
the Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

By

William Carroll Ross, B.S. Mil. E.
Capt USAF

Graduate Astronautics

August 1961

Preface

The purpose of this individual study is to demonstrate the feasibility of using neuristor logic circuits in binary coded digital computers.

This study was suggested by Professor J. Lubelfeld as an extension of the work begun by Mr. H. D. Crane, (Ref. 1.). In addition to Professor J. Lubefeld, I wish to thank Captain F. M. Brown and Lieutenant M. Kabrisky for the encouragement and help that they gave to me in preparing this study.

At this time, a practical neuristor circuit is not available; hence, the circuits shown may not be optimum. More information as to neuristor costs, weight, size, speed of propagation and refractory time must be available in order to construct the best circuits for a specific purpose. However, this report should demonstrate the ease of accomplishing digital computer operations with neuristor circuits.

William C. Ross

Contents

List of Figures

List of Figures

Abstract

Digital computer circuits are constructed using neuristors. Using H. D. Crane's definition, a neuristor is a one-dimensional channel along which pulses may propagate, the pulses taking the form of propagating discharges having the following properties:

1. Threshold stimulability
2. Attenuationless propagation
3. Uniform velocity of propagation
4. Refractive period following the passage of a discharge past any point of a channel.

Two neuristor pulse sequences representing binary numbers are fed simultaneously into a neuristor add circuit; a pulse sequence emerges which represents the sum of the two pulse sequences. Similar circuits are developed for subtraction, multiplication, division, and other elementary operations.

NEURISTOR CIRCUITS TO ACCOMPLISH SOME DIGITAL COMPUTER OPERATIONS

I. Introduction

"We now formally define a neuristor as a device having the form of a one-dimensional channel along which signals may propagate, the signals taking the form of propagating discharges having the following properties:

1. Threshold stimulability
2. Attenuationless propagation
3. Uniform velocity of propagation
4. Refractive period following the passage of a discharge past any point of a channel. (during this-period, that portion of the channel cannot propagate a second discharge)

For the reader who is unfamiliar with neuron properties, a suitable thinking-piece is the propagation of a chemical burning-discharge along a fuse, except that neuristors exhibit perfect "healing", and may consequently be used an arbitrary number of times. The refractory period may be thought of as the time of healing." (Ref 1:5)

The human nerves also operate as neuristor lines. After a

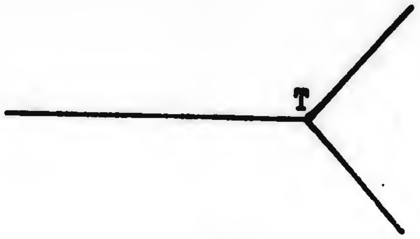


Figure 1
T Circuit

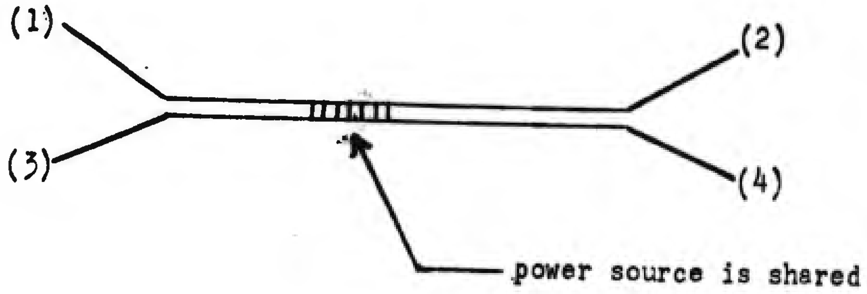


Figure 2
S Circuit

pulse is transmitted by a nerve, a certain refractory time is required for the body to build back up the power required to transmit another pulse.

Although there are examples of neuristor lines in nature, a practical neuristor line for use in digital computer circuits is not as yet available. If such a line is near to the animal nerves in weight, power requirements, and efficiency, this line would be well suited for construction of airborne computers where weight and power requirements are at a premium.

In the next paragraph, Crane's T and S junctions will be explained; after that some simple conventions used in the figures will be stated; then, some digital operations which are easily accomplished by neuristor lines will be covered. These digital operations are extracting a subsequence from a sequence of pulses in a neuristors line, and sorting, adding, subtracting, multiplying, or dividing binary coded sequences. The design of a complete computor is not attempted in this study, but rather a number of subcircuits are presented.

The T junction is shown in Figure 1. Any pulse started at any of the three ends will proceed to the junction and out both of the other ends.

The S (share) junction is shown in Figure 2. A pulse applied at end (1) will proceed out end (2). The series of vertical lines

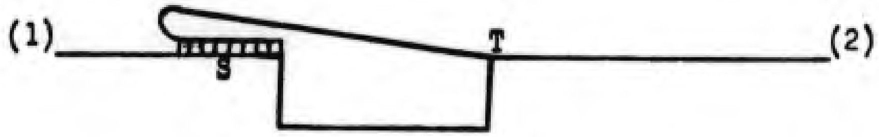


Figure 3
T-S One Way Line Circuit

indicate that the power is shared over that interval. It is not possible for a pulse in one line to get into the other line. When a pulse entering (1) and another pulse entering (4) meet in the share portion, both pulses will disappear as each will have depleted the power in the line ahead of the other. Simultaneous pulse entry at (1) and (4) insures a meeting in the share portion of the circuit.

Some special arrangements of S and T junctions are helpful. Consider the construction of a line which only allows passage of a pulse in only one direction. (See Figure 3.) Here a pulse proceeds from (1) to (2) through the S and T junctions. The pulse that goes back through the T junction does not meet a pulse in the S junction. In a sequence of pulses, the interval between pulses could be adjusted so that this returning pulse would encounter no pulses in the S junction.

However, a pulse proceeding from (2) to (1) would branch at the T junction. The two branches would meet in the S junction, thus inhibiting propagation of the pulse to (1).

A few conventions will be used in all circuits to follow. They are adapted to make the circuits simpler and easier to follow.

Pulses will travel in these circuits only from left to right unless an arrow indicates travel from right to left. An arrow will be used to indicate the direction of travel on circular paths and in any other cases of possible ambiguity. A S-T junction of the type

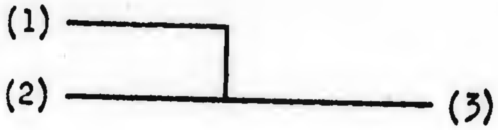


Figure 4
One Way Circuit Illustration

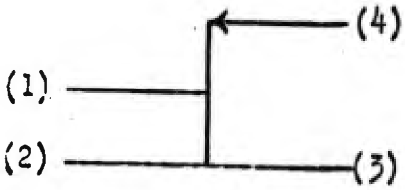


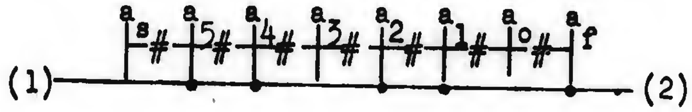
Figure 5
One Way Circuit Illustration

just mentioned can be used to insure one way travel.

Example (a). (See Figure 4.) Pulses entering at (1) or (2) will depart at (3). No pulse can enter at (3).

Example (b). (See Figure 5.) Pulses entering (1), (2), or (4) will depart at (3).

Capital letters A, B, and C, will be used to designate binary coded pulse sequences. To describe a binary coded pulse sequence consider a sequence of pulses moving from (1) toward (2) in the neuristor line below:



There is a uniform time interval labeled (#) between information positions. The time interval is greater than the refractory time of the line. Sequences of this type will be used to represent signed binary numbers.

a_f is the first information position. A pulse is always in this position.

Positions a_o through a_5 are used to represent a binary number. This number equals:

$$a_5 2^5 + a_4 2^4 + a_3 2^3 + a_2 2^2 + a_1 2^1 + a_o 2^o,$$

where a_i is 1 if a pulse is in the $i^{\underline{th}}$ position and a_i is o if no pulse is in the $i^{\underline{th}}$ position.

a_s is the sign position for the number, when there is a pulse

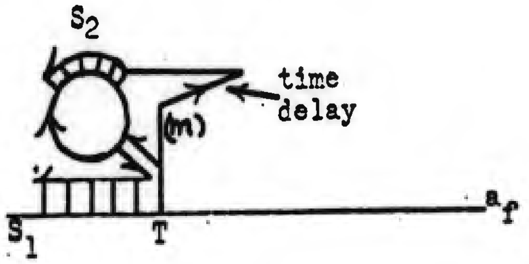


Figure 6
Circuit (x), A First Pulse Extractor

in this position the number is negative, otherwise the number is positive.

At any time, the statement "a_j is 1" will imply a pulse is in the $j^{\underline{th}}$ information position; the statement "a_j is o" will imply no pulse is in the $j^{\underline{th}}$ information position.

Generally a much longer sequence of pulses will be needed to represent a large number. Other information than one number may be present in the pulse sequence. Thus:

$$A = a_m \cdots a_n a_s a_\ell \cdots a_1 a_o a_f \longrightarrow$$

will mean that A is a binary coded pulse sequence containing:

a_f, the pulse indicating the beginning of the sequence. a_ℓ through a_o, a binary number $\ell+1$ bits in length where a_ℓ is the last information position in the number. a_s is the sign position of the number.

$a_m \ldots a_n$ are other information positions in the sequence. They may represent other binary coded numbers or other information. The arrow is a reminder of the direction of pulse propagation.

All neuristor branches in any of the following circuits will have the same properties unless specifically stated otherwise.

A few simple circuits for later use will now be shown. Circuit (x), (Figure 6.) allows only the a_f pulse to proceed out. The a_f which is always a "1", turns back to inhibit passage of succeeding pulses; in circulating around in the circular path a_f

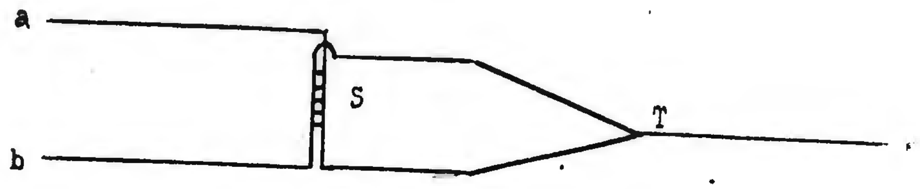


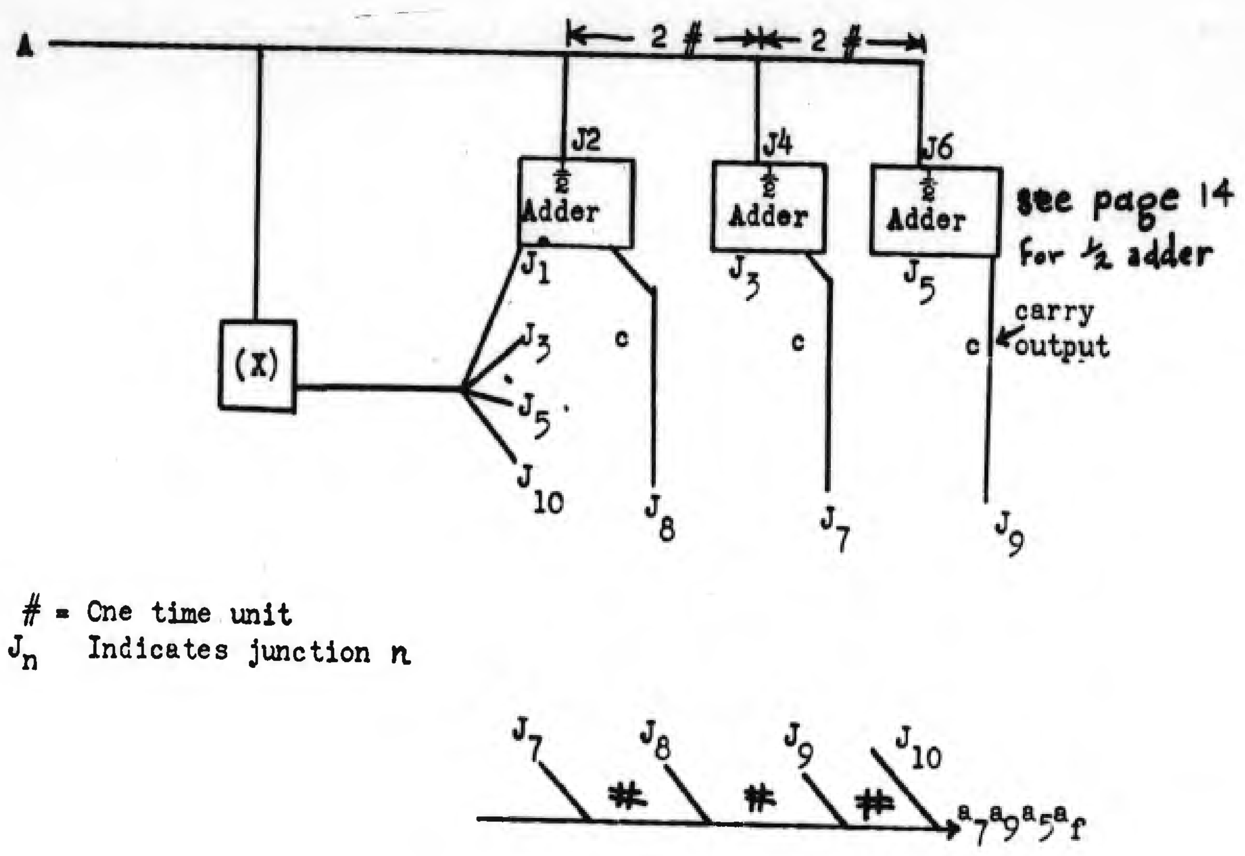
Figure 7
Unequal Circuit

furnishes a sequence of steady pulses to inhibit this passage. Such a circular path will be referred to as an inhibitor (symbol I) in the future. To stop these inhibiting pulses for a new sequence, the (m) output of the T junction inhibits the bit in the ring at S_2 after all of A has reached junction S_1. An N pulse generator (see Appendix A) actuated by the a_f bit could have been used to generate pulses which could inhibit the passage of the remainder of A through share junction S_1.

An even simpler device could serve as a circuit (x) if reliable neuristors with variable refractory times were available. A neuristor, with a refractory time greater than the propagation time of a sequence past a given point, will allow only the first pulse of the sequence to propagate past the point. As circuit (x) serves to allow only the first pulse to propagate, insertion into a line of a neuristor with the correct refractory time will have the effect of inserting a circuit (x) into the line.

Another simple circuit is the unequal circuit (Figure 7.). When the inputs a and b are not the same, one of them will be a pulse and proceed out the output. When a and b are both "1", they are stopped in the S junction. When they are both "0" nothing can proceed to the output. This may also serve as the sum output of a half adder.

Numerous other elementary circuits are listed in Crane's report. See Appendix A for his N pulse generator.



# = One time unit

J_n Indicates junction n

Figure 8
Extractor I Circuit

II. Extractors

Extractors are needed to obtain portions of an information train for future operations. Three types will be discussed here. They are:

Extractor I. -- One input and any desired fixed output.

Extractor II.-- Two inputs, the first contains the information which is extracted, the other input causes a consecutive subsequence of the first to be extracted.

Extractor III.- Two inputs, one contains the information and the other contains instructions needed to obtain the desired compressed sequence from the first input as output.

A description of the construction of these extractor circuits and a more detailed functional description follows:

Extractor I

Let sequence A represent the bit train input. Let us suppose that we wish to extract certain bits so that the output is $a_7a_9a_5a_f \longrightarrow$. The circuit in Figure 8., will do this. In Figure 8., the lengths of the lines from circuit (x) to the halfadders is adjusted so that the inputs arrived when a_9, a_7, and a_5 are at input positions J_2, J_4, and J_6 respectively. For example, if it took a_9 ten time units to get from J_0 to J_2, then the time length of a_f from J_0 to J_1, J_3, and J_5, would be ten plus the ten

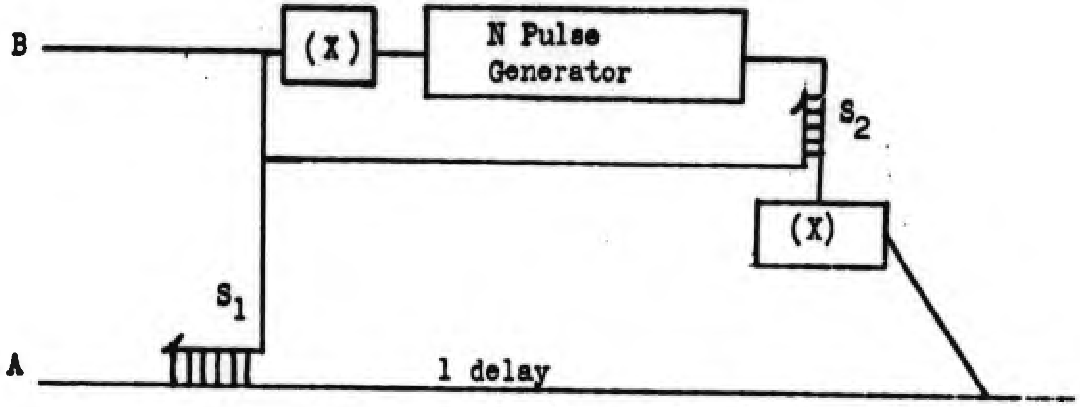
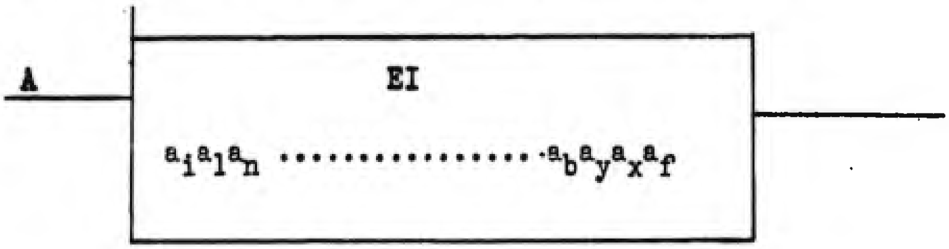


Figure 9
Extractor II. Circuit

time units a_f is ahead of a_9. (i.e., $a_9a_8a_7a_6a_5a_4a_3a_2a_1a_oa_f$ has ten intervals between a's.) Now, the output of all half adders will occur simultaneously. Combine the carry outputs as indicated in the diagram with a properly delayed output from circuit (x) (to give an a_f), and the required output results. The output is: $a_7a_9a_5a_f \longrightarrow$

Using the same principle, an Extractor I that gives any desired fixed output sequence may be easily realized. Even an output such as: $a_1a_9a_1a_7a_7a_{11}a_f \longrightarrow$ should cause no difficulty.

For use in block diagrams, Extractor I will be shown as:



where A is the input sequence and the subscripts of "a" give the fixed output sequence.

<u>Extractor II</u>

Extractor II extracts a_n through a_n+m for use in later operations and also puts out an a_f to head this sequence so that the output is:

$$a_{m+n} \cdots\cdots a_{n+2}\ a_{n+1}\ a_na_f \longrightarrow$$

The positions n and m are variable and are given in the second input.

See Figure 9., to follow the circuit description that follows.

Bit sequence B is generated by a programmed instruction; it enters the upper input. This bit sequence consists of:

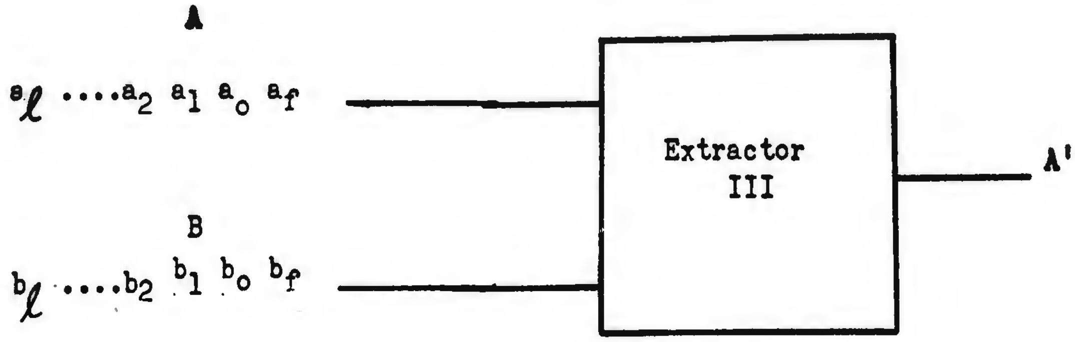

A
$a_\ell \ldots a_2\ a_1\ a_0\ a_f$
B
$b_\ell \ldots b_2\ b_1\ b_0\ b_f$
Extractor III
A'

Figure 10a
Block Diagram of Extractor III Circuit

Information Position	b_ℓ	$b_{m+n}\ b_n$	$b_{n-1}\ b_o b_f$
Condition of line in the position	1...11	00...0	1....1 1
	all 1's	all 0's	all 1's

The times of propagation of the sequences are adjusted so that b_1 enters the share junction, S_1, as a_i enters S_1. Thus only $a_{m+n}\ldots a_{n+1}a_n$ emerges from S_1 for only these positions are not inhibited by the corresponding pulses of B.

The remainder of the circuitry provides a pulse for the a_f position of the output. An (x) circuit provides a pulse for the N Pulse Generator. The sequence of pulses provided by the generator meets B in S_2. The first pulse will get through S_2 when b_n enters S_2, for b_n is the first position with no pulse in the B sequence. Another (x) circuit allows only the first pulse to get back to the output. The $a_{m+n}\ldots a_{n+1}a_n$ output is delayed so that this first pulse is one time unit ahead of a_n. Thus the output is $a_{m+n}\ldots\ldots a_n a_f$.

Extractor III

The positions in input sequence A are extracted when the corresponding position in input sequence B is occupied by a pulse. The output is compressed so that the output sequence contains only as many time bits as there are pulses in B. With A' as output,

$$\begin{array}{lll} A \text{ is} & \overset{a_\ell}{1}1\ldots0101\overset{a_o}{0}\overset{a_f}{1} & \\ & & \text{implies } A' \text{ is } \overset{a_{\ell-1}}{1}\ldots011\overset{a_o}{0}\overset{a_f}{1} \\ B \text{ is} & 01\ldots1100111 & \end{array}$$

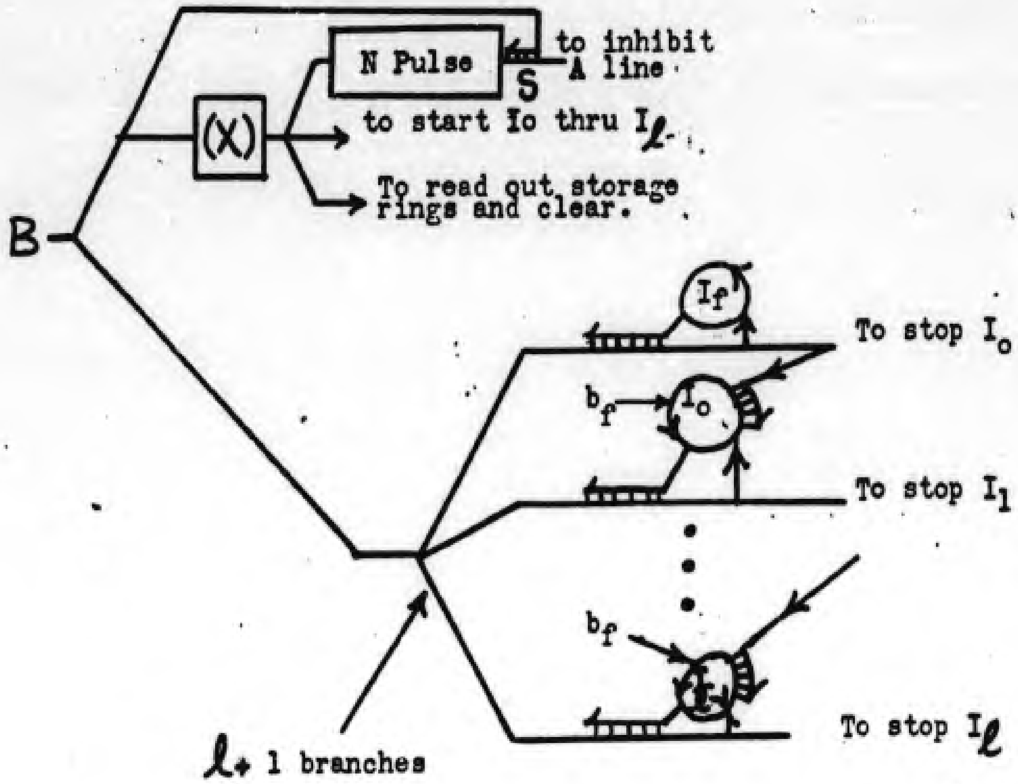


Figure 10b
Output control of Extractor III Circuit

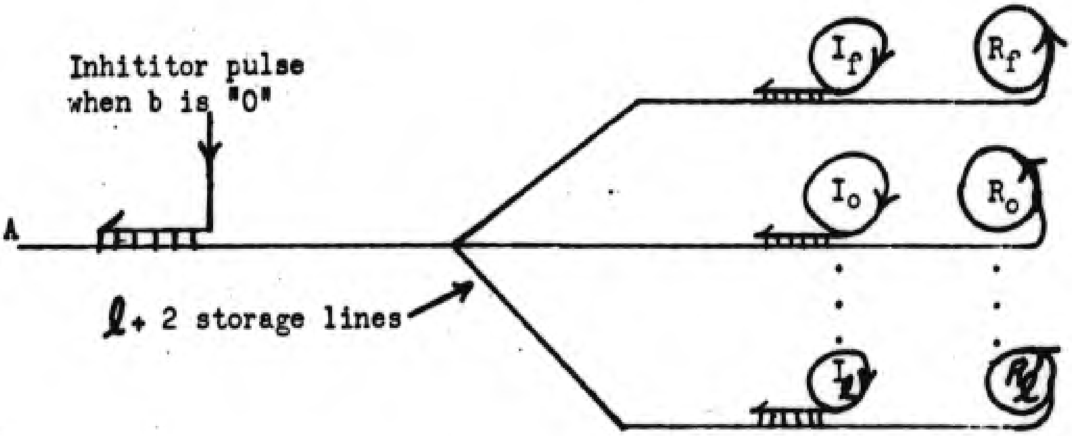


Figure 10c
Storage Part of Extractor III

First let us examine what happens to the B sequence as it enters Extractor III. (See Figure 10b.)

Sequence B enters and splits into three paths.

Circuit (x) sends one b_f pulse to start all inhibitors except I_f and to later cause storage ring output. Circuit (x) triggers a continuous pulse sequence to inhibit sequence A.

The upper path is set up so that the continuous inhibition of sequence A is interrupted when there is a pulse in the B sequence. Thus, all pulses of the A sequence that passed must be 0 where the corresponding position in B is 0.

Now consider what happens to A sequence. (See Figure 10c.)

The inhibitors shown are the same ones which are actuated by B sequence (the connections to the inhibitors via the B sequence are shown in Figure 10b).

A and B pulses enter Extractor III simultaneously. The "1"s in the B sequence have a cascading effect in that they close the open ring and open the next higher ring.. (See Figure 10b.) There is no problem of storage in a ring when b_i is "0" as the corresponding a_i is "0" due to action of the inhibitor pulse.

To illustrate storage of the correct positions in A in storage rings, consider the following example:

A	is	101101
B	is	010101

a_f proceeds to all storage rings but is blocked by inhibitors

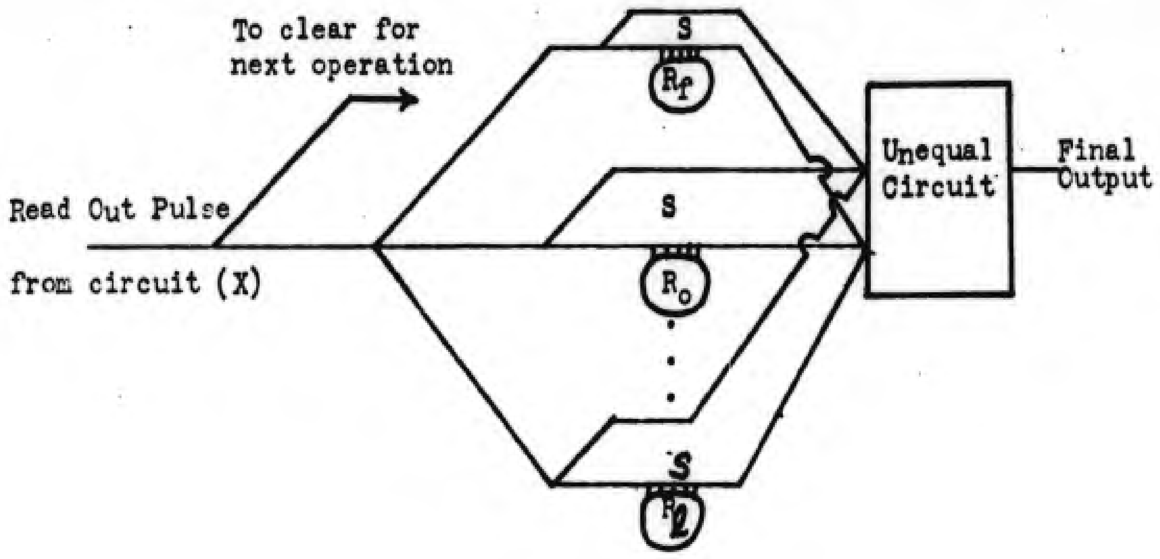


Figure 10d
Read Out Part Extractor III

I_o thru I_ℓ. Hence a_f is stored in R_f only.

b_f starts I_f and stops I_o after a_f has passed through.

a_o is zero, hence nothing is stored in R_o at this time. I_o remains stopped as b_o is "0".

a_1 is stored in R_o. b_1 starts I_o and stops I_1.

a_2 passage is inhibited as b_2 is "0". R_1 remains the open ring.

a_3 (a"0") is stored in R_1; b_3 starts I_1 and stops I_2, thereby making R_2 the open ring.

a_4 passage is inhibited as b_4 is "0". R_2 remains the open ring.

Thus, we have a_f in R_f, a_1 in R_o and a_3 in R_1. The final output results when a single pulse from circuit (x) reaches the storage rings. (See Figure 10d.) This output producing pulse reaches the storage rings at lease $\ell+2$ time intervals after a_f.

The length of the lines are adjusted so that the outputs from paths passing by R_f, R_o, R_1, etc., will reach the output line one time interval between each.

An a_f pulse branches to clear the entire circuit inhibiting any pulses remaining after the final output emerges.

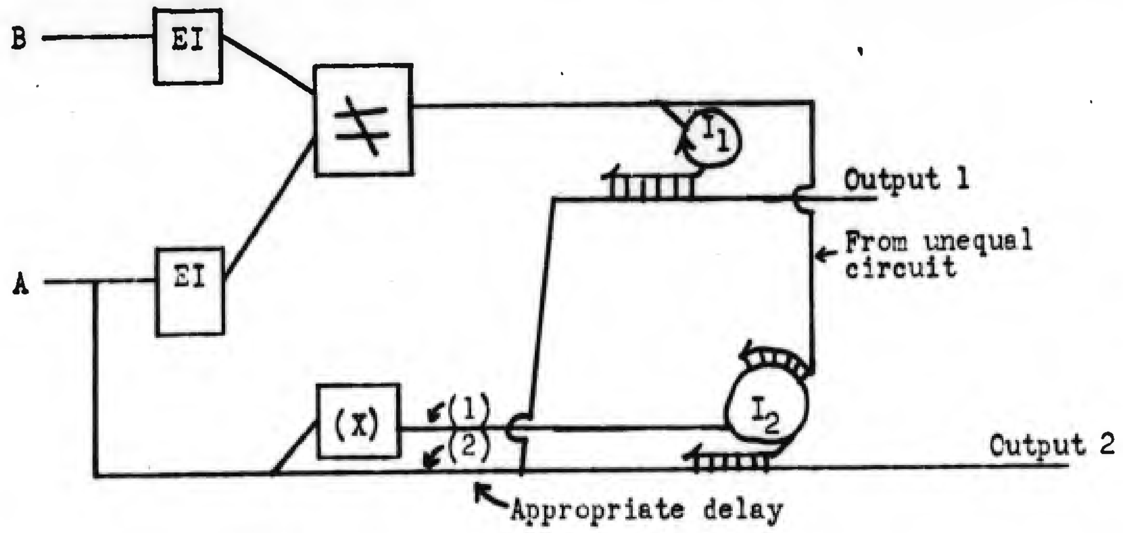


Figure 11
Sorter

III. Sorters

Sorting is an operation that is suited to neuristor circuitry.

Consider the problem, if certain prescribed bits in sequence A match those of sequence B, have sequence A exit by output 1 of the circuit; if they do not match, have sequence A exit by exit 2. (Sequence B could easily be sequence stored in storage rings and triggered each time by the a_f's of a series of different sequence A's). See Figure 11 to follow the solution in the next paragraph.

Sequences A and B feed into identical EI extractors. Here the bits prescribed for matching are extracted and feed into an unequal circuit. An a_f is extracted by circuit (x) to start I_2. Any output of the unequal circuit closes the path to output 1 by starting I_1. Any unequal output also opens path 2 by stopping I_2. When there is no unequal output, only the path to output 1 is open. Sequence A then proceeds out the correct path. Appropriate delays are used to insure that pulses reach the respective parts of the circuit in the sequence listed.

IV. The Add Circuit

Many alternate methods exist for construction of a circuit to add two signed binary numbers. The add circuit shown here is developed in three parts; these parts are the half adder, the basic adder, and finally the full adder of signed fields.

Basic Add Circuit

This circuit adds two binary sequences representing binary numbers. There are two inputs (note a_f and b_f have been removed).

Sequence A_o

$$a_\ell \ldots a_3 a_2 a_1 a_o \longrightarrow$$

Sequence B_o

$$b_\ell \ldots b_3 b_2 b_1 b_o \longrightarrow$$

Where as previously described:

$$A = a_\ell 2^\ell + a_{\ell-1} 2^{\ell-1} + \ldots + a_1 2^1 + a_o 2^o$$

$$A = b_\ell 2 + b_{\ell-1} 2^{\ell-1} + \ldots + b_1 2^1 + b_o 2^o$$

All a's and b's are either zero or one. If there is a pulse in the information position a_i, then a_i is 1; if there is no pulse in the information position a_i, then a_i is 0.

Two half adders are needed as basic sub-circuits for this circuit. A description of a half adder as shown in Figure 12., follows:

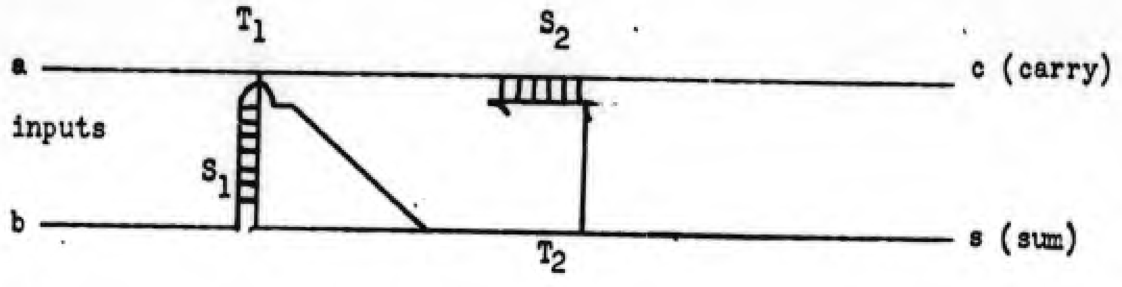


Figure 12
Half Adder

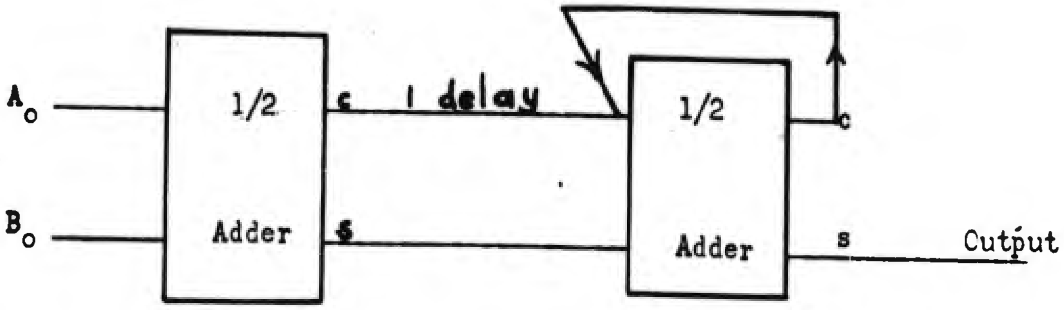


Figure 13
Basic Add Circuit

Possible inputs	Outputs
(1) a =1, b =1	C = , S = 0
(2) a =1, b =0	C = 0, S =
(3) a =0, b = 1	C = 0, S =
(4) a =0, b = 0	C = 0, S = 0

Analysis of operation in Figure 12 for the four cases listed:

(1) Pulse a follows two paths from junction T1. One pulse goes down to share junction S_1, where it meets pulse b travelling through this junction in the opposite direction. Due to the meeting of these pulses in the share junction, neither can proceed any farther. The other "a" pulse proceeds out the carry output.

(2) The "a" pulse travelling through the share junction S_1, proceeds to junction T_2. One path from T_2 allows the "a" pulse to proceed out the S (sum) output. The other path from T_2 leads to share junction S_2. The time of propagation is so adjusted that the "a" pulse from junction T_2 stops the "a" pulse in carry line as they meet in share junction S_2.

(3) The b pulse travels through S_1 to junction T_2 and out the S (sum) output.

(4) No input, hence no output.

Two of these half adders with appropriate carry delays and carry feedback may be combined to give a basic add circuit. (Figure 13.)

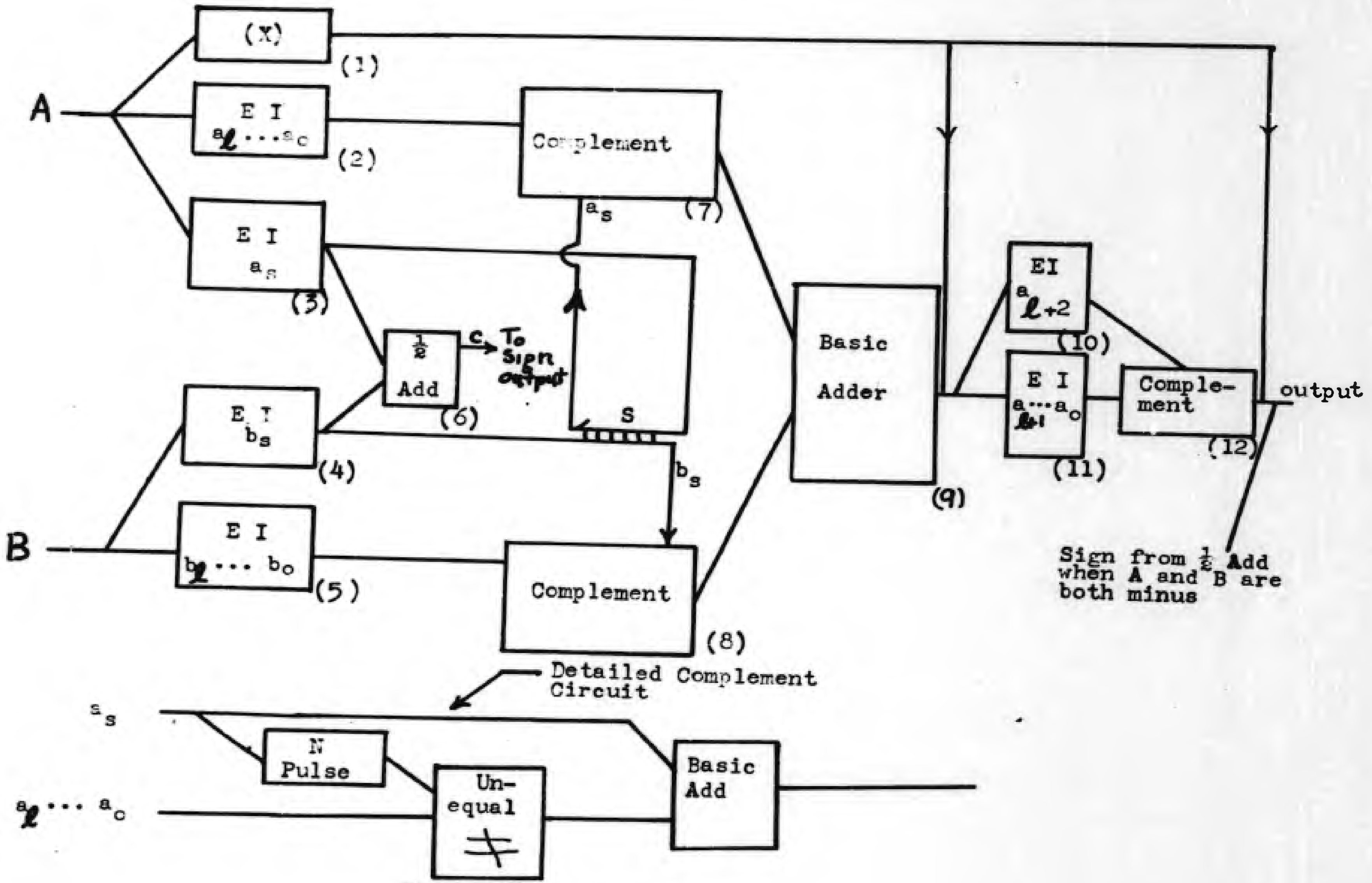


Figure 14 - Full Adder

Note that there is no possibility of two bits arriving simultaneously at the upper input to the second half adder for:

(1) If there is a carry feedback from the second half adder, there must have been a sum input one time cycle earlier. As both a sum and a carry cannot be simultaneously produced by a half adder, there can be no delayed carry from the first half adder.

(2) Similarly, if the carry from first half adder is 1, the sum input to second adder is not 1, hence no feedback arrives when the delayed carry from the first half adder arrives.

The Full Adder of Signed Numbers

The circuit shown in Figure 14 will add signed binary numbers. The inputs are two signed binary numbers represented by pulse sequences A and B of equal length. The last place in each sequence is the sign position. When this position is occupied by a 1, the number is negative; when it is occupied by a 0, the number is positive. The mathematical principle used is to complement negative numbers and add. If the answer to this addition is negative, complement the answer for final numerical output; if the answer to this addition is positive, subtract the base for complementation for the numerical output.

For easy reference to the figure each block is given a number at the lower right hand corner. Following is a list of functions of each part of the figure:

Block one extracts a 1 to serve as the first bit of the basic adder output (block nine) and to serve as the first bit of the final output.

Blocks three and four extract the sign bits.

Blocks two and five extract the binary numbers for sequence A and B.

Block six yields the final sign output when both a_s and b_s are 1. It has no output unless they are both 1.

The share circuit blocks input to the complimenters when both a_s and b_s are 1.

Blocks seven and eight complement the incoming binary number only when the sign input is 1. The base for complementation is a number three bits longer than the incoming binary number. This is accomplished by first considering a binary number two positions longer than the input (put two 0's in next higher positions of incoming sequence). Now the input considered is $a_{\ell+2}$ $a_{\ell+1}$ $a_{\ell} \ldots a_o$ where $a_{\ell+2}$ and $a_{\ell+1}$ are 0.

Next the incoming sequence is combined in unequal circuit with a series of $(\ell+2)$ 1's. This series of 1's is generated by the 1 sign input to the complementer.

Finally, the output of the unequal circuit and a 1 is fed into a basic adder to yield the complemented number. Note that the effect of taking the output of the unequal circuit is to change the 1's of the incoming sequence to 0's and the 0's to 1's.

Block nine adds the inputs from the incoming complementers.

Block eleven extracts the binary number for possible final complement action.

Block ten extracts the sign input for final complementation.

Block twelve complements its input when it has a 1 sign input. The output of block twelve is sequence representing the sum of signed binary numbers A and B.

By enumeration of possibilities of signs A and B, the following depicts the operation of the adder of signed numbers.

Case I, A and B are both positive.

There is no output from block six as both inputs are 0. No complementation takes place in blocks seven or eight as the sign inputs are 0. As inputs to the basic adder are only ℓ positions long, the highest position possible for a 1 is in the $(\ell+1)^{st}$ position. Hence, the $(\ell+2)$ position is zero and no complementation will take place in block twelve and no "1" sign output will come from block ten.

Case II, A and B are both negative.

Block six provides a 1 as the sign of the final since the combination of two 1's (a_s and b_s both 1 as both negative) in the half adder will yield a carry of 1.

Other than this, action proceeds as in Case I for the sign inputs to blocks seven and eight are stopped in the share circuit.

Case III, A negative and B positive (as the whole diagram is symmetric the case of B negative and A positive will be omitted).

Here we have two possibilities, $|A| > B$ and $|A| < B$. In both cases A is complemented in block seven.

Let C be the base of complementation, thus:

$$C = 1\overbrace{0000\ldots\ldots0}^{(\ell+2)\ 0\text{'s}} \qquad C - |A| \text{ is the output of block seven.}$$

When $|A| < B$, the output of the basic adder is greater than C. But there can be no 1 in the $(\ell+2)$ position as $B < C/2$. Hence the first $\ell+1$ positions will be the answer. Example:

$$\begin{array}{lrr} A = 0011 \text{ then } & C - |A| = & 111101 \\ & B = & \underline{\ \ 0100} \\ & C - |A| + B = & \underset{s}{1}\underbrace{000001}_{\text{answer}} \end{array}$$

When $|A| > B$, the output of basic adder is less than C. But there must be a 1 in the $(\ell+2)$ position as $|A| < C/2$. Hence complement first $\ell+1$ positions for answer. Put out 1 for sign of answer.

Example:

$$\begin{array}{lrr} A = 0100 & C - |A| = & 111100 \\ B = 0011 & B = & \underline{\ \ \ \ 11} \\ & C - |A| + B = & \underset{\text{sign}}{1}\underbrace{11111}_{\text{complement of answer}} \end{array}$$

Complementing the five positions with C, the output of complementor is $\overset{s}{1}00001$. This is what was desired. The extra 1 takes care of minus sign for output. A base for complementation, one-half of the C used might be employed if we are sure that A and B are of opposite sign or that A + B will not cause a carry into a bit position outside the highest allowable input position, $\ell+1$ (a_ℓ occupies the (ℓ+1) position). This would give us the advantage of having output sequence length equal to input sequence length.

As previously stated, the mathematical principal used here is to accomplish subtraction by complementation and addition. When the addition results in a sum which exceeds the base of complementation, the desired difference is realized by ignoring all places in sum higher than those present in the original factors. When the addition results in a sum less than the base of complementation; a negative sum requires another complementation to obtain the answer. (Ref 4:124)

V. Subtraction

To subtract one signed number from another, it is only necessary to:

(1) extract the sign field of the first number

(2) add this sign field to a 1 in a half adder

(3) replace the original sign with the sum output of the half adder

(4) proceed as in addition of signed numbers.

This procedure changes the sign of the number to be subtracted and then adds.

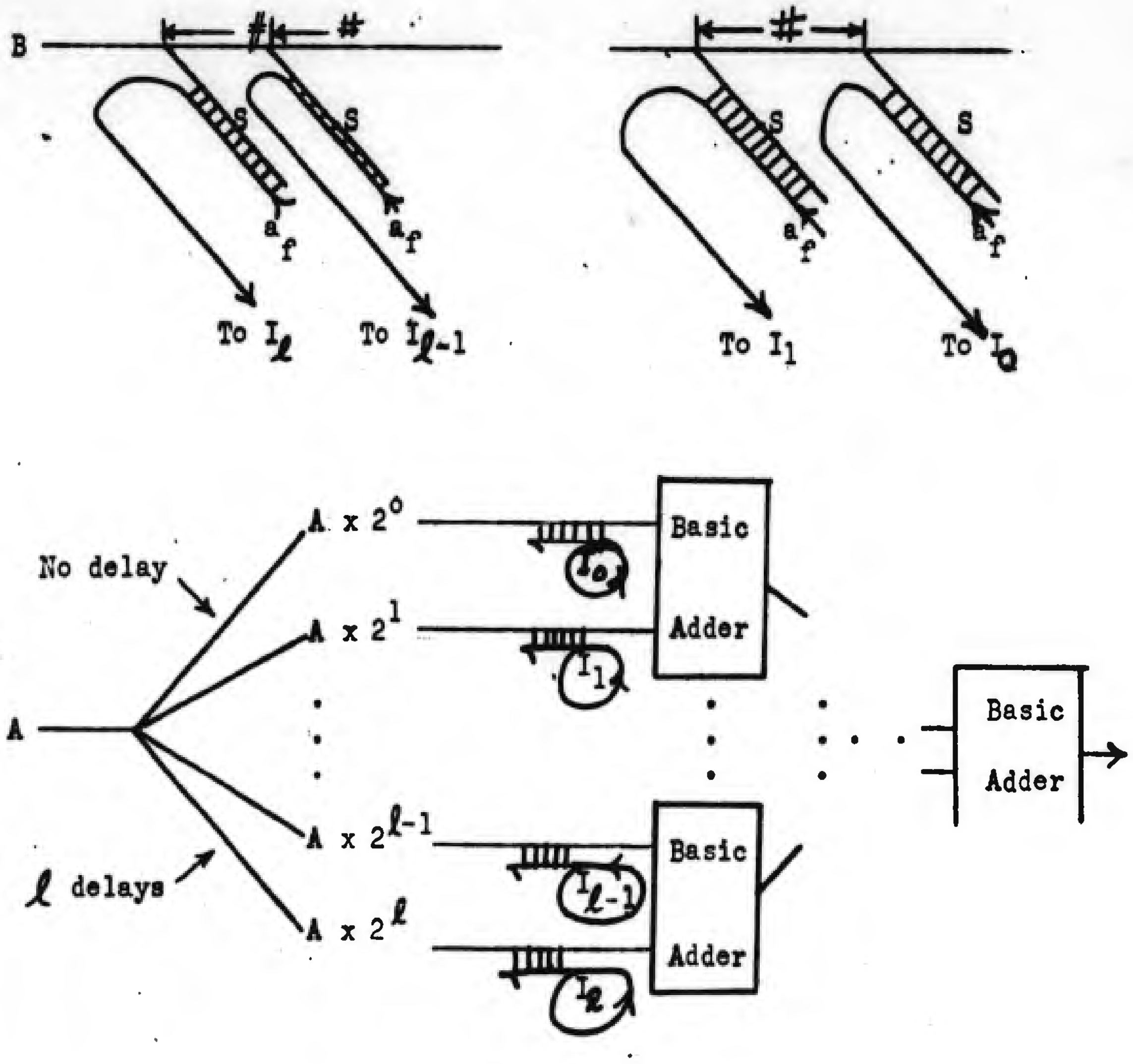


Figure 15
Multiply Circuit

VI. Multiplication

The multiplication of two signed binary numbers is done in two parts. First, the sign of the product is computed by extracting the two signs and combining them in an unequal circuit. The output of unequal circuit is the sign of the product. Second, basic multiplication is performed in a basic multiplier. An explanation of the basic multiplier follows.

A circuit diagram for the basic multiplier is shown in figure 15. The mathematical principle is to first multiply the multiplicand separately by all (ℓ+1) powers of two; second, discard all products where the corresponding bit in the multiplier is 0; and finally, add together the remaining products for the answer.

The first task is accomplished by branching the multiplicand into ℓ+1 branches and delaying each line by one more time unit than the next lower numbered line. As delaying a binary sequence multiplies the corresponding binary number by two, the multiplicand has been separately multiplied by all (ℓ+1) powers of two, (i.e., from 2^0 to 2^ℓ).

The second task is accomplished (see Figure 15.) by intersecting an a_f with each of $b_\ell \ldots b_o$'s then sending the a_f's which get through to start corresponding inhibitors of the multiplicand branches. Thus if $b_i = 1$, a_f does not get to start I_i, and A times

2^i proceeds; but if $b_i = 0$, a_f does get to I_i, and A times 2^i is stopped by I_i.

The remaining products are added in successive groups of basic adders to yield the final answer. (See figure 15)

VII. Division

Plan for Division

$A \div B$ (Neither A nor B = 0)

1. The sign fields of A and B are extracted and combined in an unequal circuit. The output of the unequal circuit is placed in the output quotient as the sign.

2. Count the number of consecutive zeros in the nonsigned binary numbers of A and B, starting with the highest bit position. Let the results of the counts be represented by the numbers m and n respectively.

3. Subtract $B \times 2^{n-m}$ from A. Store a bit in the 2^{n-m} storage ring.

4. a. If the remainder is positive, subtract $B \times 2^{n-m-1}$ from the remainder. Store a bit in the 2^{n-m-1} storage ring.

b. If the remainder is negative, add $B \times 2^{n-m-1}$ to the remainder. Store nothing in the 2^{n-m-1} storage ring.

5. Diminish the power of 2 in step 4 and repeat the operations in step 4 with this new power of two. Continue this process, each time decreasing the previous power of 2 by one, until the last available storage ring has been reached.

6. Generate two binary numbers M and N with the information in the storage ring. So that

$$M = k_{n-m}2^{n-m} + k_{n-m-1}2^{n-m-1} + \dots\dots k_{n-m-1}2^{n-m-\ell}$$

$$N = \bar{k}_{n-m}2^{n-n} + \bar{k}_{n-m-1}2^{n-m-1} + \dots\dots \bar{k}_{n-m-1}2^{n-m-\ell}$$

where

$k_i = 1$ and $\bar{k}_i = 0$ if there is a bit in the i <u>th</u> storage ring.

$k_i = 0$ and $\bar{k}_i = 1$ if there is not a bit in the i <u>th</u> storage ring.

7. Subtract N from M, the result is the quotient desired.

Reason-

$$A + B\,(-M+N) = \text{final remainder}$$

or

$$\frac{A}{B} = M-N+ \quad \frac{\text{final remainder}}{B}$$

As the remainder can be made as small as desired by using storage rings representing smaller powers of two, then

$$A \div B = M-N \pm \text{ any desired limit.}$$

thus the quotient of $A \div B$ is M-N with remainder limited by the power of two represented by the lowest storage ring and the magnitude of B. A round off by taking extra places will not be considered as a round off in binary arithmetic can only decrease the average error but not the maximum error.

Division Circuitry

The circuit will accomplish division as set forth in this plan is shown in Figure 16g. The possibility of A = 0 or B = 0 is also provided for in this circuit. A more detailed picture of the sub-

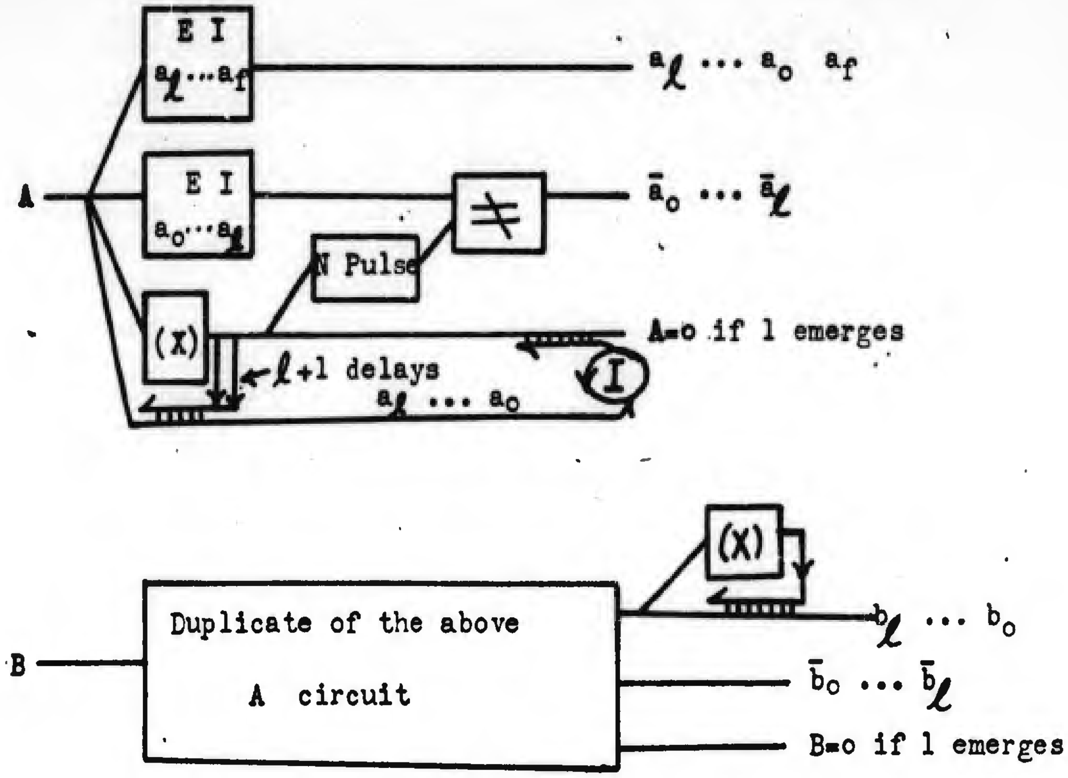


Figure 16a
Subcircuit a.

circuits is shown in Figures 16a, b, c, d, and e. Figure 16f explains in more detail the variable pulse generator shown in Figure 16e.

Following is a more detailed explanation of the action that takes place in each subcircuit.

1. Action in Figure 16a. (Proceeding from the top to bottom.) The upper EI extractor removes the sign bit from sequence A, yielding an output of $a_{\ell}\ldots a_o a_f$.

The other EI extractor removes a_s and a_f as well as reversing the order of the sequence. This reversed order sequence is then combined in an unequal circuit with a sequence of $(\ell+1)$1's. The output of the unequal circuit will then be $\bar{a}_o\ldots\bar{a}_{\ell}$ (where $a_i+\bar{a}_i = 1$). Here the number of consecutive 1's in this output is a measure of the number of consecutive zeros in the non-signed binary number A, starting with the highest bit position.

The (x) circuit extracts the a_f bit. This a_f is fed into the N Pulse Generator to provide the $(\ell+1)$ 1's for the unequal circuit. The a_f is also fed back at proper times into the lower line to inhibit passage of a_s and a_f. The a_f also provides the A = 0 signal.

If A is not zero, at least one of the bits in the lower line sequence $a_{\ell}\ldots a_o$ will be a 1. Then the inhibitor blocking the A = 0 output will not let the a_f signal out. However, when A is zero, the inhibitor will not operate and the a_f signal will proceed out.

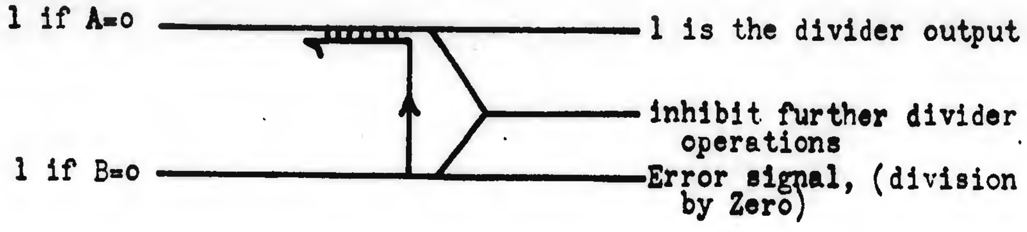

1 if A=o
1 if B=o
1 is the divider output
inhibit further divider operations
Error signal, (division by Zero)

Figure 16b
Subcircuit b.

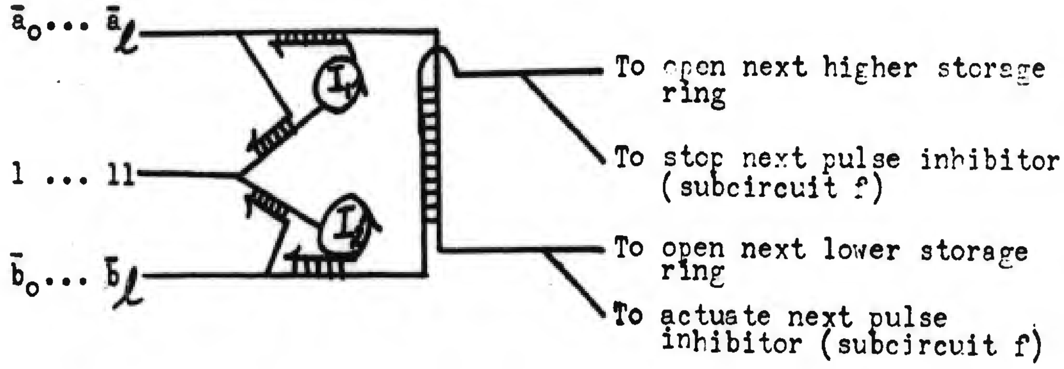

$\bar{a}_o \ldots \bar{a}_\ell$
1 ... 11
$\bar{b}_o \ldots \bar{b}_\ell$
I
I
To open next higher storage ring
To stop next pulse inhibitor (subcircuit f)
To open next lower storage ring
To actuate next pulse inhibitor (subcircuit f)

Figure 16c
Subcircuit c.

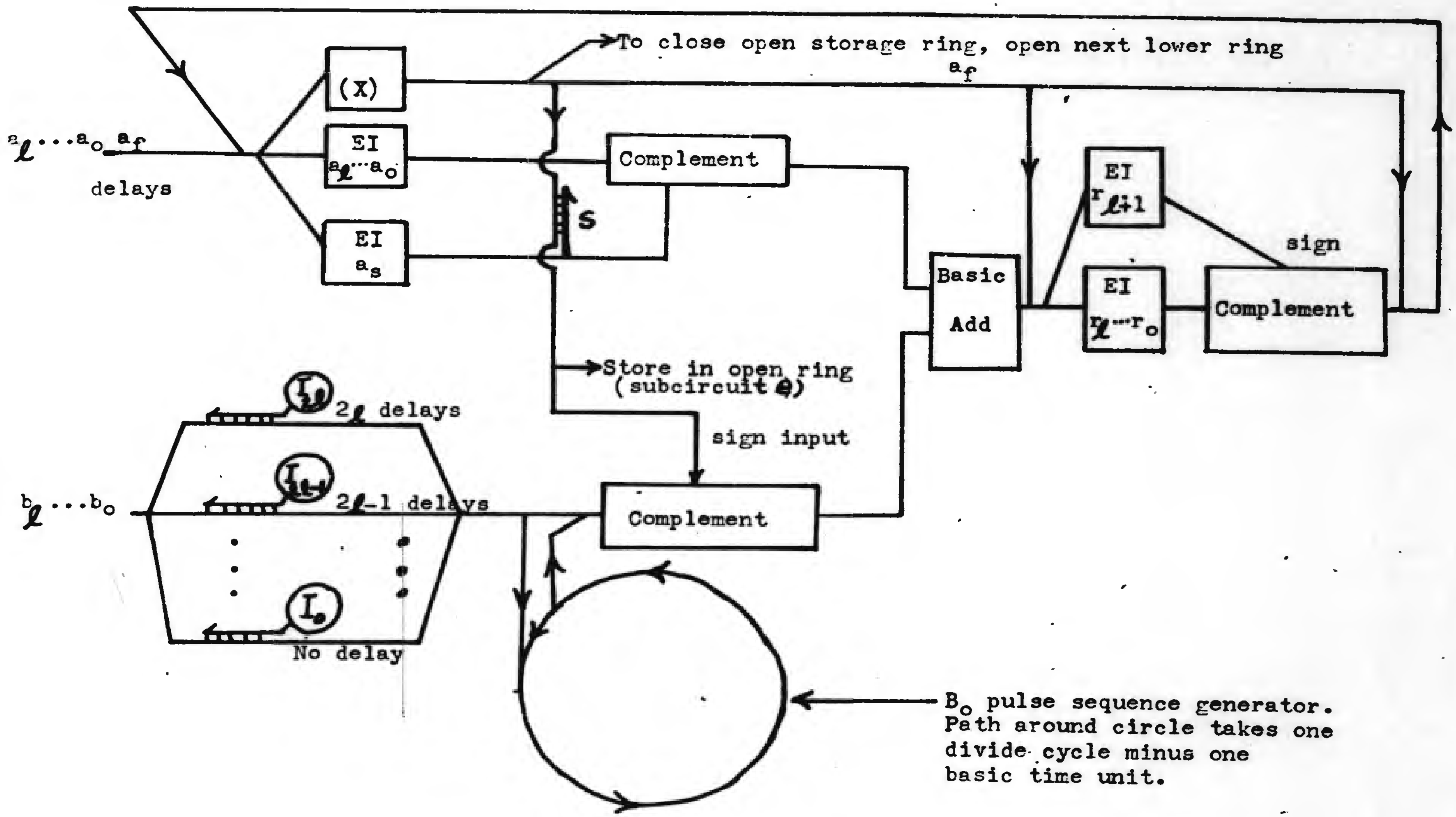
The lower half of the figure is the same as the upper half except that b_f is not included in the first output. The (x) circuit feeds back the b_f bit to inhibit itself.

2. Action in Figure 16b.

When $B = 0$, there is a 1 input to lower line from subcircuit a. This 1 signals error as division by zèro is not possible. This one also inhibits any further operations in the entire divider. When $A = 0.$, but $B \neq 0$. The 1 input from subcircuit "a" proceeds to the right to yield a 1 as the divider output and to inhibit further division circuit operation. This is what is desired for 0...001 is $q_s q_\ell \cdots q_1 q_0 q_f$, which means the quotient is zero.

3. Action in subcircuit C. (Figure 16c.)

This circuit is timed so that as long as a sequence of 1's come in the upper and the lower lines, I_1 and I_2 cannot start inhibitor action. After operation starts, the first zero that appears allows the middle input to start the inhibitor of the line containing this zero. Thus the number of 1's that pass I_1 and I_2 respectively are the number of consecutive zeros in A and B counting from the highest bit position. By passing the upper and lower lines through the share junction, the difference between the number of consecutive zeros in A and B is obtained. The difference only will then emerge out the line which contained the larger number of consecutive 1's. (These 1's represent consecutive 0's in the original sequence.)



B_o pulse sequence generator. Path around circle takes one divide cycle minus one basic time unit.

Figure 16d
Subcircuit d.

There can be only one output. An upper output proceeds to open next higher storage ring; a lower output proceeds to open next lower storage ring. (See Figure 10b for inhibitor action in regulating open storage rings.) The 2^o ring is initially the open ring, all others are closed by an a_f pulse. These outputs also go to adjust the N+x-y Pulse Generator.

4. Subcircuit d. (Figure 16d.)

The $b_\ell \ldots b_1 b_0$ sequence enters the lower input and branches into $2\ell+1$ paths, each one is one time unit longer than the next lower numbered path. Thus a binary number emerging from one of these lines would be 2 times a number emerging from the next lower numbered line. Initially all the inhibitors except I_1 on these lines are set in action by an a_f bit. These are the same inhibitors that regulate the open storage rings in Figure 16e. The correspondence is such that I_i regulates passage to $R_{i-\ell}$. The output of subcircuit C has operated to open the correct storage ring; hence, the correct line in the multiple branch is open. The following example illustrates this:

Suppose:

$\bar{a}_o \ldots \bar{a}_{\ell-1} \bar{a}_\ell$ has first 0 in $\bar{a}_7$.

$\bar{b}_o \ldots \bar{b}_{\ell-1} \bar{b}_\ell$ has first 0 in $\bar{b}_{12}$.

Then $\bar{b}$ output is 5 consecutive 1's. These 5 consecutive 1's close I_ℓ and open $I_{\ell+5}$. This delays the B number five time units more than the A number which has fixed ℓ delays. Thus B has been multiplied by 2^{n-m} so that subtraction may now begin.

Next, after the multiple branching, the B sequence sends a branch into a time delay ring. This ring returns the B sequence advanced by one time cycle at the same time the remainder feedback returns to the upper circuit. This diminishes the power of two (in 2^{n-m} xB) by one each time.

Action in the remainder of the circuit is much the same as in full adder of signed fields. There are some important exceptions.

One exception is that the sign of B x 2^{n-m-i} is always opposite that of the upper input. This is accomplished by allowing a branch containing the sign of the basic adder upper input to inhibit passage of an a_f bit that is going to serve as the sign of the lower input. This helps accomplish step 4 of the Plan for Division by subtracting B x 2^{n-m-i} when the remainder is positive and adding when the remainder is negative. The rest of step 4 is accomplished by storing the sign of the B x 2^{n-m-i} input in the open ring. An a_f from each feedback goes to open the next lower ring and to close the open ring.

Another important exception is that the base for complementation is two bits longer than the allowed length of incoming binary numbers. Trouble would result in the original full adder if 1's were allowed in the high order positions and both numbers were positive.

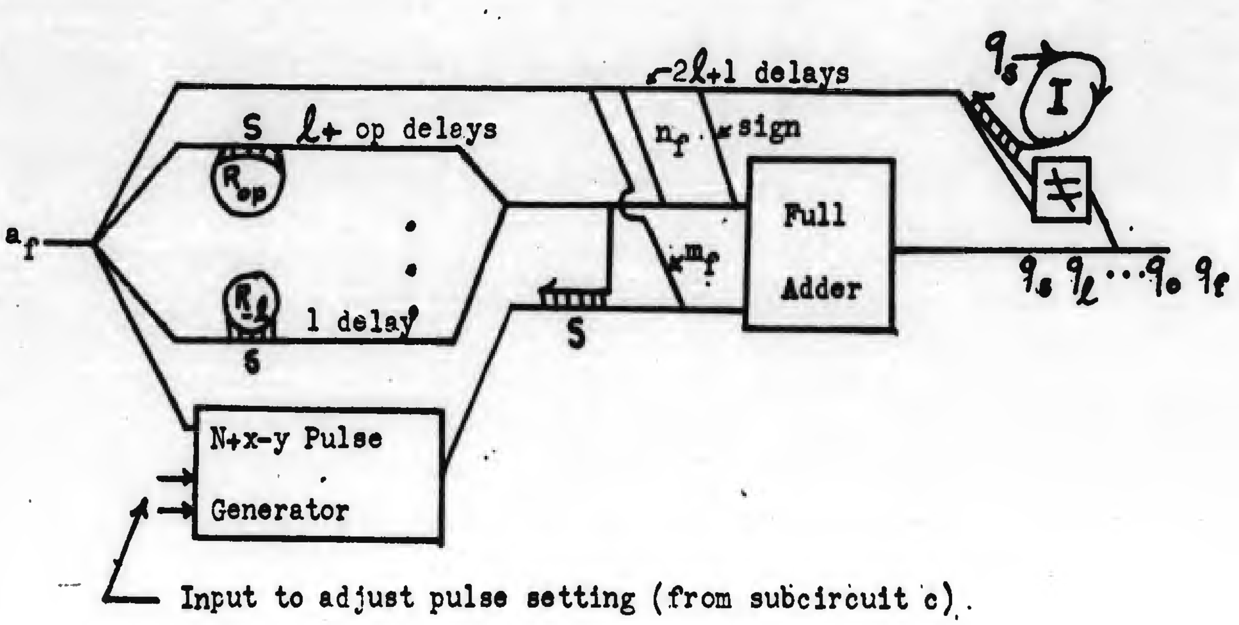
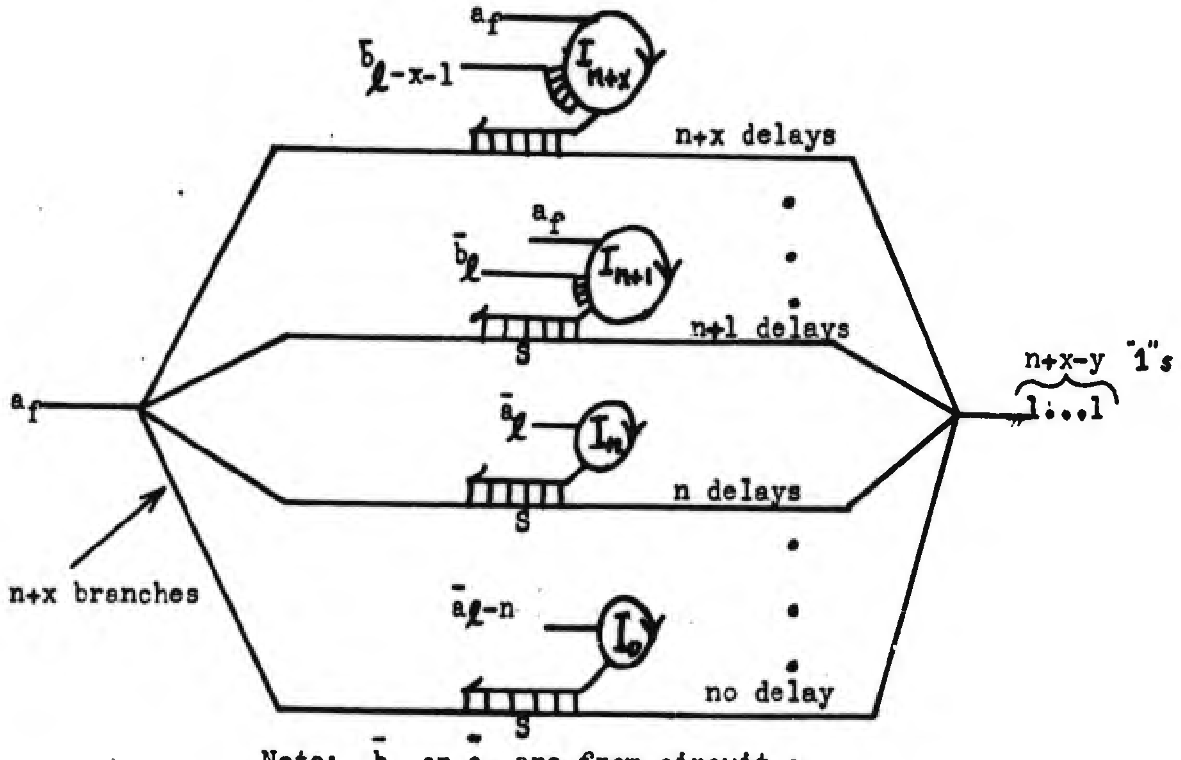


Figure 16e
Subcircuit e.



Note: $\bar{b}_i$ or $\bar{a}_i$ are from circuit c

Figure 16f
N+x-y Pulse Generator

Then a 1 would appear in the sign position even though both numbers were positive; $(\ell+1)$ is the sign position for complementation with $(\ell+2 \text{ positions}) \rightarrow 1000\ldots0$.

As we are always combining two numbers of opposite sign in this subcircuit, there is no danger of an addition causing a sign difficulty.

5. Subcircuit e. (Figure 16e.)

The a_f bit from subcircuit d which opens the last storage ring branches to become the input to subcircuit e. Of course there is a slight delay to allow time for storage in the last ring $(R\text{-}\ell)$.

The upper branch line goes through to provide a first bit for the N and M binary number sequences. The sequence that emerges from the branching and a_f intersection with the storage rings is N, the number of times that B was added. The inhibitor action of N on a sequence of continuous pulses of the same length as N produces M.

R_{op} indicates the first open storage ring when the division begins. More branches may exist for higher number storage rings but these are blocked by a method described in subcircuit f.

The a_f bit also provides a 1 for the sign of the N binary number. Here again the base for complementation could be $\overbrace{100\cdots0}^{\ell+2 \text{ positions}}$. For uniformity this base should be used in all full adders; however, care must be not to exceed the capacity of the circuit by addition or an erroneous answer will result.

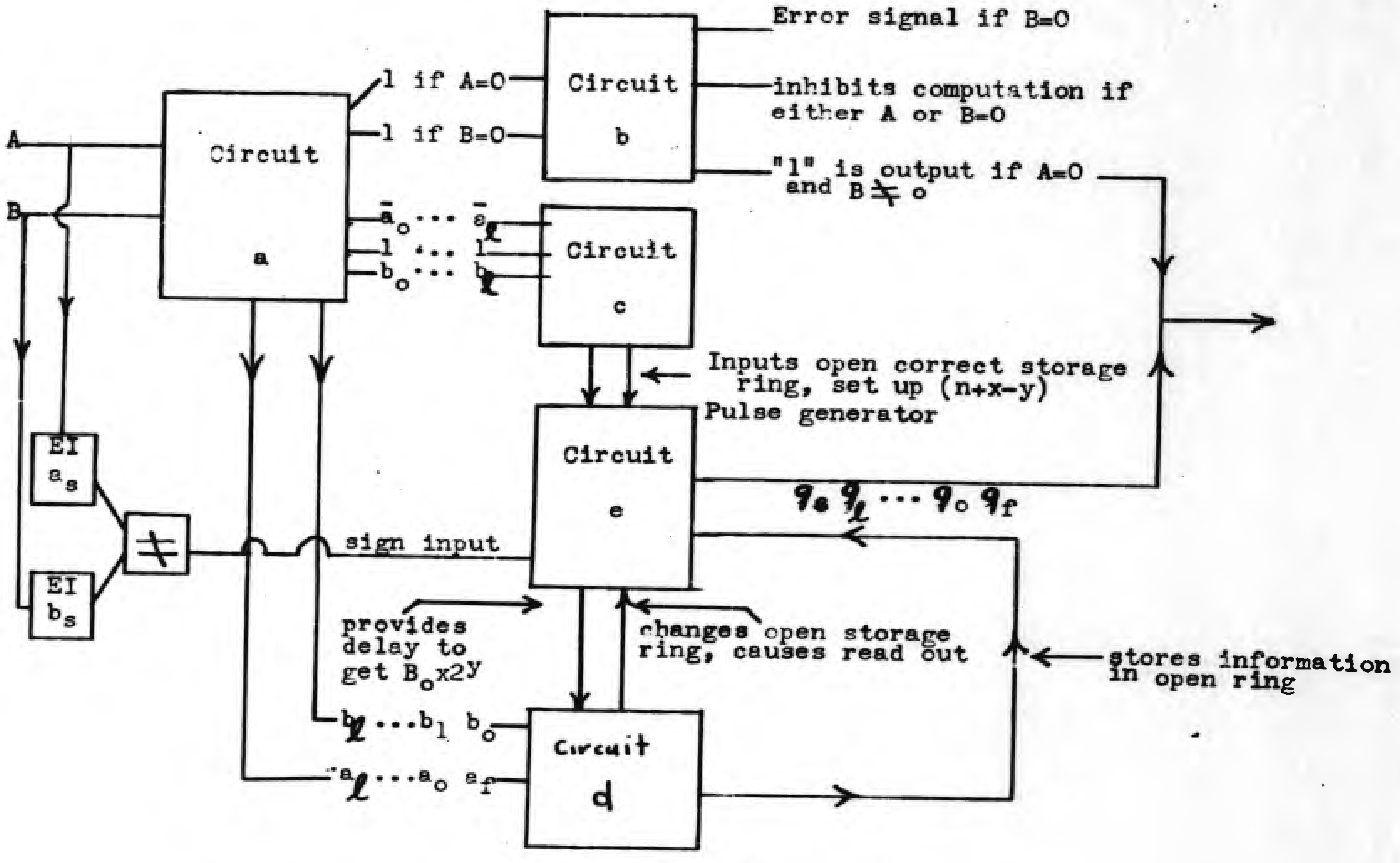

Error signal if B=O
inhibits computation if either A or B=O
"1" is output if A=O and B ≠ o
1 if A=O
1 if B=O
A
B
Circuit a
Circuit b
Circuit c
Circuit e
Circuit d
$\bar{a}_o \cdots \bar{a}_\ell$
$1 \cdots 1$
$b_o \cdots b_\ell$
EI a_s
EI b_s
≠
sign input
Inputs open correct storage ring, set up (n+x-y)
Pulse generator
$q_s\ q_\ell \cdots q_o\ q_f$
provides delay to get $B_o x2^y$
changes open storage ring, causes read out
stores information in open ring
$b_\ell \cdots b_1\ b_o$
$a_\ell \cdots a_o\ a_f$

Figure 16g
Division Circuit Block Diagram

As the time of division is variable, the sign bit of the final answer is stored and triggered by a properly delayed a_f bit. This sign bit is computed as in the multiplication circuit.

6. Subcircuit f. (Figure 16f · N+x-y Pulse Generator.)

This subcircuit is a part of the previous subcircuit. All of the inhibitors shown block the paths of the a_f read out "1" to the storage rings as well as the paths shown in the N+x-y Pulse Generator. These inhibitors are in parallel with the normal ones which allow only one storage ring to be open. Initially, I_{n+1} thru I_{n+x} are closed by an a_f bit. Each $\bar{b}$ from subcircuit c opens the next higher line; each $\bar{a}$ closes the next lower line. It is impossible to get both an $\bar{a}$ and $\bar{b}$ output from subcircuit c. The $\bar{a}$ and $\bar{b}$ bits which accomplish this action come through the normal inhibitor lines which allow only one storage ring to be open. However, after $\bar{a}$ and $\bar{b}$ output ceases, input to the inhibitors of Figure 16f, is blocked by an a_f bit. Thus the division operations will not alter the configuration of these inhibitors.

Example: Suppose storage ring configuration is 1101011. Then upper input to full adder is $\overset{s}{1}0010100\overset{f}{1}$. This is -N. The lower input to full adder $\overset{s}{0}1101011\overset{f}{1}$. This is M. The output of the full adder is $01010111\overset{f}{1}$. This is the quotient; the sign is added in the next operation.

7. Division Circuit Block Diagram (Figure 16g.).

This diagram ties together all of division subcircuits. In addition it shows the combination of the sign bits in an unequal circuit to yield the sign of the quotient.

VIII. Conclusion

The ease with which the neuristor nets may be combined to perform digital computer arithmetic operations (addition, subtraction, multiplication and division), indicates a need for further study of neuristor capabilities in performing digital computer operations. The existence of a more detailed plan for construction of a digital computer using neuristor nets would save much time in the construction of a practical digital computer employing neuristor capabilities. Programming or checking are not included in this study, nor are plans for many other mathematical operations. Further work in these areas is recommended to further develop neuristor circuits for digital computer operations.

Bibliography

1. Crane, H.D. Neuristor Studies. Technical Report No. 1506-2, July 11, 1960. Office of Naval Research.

2. Biezer, Boris, and S.W. Leibholz. "Engineering Applications of Boolean Algebra". (Electrical Manufacturing Combined Reprint) New York: Gage Publishing Co., (No date).

3. Murphy, J.S. Basics of Digital Computers Vol. I, II and III, J.F. Rider, Publisher, 1958.

4. Richards, R.K. "Arithmetic Operations in Digital Computers", Princeton, N.J., D. VanNostrand Co., 1955.

Appendix A

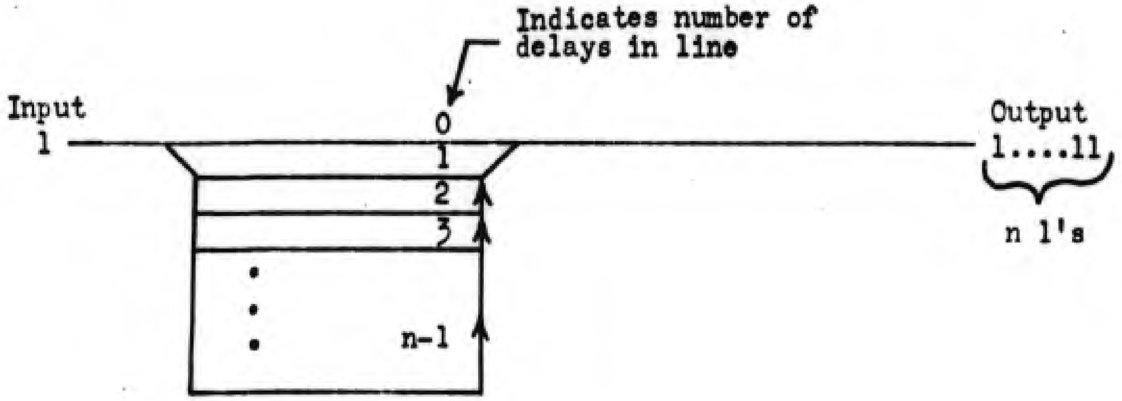


Figure 17
N Pulse Generator

Vita

William Carrol Ross was born [redacted] From 1943 to 1945, he was an enlisted man in the United States Army. He graduated from the United States Military Academy in 1949 and did graduate work in Mathematics at Columbia University in 1952-1953 prior to instructing in Mathematics at the United States Military Academy. His assignment prior to coming to the Institute of Technology was as Chief of Operations Research at Memphis Air Force Depot.

Permanent address: [redacted]

This thesis was typed by Mrs. Jeanette M. Barnett