

I ILLIAC DESIGN TECHNIQUES

by James E. Robertson*

1.0 General Description of the Illiac

The Illiac is one of several binary digital computers patterned after the Institute for Advanced Study Computer at Princeton, New Jersey. These computers are

NAME	BUILT BY	BUILT FOR	COMPLETION DATE
ORDVAC	Univ. of Ill.	Aberdeen Proving Ground, Md.	Nov., 1951
ILLIAC	Univ. of Ill.	Univ. of Ill.	Sept., 1952
I. A. S.	I. A. S.	Institute for Advanced Study, N. J.	Jan., 1952
AVIDAC	Argonne	Argonne Nat'l Lab., Tenn.	1953
ORACLE	Argonne	Oak Ridge Nat'l Lab., Tenn.	June, 1953
MANIAC	Los Alamos	Los Alamos Scientific Lab., N. Mex.	March, 1952
JOHNNIAC	Rand Corp.	Rand Corp., Santa Monica, Calif.	Jan., 1954
SILLIAC	Univ. of Sydney	Univ. of Sydney, Australia	1956?

Although the computers in the I. A. S. group differ in design details, many of the comments in the discussions of the Illiac which follow are applicable to the group.

The major characteristics of the Illiac are summarized in the following table:

1. The Storage Unit

Capacity -- 1024 words of 40 binary digits each

Type of Storage -- parallel electrostatic (Williams)

Access time -- 18 μ sec minimum 36 μ sec maximum

2. The Arithmetic Unit

Mode of operation -- parallel

Representation of numbers

Binary, fixed point

40 binary digits; a number x lies in the range $-1 \leq x < 1$

Negative numbers represented as two's complement

* Digital Computer Laboratory, University of Illinois

Arithmetic operations with time required for execution (including access time)

Addition, subtraction	72 μ sec
Multiplication	620 - 800 μ sec
Division	900 μ sec

3. The Control Unit

Type of instruction -- single address
Function digits -- 8
Address digits -- 10 for each instruction
Unused digits -- 2
Instructions per word -- 2
Mode of operation -- asynchronous

4. Input-output Mechanisms

Input-output medium -- 5 hole punched paper tape
Input reader -- photoelectric -- 300 characters per second
Output units:
a) Tape punch -- 60 characters per second
b) Page printer -- 5 characters per second
c) Cathode ray tube display unit -- 1 millisecond per spot
with raster of 256 x 256 spots

The differences with MIDAC type computers which will be emphasized are the following:

1. Parallel mode of operation
2. Two's complement representation of negative numbers
3. Asynchronous control design
 - a) Use of flipflops
 - b) Techniques of shifting
 - c) Design of counting circuits

2.0 Elementary Illiac Circuits

There is a direct correspondence between the elementary circuits used in the Illiac and the operations of Boolean algebra and the connectives of formal logic. The correspondence is indicated in the table below:

ILLIAC	BOOLEAN ALGEBRA	LOGIC
signal with two voltage levels	element	statement
positive voltage (0 v)	0 element	false statement
negative voltage (-20 v)	1 element	true statement
"and" circuit	operation \cdot	"and" connective
"or" circuit	operation \vee (+)	"or" connective
"not" circuit (inverter)	operation $\bar{}$ (')	"not" connective

The elementary circuits used in Illiac corresponding to the logical operations are shown in Figure 2.0.

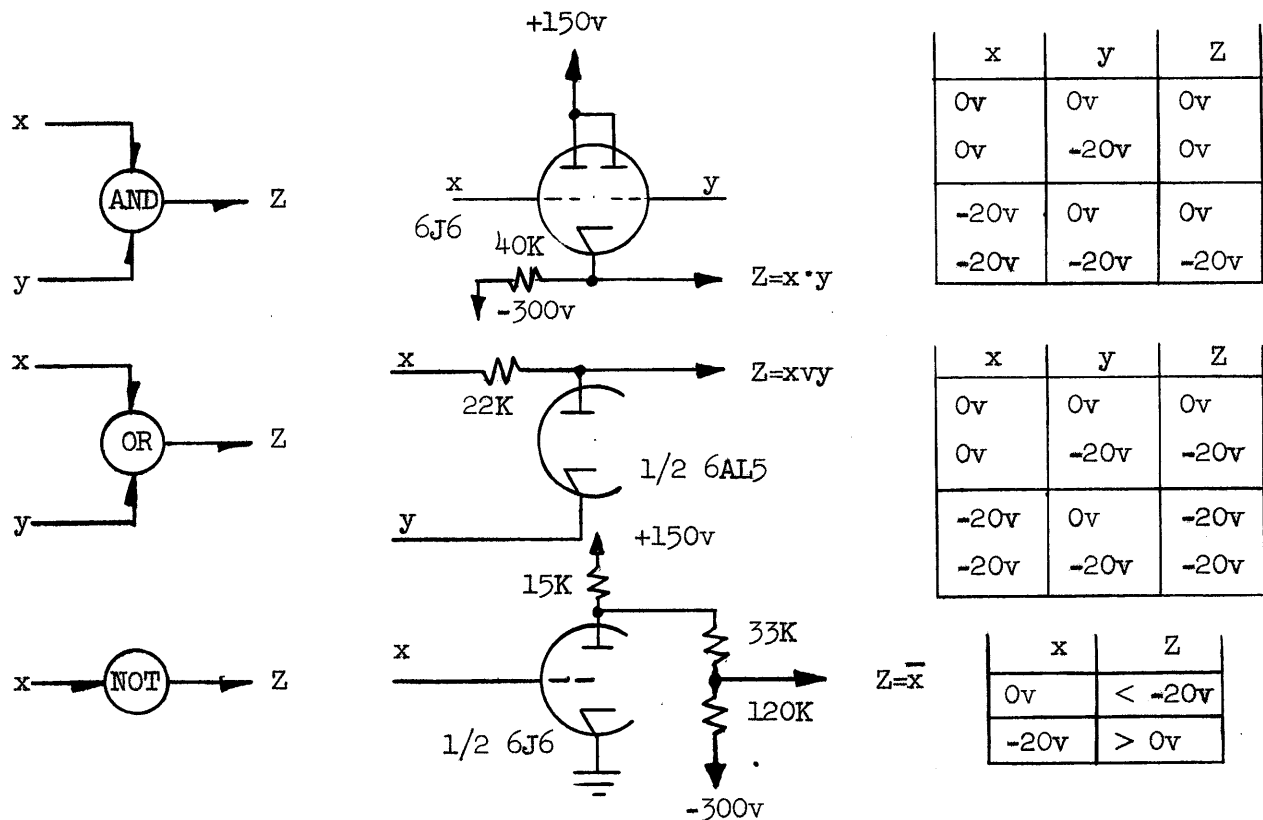


Figure 2.0 ELEMENTARY ILLIAC CIRCUITS

2.2 The Illiac Register; Representation of Numbers

A number is stored in the Illiac arithmetic unit in a set of forty flipflops, called a register. The binary point is fixed to the right of the most significant digit, which serves as the sign digit. (Figure 2.2)

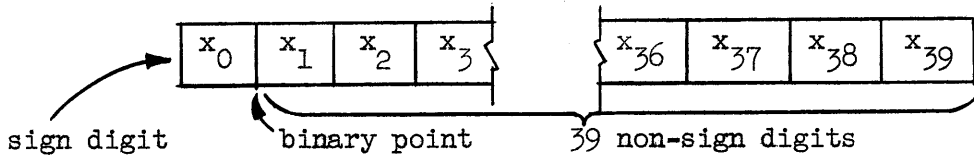


Figure 2.2 THE ILLIAC REGISTER

Negative numbers are represented as complements with respect to two. The process of forming the complement can be visualized by counting downward by a unit in the least significant place:

0.010	+ 1/4
0.001	+ 1/8
0.000	0
1.111	- 1/8
1.110	- 1/4

Complementation can also be performed by replacing 1's by 0's and 0's by 1's and subsequently adding a unit in the least significant place, as follows:

0.010	+ 1/4
1.101	
1	
<hr/>	
1.110	- 1/4

If we call the digits of a number x_0, x_1, \dots, x_{39} we can determine the algebraic value x of the number from the machine representation by the formula:

$$x = -x_0 + \sum_{i=1}^{39} 2^{-i} x_i \quad (2.2)$$

For example, $-1/4$ is represented as 1.110 and is equal to $-1 + 1/2 + 1/4$.

2.3 Addition and Subtraction in the Illiac

The process of binary addition is illustrated by the following example:

```

  1 1 1 1
  0 0 1 1 1 1 0 0
  0 1 0 1 1 0 1 0
  -----
  1 0 0 1 0 1 1 0
  
```

If one inspects the possible combinations which may arise in any one digital position of an adder, one sees that it is necessary to find two output functions--the sum digit and the carry digit--of three binary variables; the addend digit, the augend digit, and the incoming carry digit.

Techniques for the logical realization of one digital position of an adder are well known; a logical diagram for one adder digital position is shown in Figure 2.31.

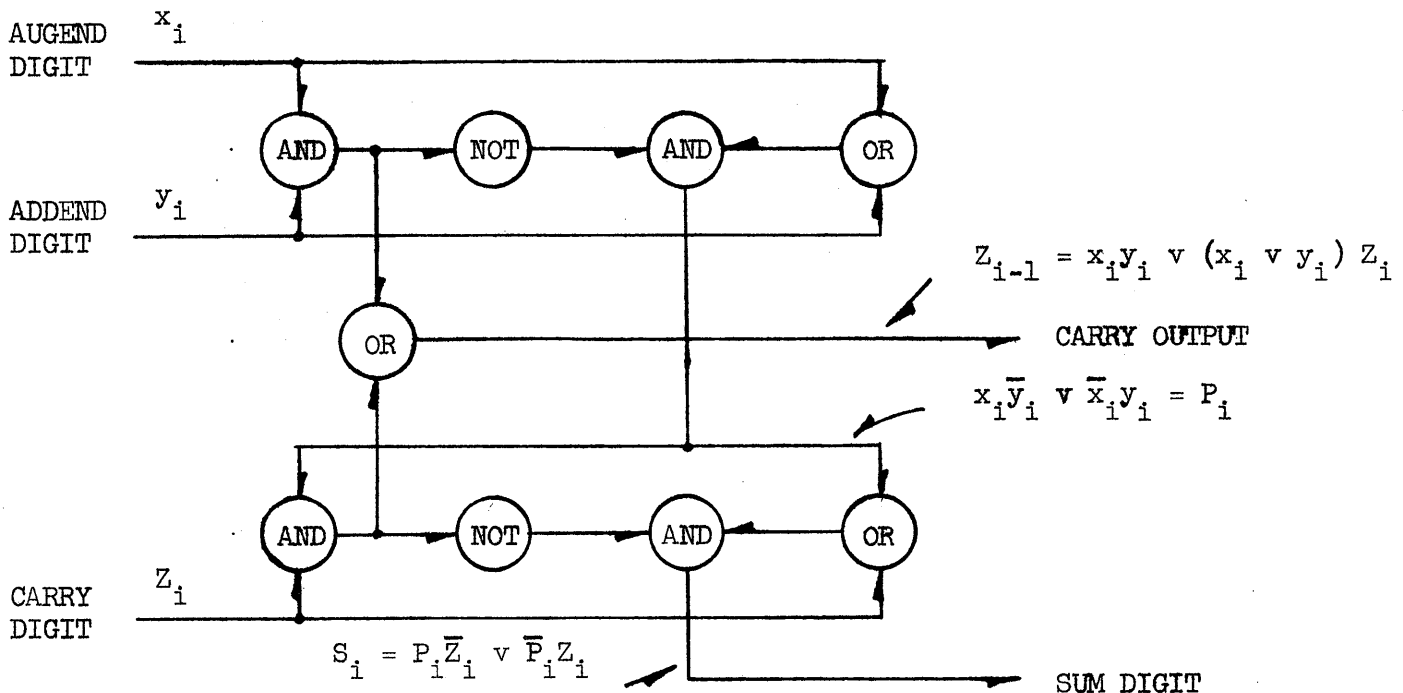


Figure 2.31 LOGICAL DIAGRAM OF ADDER

For a serial computer, the adder requires one circuit of the type shown in Figure 2.31, or its Boolean equivalent. For a parallel computer with 40 binary digits per number, such as the Illiac, 40 circuits of the type shown must be constructed.

If we consider the addition of two numbers, one or both of which may be negative, we must know what system of representation of negative numbers is used. To choose an example which is most difficult in the absolute value system, consider the addition $+ 1/2 + (- 3/4) = - 1/4$

The addition requires the following steps:

- 1) Convert $- 3/4$ to complementary form
- 2) Perform the addition
- 3) Check the sign of the result, and complement again if negative.

Thus: 1.11 is converted to 1.01
 0.10 is added: 0.10

 1.11

The result is complemented: 1.01

In contrast, in the complementary system of representation, the adder circuit suffices for the process; for example:

$- 3/4$	1.01
$+ 1/2$	$+0.10$
<hr/>	<hr/>
$- 1/4$	1.11

For addition in the absolute value system, an arrangement of equipment similar to that shown in Figure 2.32 is required.

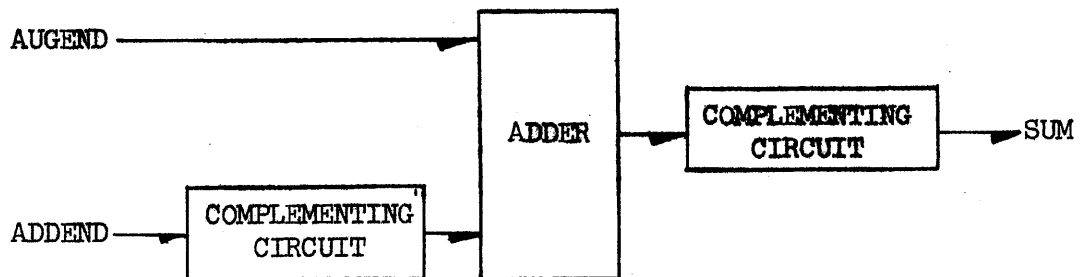


Figure 2.32 EQUIPMENT FOR ABSOLUTE VALUE ADDITION

The same equipment suffices for subtraction. For addition and subtraction in the complementary system, only one complementing circuit is required, as indicated in Figure 2.33.

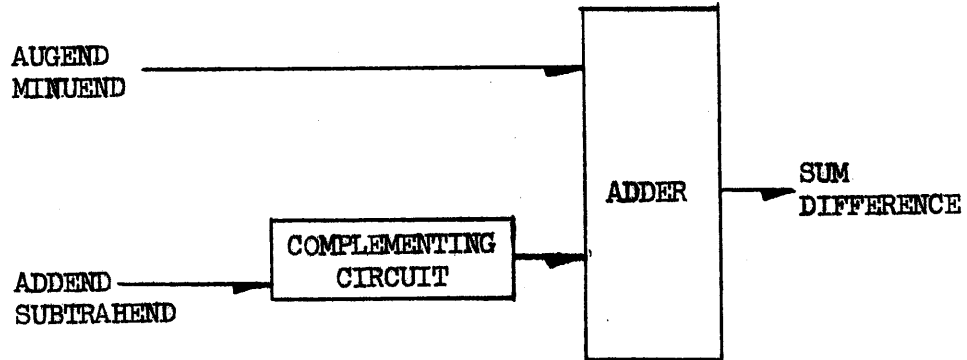


Figure 2.33 EQUIPMENT FOR ADDITION AND SUBTRACTION IN THE ILLIAC

The Illiac method has the additional advantage that the sign digit is treated in the same manner as the non-sign digits; whereas in the absolute value system, special circuits for the sign digits are required.

2.4 Shifting Circuits

In order to perform multiplication and division automatically as sequences of additions or subtractions, facilities for shifting the digits of a number with respect to the fixed binary point are required. The left shift corresponds to a multiplication by two; the right shift to a division by two; for example:

0.001	shifted left becomes	0.010	$1/8 \times 2 = 1/4$
1.101	shifted left becomes	1.010	$-3/8 \times 2 = -3/4$
0.010	shifted right becomes	0.001	$1/4 \div 2 = 1/8$
1.010	shifted right becomes	1.101	$-3/4 \div 2 = -3/8$

For the complementary system, the sign digit is propagated and preserved during the right shift.

To perform a shift in the Illiac, two registers are used. The digits of the number to be shifted are transferred from the first register to the second, and then back to the first register with a displacement of the digits. The steps used for shifting one digit one digital position to the right are indicated in Figure 2.41.

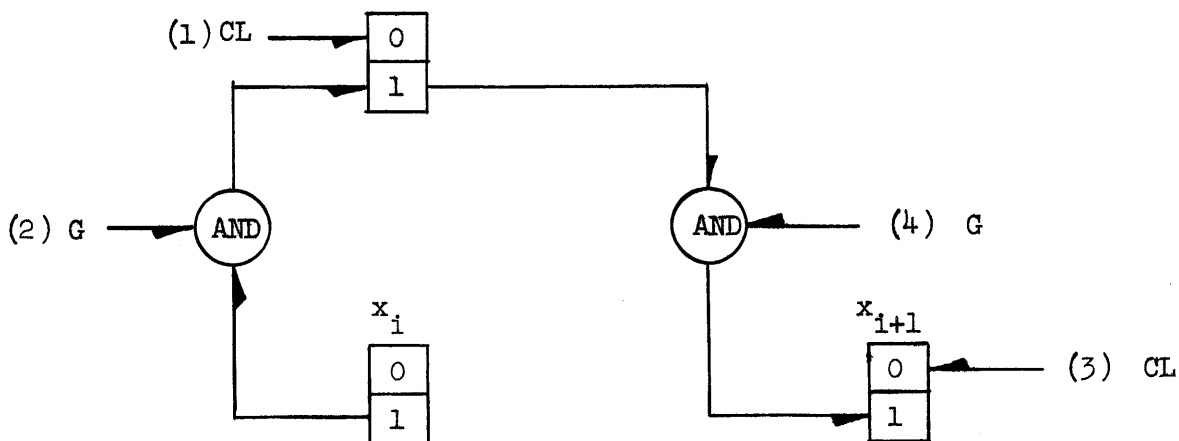


Figure 2.41 THE ILLIAC RIGHT SHIFT

The transfer of a digit from one flipflop to another requires two steps. As an example, to transfer from x_i to y_i (Figure 2.41), first clear y_i to 0, then gate a one to y_i if x_i is one. Electronically (Figure 2.1) the voltage supplied at point B is lowered to clear y_i to 0, then the cathode of a gate tube is lowered from +10 to -10v. If the grid of the gate tube is positive ($x_i = 1$) then the flipflop y_i will be set to 1. A similar transfer from y_i to x_{i+1} completes the shift.

The shifting process illustrates one design feature of the Illiac. During the shift, the digital information is always in one flipflop or another; in a transfer, the information reaches its destination before any change in the source is allowed to occur. The Illiac shift is to be contrasted with

the shifting method of Figure 2.42,

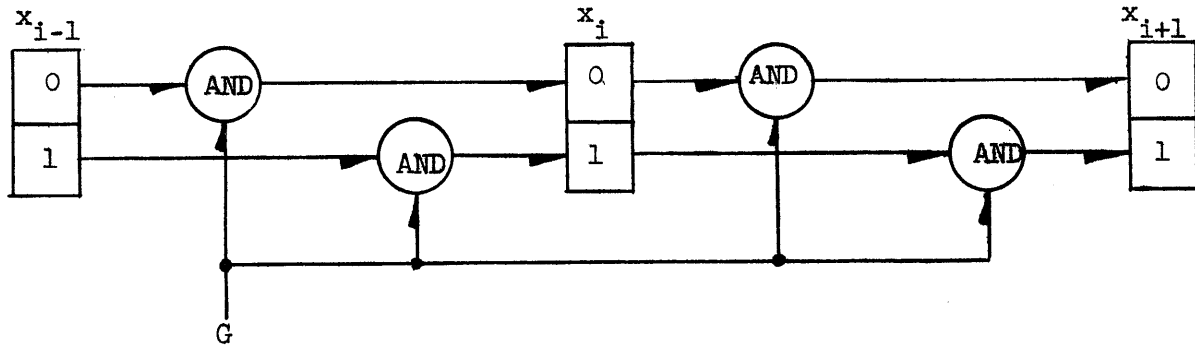


Figure 2.42 SHIFTING WITH A SINGLE REGISTER

which employs only one register. The duration of the gating signal G must be carefully controlled; if G is too short, no transfer will occur; if G is too long, a transfer over two digits may occur. Furthermore, the information is in transit; the original digital information in x_i must be on its way to x_{i+1} before the information from x_{i-1} arrives at x_i .

In the Illiac shift, each of the clear and gate signals have a minimum duration; but there is no limitation on their maximum duration. There is only the requirement that they not overlap timewise; this requirement is met by adding logical circuitry in the control.

2.5 Structure of the Illiac Arithmetic Unit

The Illiac arithmetic unit consists of five registers, an adder, and a complementing circuit, with perhaps a dozen gating circuits for number transfers and shifts. (Figure 2.5)

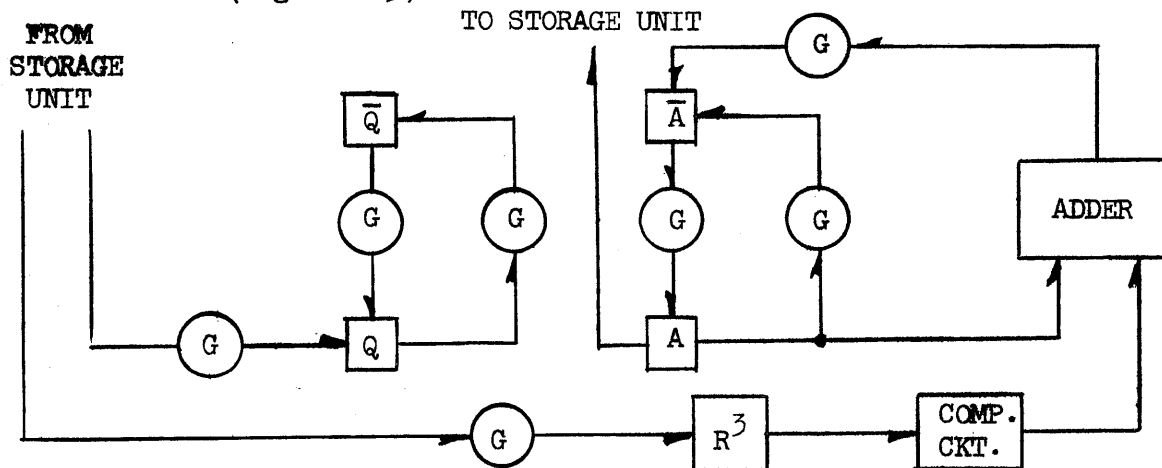


Figure 2.5 STRUCTURE OF ILLIAC ARITHMETIC UNIT

To perform an addition, the augend is initially in A. The addend is brought from storage to R^3 , the sum is formed by the adder and gated to \bar{A} . The sum in \bar{A} is then gated to A, replacing the augend. Subtraction is similar, except that the subtrahend is brought to R^3 and its complement is formed and added to the minuend in A.

For multiplication and division, shifting facilities are required in A and \bar{A} , and the multiplier-quotient registers Q, \bar{Q} with shifting facilities are also needed.

2.6 Illiac Multiplication

Multiplication is sequenced as a series of additions and right shifts, as illustrated in examples 2.61 and 2.62.

	0.111	7/8		1.001	-7/8		
	<u>0.101</u>	5/8		1.011	-5/8		
P_0	0.000		101	0.000		011	A, Q
	<u>0.111</u>			<u>1.001</u>			
	0.111			1.001			\bar{A}
P_1	0.011	1	10	1.100	1	01	A, Q
	<u>0.000</u>			<u>1.001</u>			
	0.011			0.101			\bar{A}
P_2	0.001	11	1	1.010	11	0	A, Q
	<u>0.111</u>			<u>0.000</u>			
	1.000			1.010			\bar{A}
P_3	0.100	011		1.101	011		A, Q
				<u>-1.001</u>			
				0.100	011		A, Q

Example 2.61

Example 2.62

At any one step a partial product is held in register A. A multiplier digit in the least significant digit of Q is sensed. If the multiplier digit is 1, the sum of the partial product in A and the multiplicand in R^3 is formed and transferred to \bar{A} . If the multiplier digit is 0, the partial product in A is transferred to \bar{A} . In either case, a right shift from \bar{A} to A follows, forming a new partial product. Simultaneously, a right shift is executed in Q, bringing the next multiplier digit into the least significant place. The gap in Q created by shifting the multiplier is filled by the digit of the final product formed during the step. The double length product is thus formed.

Two problems exist when the process is generalized for negative multiplier or multiplicand.

(1) The partial product p_i and multiplicand x lie in the range $-1 \leq p_i < 1$, $-1 \leq x < 1$. The sum in \bar{A} is therefore in the range $-2 \leq x + p_i < 2$. As a consequence, the sign digit of \bar{A} is not a true indication of the sign of $p_i + x$. The right shift to A will, however, form $1/2 (p_i + x)$ correctly if the proper sign digit is inserted. The solution of the problem is exceedingly simple. If p_i and x have the same sign, then that should be the sign of $1/2 (p_i + x)$. If p_i and x are of different sign, then $p_i + x$ is within the range $-1 \leq p_i + x < 1$ and the sign digit in \bar{A} is correct.

(2) It was previously noted that the Illiac representation and the algebraic value of a number are related by

$$y = -y_0 + \sum_{i=1}^{39} 2^{-i} y_i ,$$

where the sum \sum is the weighted sum of the non-sign digits. If the multiplier is positive (i.e., $y_0 = 0$), the process of multiplication described forms

$$x \left[\sum_{i=1}^{39} 2^{-i} y_i \right] = xy \text{ after 39 steps.}$$

If, however, the multiplier is negative

($y_0 = 1$) a correction consisting of the subtraction of the multiplicand x must be performed.

2.7 Division

The Illiac division is very similar to ordinary long division as taught in grammar school, except that binary numbers are used. Example 2.7 illustrates the process.

divisor = 5/8	0.011	quotient = 3/8
0.101	0.001,111	dividend = 15/64
	<u>-0.101</u>	
	-0.100	
	0.001	
	0.011	
	<u>-0.101</u>	
	-0.010	
	0.011	
	0.111	
	<u>-0.101</u>	
	0.010	
	0.101	
	<u>-0.101</u>	
	0.000	

Example 2.7 THE ILLIAC DIVISION

At each step, the divisor in R^3 is subtracted from the partial remainder in A. The sign digit of the difference appearing in the adder is sensed. If the difference is negative, the partial remainder in A is shifted left to form the new partial remainder and a 0 is inserted as a quotient digit. If the difference is positive, the difference of partial remainder and divisor is shifted left to form the new partial remainder and a 1 is inserted as quotient digit.

The process shown for positive numbers is easily generalized for all cases of signs of divisor and dividend.

The Illiac division method cannot be performed on most serial computers. Usually, the partial remainder is destroyed in the process of forming the difference between partial remainder and divisor. In the Illiac both the partial remainder and the difference are available after the subtraction, and one or the other is selected on the basis of the sign of the difference.

3.0 Asynchronous Sequencing Circuits

Each Illiac shift, each addition or subtraction, and each step of a multiplication or division requires the following control sequence:

- 1) Clear \bar{A}
- 2) Gate to \bar{A}
- 3) Clear A
- 4) Gate to A

A sequence of four steps can be executed with two flipflops, as shown in Figure 3.01.

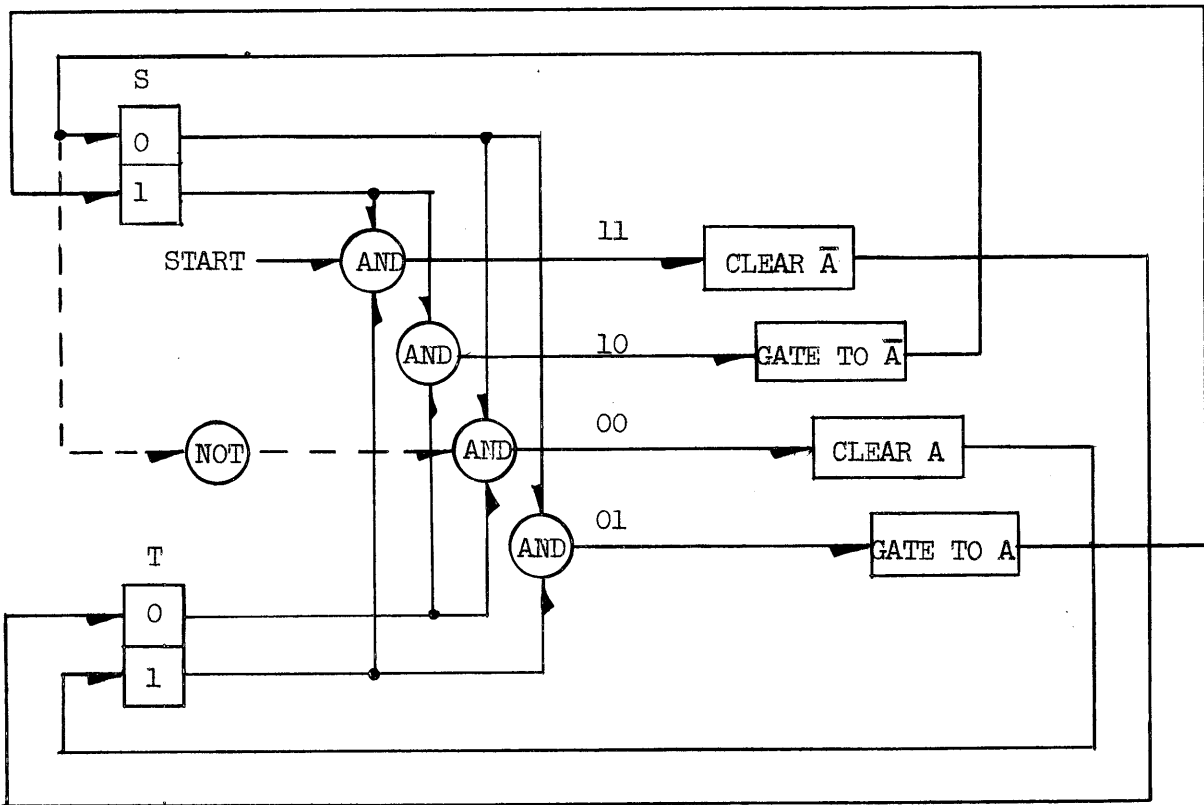


Figure 3.01 SEQUENCING OF FOUR STEPS

If the two flipflops are in the state $S = 1, T = 1$, and a start signal is given, the "Clear \bar{A} " signal will be enabled. The "Clear \bar{A} " signal also clears T to 0, thereby enabling the Gate to \bar{A} , which in turn gates flipflop S to 0. The four steps will be executed in sequence, repetitively, until the start signal is removed.

If it is desired that no two successive signals overlap, a "not" circuit can be added as shown by the dotted lines. Assume that the "Gate to \bar{A} " signal is present. Flipflop S will be gated to 0, disabling the "Gate to \bar{A} ". The disabling of the "Gate to \bar{A} " then enables, through the "not" circuit, the "Clear A " signal. Thus, there is a logical requirement that "Gate to \bar{A} " be off before "Clear A " is initiated.

For asynchronous control circuits, it is important that only one flipflop be turned over at a time; or if more than one is turned over, it must be immaterial which is turned over first.

Consideration of such restrictions leads to consideration of both Gray code sequences and of geometrical models as aids to control design.

The Gray code sequence is a means of numbering whereby all possible configurations of binary digits are exhausted, one digit being changed between successive numbers of the sequence. In Figure 3.0, for example, the sequence of states is

11 , 10 , 00 , 01 .

The Gray code sequence for three digits is:

000 001 011 010 110 111 101 100 .

A geometrical model can be set up with the following correspondences:

- 1) flipflop . . . dimension
- 2) state of n flipflops . . . vertex of n -dimensional cube
- 3) single flipflop change . . . edge of n -dimensional cube

The three-dimensional model for a sequence of states of three flipflops corresponding to the Gray code sequence is shown in Figure 3.02.

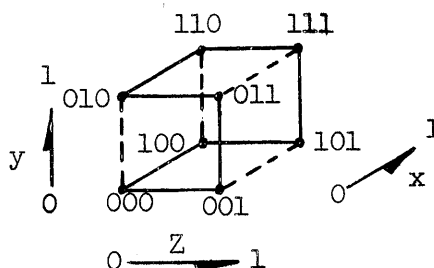


Figure 3.02 GEOMETRICAL MODEL FOR THREE FLIPFLOPS

The geometrical model can be of considerable assistance in asynchronous control design.

3.1 Counting Circuits

Conventional binary counting is unexpectedly difficult when asynchronous circuits are employed. Consider the circuit shown in Figure 3.1.

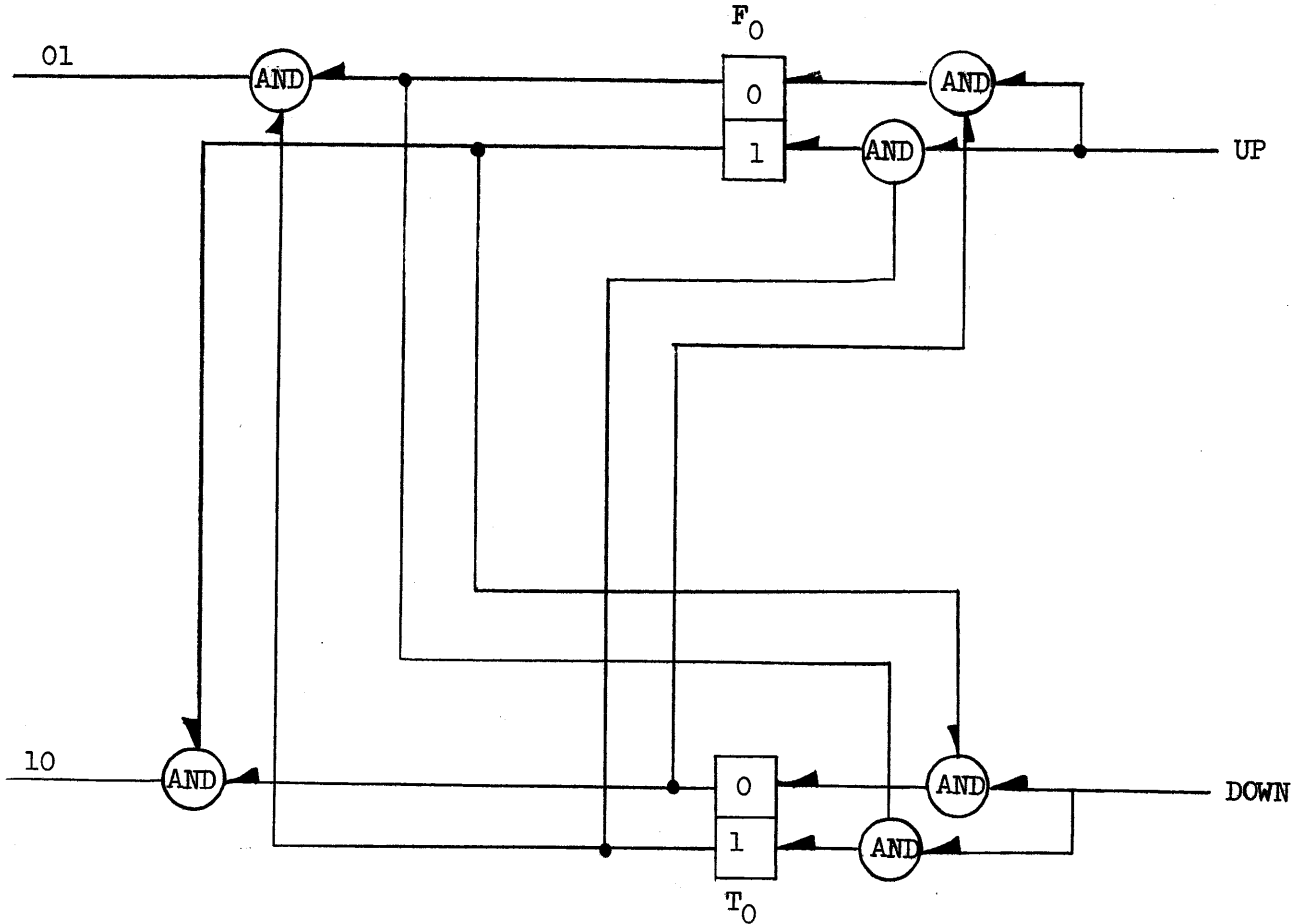


Figure 3.1 ONE STAGE OF BINARY COUNTER

The "UP" pulse sets flipflop F_0 to agree with T_0 , the "DOWN" pulse sets flipflop T_0 to disagree with F_0 . If we connect the 01 and 10 output signals as up and

down inputs to a similar circuit with flipflops F_1 and T_1 , the flipflops will follow the sequence of Table 3.1.

T_1	T_0	F_1	T_1	F_0	T_0	Input Pulse
0	0	1	0	1	0	Up
				0	0	Down
0	1	0	0	0	1	Up
				1	1	Down
1	0	0	1	1	0	Up
				0	0	Down
1	1	1	1	0	1	Up
				1	1	Down
0	0	1	0	1	0	Up
				0	0	

Table 3.1 SEQUENCE OF STATES IN AN ASYNCHRONOUS BINARY COUNTER

It is to be noted that for successive pairs of input pulses, the states of T_1 and T_0 are those of a conventional binary counter.

It the leads marked 01 and 10 are reversed between successive stages in the circuit of Figure 3.1, the counter counts downwards, rather than upwards.

4.0 Summary

The Illiac is a binary parallel asynchronous computer in which negative numbers are represented as two's complements. As in all computers, the major characteristics of the Illiac have a profound influence on the structural details.

The choice of the two's complement method of negative number representation simplifies the processes of addition and subtraction and causes some complication for multiplication. The handling of digits in parallel makes possible higher

computational speeds at the expense of additional circuitry in the arithmetic unit.

The direct-coupled asynchronous control of the Illiac permits wide variation in the duration of steps necessary for arithmetic operations; in theory no upper limit on the duration of any step is required. The shifting and counting operations are complicated by adherence to the asynchronous principle; on the other hand, diverse and sometimes unexpected benefits are obtained. A relatively simple method of division and ease in trouble-shooting are examples.

References:

1. "Preliminary Discussion of the Logical Design of an Electronic Computing Instrument", by Burks, Goldstine, and von Neumann.
2. "Arithmetic Operations in a Binary Computer", by R. F. Shaw; Review of Scientific Instruments, Volume 21; August, 1950, pp. 687-693.
3. "The Ordvac", by R. E. Meagher and J. P. Nash; Review of Electronic Computers, (Joint AIEE-IRE Computer Conference) published by AIEE, February, 1952.
4. "The Logical Principles of a New Kind of Binary Counter" by Willis Ware; Proceedings of the IRE, Volume 41, No. 10; October, 1953.

May 23, 1955

mpf