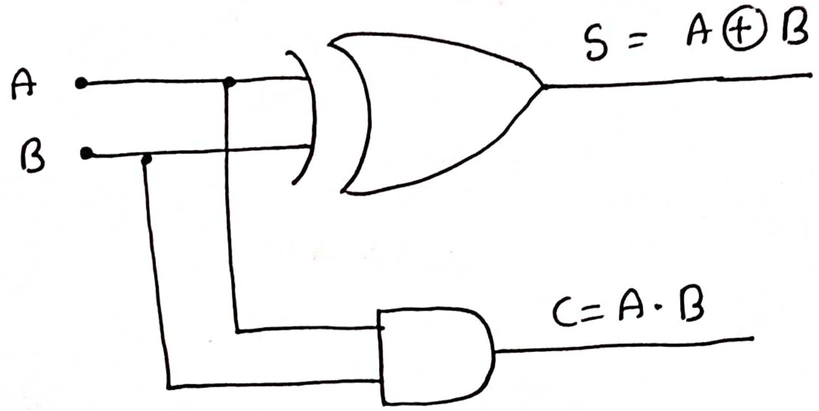
Monika Sharma

UNIT - 2

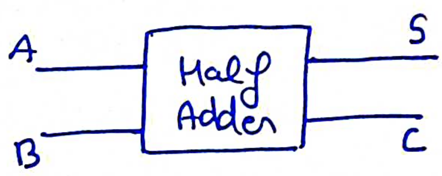
ARITHMETIC AND LOGIC UNIT

HALF ADDER:- A combinational circuit that Performs the addition of two bits is called a Half Adder.



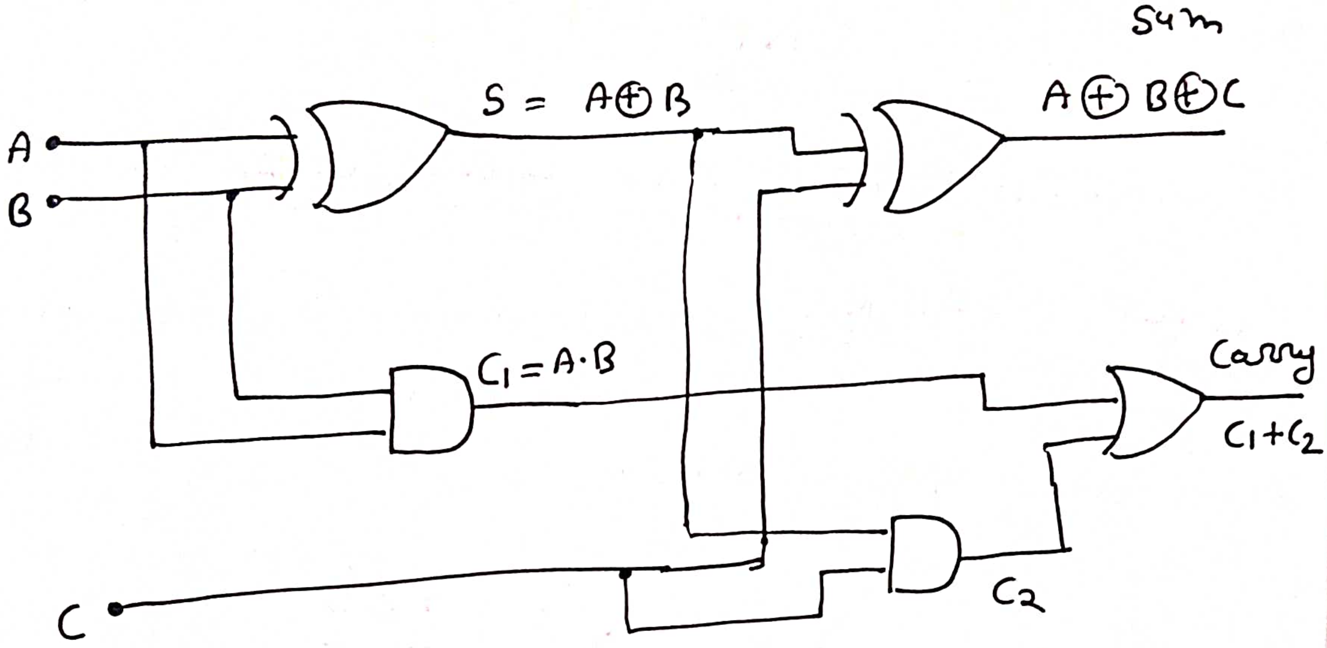
Half Adder truth table:-

A	B	S = (A ⊕ B) sum	C = A · B carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



FULL ADDER:- A combinational circuit that Performs the addition of three bits two significant bits and a carry bit is called a full Adder.

M.S



Truth table of full Adder :-

Input			Output	
A	B	C	Sum - S	Carry - C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

PARALLEL BINARY ADDER:-

In the most of Logic circuit addition of more than 1-bit is carried out. The Addition of multibit numbers can be performed using several full adder.

The sum of 4-bit number can be done using 4-full Adders as shown in fig

Let two 4-bit numbers A & B

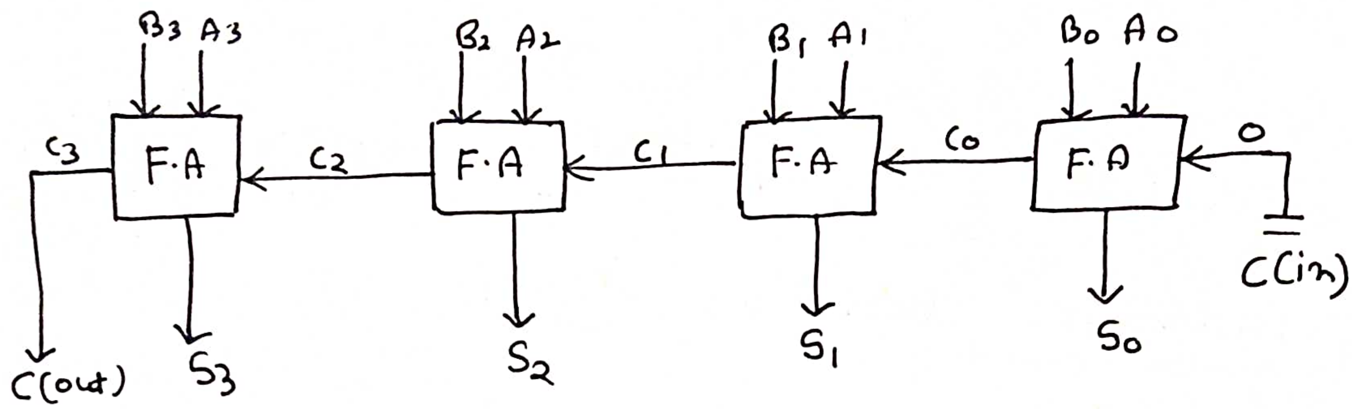
A = 1111
B = 0011

A_3 A_2 A_1 A_0 = 1111
B_3 B_2 B_1 B_0 = +0011

1 0010

4-Bit Binary Parallel Adder (Ripple carry Adder):-

M.S



⇒ Since all the bits of the two numbers are fed into the adder circuit simultaneously and the addition in each Position is taking place at the same time therefore this circuit is known as Parallel Adder.

⇒ This type of Adder is called ripple carry Adder. Because the output carry of lower stage is connected to the input carry of the next Higher stage.

⇒ It is speed of operation is limited by the carry Propagation delay through all stage.

⇒ to overcome carry Propagation delay a new adder is used which is known as carry look ahead (CLA) Adder.

M.S

$G_i \Rightarrow$ Carry generate $\Rightarrow$ It produces carry

$P_i \Rightarrow$ Carry Propagate $\Rightarrow$ Propagation of the carry from C_i to C_{i+1}

The carry output of each stage:

$$\boxed{C_{i+1} = G_i + P_i \cdot C_i} \longrightarrow ①$$

for $i = 0$

$$C_1 = G_0 + P_0 \cdot C_0$$

$$\boxed{C_1 = A_0 B_0 + (A_0 \oplus B_0) \cdot C_0}$$ All known terms

for $i = 1$

$$C_2 = G_1 + P_1 \cdot C_1$$

$$C_2 = G_1 + P_1 (G_0 + P_0 \cdot C_0)$$

$$C_2 = G_1 + P_1 G_0 + P_1 P_0 \cdot C_0$$

$$\boxed{C_2 = (A_1 \cdot B_1) + (A_1 \oplus B_1) \cdot (A_0 B_0 + A_0 \oplus B_0) \cdot C_0}$$

for $i = 2$

$$C_3 = G_2 + P_2 \cdot C_2$$

$$C_3 = (A_2 \cdot B_2) + (A_2 \oplus B_2) \cdot C_2$$

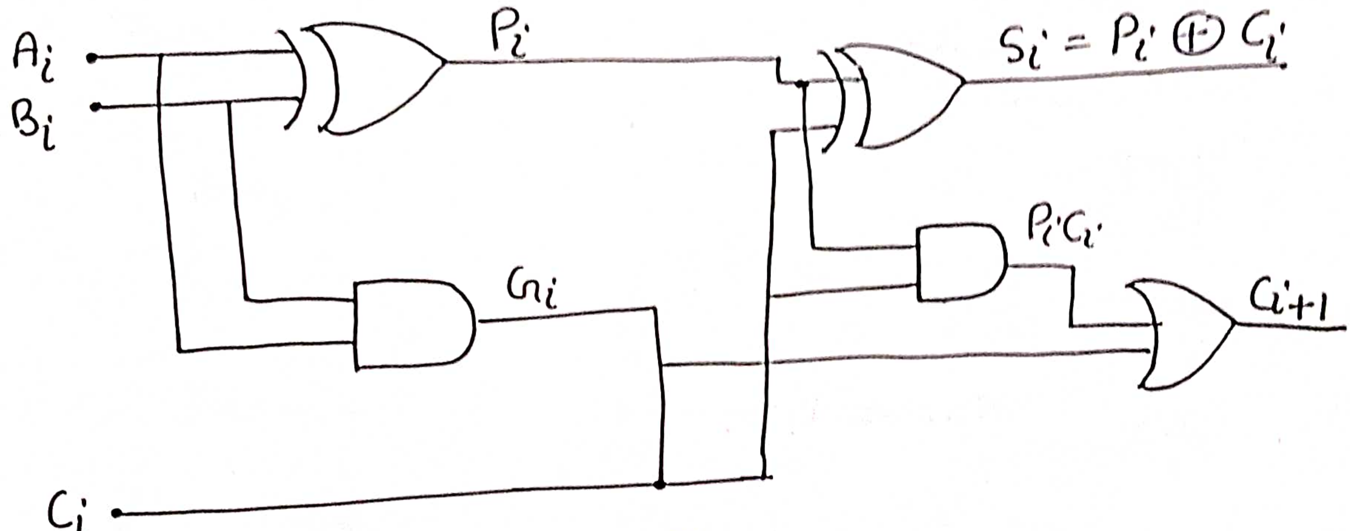
M.S

Carry Look Ahead Adder: It is based on the principle of looking at the lower order bits of the inputs if a higher order carry is generated.

⇒ two functions:

(1) Carry Propagation (P_i)

(2) Carry generation (G_i)



$$P_i = A_i \oplus B_i$$

$$G_i = A_i \cdot B_i$$

Sum $\quad S_i = P_i \oplus C_i$.

carry $\quad C_{i+1} = G_i + P_i C_i$

where:-

M.S

$$\boxed{C_3 = G_2 + P_2\ (G_1 + P_1\ (G_0 + P_0\ (C_0)))}$$

$$\boxed{C_3 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0}$$

C_3 does not wait for C_2 & C_1 to Propagate.

C_3 is estimated at the same time as C_2 & C_3 are estimated.

The sum output of each stage:

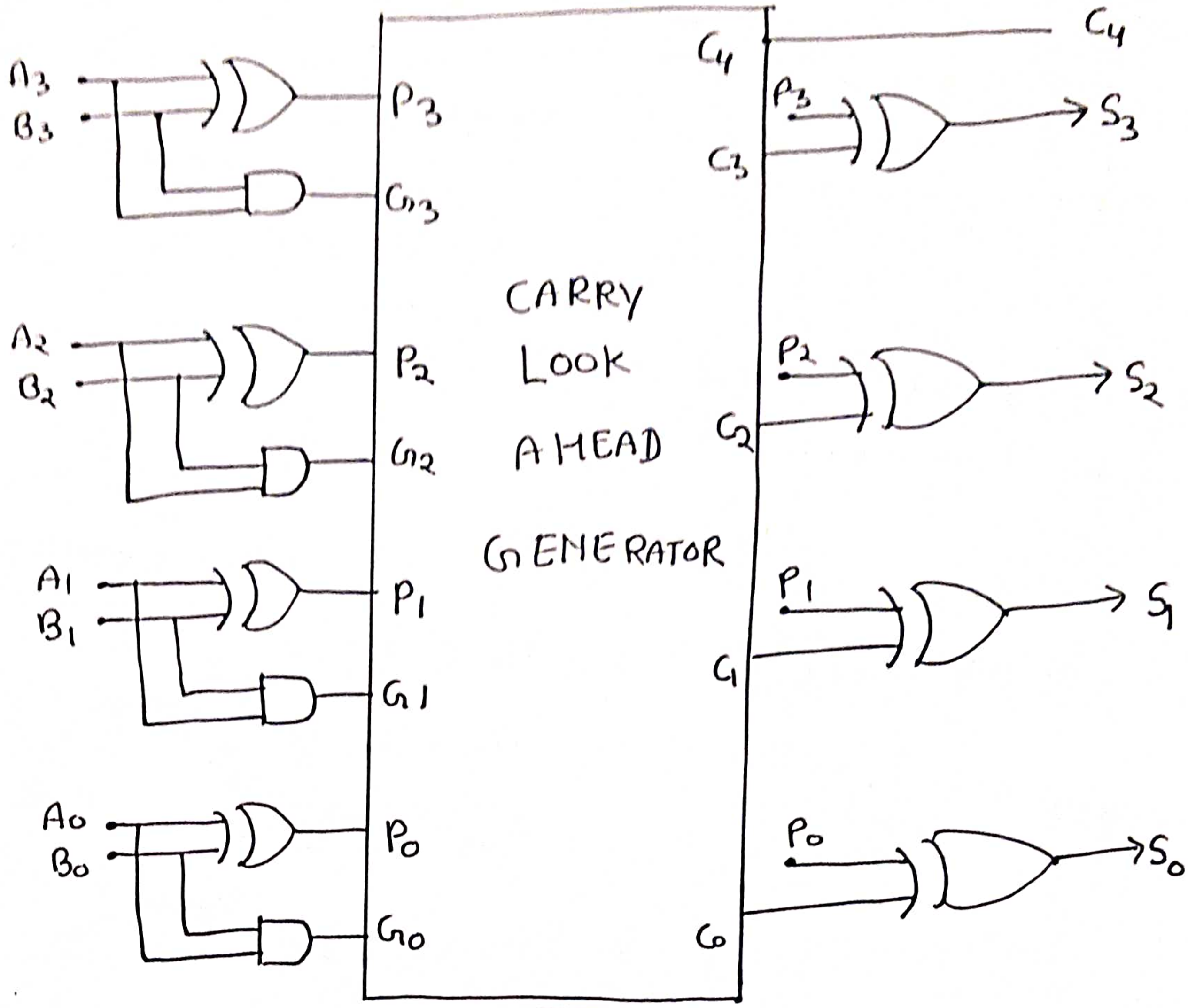
$S_0 = P_0 \oplus C_0 \Rightarrow A_0 \oplus B_0 \oplus C_0$

$S_1 = P_1 \oplus C_1 \Rightarrow A_1 \oplus B_1 \oplus C_1$

$S_2 = P_2 \oplus C_2 \Rightarrow A_2 \oplus B_2 \oplus C_2$

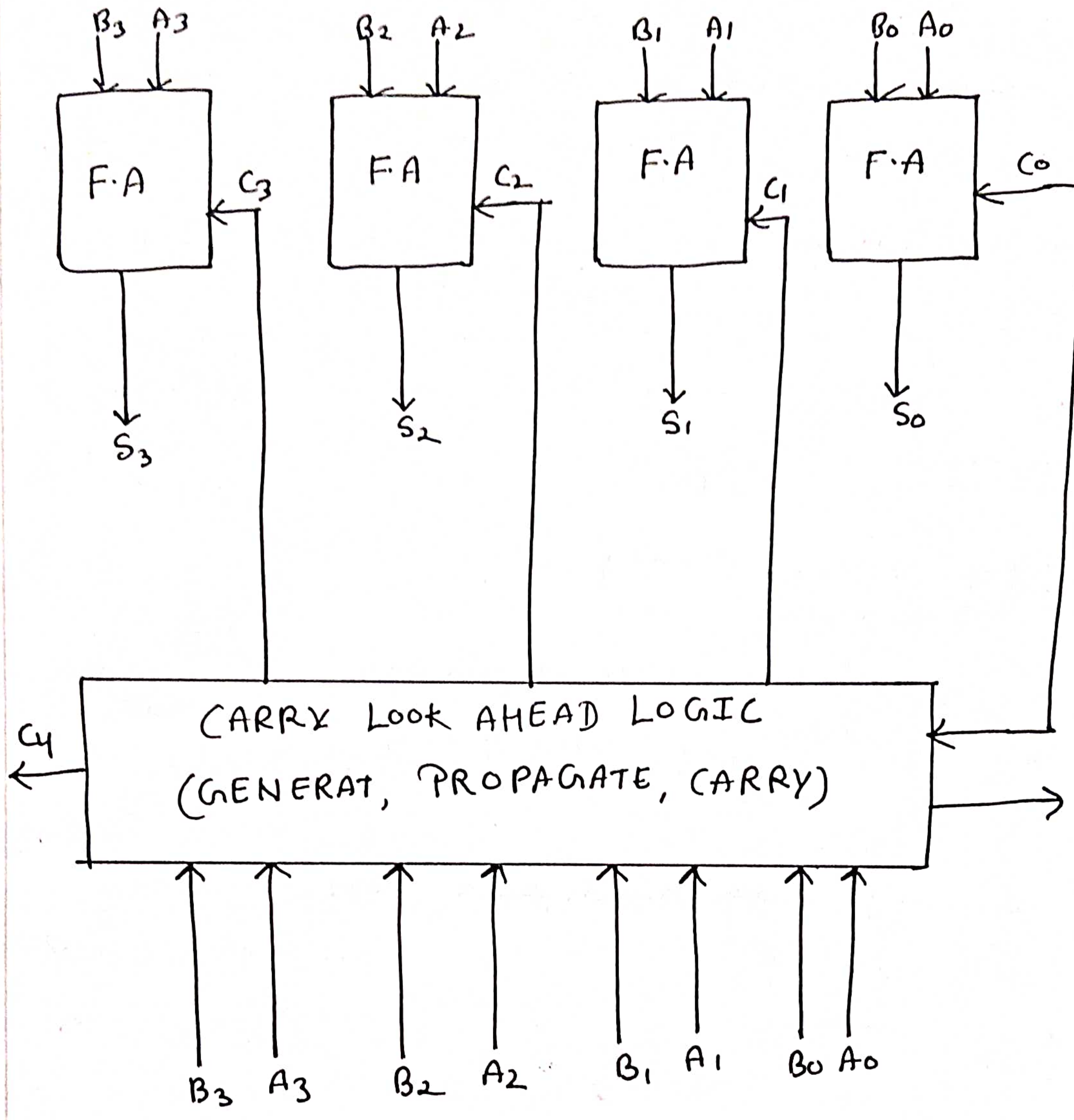
$S_3 = P_3 \oplus C_3 \Rightarrow A_3 \oplus B_3 \oplus C_3$

Logic Diagram carry Look Ahead Adder:-



(Logic Diagram carry lookahead Adder)

Special Look Ahead carry generator CKT :-



(Special Look AHEAD carry generator CKT)

ARRAY MULTIplier :- Array multiplier has regular structure. multiplier circuit is based on add and shift algorithm. Each Partial Product in generated by multiplication of multiplicand with one multiplier bit. The Partial Product are shifted according to their bit orders and then added.

The addition can be Performed with normal carry Propagate adder.

N - 1 adders are required where N is multiplier length.

4x4 Array multiplier:-

$x \times y$ $x \Rightarrow$ multiplicand

$y \Rightarrow$ multiplier

4x4 $\Rightarrow$ Array multiplier $\Rightarrow$ Hardware circuit for fast multiplication

x $\downarrow$ 4-bit

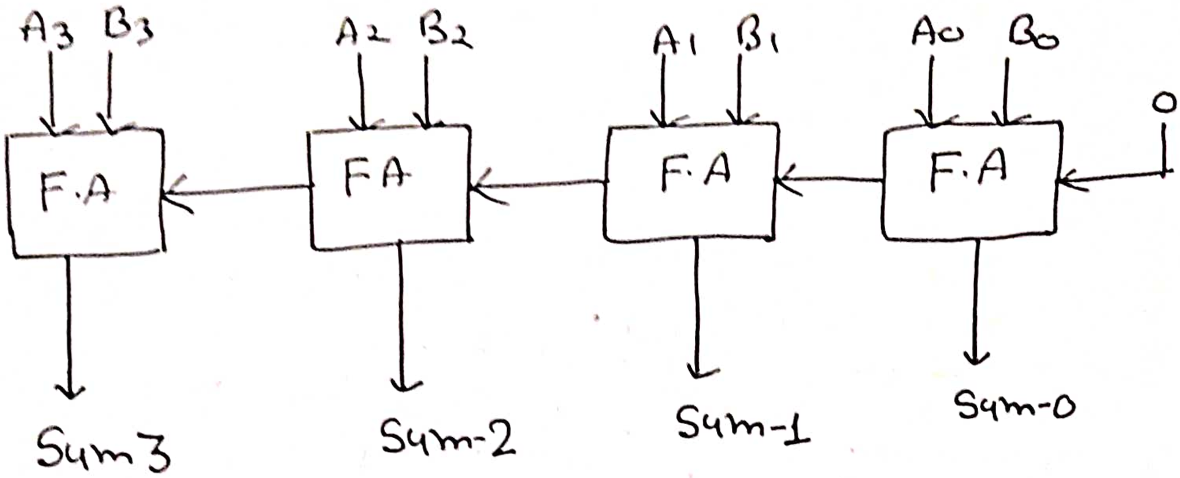
y $\downarrow$ 4-bit

(full Adder / Half Adder / AND ~~gate~~ gate)

Carry Propagate Adder:-

connecting full-adders to make a multi-bit carry Propagate adder.

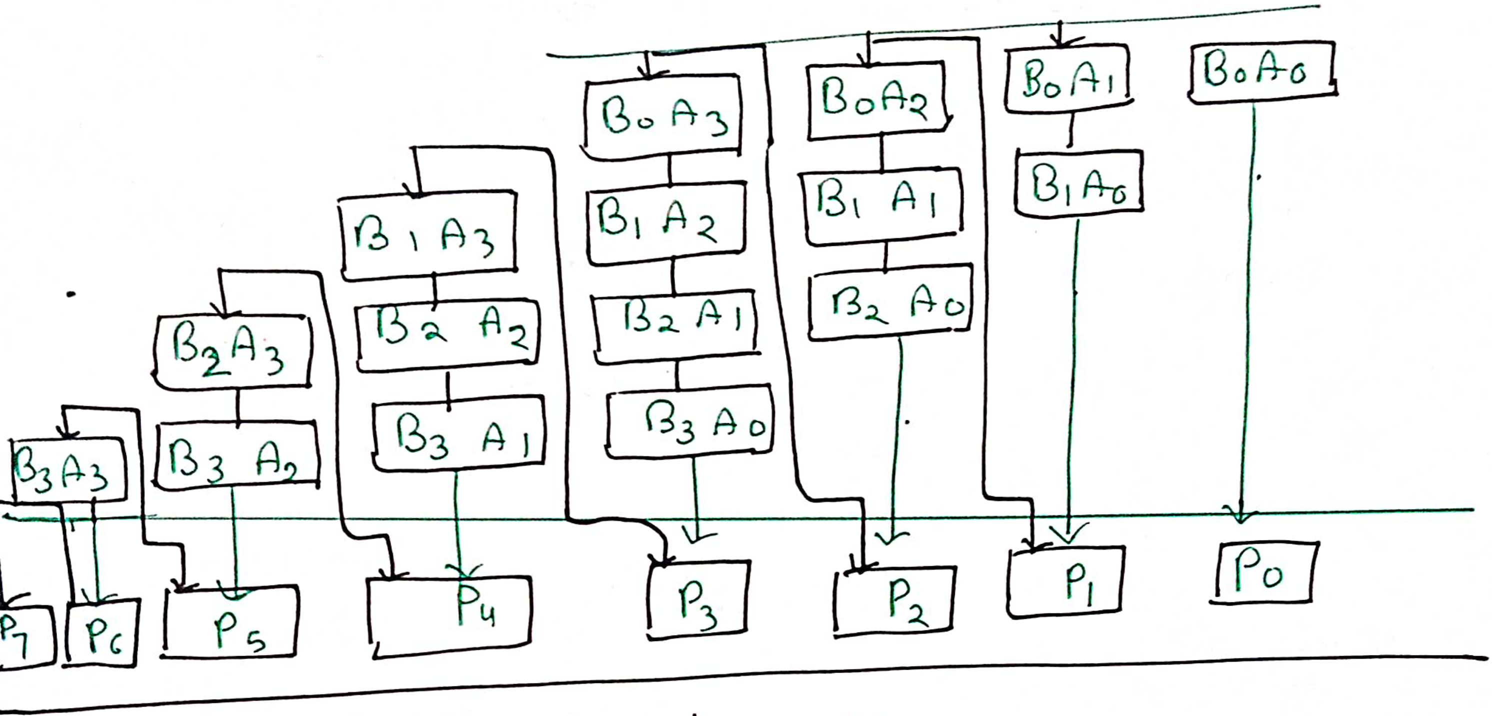
M.S



$x = x_3 x_2 x_1 x_0 \quad y = y_3 y_2 y_1 y_0$

$$x \times y = \begin{array}{cccccccc}
 & & & & x_3 & x_2 & x_1 & x_0 \\
 & & & \times & y_3 & y_2 & y_1 & y_0 \\
\hline
 & & & & x_3y_0 & x_2y_0 & x_1y_0 & x_0y_0 \\
 & & & x_3y_1 & x_2y_1 & x_1y_1 & x_0y_1 & \\
 & & x_3y_2 & x_2y_2 & x_1y_2 & x_0y_2 & & \\
 & x_3y_3 & x_2y_3 & x_1y_3 & x_0y_3 & & & \\
\hline
C & S_6 & S_5 & S_4 & S_3 & S_2 & S_1 & S_0
\end{array}$$

$$\begin{array}{r} A_3\ A_2\ A_1\ A_0 \\ \times\ B_3\ B_2\ B_1\ B_0 \end{array}$$



Hardware Requirement

16 – AND Gate

4 – Half Adder

8 – full Adeer

total 12 – Adders

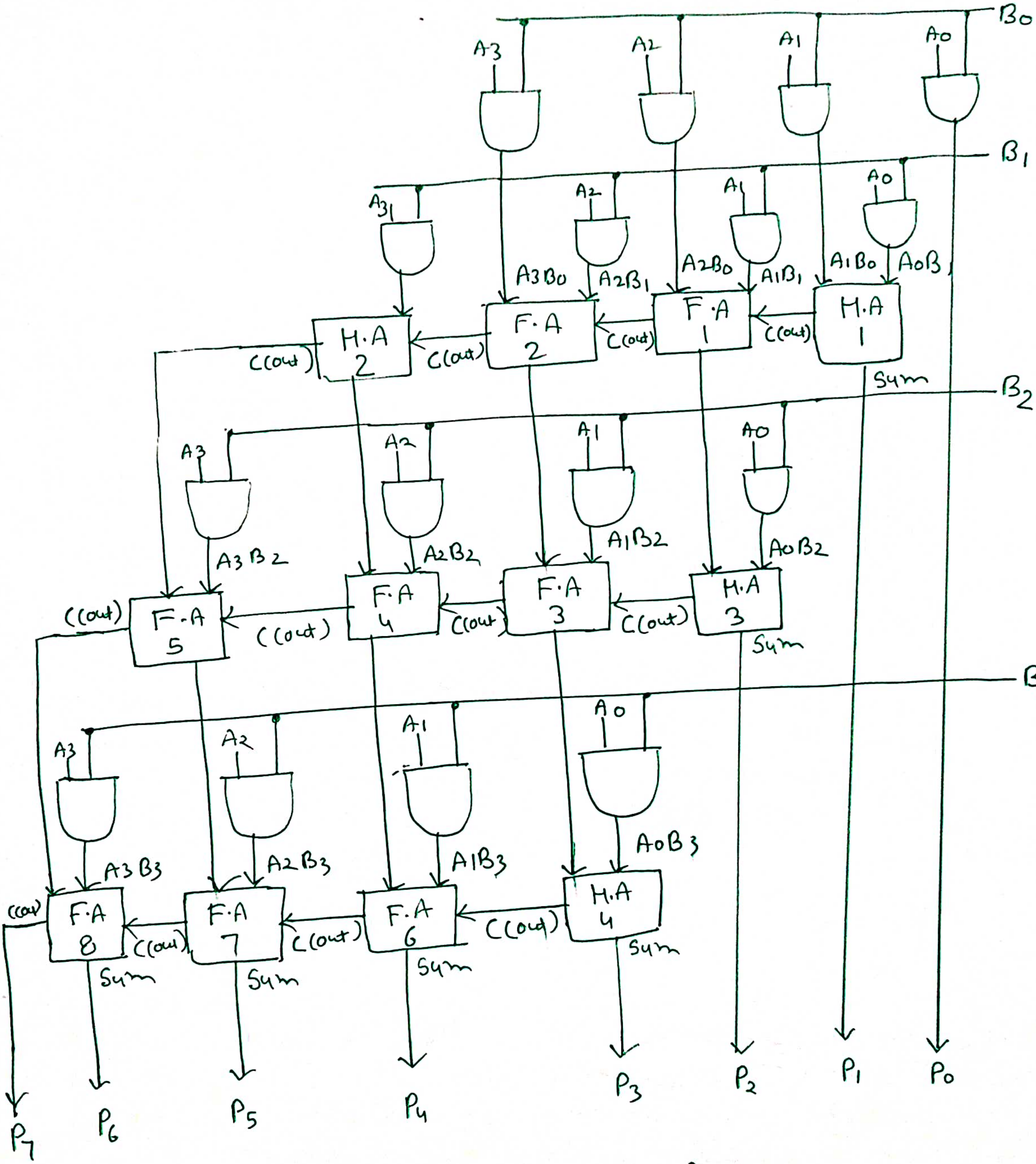
$m \times n$ = $m \Rightarrow 4$ (No of Bits in A)

$n \Rightarrow 4$ (No of Bits in B)

$m \times n$ = AND Gates

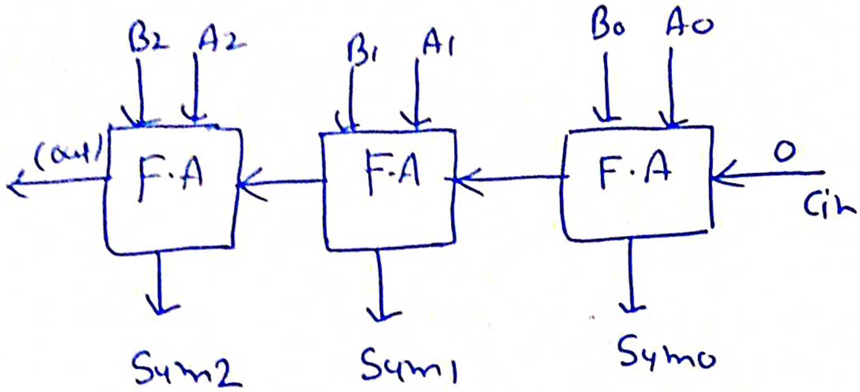
n = No of H.A (Half Adder)

$m(n-2)$ = No of F.A (full Adder)



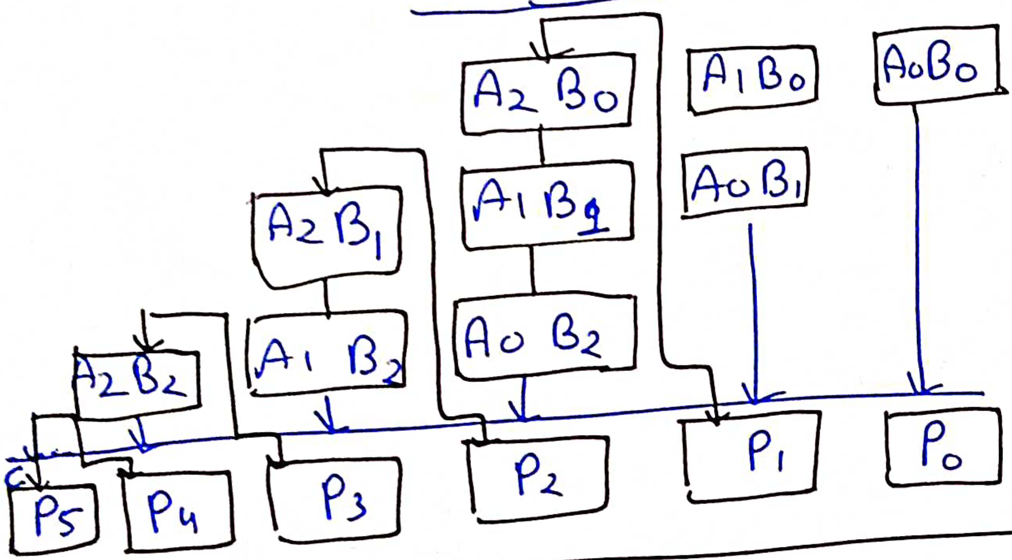
(4X4 Array multiplier)

3 - bit Binary multiplier:-

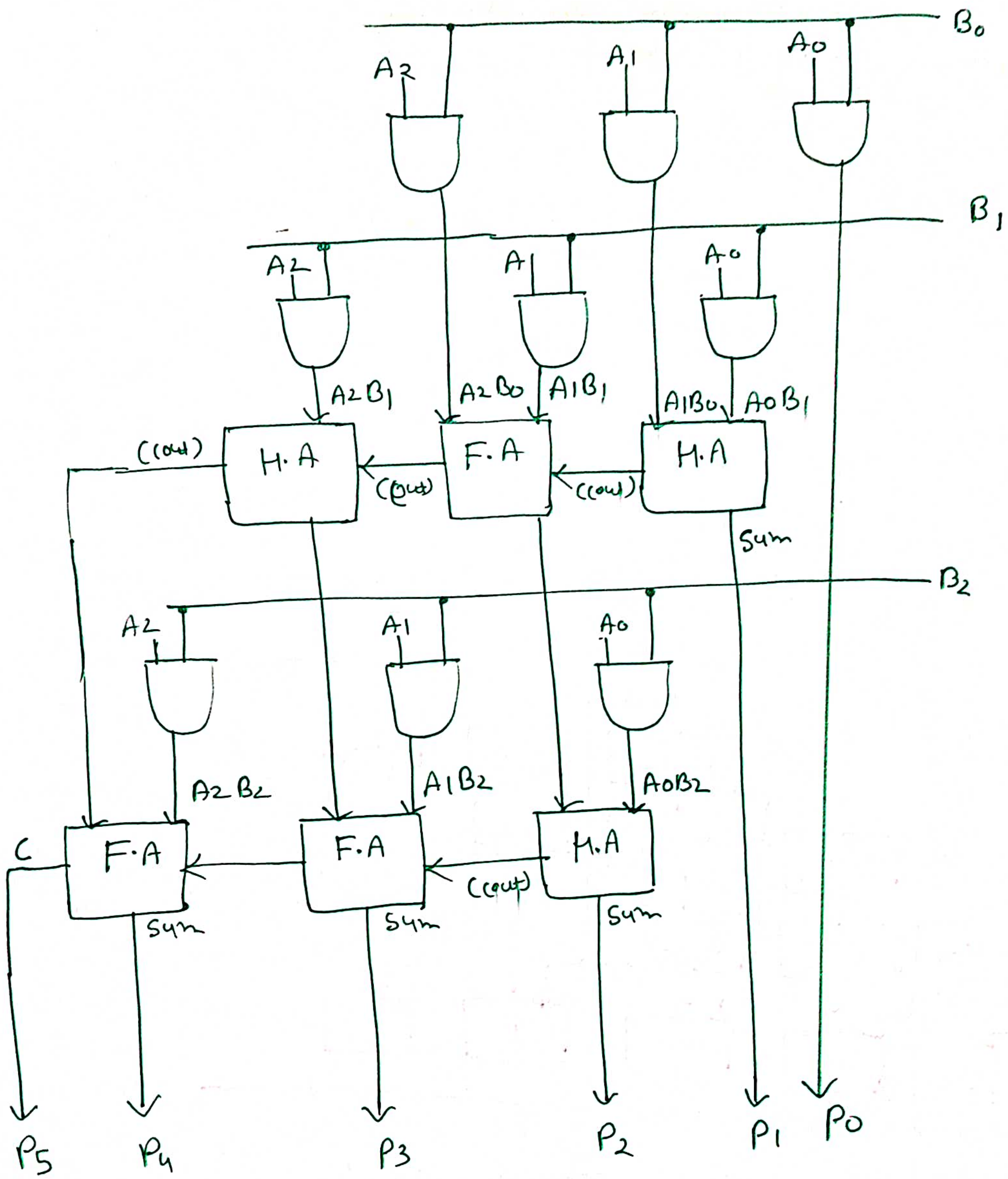


$A = A_2 A_1 A_0 \qquad B = B_2 B_1 B_0$

$$A \times B = \begin{array}{r} A_2 \; A_1 \; A_0 \\ \times \; B_2 \; B_1 \; B_0 \\ \hline \end{array}$$



(3x3) Array multiplier:-



Hardware Requirement

N- bit multiplication

AND Gates $\Rightarrow N^2 = 9$

Half Adder $\Rightarrow N = 3$

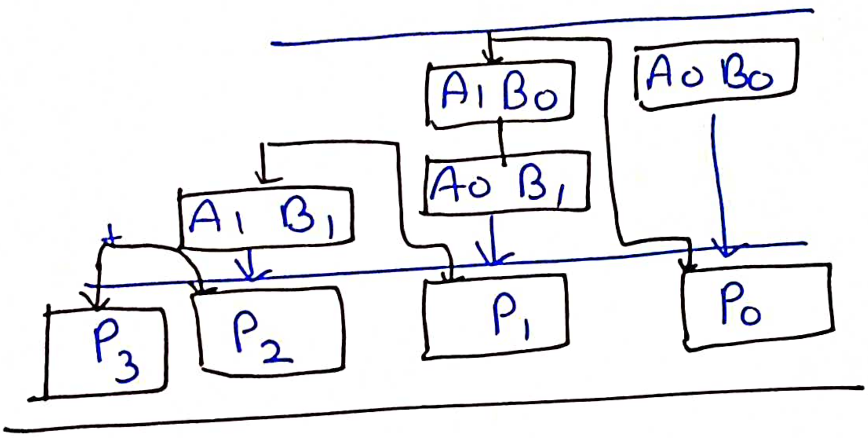
Full Adder $= N(N-2) = 3$

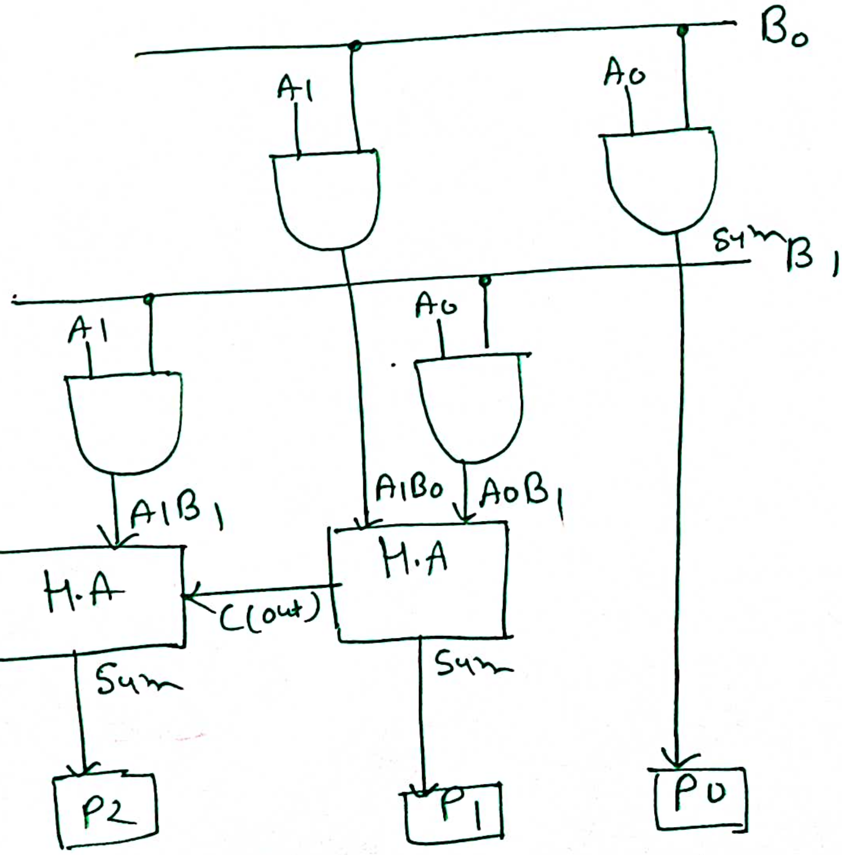
2- bit Binary Array multiplier:-

$A = A_1 A_0 \qquad B = B_1 B_0$

$A \times B = \cancel{A_2}\, A$

$$\begin{array}{rcc} & A_1 & A_0 \\ \times & B_1 & B_0 \end{array}$$





(2x2 Array multiplier)

BOOTHS ALGORITHM

Booths multiplication Algorithm:-

multiplication of signed 2's complement integers:-

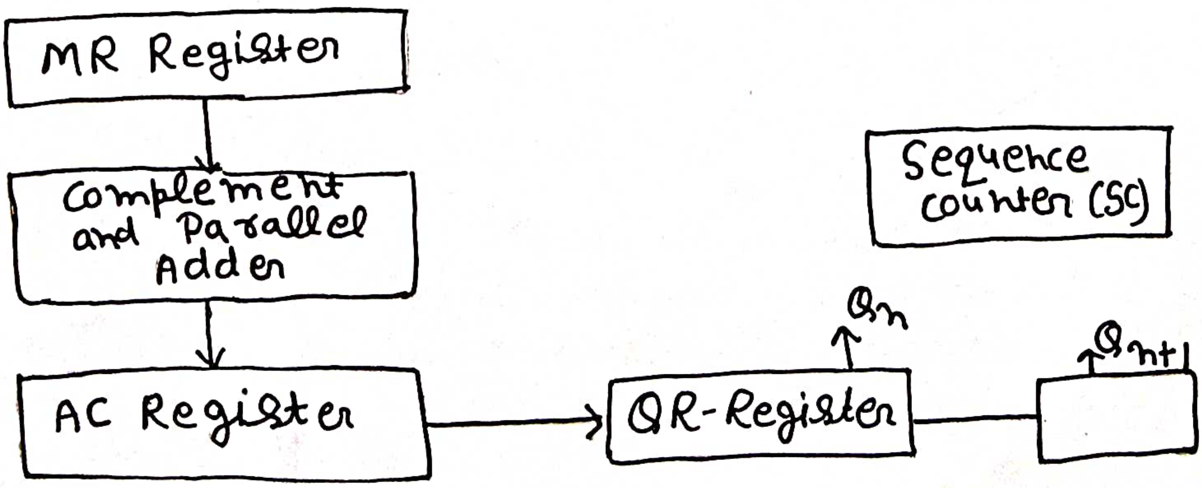
Shifting and addition / Subtraction rules for multiplicand in Booths Algorithm.

1. The multiplicand is subtracted from the Partial Product upon encountering the first least significand 1 in a string of 1's in a multiplier.

2. The multiplicand is added to the Partial Product upon encountering the first 0 (Provided that there was a Previous 1) in a string of 1's in the multiplier.

3. The Partial Product does not change when the multiplier bit is identical to the Previous multiplier bit.

Hardware Implementation for Booth Algorithm:-



M.S

⇒ Here sign is not separated from register.

⇒ QR Register contains the multiplier registers and Q.

⇒ Q_n represent LSB of the multiplier in QR.

⇒ Q_{n+1} is an extra flip-flop appended to QR to facilitate a double bit inspation of the multiplier.

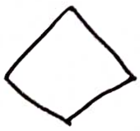
⇒ AC Register and Q_{n+1} are initially cleared to 0

⇒ SC is a set to the number of Bits in the multiplier.

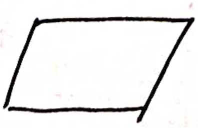
⇒ Q_n Q_{n+1} are two Successive bits in the multiplier.

FLOW CHART for multiplication of signed 2's complement numbers

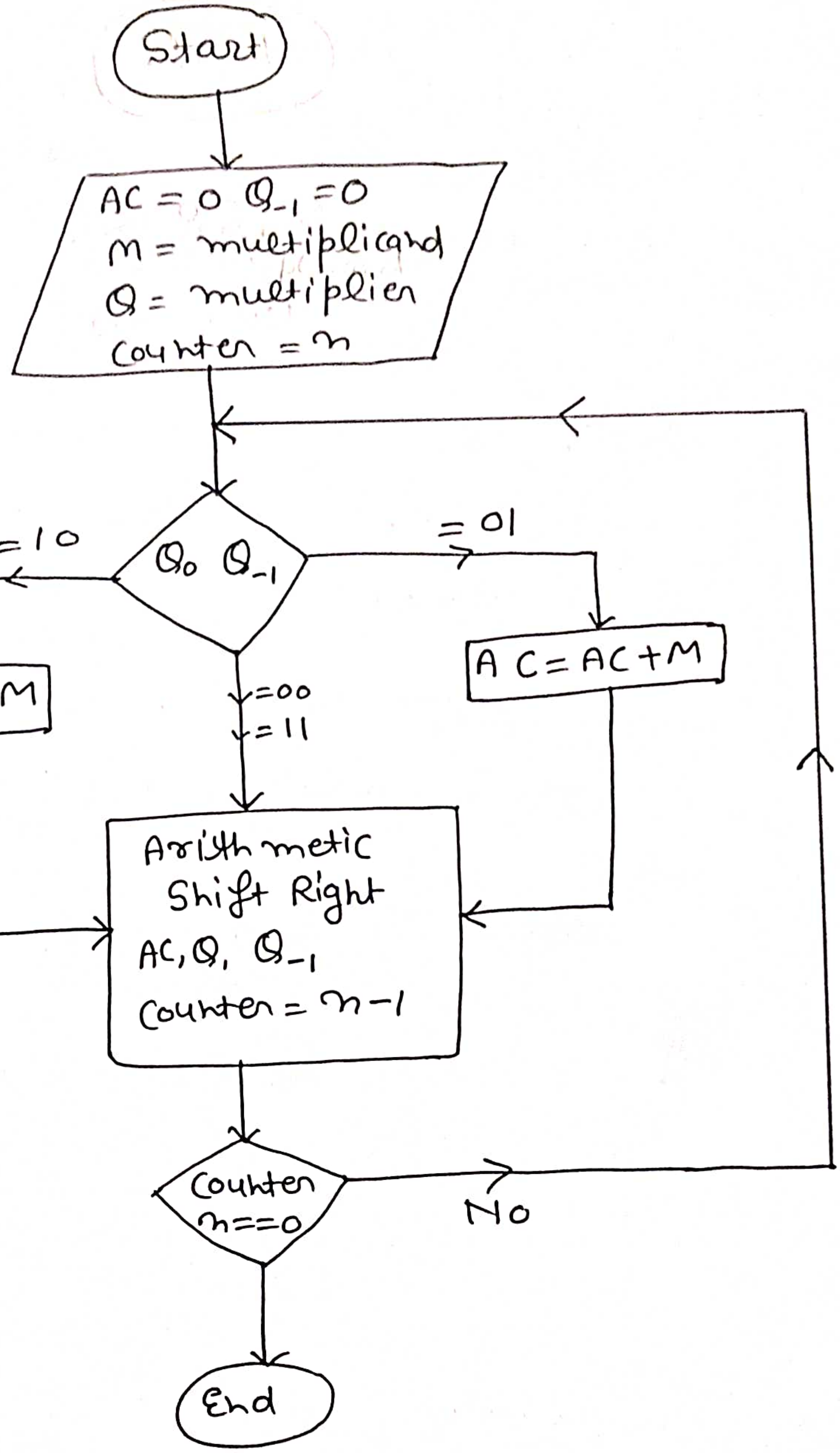
FLOW CHART FOR BOOTHS MULTIPLICATION :-

 ⇒ Decision box or Condition box

 ⇒ Start

 ⇒ Processing box

M.S



For signed number multiplications three cases is Possible.

(1) + +

(2) + −

(3) − −

M.S

NOTE:- if n (Register bit) value is not given M+1 Add one bit in multiplicand otherwise n value is depend given the Number of bit in the Questions.

Q1 +7 X +3 Using Booths multiplications.

Sol $M = 7 \quad Q = 3$

$M = 0111$ $\qquad$ n = number of bit in M+1

$Q = 0011$

1's complement of M = 1000

2's complement of (−M) = $+ \quad 1$

$\qquad 1001$

$(-M) = (1001)_2$

$AC = 0000 \quad Q_{-1} = 0 \quad n = 4$

Step-1:-

AC	Q	Q_{-1}	operation
0000	$Q_3Q_2Q_1Q_0$ 0011	0	Initially
0000	0011	0	AC = AC − M

$\underbrace{1 \quad 0}_{10}$

AC = 0000
$\quad$ 1001
$\quad$ ————
$\quad$ 1001

M.S

1001 0011 0 → Ignore ASR

1100 1001 1

1100 1001 1 $n = n-1 = 4-1 = 3$

Step-2:

AC	Q	Q_{-1}	Operation
1100	1001	1	ASR

11

1100 1001 1 → Ignore ASR

1110 0100 1

1110 0100 1 $n = 3-1 = 2$

Step 3:

AC	Q	Q_{-1}	operations
1110	0100	1	AC = AC+M

01

AC = 1110
  0111
  -----
  1 0101

1 ↓ Discard cary

0101 0100 1 → Ignore ASR

0010 1010 0

0010 1010 0 $n = 2-1 = 1$

M.S

Step-4:

AC	Q	Q_{-1}	operation
0010	1010	0	ASR
	00		
0010	1010	0 → Ignore	ASR
0001	0101	0	
0001	0101	0	$n = 1-1$ $= 0$

$\Rightarrow$ AC. Q

$\Rightarrow$ 0001 0101

merge

00010101

$\Rightarrow (21)_{10}$

$(+7) \times (+3) = (21)_{10}$ Ans

Q2 $(-7) \times (+3)$ using Booths multiplication
$n = 5$ bit.

Sol $M = -7$ $\quad Q = 3$

$M = (-7) = 00111$

$Q = (3) = 00011$

$-M = 00111$

M.S

1's complement of M = 11000

$$
\begin{array}{lr}
\text{2's complement of M} = & + \quad\quad 1 \\
\hline
 & 11001
\end{array}
$$

$-(-M) = 11001$

$M = (11001)_2$

$Q = (00011)_2$

$AC = 00000 \quad Q_{-1} = 0$

Step 1:-

AC	Q	Q_{-1}	operation
00000	$Q_4Q_3Q_2Q_1Q_0$ 00011	0	Initially
	10		
00000	00011	0	AC= AC+(-M)
			AC= 00000 + 00111 00111
00111	00011	0 → Ignor	ASR
00011	10001	1	ASR
00011	10001	1	$n = n-1$ $= 5-1$ $n = 4$

Step 2!

AC	Q	Q_{-1}	operation
00011	10001	1	ASR
	11		

M.S

00011	10001	1 → Ignore	ASR
00001	11000	1	
00001	11000	1	

$n = n-1$
$= 4-1 = 3$
$n = 3$

Step-3

AC	Q	Q_{-1}	operation
00001	11000	1	AC = AC + M

01

AC = 00001
  11001

  11010

11010	11000	1 → Ignore	ASR
11101	01100	0	
11101	01100	0	

$n = n-1$
$n = 3-1$
$n = 2$

Step-4:

AC	Q	Q_{-1}	operation
11101	01100	0	ASR

00

11101	01100	0 → Ignore	ASR
11110	10110	0	ASR
11110	10110	0	

$n = n-1$
$n = 2-1$
$n = 1$

$\boxed{M.S}$

Monika Sharma

Step-5

AC	Q	Q_{-1}	operation
11110	10110	0	ASR
	00		
11110	10110	0 → Ignore	ASR
11111	01011	0	
11111	01011	0	$n = n-1$ $n = 1-1$ $n = 0$

⇒ AC, Q

11111 01011

merge

P ⇒ $(1111101011)_2$

1's complement of P = 0000010100

2's complement of P = + 1

0000010101

$-P = (10101)_2$

$-P = 21$

$(-7) \times (+3) = (-21)_{10} = (10101)$ Ans

M.S

Q3 -7×-3 using Booths multiplication. Register bit is 4.

$\Rightarrow$ (-ve) x (-ve) $\Rightarrow$ + (Product)

Sol

$$M = (-7)_{10} \Rightarrow (0111)_2$$

1's complement of M = 1000

2's complement of (-M) = $\begin{array}{r} 1000 \\ + \quad 1 \\ \hline 1001 \end{array}$

$$(-M) = (1001)_2$$

$$M = (-7)_{10} = (1001)_2$$

$$-M = (7)_{10} = (0111)_2$$

$$Q = (-3)_{10} = (0011)_2$$

1's complement of Q = 1100

2's complement of (-Q) = $\begin{array}{r} 1100 \\ + \quad 1 \\ \hline 1101 \end{array}$

$$Q \Rightarrow (-3)_{10} = (1101)$$

AC = 0000 $\quad Q_{-1} = 0 \quad -M = (0111)_2$

Q = 1101

Step1:

AC	Q ($Q_3 Q_2 Q_1 Q_0$)	Q_{-1}	operation
0000	1101	0	Initially
	10		
0000	1101	0	AC = AC − M

M.S

$AC = 0000$
$\quad\; 0111$
$\quad\; \overline{0111}$

0111   1101   0 → Ignore   ASR

0011   1110   1

0011   1110   1

$n = n-1 = 4-1$
$n = 3$

Step-2:

AC	Q	Q_{-1}	Operation
0011	1110	1	$AC = AC + M$

01

$AC = 0011$
$\quad\;\; 1001$
$\quad\;\; \overline{1101}$

1101   1110   1 → Ignore   ASR

1110   1111   0   ASR

1110   1111   0

$n = n-1 = 3-1$
$n = 2$

Step 3:

AC	Q	Q_{-1}	operation
1110	0111	0	$AC = AC - M$

10

$AC = \;\; 1110$
$\quad\quad\; 0111$
$\quad\quad \overline{1\,0101}$

1 → Discard cary

0101   0111   0   ASR

$\boxed{M.S}$

0101	0111	0	Ignore	ASR
0010	1011	1		
0010	1011	1		$n = n-1 = 2-1 = 1$

Step 4:

AC	Q	Q_{-1}	operation
0010	1011	1	ASR

11

0010	1011	1	Ignore	ASR
0001	0101	1		ASR
0001	0101	1		$n = n-1 = 1-1 = 0$

$P \Rightarrow AC, Q$

$\Rightarrow 0001 \quad 0101$

merge

00010101

$\Rightarrow (21)_{10}$

$(-7) \times (-3) = (21)_{10}$ Ans

$\boxed{M.S}$

Example Show the multiplication Process using Booth's algorithm when the following numbers are multiplied (−13) by (−8).

Sol $M = -13$ $Q = 8$

$-M = 01101$ $Q \Rightarrow 01000$

number of bit $n = 5$

1's complement of M = 10010
2's complement of −(−M) = + 1

$$\begin{array}{r} 10010 \\ +\quad 1 \\ \hline 10011 \end{array}$$

$-(-M) = 10011$

$M = 10011$

$-M = 01000$

$AC = 00000$

$Q_{-1} = 0$

$n = 5$

Step 1

AC	Q	Q_{-1}	operation
00000	01000 ($Q_4 Q_3 Q_2 Q_1 Q_0$)	0	Initially
	00		
00000	01000	0 — Ignore	ASR
00000	00100	0	

M.S

000000	00100	0	$n = n-1$ $= 5-1$ $= 4$

Step 2:-

AC	Q	Q_{-1}	Operation
00000	$0010\underbrace{0 \quad 0}_{00}$		ASR
00000	00100	0 → Ignore	ASR
00000	00010	0	
00000	00010	0	$n = n-1 = 4-1$ $= 3$

Step 3:

AC	Q	Q_{-1}	Operation
00000	$0001\underbrace{0 \quad 0}_{00}$		ASR
00000	00010	0 → Ignore	ASR
00000	00001	0	
00000	00001	0	$n = n-1 = 3-1$ $= 2$

Step-4

AC	Q	Q_{-1}	Operation
00000	$0000\underbrace{1 \quad 0}_{10}$		AC= AC−M

AC = 00000
      01101

      01101

01101	00001	0 → Ignore	ASR
00110	10000	1	
00110	10000	1	$n = n-1 = 2-1$ $\Rightarrow 1$

Step-5

AC	Q	Q_{-1}	operation
00110	$\underbrace{10000\ \ 1}_{01}$		AC = AC+M

$$
\begin{array}{r}
AC = 00110 \\
10011 \\
\hline
11001
\end{array}
$$

11001	10000	1 → Ignor	ASR
11100	11000	0	
11100	11000	0	$n = n-1$ $= 1-1$ $= 0$

Result = AC, Q.

$\Rightarrow$ 11100 11000

merge

$R \Rightarrow (1110011000)_2$

$$
\begin{array}{lr}
1\text{'s complement of } R = & 00011\ 00111 \\
2\text{'s complement of } (R) = & +\qquad\qquad 1 \\
\hline
 & 00011\ 01000
\end{array}
$$

M.S

$(-R) \Rightarrow \quad 1101000$

$R = -(104)_{10}$ Ans

Ex Show the multiplication Process using Booth's algorithm when the following numbers are multiplied (+13) by (+8).

Ex Show the multiplication Process using Booth's algorithm when the following numbers are multiplied (−13) by (−8).

$\boxed{\text{Do your self}}$

Example

Show the step by step the multiplication Process using Booth's algorithm when (+15) and (−13) numbers are multiplied. Assume 5- bit registers that Hold signed numbers.

Sol

$M = 15 \quad Q = -13 \quad n = 5$

$M \Rightarrow 01111 \quad Q = 01101$

1's complement of M = 10000

2's complement of (-M) = $\begin{array}{r} + \quad 1 \\ \hline 10001 \end{array}$

$-M = 10001$

$-Q = 01101$

1's complement of (-Q) = 10010

2's complement of -(-Q) = $\begin{array}{r} + \quad 1 \\ \hline 10011 \end{array}$

$Q = 10011$

$M = 01111 \quad Q = 10011 \quad -M = 10001$

$AC = 00000 \quad Q_{-1} = 0$

Step-1

AC	Q	Q_{-1}	operation
00000	10011	0	Initially
			AC = AC − M

10

M.S

$$
\begin{array}{r}
AC = 00000 \\
10001 \\
\hline
10001
\end{array}
$$

10001	10011	0 → Ignore	ASR
11000	11001	1	ASR
11000	11001	1	$n = n-1 = 5-1 = 4$

Step-2

AC	Q	Q_{-1}	operation
11000	11001 (11)	1 → Ignore	ASR
11100	01100	1	ASR
11100	01100	1	$n = n-1 = 4-1 = 3$

Step-3

AC	Q	Q_{-1}	operation
11100	01100 (01)	1	AC = AC + M

$$
\begin{array}{r}
AC = 11100 \\
01111 \\
\hline
(1)01011
\end{array}
$$

Discard carry

01011	01100	1 → Ignore	ASR
00101	10110	0	ASR

$\boxed{M.S}$

00101	10110	0	$n = n-1 = 3-1 = 2$

Step-4

AC	Q	Q_{-1}	operation
00101	10110	0	ASR
	$\underbrace{0 \quad 0}_{00}$		
00101	10110	0 → Ignore	ASR
00010	11011	0	
00010	11011	0	$n = n-1 = 2-1 = 1$

Step-5

AC	Q	Q_{-1}	operation
00010	11011	0	AC= AC-M
	$\underbrace{1 \quad 0}_{10}$		AC= 00010 + 10001 = 10011
10011	11011	0 → Ignore	ASR
11001	11101	1	ASR
11001	11101	1	$n = n-1 = 1-1 \Rightarrow 0$

Result ⇒ R = AC. Q

11001 11101

merge

1100111101

M.S

1's complement of R = 0011000010

2's complement of (-R) = + 1

$$\begin{array}{r} 0011000010 \\ +\qquad\qquad 1 \\ \hline 0011000011 \end{array}$$

(-R) = 0011000011

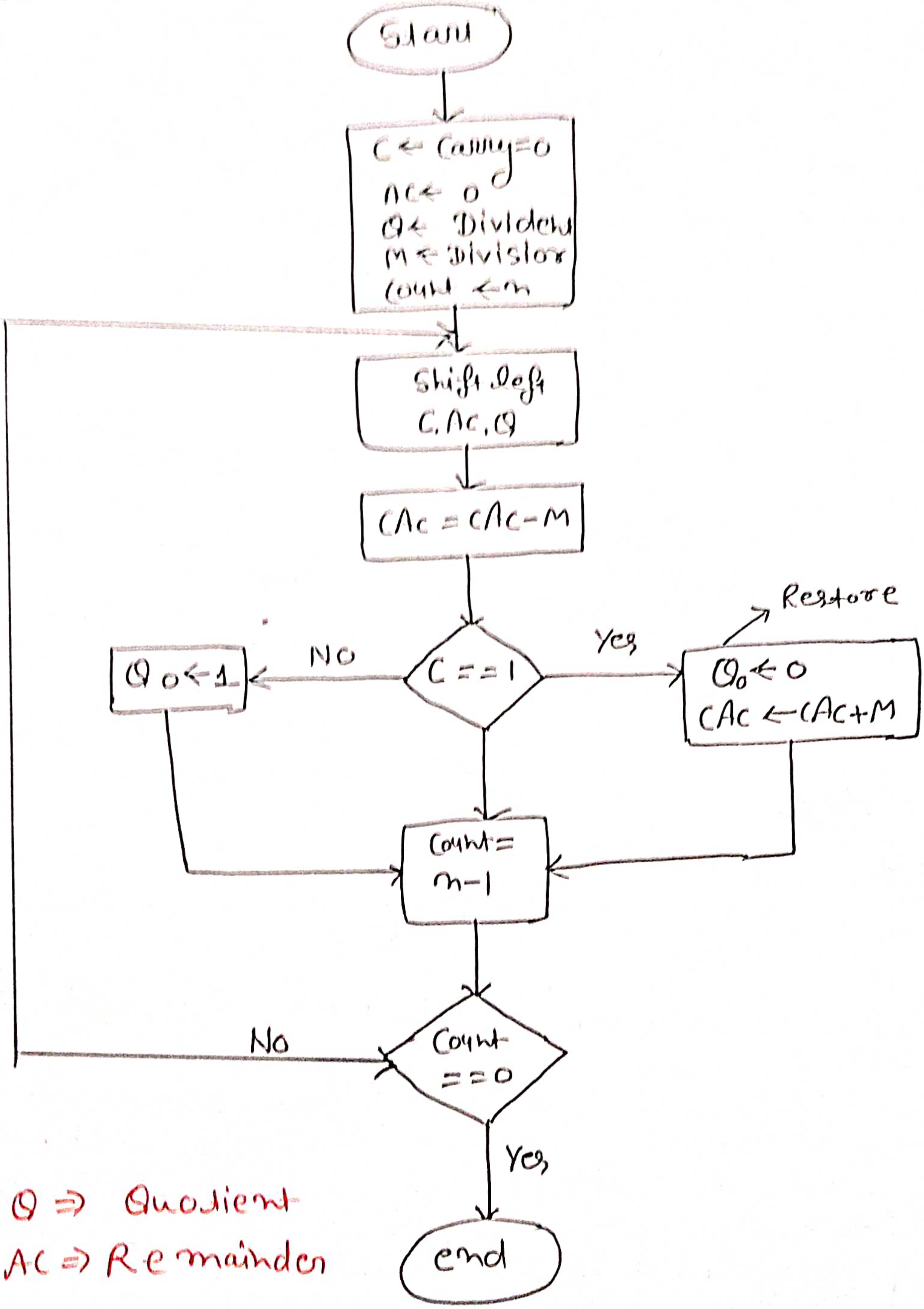
R = $(-195)_{10}$ Ans

Ex Show step by step the multiplication process using Booth algorithm when (+15) and (+13) numbers are multiplied. Assume 5-bit registers that hold signed numbers.

Ex Show step by step the multiplication process using Booth Algorithm when (-15) and (-13) numbers are multiplied. Assume 5-bit registers that hold signed numbers.

$\boxed{\text{Do your self}}$

Restoring Division method: FLOW CHART



Q ⇒ Quotient

AC ⇒ Remainder

M.S

Q1 Perform Division of the following 12/3.

Sol. Q = 12

M = 3

Q = 12 ⇒ $(1100)_2$

M = 3 ⇒ (0 0011)

Add

1's complement of M = 11100

2's complement of (-M) = + 1

11101

$(-M) = (11101)_2$

AC = 0000   n = 4

Step 1:

C	AC	Q	operation
0	0000	1100	Shift left
0	0001	100□	CAC = CAC − M
			= 00001
			11101
			11110
1	1110	100$\boxed{0}$	CAC = CAC + M
			= 11110
			Discard 00011
			1 00001
1	1110	100$\boxed{0}$	n = n−1 = 4−1 = 3

M.S

Step-2:

C	AC	Q	Operation
0	0001	1000	Shift left
0	0011	000$\square$	CAc = CAc − M
			= 00011
			11101
			1 00000
			Discard carry
0	0000	000$\boxed{1}$	n = 3−1
			= 2

Step-3:

C	AC	Q	Operation
0	0000	0001	Shift left
0	0000	001$\square$	CAc = CAc − M
			= 00000
			11101
			11101
1	1101	001$\boxed{0}$	CAc = CAc + M
			= 11101
			00011
			1 00000
			Discard
0	0000	001$\boxed{0}$	n = 2−1
			= 1

$\boxed{\text{M.S}}$

Step-4:

C	AC	Q	operation
0	0000	0010	Shift left
0	0000	010$\square$	CAC = CAC − M
			= 00000
			11101
			11101
1	1101	010$\boxed{0}$	CAC = CAC+M
			= 11101
			00011
			1 00000
			Discard
0	0000	010$\boxed{0}$	n = 1−1 = 0

AC = Remainder ⇒ 0

Q = 0100

Q = $(4)_{10}$ Ans

Q2 Perform division of the following 17/3.

Sol Q = 17 M = 3 n = 5

Q = $(17)_{10}$

⇒ $(10001)_2$

M = $(3)_{10}$ ⇒ $(11)_2$

M = 000011

c

M.S

1's complement of M = 111100

2's complement of M = + 1

111101

$(-M) = (111101)_2$

Step-1

C	AC	Q	operation n=5
0	00000	10001	Shift-Left
0	00001	0001☐	CAC = CAC − M
			= 000001 + 111101 = 111110
1	11110	0001$\boxed{0}$	CAC = CAC + M
			= 111110 + 000011 = 1000001
			Discard carry
0	00001	00010	n = 5−1 = 4

Step-2

C	AC	Q	Operation n=4
0	00001	00010	Shift Left

M.S

0	0 0010	0010$\square$	CAc = CAc − M
			= 000010
			  111101
			———
			  111111

1	11111	0010$\boxed{0}$	CAc = CAc + M = 111111 000011 ——— 1 000010 Discard carry
0	00010	00100	n = 4−1 = 3

Step-3

C	Ac	Q	operation n=3
0	00010	00100	Shift left
0	00100	0100$\square$	CAc = CAc − M = 000100 111101 ——— 1 000001 Discard carry
0	00001	0100$\boxed{1}$	n = 3−1 = 2

Step-4

C	AC	Q	operation
0	00001	01001	Shift left
0	00010	1001$\square$	CAc = CAc−M

M.S

$$\begin{array}{r} =000010 \\ 111101 \\ \hline 111111 \end{array}$$

1	11111	1001$\boxed{0}$	CAC = CAC + M

$$\begin{array}{r} =111111 \\ 000011 \\ \hline 1\,000010 \end{array}$$

Discard carry

0	00010	10010	n = 2-1 = 1

Step-5

C	AC	Q	operation
0	00010	10010	Shift Left
0	00101	0010$\square$	CAC = CAC − M

$$\begin{array}{r} =000101 \\ 111101 \\ \hline 1\,000010 \end{array}$$

Discard carry

1	00010	0010$\boxed{1}$	n = 1-1 = 0

Q = 00101 = $(5)_{10}$

AC = 00010 = $(2)_{10}$

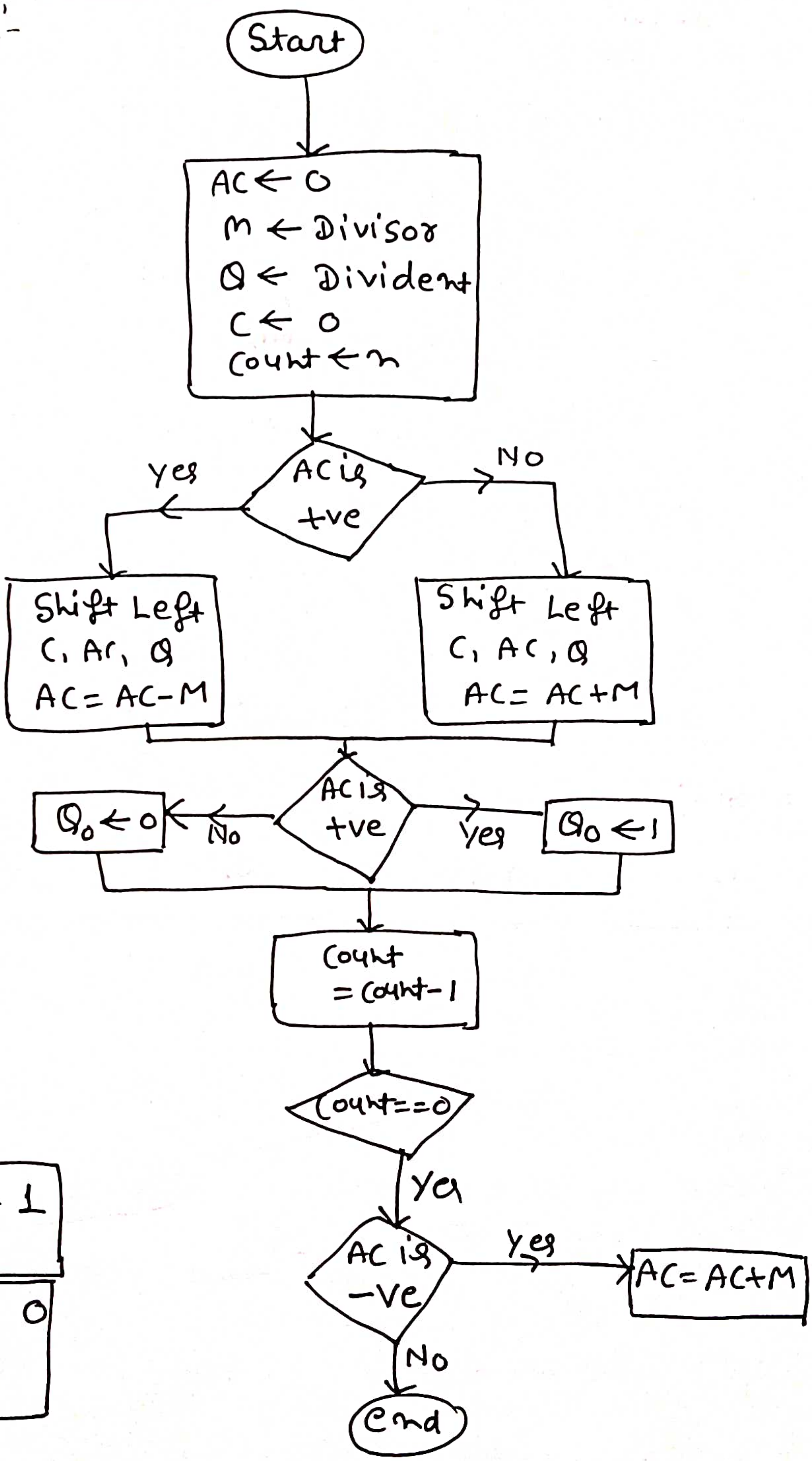
Remainder = $(2)_{10}$

Quotient = $(5)_{10}$ **Ans**

M.S

Non- REstoring Division method:-

Flow- chart:-



if c bit is 1 −ve

if c bit is 0 +ve

Q1 Perform division using non-restoring division $(10)_{10} \div (3)_{10}$

Sol

$Q = (10)_{10} \Rightarrow (1010)_2$

$M = (3)_{10} \Rightarrow (11)_2$ $n = 4$

0 0011

C(bit)

$M = (00011)$

1's complement of M = 11100

2's complement of (−M) = 11100 + 1 = 11101

$-M = (11101)_2$

Step 1:

C	AC	Q	operation
0	0000	1010	Initially
0	0001	010□	Shift left
0	0001	010□	AC = AC − M
			= 00001 + 11101 = 11110
1	1110	010$\boxed{0}$	C(bit)
			n = n−1 = 4−1 = 3

Step-2

C	AC	Q	Operation
1	1110	0100	Shift left
1	1100	100□	AC = AC+M
			= 11100
			00011
			(1)1111
			(C bit)
1	1111	100[0]	n = n-1
			= 3-1 = 2

Step 3:

C	AC	Q	Operation
1	1111	1000	Shift left
1	1111	000□	AC = AC+M
			= 11111
			00011
			(1)00010
			Discard carry
0	0010	000[1]	n = n-1 = 2-1
			= 1

Step 4:

C	AC	Q	Operation
0	0010	0001	Shift left
0	0100	001□	AC = AC-M
			= 00100
			11101
0	0001	001[1]	(1)00001
			Discard carry
			n = 1-1 = 0

Remainder = C, AC Quotient = Q

$R = (00001)_2$ $Q = (0011)_2$

$= (1)_{10}$ $Q = (3)_{10}$

Quotient $= (3)_{10}$ $10 \div 3$

Remainder $= (1)_{10}$ Ans

Q3 Divide (1011) with $(0011)_2$ using non-restoring divison methods.

sol

$Q = (1011)_2$

$M = (0011)_2$

$M = (00011)_2$

CCbit

1's complement of M = 11100

2's complement of (−M) = 1

$$\begin{array}{r} 11100 \\ +\quad 1 \\ \hline 11101 \end{array}$$

$-M = (11101)_2$

$Q = (1011)_2$

n noof bit = 4

Step -1

C	AC	Q	operations
0	0000	1011	Initial
0	0001	011 $\square$	Shift left
			AC= AC-M
			= 00001
			11101
1	1110	011 $\boxed{0}$	11110
			n= 4-1=3

Step-2

C	AC	Q	Operations
1	1110	0110	Shift left
1	1100	110 $\square$	AC= AC+M
			= 11100
			00011
1	1111	110 $\boxed{0}$	11111
			n= n-1
			= 3-1
			= 2

Step-3

C	AC	Q	operations
1	1111	1100	Shift left
1	1111	100 $\square$	AC= AC+M
			= 11111
			00011
0	0010	100 $\boxed{1}$	1 00010
			Discard carry
			n= n-1 = 2-1
			= 1

Step-4

C	AC	Q	Operations
0	0010	1001	Shift left
0	0101	001 $\square$	AC = AC - M = 00101 11101 $\overline{}$ 1 00010 discard carry
0	0010	001 $\boxed{1}$	n = n-1 = 1-1 = 0

Remainder = C, AC

R = $(00010)_2$

= $(2)_{10}$

Quotient = Q

Q = $(0011)_2$

Q = $(3)_{10}$

Q = $(3)_{10}$

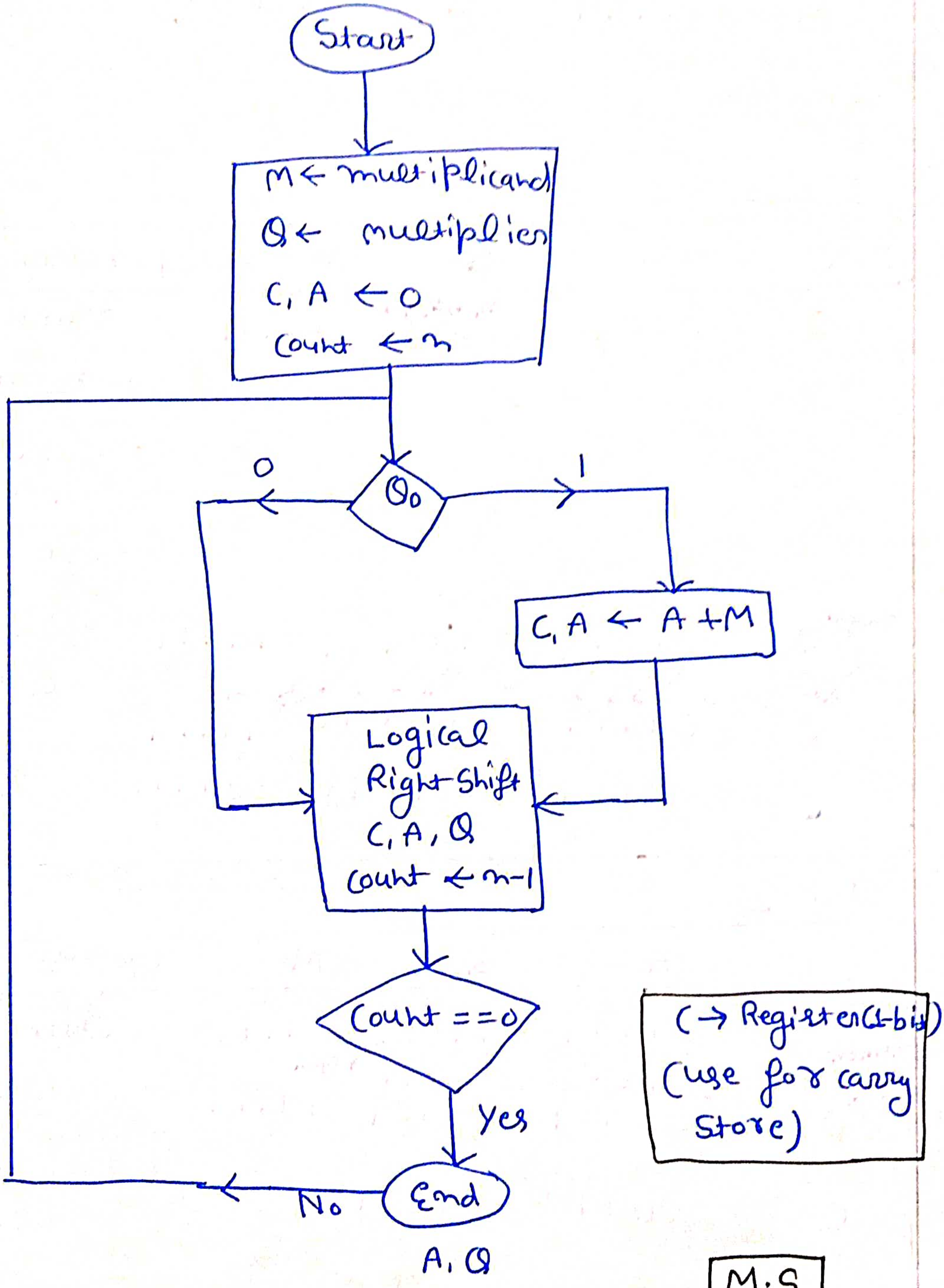
$(11)_{10} \div (3)_{10}$

Q = $(3)_{10}$

R = $(2)_{10}$ Ans

Unsigned number multiplication:-

FLOW CHART FOR UNSIGNED MULTIPLICATION



M·S

Ex-1 Show the contents of the Registers E, A, Q, SC during Processes of multiplication of two binary no.

M = 11111   Q = 10101   n=5

Sol

Step-1

C	A	Q	M	operations
0	00000	$Q_4Q_3Q_2Q_1Q_0$ 10101	11111	Initially
				C, AC = A+M = 000000 + 11111 = 011111
Insert 0 → 0	01111	11010	11111	Shift Right C, A, Q (Last)
0	01111	11010	11111	n = n-1 = 5-1 = 4

Step-2

C	A	Q	M	operations
0	01111	11010	11111	Shift right
Insert 0 → 0	00111	11101	11111	(Ignore)
0	00111	11101	11111	n = 4-1 = 3

Step-3

C	A	Q	M	operation
0	00111	11101	11111	C, A = A+M
				= 000111 11111 100110
1	00110	11101	11111	Shift-Right C, A, Q
0	10011	01110	11111	
0	10011	01110	11111	n = n-1 = 3-1 = 2

Insert 0

Ignore

Step-4

C	A	Q	M	operations
0	10011	01110	11111	Shift Right
0	01001	10111	11111	n = n-1 = 2-1 = 1

Insert 0

Last Ignore

M.S

Step-5

C	A	Q	M	operation
0	01001	10111	11111	C, A = A+M = 001001 11111 101000
1	01000	10111 Last Ignor	11111	Shift Right
Insert 0	10100	01011	11111	n = 1-1 = 0

Result R = A, Q

= 10100 01011

merge

= 1010001011

R ⇒ $(651)_{10}$ <u>Ans</u>

<u>Ex-2</u> Show the contents of the Registers E, A, Q, SC during Process of multiplication of two binary No

M = 1011 Q = 1101 n = 4