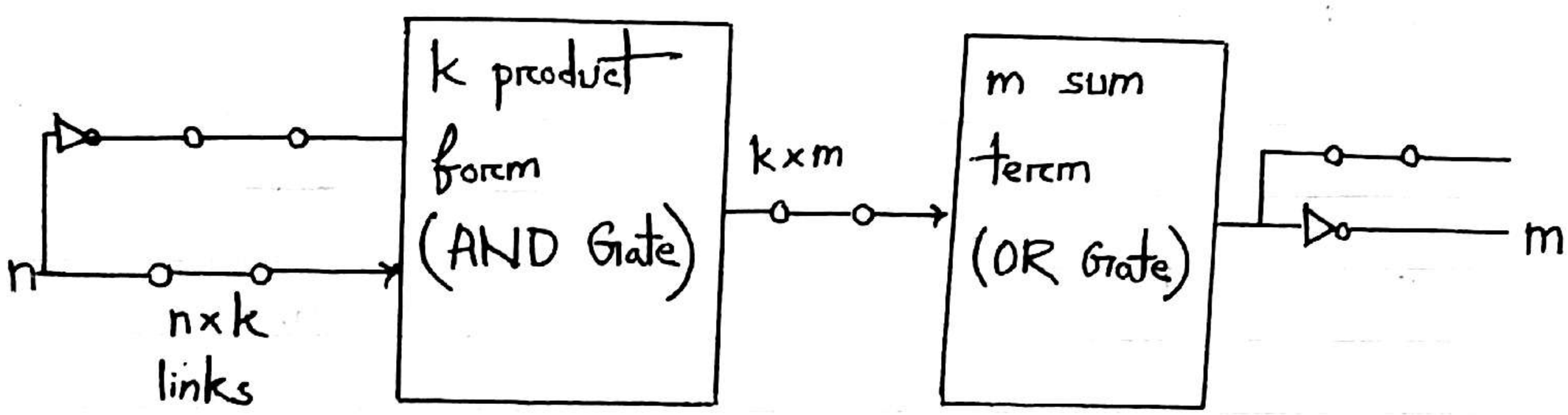
Digital System Design

Date: 23 April

Book : Digital Logic and Computere Design — M. Morris Mano



A combinational circuit is defined by the functions :

$$F_1(A, B, C) = \Sigma(4, 5, 7)$$

$$F_2(A, B, C) = \Sigma(3, 5, 7)$$

Q. Implement the circuit with a PLA having 3 inputs, 3 product terms and two outputs.

(a) truth table :

A	B	C	F_1	F_2
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	0	0
1	1	1	1	1

(b) map simplification :

A \ BC	00	01	11	10
0				
1	1	1	1	

$F_1 = AB' + AC$

A \ BC	00	01	11	10
0			1	
1		1	1	

$F_2 = AC + BC$

(c) PLA Program table :

Product term	A	B	C	Output F_1	Output F_2
AB'	1	0	—	1	—
AC	1	—	1	1	1
BC	—	1	1	—	1
				T	T

Date.....

Q. [Same Question]

$F_1(A,B,C) = \Sigma(3,5,6,7)$
$F_2(A,B,C) = \Sigma(0,2,4,7)$

Map Implementation :

A \ BC	00	01	11	10
0			1	
1		1	1	1

$F_1 = AC + AB + BC$

A \ BC	00	01	11	10
0	1			1
1	1		1	

$F_2 = A'C' + B'C' + ABC$

A \ BC	00	01	11	10
0	0	0		0
1	0			

$F_1' = A'C' + B'C' + A'B'$

JMI SYRINGES & MEDICAL DEVI

Scanned by CamScanner

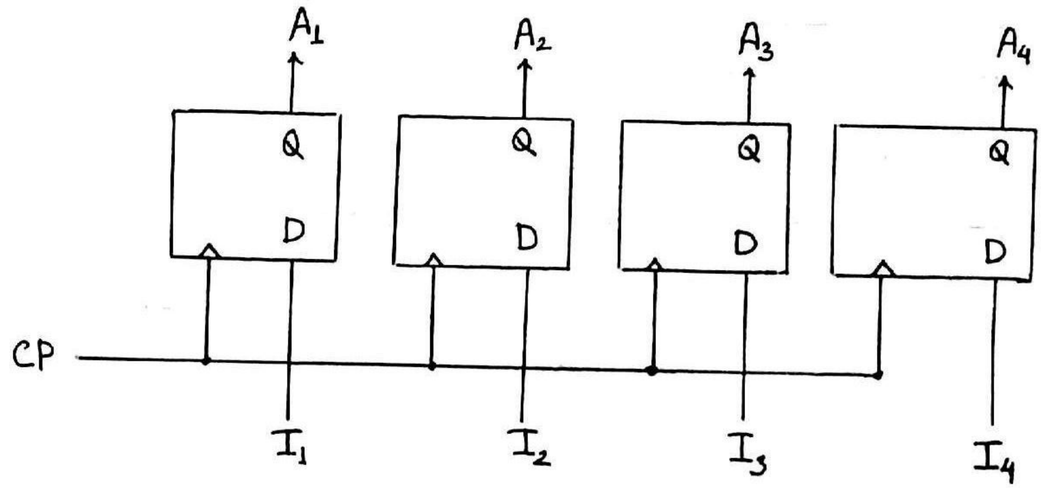
PLA Program table :

Product term	A	B	C	F_1	F_2
$A'C'$	0	—	0	1	1
$A'B'$	0	0	—	1	1
$B'C'$	—	0	0	1	—
ABC	1	1	1	—	1
				C	T

Date 26 April

Register :

A register is a group of binary storage cells suitable for holding binary information. A group of flipflop constitutes a register, since flipflop is a binary cell capable of storing one bit of information. A n bit register has a group of n flipflops and is capable of storing any binary containing n bits.

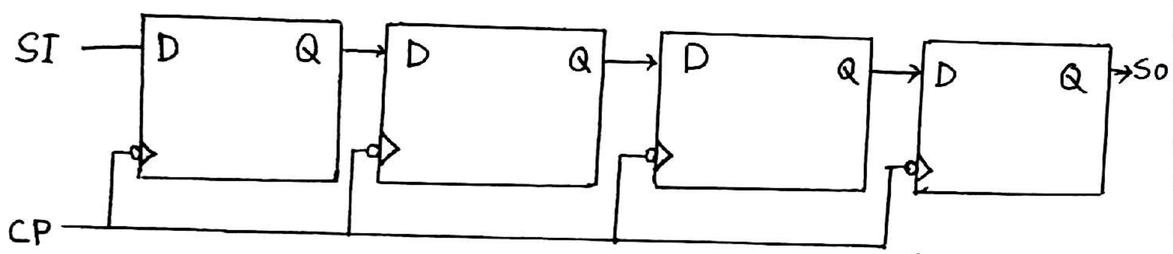


Register is enabled for as long as CP=1. When CP goes to 1, input information is loaded into the register. If CP remains to 0, content of register is not changed.

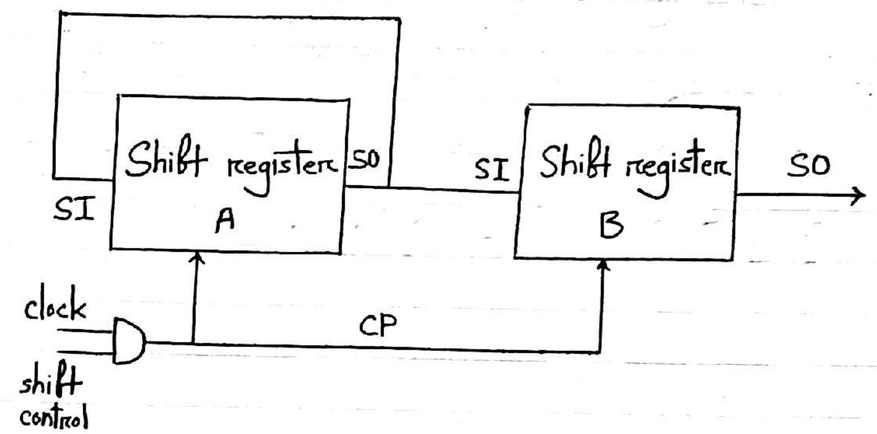
Scanned by CamScanner

Shift register :

A register is capable of shifting its binary information either to the right or to the left is called a shift register. The logical configaration of a shift register consists of a chain of flipflops connected in cascade with the output of one flipflop connected to the input of the next flipflop. All flipflop recieve a common clock-pulse which causes the shift from one stage to the next.

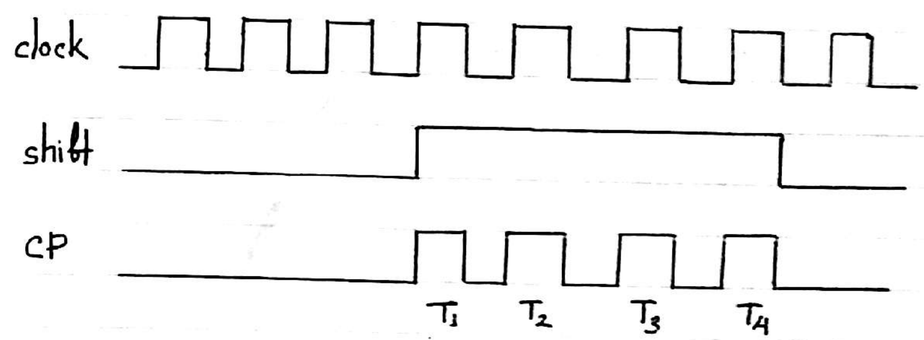


Serial Transfer :



The shift control input determines when and by how many times the registers are shifted suppose the shift register has 4 bits each.

Timing Diagram :



JMI SYRINGES & MEDICAL DEVICES LTD.
Scanned by CamScanner

Serial Tranfer Example :

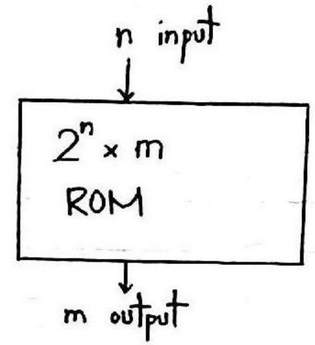
Timing Pulse	Shift register A	Shift register B	Serial Output B
Initial Value	1 0 1 1	0 0 1 0	0
After T_1	1 1 0 1	1 0 0 1	1
After T_2	1 1 1 0	1 1 0 0	0
After T_3	0 1 1 1	0 1 1 0	0
After T_4	1 0 1 1	1 0 1 1	1

After 4th shift control, it goes to 0 and Both register A and B have the same value 1011 :

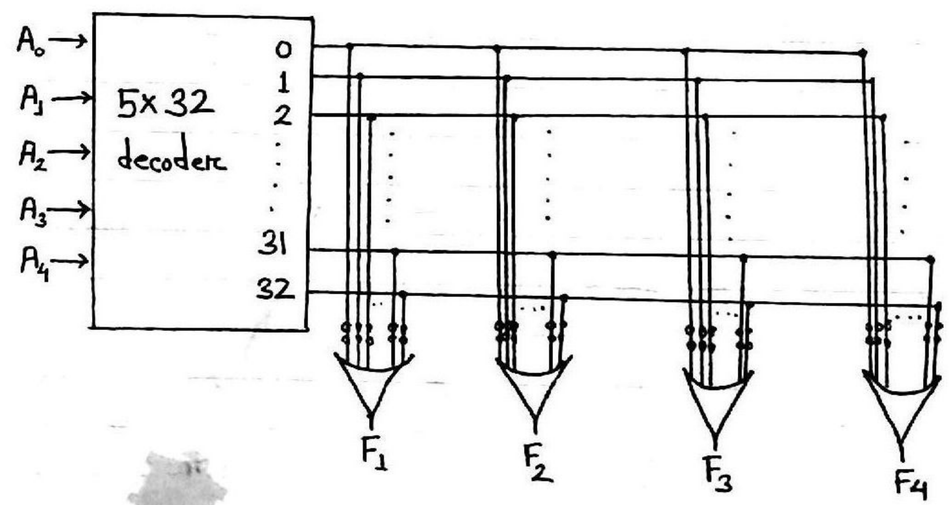
Date

Read Only Memory (ROM) :

A ROM is essentially a memory device in which a fixed set of binary info is stored. The binary info must be specified by the user and is then embedded in the unit to form the required interconnection pattern.



Consider a 32×4 ROM. the unit consist of 32 words of 4 bits each. If the input address is 00000, word no. 0 is selected and it appears on the output line.



Scanned by CamScanner

29 April

Random Access Memory (RAM) :

RAM is the place in a computer where the operating system, application programs and data is current use are kept so that they can be quickly reached by Computer's Processor.

ROM :

ROM is a type of memory that is fast as RAM, but has 2 Important differences. It can not be changed and it remains its content when the computer is shut off. It is generally used to start your computer up and load OS.

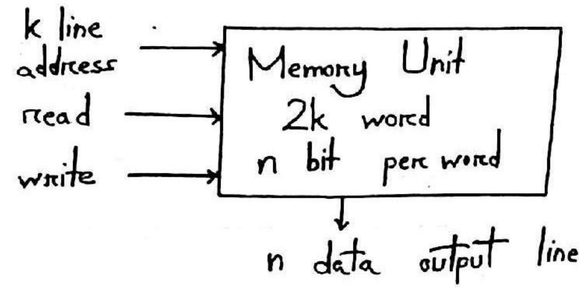
RAM :

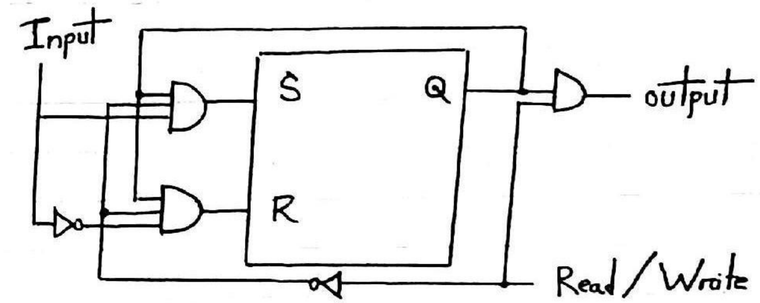
Memory unit : stores ~~bit~~ binary information in group of ~~word~~ bits called word.

Memory words : Group of 1's and 0's may represent a number, characters, instructions or other binary information. Most computers use words that are multiple of 8 bits.

Ex. 32 bit word = 4 byte

Date





Each word in memory is assigned as address 0 upto 2^k-1
(k = no. of address line)

Binary	Decimal	Memory Address
0000 0000 01	1	1011 1100 11
0000 0000 10	2	1010 0010 01
⋮	⋮	⋮
1111 1111 10	1022	1000 1001 01
1111 1111 11	1023	1111 1011 10

To transfer a new word to be stored into memory :

(1) Apply the binary address of the word to address lines.

(2) Apply the data bits that must be stored in memory to the data input lines.

(3) Activate the "write" input.

To transfer a stored word out of memory :

(1) Apply the binary address of the word to address lines.

(2) Activate the "Read" input.

Types : RAM units are mainly available in two operating modes.

- ★ Static
- ★ Dynamic

Date.

Static RAM (SRAM) :

Consits of internal latches that stores the binary information that remains valid as long as power is applied to the unit.

Dynamic RAM (DRAM) :

stores the binary into the form of electric charges on capacitor provided by the mos transistor. The capacitor must be periodically reached by refreshing of the dynamic memory every few seconds.

Vs :

★ DRAM offers reduced power consumption, large of units on chip.

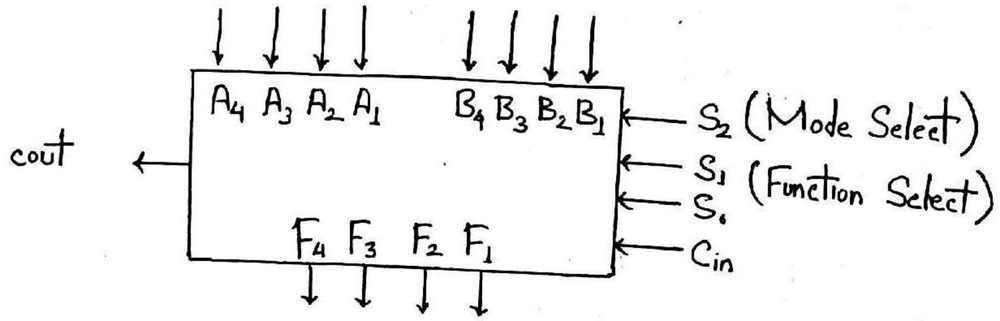
★ SRAM is faster, has shorter read and write cycle. SRAM is used in cache.

Disadvances : high power consumption, low density, expensive.

Scanned by CamScanner

30 April

ALU Design :



S_1	S_0	C_{in}	X	Y	Output, F	Function
0	0	0	A	0	A	Transfer A
0	0	1	A	0	A+1	Increment A
0	1	0	A	B	A+B	Add A and B
0	1	1	A	B	A+B+1	Add A and B and 1
1	0	0	A	B'	A−B−1	Subtract A and B and 1
1	0	1	A	B'	A−B	Subtract A and B
1	1	0	A	All 1	A−1	Decrement A
1	1	1	A	All 1	A	Transfer A

$$X_i = A_i$$

$$Y_i = S_1' S_0 B_i + S_1 S_0' B_i' + S_1 S_0$$

$$= B_i S_0 + B_i' S_1$$

Date: 3 May

Effect of output carry :

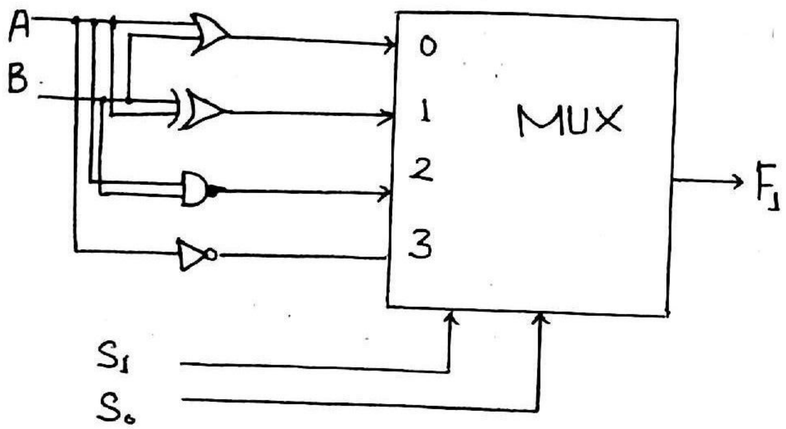
To investigate the effect of output carry, we expand the arithmatic circuit to n bits so that cout = 1 when the output of the circuit is equal to greater than 2^n. An output carry of 1 after an addition operation denotes an overflow condition.

S_1	S_0	C_{in}	Arithmatic Function	Cout = 1 if
0	0	0	F = A	
0	0	1	F = A+1	$A = 2^n - 1$
0	1	0	F = A+B	$(A+B) \geq 2^n$
0	1	1	F = A+B+1	$(A+B) \geq 2^n - 1$
1	0	0	F = A−B−1	$A > B$
1	0	1	F = A−B	$A \geq B$
1	1	0	F = A−1	$A \neq 0$
1	1	1	F = A	

Scanned by CamScanner

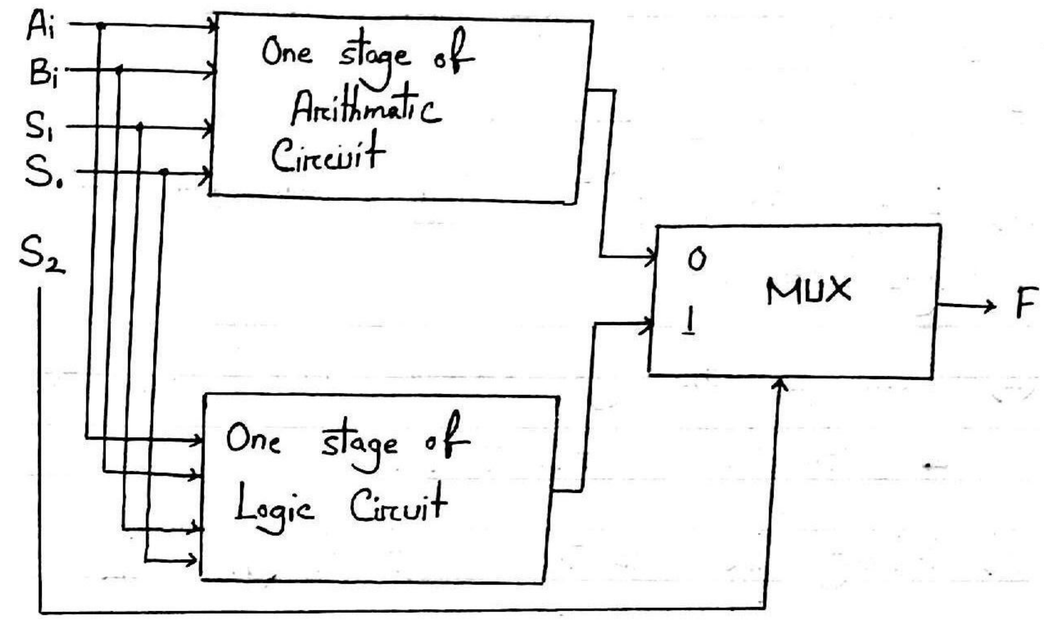
6 May

Design of Logic Circuit :



S_1	S_0	Output
0	0	$A_i + B_i$
0	1	$A_i \oplus B_i$
1	0	$A_i B_i$
1	1	$\overline{A_i}$

Date



Logic Operation in one stage of ~~a~~ arithmatic circuit :

S_2	S_1	S_0	X_i	Y_i	C_{in}	$F_i = X_i \oplus Y_i$	Operation	Required
1	0	0	A_i	0	0	A_i	Transfer A	OR
1	0	1	A_i	B_i	0	$A_i \oplus B_i$	XOR	XOR
1	1	0	A_i	B_i'	0	$A_i \odot B_i$	Equivalence	AND
1	1	1	A_i	1	0	A_i'	NOT	NOT

Scanned by CamScanner

Design of ALU : The steps involved in the design of ALU :

(1) Design the arithmatic section, independent of logic section.

(2) Determine the logic operation obtained from the arithmatic circuit in step 1.

(3) Modify the arithmatic circuit to obtain the required logic operation.

Date....

S_2	S_1	S_0	C_{in}	Output	X_i	Y_i	Function
0	0	0	0	A	A	0	
0	0	0	1	A+1	A	0	
0	0	1	0	A+B	A	B	
0	0	1	1	A+B+1	A	B	
0	1	0	0	A−B−1	A	B′	
0	1	0	1	A−B	A	B′	
0	1	1	0	A−1	A	1	
0	1	1	1	A	A	1	
1	0	0	×	$A \vee B$	$A \vee B$	0	
1	0	1	×	$A \oplus B$	A	B	
1	1	0	×	$A \wedge B$	A+B′	B′	
1	1	1	×	$\bar{A}$	A	1	

JMI SYRINGES & MEDICAL DEVICES

Scanned by CamScanner

07 May

Q. Problem : 9.17

$$X_i = A_iB_i + (S_2S_1'S_0')'A_i + S_2S_1S_0'B_i$$

$$Y_i = S_0B_i + S_1B_i'(S_2S_1S_0')'$$

$$Z = S_2'C_{in}$$

Sol^n : If $S_2=0$ then function reduce to

$$X = AB + A$$
$$= A(B+1)$$
$$= A$$

$$Y = S_0B + S_1B'$$

$$Z = C_{in}$$

when $(S_1S_0) = (0,0)$

$X = A$

$Y = 0$

$Z = C_{in}$

now if $C_{in} = 0$	now if $C_{in} = 1$
$F = A$	$F = A+1$

Date:

when $(S_1S_0) = (0,1)$

$X = A$

$Y = B$

$Z = C_{in}$

now if $C_{in} = 0$	now if $C_{in} = 1$
$F = A+B$	$F = A+B+1$

when $(S_1S_0) = (1,0)$

$X = A$

$Y = B'$

$Z = C_{in}$

now if $C_{in} = 0$	now if $C_{in} = 1$
$F = A-B-1$	$F = A-B$

when $(S_1S_0) = (1,1)$

$X = A$

$Y = 1$

$Z = C_{in}$

now if $C_{in} = 0$	now if $C_{in} = 1$
$F = A-1$	$F = A$

Scanned by CamScanner

when $S_2 = 1$, $X \oplus Y$ operation will occur each operation will occur in each stage. now if $(S_2\ S_1\ S_0) = (1\ 0\ 0)$

$X = AB$

$Y = 0$

$Z = 0$

$\therefore F = AB \oplus 0$

$= AB$

now if $(S_2\ S_1\ S_0) = (1\ 0\ 1)$

$X = AB + A = A$

$Y = B$

$Z = 0$

$\therefore F = A \oplus B$

now if $(S_2\ S_1\ S_0) = (1\ 1\ 0)$

$X = AB + A + B = A + B$

$Y = 0$

$Z = 0$

$\therefore F = A + B$

Date

now if $(S_2\ S_1\ S_0) = (1\ 1\ 1)$

$X = AB + A = A$

$Y = B + B' = 1$

$Z = 0$

$\therefore F = A \oplus 1$

$= A'$

S_2	S_1	S_0	C_{in}	X	Y	Z	F
0	0	0	0	A	0	0	A
0	0	0	1	A	0	1	A+1
0	0	1	0	A	B	0	A+B
0	0	1	1	A	B	1	A+B+1
0	1	0	0	A	B'	0	A−B−1
0	1	0	1	A	B'	1	A−B
0	1	1	0	A	1	0	A−1
0	1	1	1	A	1	1	A
1	0	0	×	AB	0	0	AB
1	0	1	×	A	B	0	A⊕B
1	1	0	×	A+B	0	0	A+B
1	1	1	×	A	1	0	A'

Scanned by CamScanner

13 May

Date:

Counter :

J	K		P.S	N.S
0	0	Hold	0	0
			1	1
0	1	Reset	0	0
			1	0
1	0	Set	0	1
			1	1
1	1	Toggle	0	1
			1	0

Excitation table : JK FlipFlop

P.S	N.S	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

0	Hold
1	Toggle

Excitation Table : T flipflop

P.S	N.S	T
0	0	0
0	1	1
1	0	1
1	1	0

3 bit binary Counter : JK FlipFlop

P.S			N.S								
A_3	A_2	A_1	A_3	A_2	A_1	J_3	K_3	J_2	K_2	J_1	K_1
0	0	0	0	0	1	0	x	0	x	1	x
0	0	1	0	1	0	0	x	1	x	x	1
0	1	0	0	1	1	0	x	x	0	1	x
0	1	1	1	0	0	1	x	x	1	x	1
1	0	0	1	0	1	x	0	0	x	1	x
1	0	1	1	1	0	x	0	1	x	x	1
1	1	0	1	1	1	x	0	x	0	1	x
1	1	1	0	0	0	x	1	x	1	x	1

JMI SYRINGES & MEDICAL DEVICES LTD.

Scanned by CamScanner

$A_3 \backslash A_2A_1$	00	01	11	10
0			1	
1	x	x	x	x

$J_3 = A_2 A_1$

$A_3 \backslash A_2A_1$	00	01	11	10
0	x	x	x	x
1			1	

$K_3 = A_2 A_1$

$A_3 \backslash A_2A_1$	00	01	11	10
0		1	x	x
1		1	x	x

$J_2 = A_1$

$A_3 \backslash A_2A_1$	00	01	11	10
0	x	x	1	
1	x	x	1	

$K_2 = A_1$

$A_3 \backslash A_2A_1$	00	01	11	10
0	1	x	x	1
1	1	x	x	1

$J_1 = 1$

$A_3 \backslash A_2A_1$	00	01	11	10
0	x	1	1	x
1	x	1	1	x

$K_1 = 1$

Date:

Booth's Multiplication Algorithm :

Booth's algorithm can be implemented by repeatedly adding one or two values A and S to a product P, then performing a rightward arithmatic ~~st~~ shift of P.

Let m and r be the multiplicand and multiplier respectively, and Let x and y represent the number of bits in m and r.

(1) Determine the value of A, S and the intial value of P. All these numbers should have length equal to $(x+y+1)$.

(I) A: Fill the most significant bits with the value of m. Fill the remaining $(y+1)$ bits with zeros.

(II) S : Fill the most significant bits with the value of $(-m)$ in two's compliment notation. Fill the remaining $(y+1)$ bits with zero's.

(III) P : Fill the most significant x bits with zeros. To the right of this, append the value of r. Fill the least significant bit with zero.

(2) Determine the ~~a~~ two least significant bits of P.

(I) If they are 01, find the value of P+A. Ignore any overflow.

(II) If they are 10, find the value of P+S. Ignore any overflow.

(III) If they are 00, do nothing. Use P directly in the next step.

(IV) If they are 11, do nothing. Use P directly in the next step.

JMI SYRINGES & MEDICAL DEVICES LTD.

Scanned by CamScanner

(3) Arithmatically shift the values obtained in the ~~next~~ 2nd step by a single place to the right. Let P now be equal to this new value.

(4) Repeat steps 2 and 3 untill they have been done y times.

(5) Drop the least significant bit from P. this is the product of m and r.

Q. Find $13 \times (-6)$ using booth's multiplication algorithm where $m = 13$ ~~is~~ $r = -6$, $x = 5$, $y = 5$

$m = 01101$ $\quad -m = 10011$

$r = 06110 \xrightarrow{2s} 11010$

A : 01101 00000 0

S : 10011 00000 0

P : 00000 11010 0

Date:

Step 1 : P : 00000 11010 0

→ P : 00000 01101 0

Step 2 : P+S : 10011 01101 0

→ P : 11001 10110 1

Step 3 : P+A : 00110 10110 1

→ P : 00011 01011 0

Step 4 : P+S : 10110 01011 0

→ P : 11011 00101 1

Step 5 : → P : 11101 10010 1

Ans. 11101 10010 = −78

JMI SYRINGES & MEDICAL DEVICES LTD.

Scanned by CamScanner

17 May

Q. 5×5 booth multiplication (For Lab)

x	y	-y
01101 (=13)	11010 (-6)	00110

	U	V	x	x_{-1}
	00000 00110	00000	01101	0
Step 1:	00110 00011 11010	00000 00000	01101 00110	0 1
Step 2:	11101 11110 00110	00000 10000	00110 00011	1 0
Step 3:	00100 00010 00000	10000 01000	00011 00001	0 1
Step 4:	00010 00001 11010	01000 00100	00001 00000	1 1
Step 5:	11011 11101	00100 10010	00000 00000	1 0

Date:

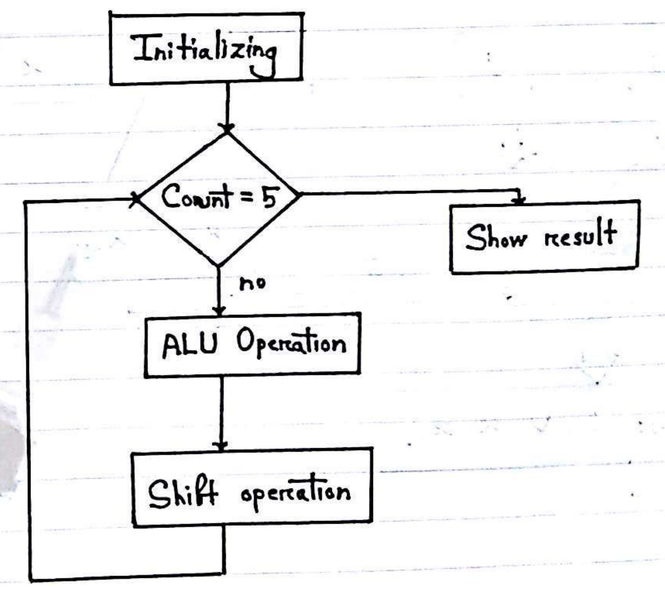
1. Initialization :

U ← 0
V ← 0
x ← input
y ← input
x_{-1} ← 0
count ← 0

ALU Operation :

x_0	x_{-1}	
0	0	U+0
0	1	U+y
1	0	U-y
1	1	U+0

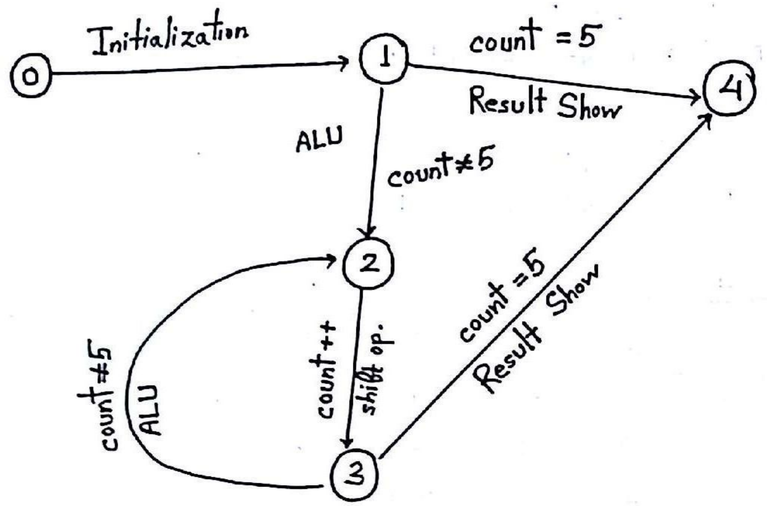
2. Flow Chart :



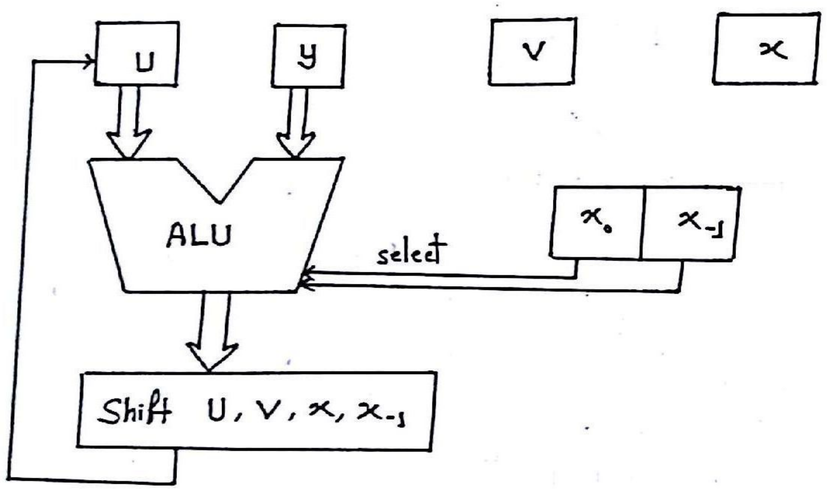
JMI SYRINGES & MEDICAL DEVICES LTD.

Scanned by CamScanner

3. State Diagram :



4. Architecture :



Date:

Hard wired Control :

Example 1 : The design is carried out in five consecutive steps.

1. The problem is stated.
2. An initial equipment configuration is assumed.
3. Algorithm is formulated.
4. The data processor part is specified.
5. The control logic is designed.

Q. The problem here is to implement with hardware the addition and subtraction of two fixed point binary numbers represented in signed magnitude form.

Equipment Configuration :

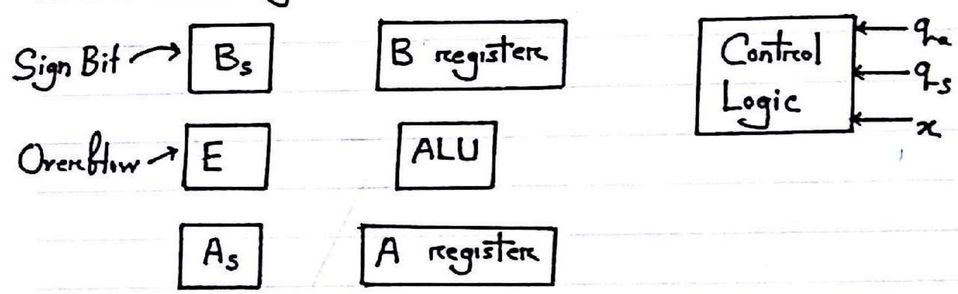


Fig 10.6 : Register Configuration for Adder and Subtractor.

Scanned by CamScanner

Derivation of Algorithm :

$$(\pm A) \pm (\pm B)$$

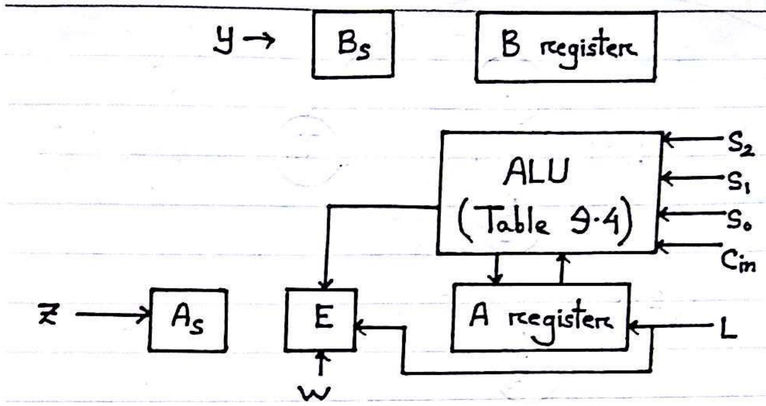
$$(\pm A) - (+B) = (\pm A) + (-B)$$

$$(\pm A) - (-B) = (\pm A) + (+B)$$

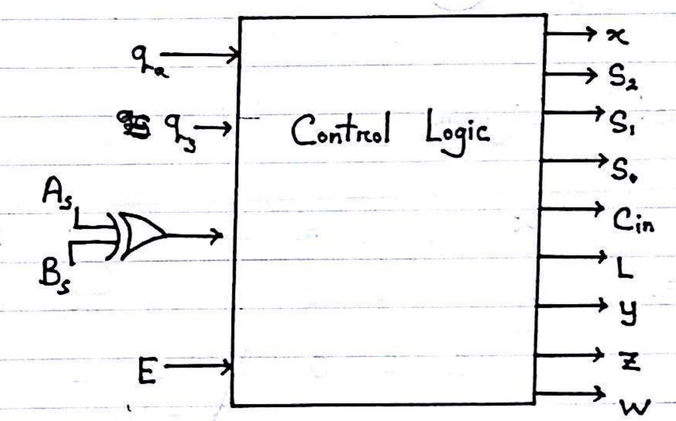
$$(\pm A) + (\pm B)$$

$(+A) + (+B) = +(A+B)$	if $A \geq B$	if $A < B$
$(+A) + (-B) = \longrightarrow$	$+(A-B)$	$-(B-A)$
$(-A) + (+B) = \longrightarrow$	$-(A-B)$	$+(B-A)$
$(-A) + (-B) = -(A+B)$		

Date:



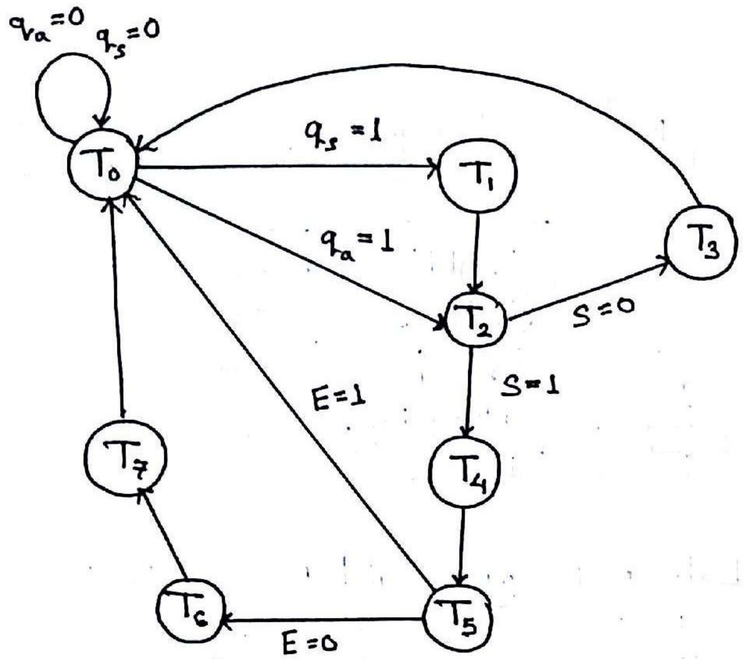
(a) Data processing register and ALU



(b) Control Block Diagram

JMI SYRINGES & MEDICAL DEVICES LTD.

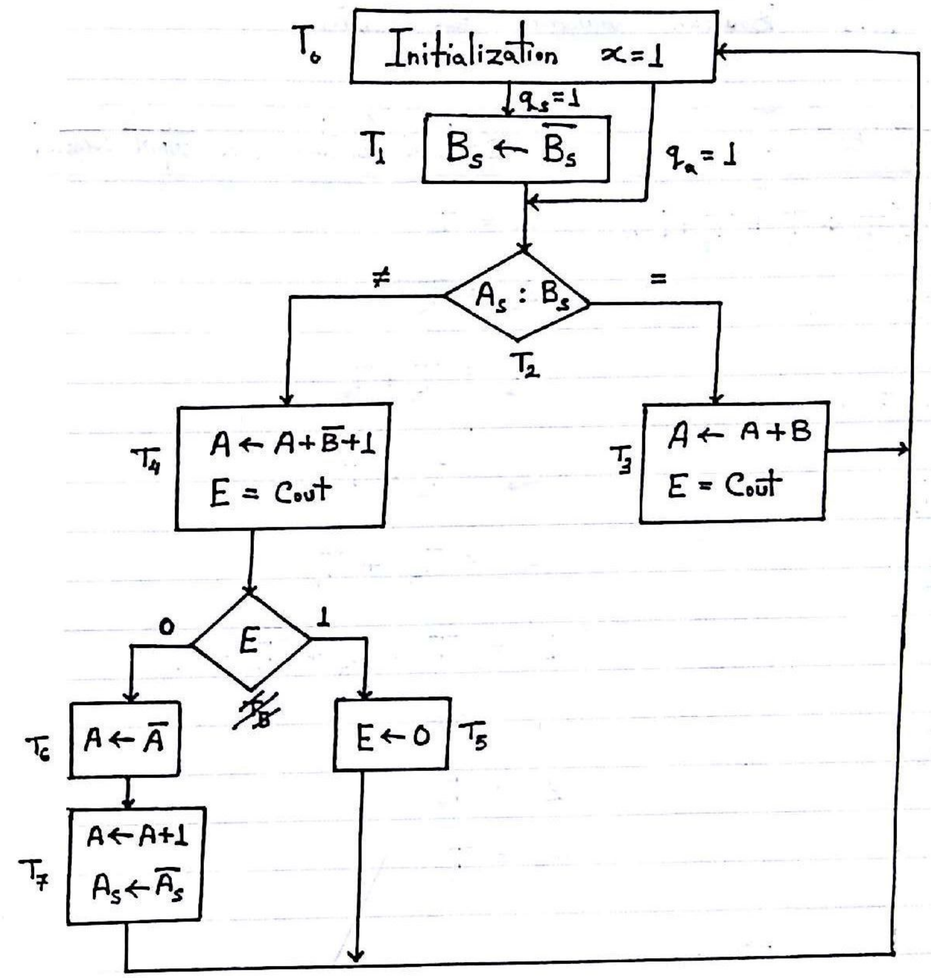
Scanned by CamScanner



(a) State Diagram

	x	S_2	S_1	S_0	C_{in}	L	y	z	w
T_0 : Initial state , $x=1$	1	0	0	0	0	0	0	0	0
T_1 : $B_s \leftarrow \overline{B_s}$	0	0	0	0	0	0	1	0	0
T_2 : nothing	0	0	0	0	0	0	0	0	0
T_3 : $A \leftarrow A+B$, $E \leftarrow cout$	0	0	0	1	0	1	0	0	0
T_4 : $A \leftarrow A+\overline{B}+1$, $E \leftarrow cout$	0	0	1	0	1	1	0	0	0
T_5 : $E \leftarrow 0$	0	0	0	0	0	0	0	0	1
T_6 : $A \leftarrow \overline{A}$	0	1	1	1	0	1	0	0	0
T_7 : $A \leftarrow A+1$, $A_s \leftarrow \overline{A_s}$	0	0	0	0	1	1	0	1	0

Date:



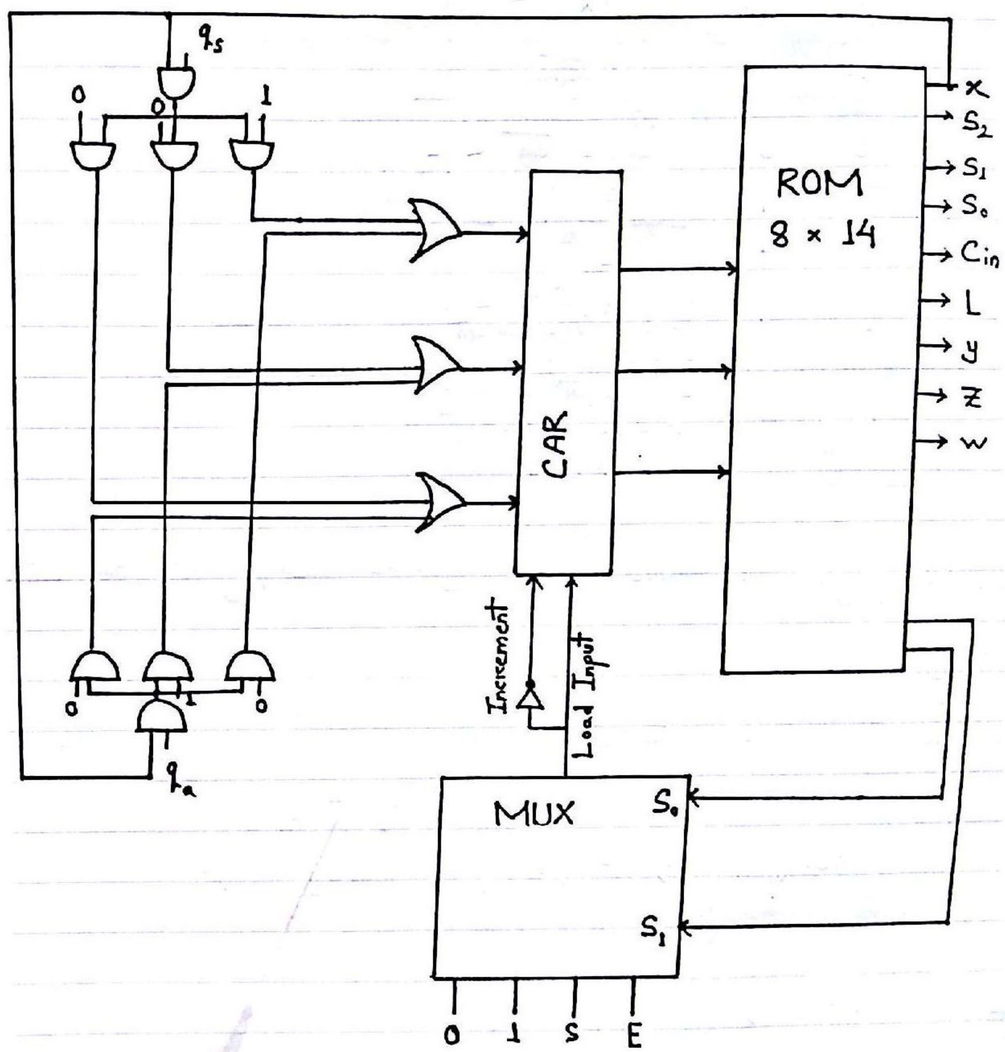
JMI SYRINGES & MEDICAL DEVICES LTD.

Scanned by CamScanner

Table 10.1 Boolean function for control

Flip flop input function	Boolean function for output control
$DT_0 = q_s' q_a' T_0 + T_5 + T_7 + ET_5$	$x = T_0$
$DT_1 = q_s T_0$	$S_2 = T_6$
$DT_2 = q_a T_0 + T_1$	$S_1 = T_4 + T_6$
$DT_3 = S' T_2$	$S_0 = T_3 + T_6$
$DT_4 = S T_2$	$C_{in} = T_4 + T_7$
$DT_5 = T_4$	$L = T_3 + T_4 + T_6 + T_7$
$DT_6 = E' T_5$	$y = T_1$
$DT_7 = T_6$	$z = T_7$
	$w = T_3$

Microprogram Control : Hardware Configuration



JMI SYRINGES & MEDICAL DEVICES LTD.

Scanned by CamScanner

ROM bits 13	14	MUX Select Function
0	0	Increament Carc
0	1	Load input to Carc
1	0	Load Input to Carc if S=1, Increament Carc if S=0
1	1	Load input to Carc if E=1, increment Carc if E=0

The microprogram : Table 10-2 : Symbolic microprogram for control memory.

ROM address	Micro instruction	Comments
0	$x=1$, if $q_s=1$ then (go to 1) if $q_a=1$ then (go to 2) if $(q_a \wedge q_s)=0$ then (go to 0)	
1	$B_s \leftarrow \overline{B_s}$	
2	If $(s=1)$ then (go to 4)	
3	$A \leftarrow A+B$, $E \leftarrow C_{out}$, go to 0	
4	$A \leftarrow A+\overline{B}+1$, $E \leftarrow C_{out}$	
5	If $E=1$ then (go to 0), $E \leftarrow 0$	
6	$A \leftarrow \overline{A}$	
7	$A \leftarrow A+1$, $A_s \leftarrow \overline{A_s}$, go to 0	

Date.....

Table 10-3 : Binary microprogram for control memory

ROM address			x	S_2	S_1	S_0	C_{in}	L	y	z	w	Address			Select	
			1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	1
0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
0	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	1
1	0	0	0	0	1	0	1	1	0	0	0	1	0	1	0	1
1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1
1	1	0	0	1	1	1	0	1	0	0	0	1	1	1	0	1
1	1	1	0	0	0	0	1	1	0	1	0	0	0	0	0	1

JMI SYRINGES & MEDICAL DEVICES LTD.

Scanned by CamScanner

10.6 : Hard wired Control : Example -2 :

1. Statement of the problem : We wish to design an arithmatic circuit that multiplies two fixed point binary numbers sign magnitude representation.

multiplicand : 10 111

multiplier : 10011

1st multiplier bit = 1 , copy multiplicand : 10 111

Shift right to obtain 1st product : 010111

2nd multiplier bit = 1 , copy multiplicand : 10111

add multiplicand to previos partial product : 100010

Shift right to obtain 2nd partial product : 1000101

3rd multiplier bit = 0 , Shift right to obtain PP : 01000101

4th " " = 0 , " " " " " : 001000101

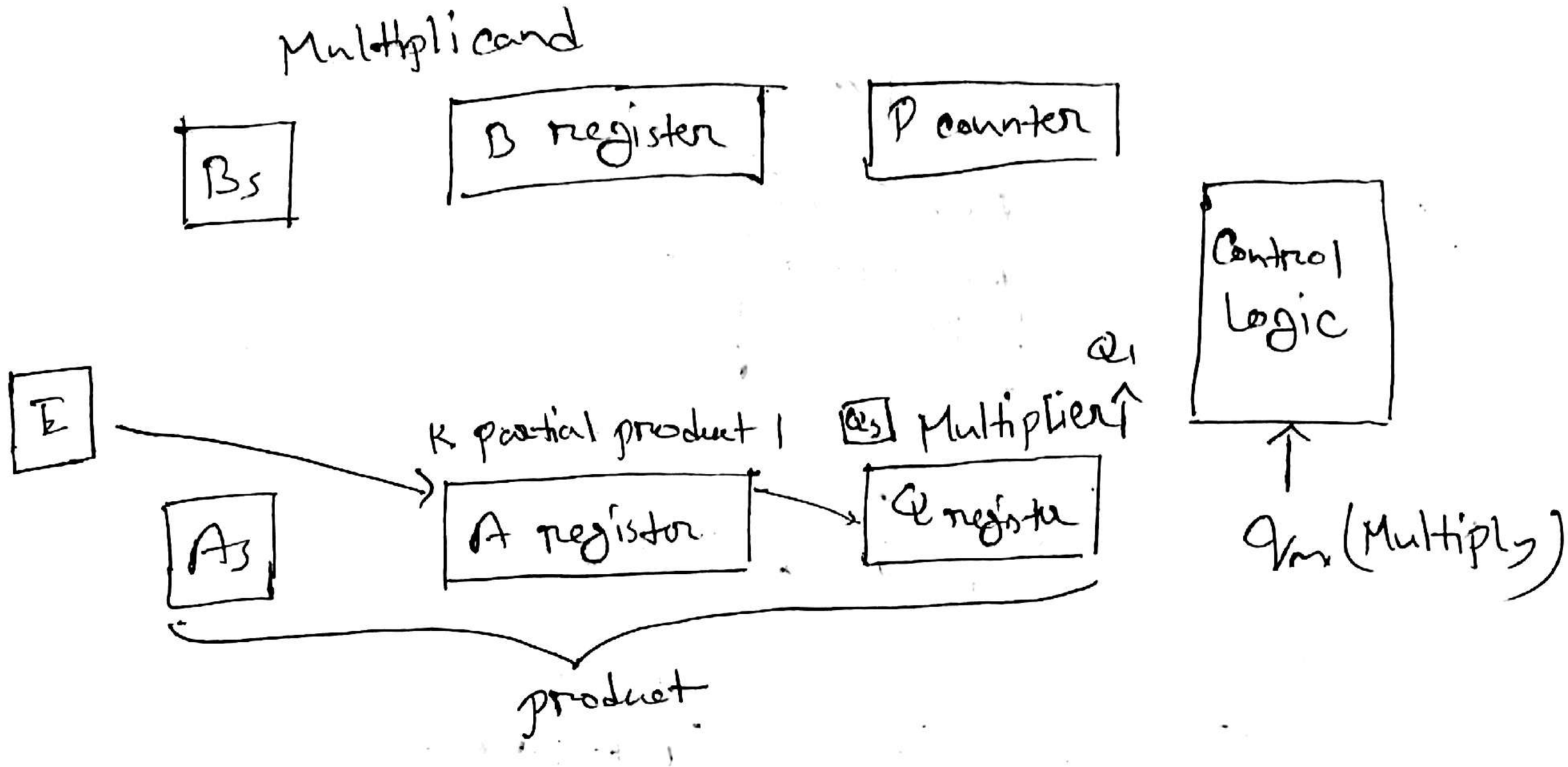
5th " " = 1 , copy multiplicand : 10111

add multicand to previous PP : 110110101

Shift right to obtain 5th PP : 01101101 01

Scanned by CamScanner

(2) Equipment Configuration



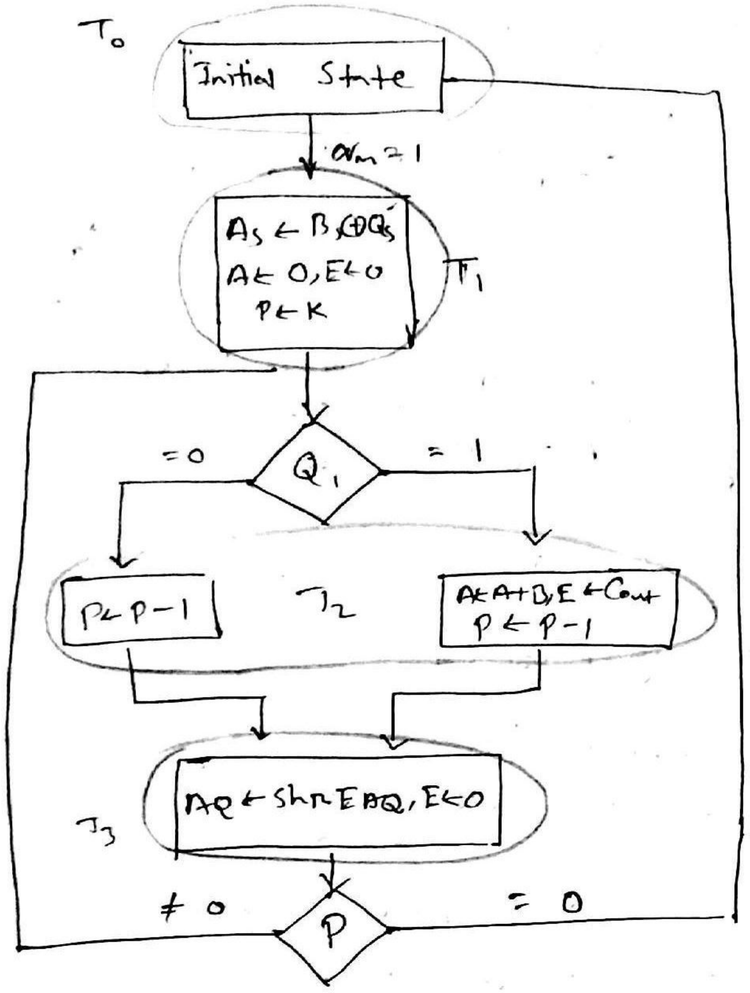
(3) Derivation of Algorithm

Initially the multiplicand is in B and the multiplier in Q. Their corresponding signs are in Bs and Qs.

P is set to a binary number k, which is equal to the number of bits in multiplier

The shift operation is symbolized

$AQ \leftarrow \text{shr } EAQ, \; E \leftarrow 0$



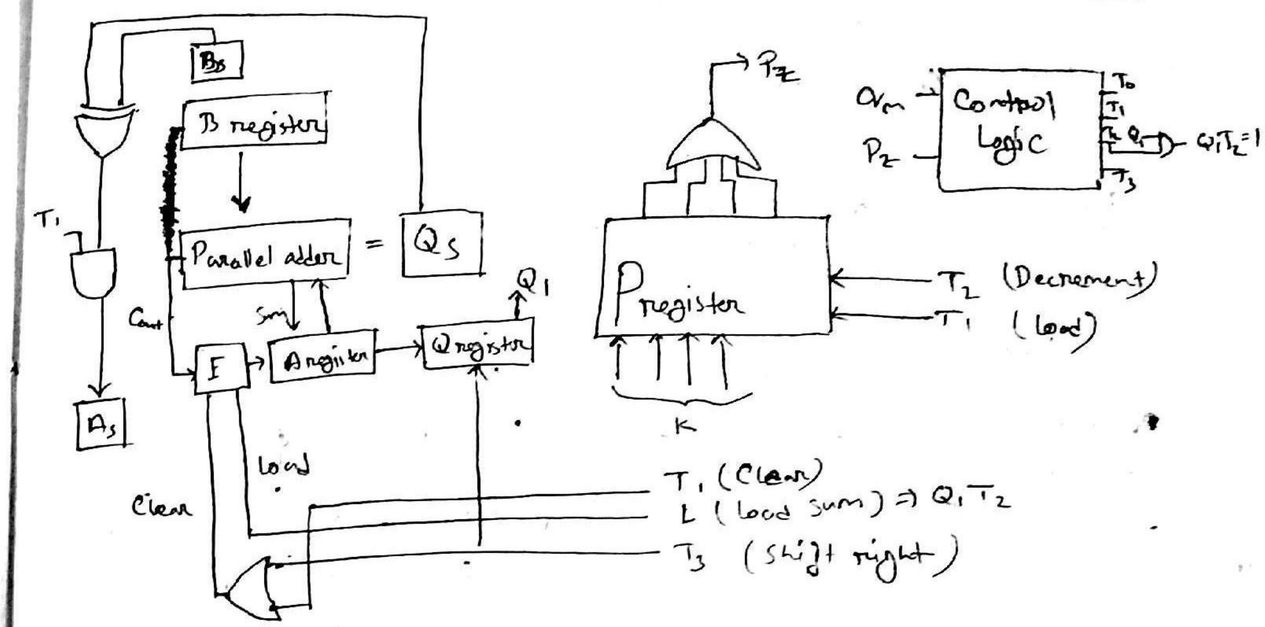
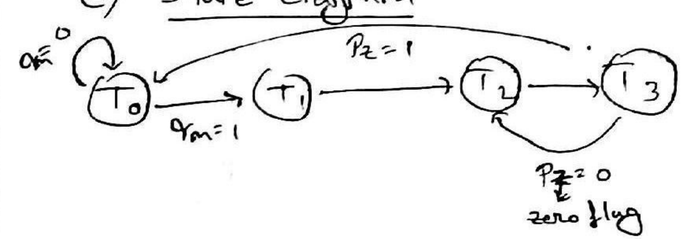


Fig 10-6 : Data Processor for binary multiplier

(a) State diagram



(b) Sequence of register transfer

T_0 : Initial state

T_1 : $A_s \leftarrow B_s \oplus Q_s$, $A \leftarrow 0, E \leftarrow 0, P \leftarrow K$

Q_1T_2 : $A \leftarrow A+B, E \leftarrow Cout$

T_2 : $P \leftarrow P-1$

T_3 : $AQ \leftarrow shr\ E\ AQ, E \leftarrow 0$

Fig-10-15 : Control state diagram and sequence of microoperation for multiplier

Lecture-17 4.05.17

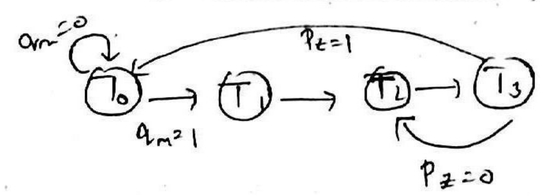
5 Design of Hardwired Control

Fig 10-17 : (a,b,c) Design of control for binary multiplier

(a) Execution Table

	Present State		Inputs		Next State		Flip-Flop inputs			
	G_2	G_1	q_m	P_z	G_2	G_1	JG_2	KG_2	JG_1	KG_1
T_0	0	0	0	X	0	0	0	X	0	X
T_0	0	0	1	X	0	1	0	X	1	X
T_1	0	1	X	X	1	0	1	X	X	1
T_2	1	0	X	X	1	1	X	0	1	X
T_3	1	1	X	0	1	0	X	0	X	1
T_3	1	1	X	1	0	0	X	1	X	1

P.S	N.S	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0



(b) Flip-Flop input function :

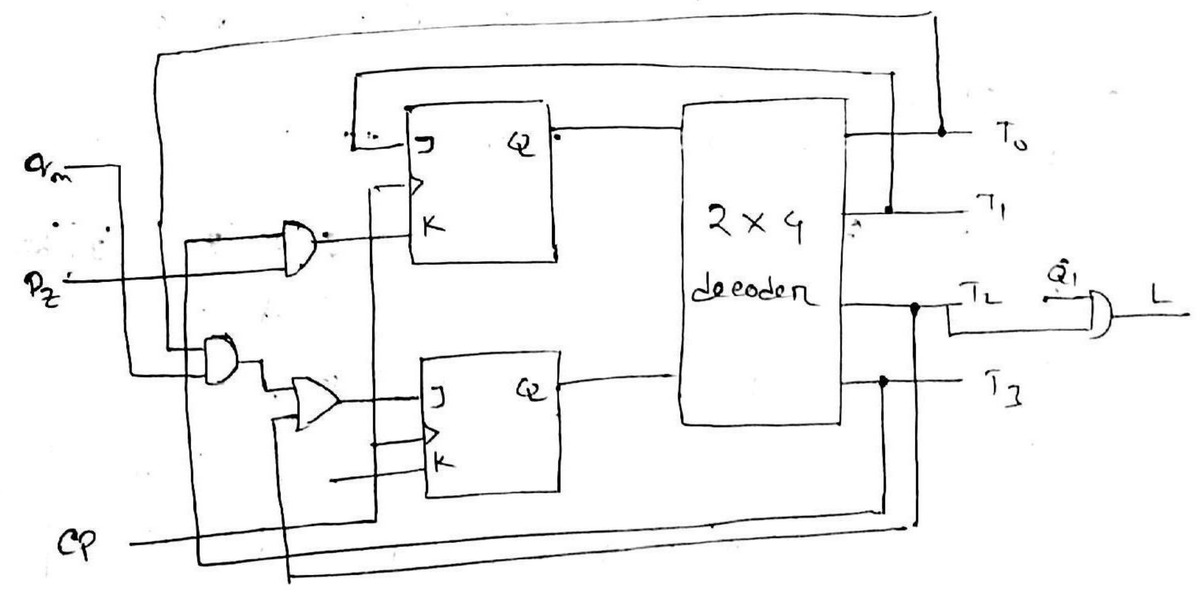
$JG_2 = T_1$

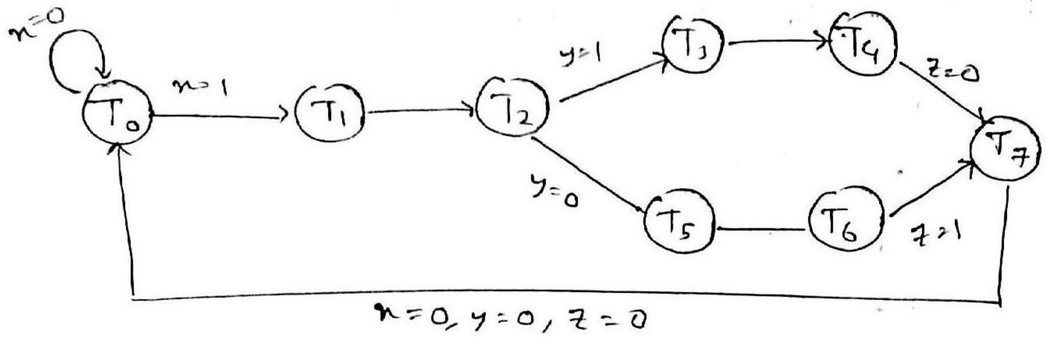
$KG_2 = T_3 P_z$

$JG_1 = T_0 q_m + T_2$

$KG_1 = 1$

(c) Logic Diagram





Design the control by the sequence register and decoder method with 3 JK flip-flop G_3, G_2, G_1

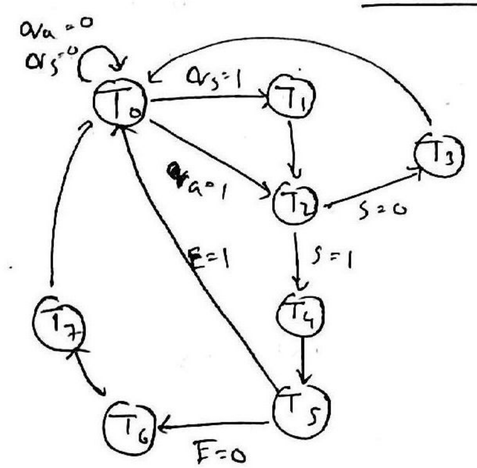
(c) Execution table

Present State			Inputs			Next State			Flip-Flop Input					
G_3	G_2	G_1	x	y	z	G_3	G_2	G_1	JG_3	KG_3	JG_2	KG_2	JG_1	KG_1
0	0	0	0	X	X	0	0	0	0	X	0	X		
0	0	0	1	X	X	0	0	1	0	X	0	X		
0	0	1	X	X	X	0	1	0	0	X	1	X		
0	1	0	X	1	X	0	1	1	0	X	X	0		
0	1	0	X	0	X	1	0	1	1	X	X	1		
0	1	1	X	X	X	1	0	0	1	X	X	1		
1	0	0	X	X	0	1	1	1	X	0	1	X		
1	0	1	X	X	X	1	1	0	X	0	1	X		
1	1	0	X	X	1	1	1	1	X	0	X	0		
1	1	1	0	0	0	0	0	0	X	1	X	1		

15-01-04-133

Lecture - 18

7.06.17



Design the control by the sequence register and decoder method Use 3 J-K flip-flop G_3, G_2 and G_1

Present State			Inputs				Next State			Flip-Flop Input					
G_3	G_2	G_1	Qs	Qa	E	S	G_3	G_2	G_1	KG_3	JG_3	KG_2	JG_2	KG_1	JG_1
0	0	0	0	0	X	X	0	0	0						
0	0	0	X	X	X	X	0	0	1						
0	0	0	X	1	X	X	0	1	0						
0	0	1	X	X	X	X	0	1	0						
0	1	0	X	X	X	0	0	1	1						

Scanned by CamScanner

15.01.04.115

	Present state	Inputs	Next state	Flip-flop inputs		
	$G_3\ G_2\ G_1$	$q_s\ q_a\ E\ s$	$G_3\ G_2\ G_1$	$JG_3\ KG_3$	$JG_2\ KG_2$	$JG_1\ KG_1$
T_0	0 0 0	0 0 X X	0 0 0	0 X	0 X	0 X
T_0	0 0 0	1 X X X	0 0 1	0 X	0 X	1 X
T_0	0 0 0	X 1 X X	0 1 0	0 X	1 X	0 X
T_1	0 0 1	X X X X	0 1 0	0 X	1 X	X 1
T_2	0 1 0	X X X 0	0 1 1	0 X	X 0	1 X
T_2	0 1 0	X X X 1	1 0 0	1 X	X 1	0 X
T_3	0 1 1	X X X X	0 0 0	0 X	X 1	X 1
T_4	1 0 0	X X X X	1 0 1	X 0	0 X	1 X
T_5	1 0 1	X X 0 X	1 1 0	X 0	1 X	X 1
T_5	1 0 1	X X 1 X	0 0 0	X 1	0 X	X 1
T_6	1 1 0	X X X X	1 1 1	X 0	X 0	1 X
T_7	1 1 1	X X X X	0 0 0	X 1	X 1	X 1

Scanned by CamScanner

$JG_3 = T_2 S$

$KG_3 = T_5 E + T_7$

$JG_2 = T_0 q_a + T_1 + T_5 E'$

$KG_2 = T_2 S + T_3 + T_7$

$JG_1 = T_0 q_5 + T_2 S' + T_4 + T_6$

$KG_1 = 1$

10.06 2017

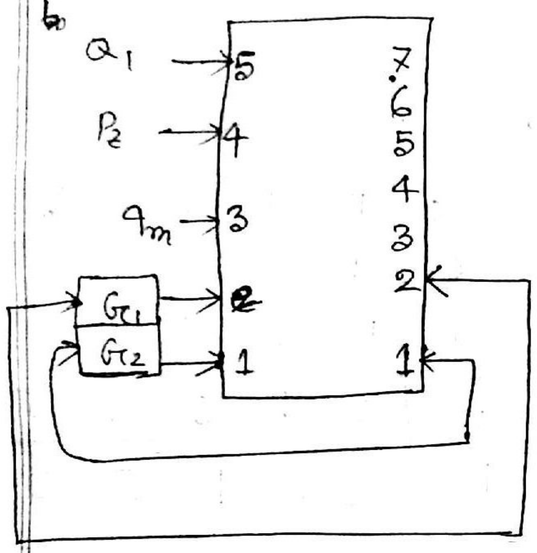
10-7 PLA control :

Table 10-6: state table for control circuit.

Present state		Inputs			Next state		outputs				
G_{c_2}	G_{c_1}	q_m	P_2	Q_1	G_{c_2}	G_{c_1}	T_0	T_1	T_2	L	T_3
0	0	0	X	X	0	0	1	0	0	0	0
0	0	1	X	X	0	1	1	0	0	0	0
0	1	X	X	X	1	0	0	1	0	0	0
1	0	X	X	0	1	1	0	0	1	0	0
1	0	X	X	1	1	1	0	0	1	1	0
1	1	X	1	X	0	0	0	0	0	0	1
1	1	X	0	X	1	0	0	0	0	0	1

Fig 10-18: PLA control for binary multiplier

(a) Block diagram



(b) PLA program table

Product terms	Inputs 1 2 3 4 5	Outputs 1 2 3 4 5 6 7	Comments
1	0 0 0 - -	- - 1 - - - - -	$T_0=1, q_m=0$
2	0 0 1 - -	- 1 1 - - - - -	$T_0=1, q_m=1$
3	0 1 - - -	1 - - 1 - - - -	$T_1=1$
4	1 0 - - 0	1 1 - - 1 - - -	$T_2=1, Q_1=0$
5	1 0 - - 1	1 1 - - 1 1 - -	$T_2=1, L=1, Q=1$
6	1 1 - 1 -	- - - - - - 1 -	$T_3=1, P_2=0$
7	1 1 - 0 -	- - - - - - - 1	$T_3=1, P_2=1$

02-07-2017
quiz-2
Chapter-10

05.07.2017

Sap - I

Simple - As - possiple computer

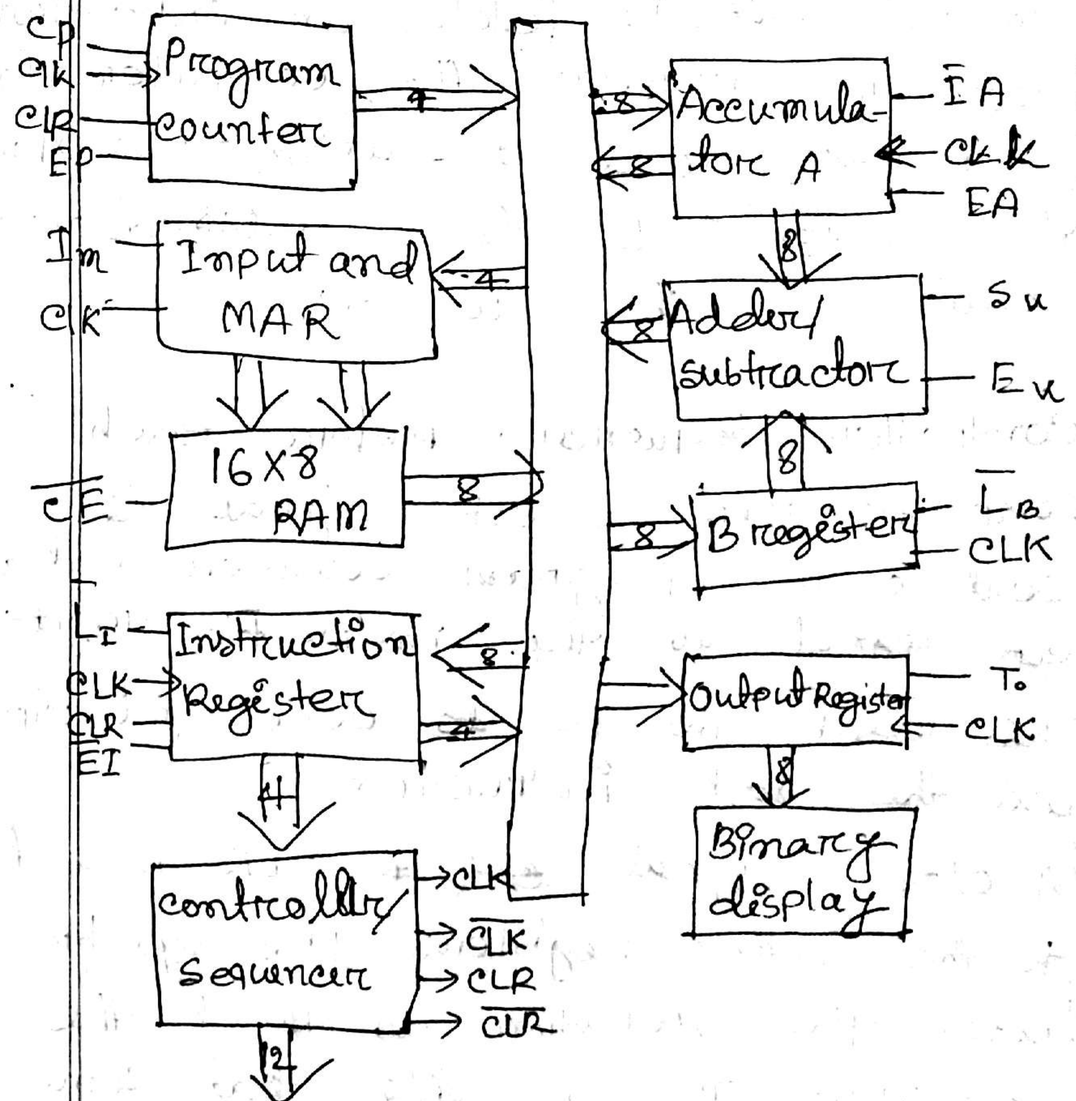
10-1 Architecture

Programm counter : is a part of the control unit, counts from 0000 to 1111. Its job is to send to the memory the address of the ~~me~~ next instruction to be fetched and executed.

Input and MAR: It includes the address and data with switch registers Allow you to send 4 addresses bits and 8 data bits to the ~~Ram~~ RAM. Instruction and data words are written into the RAM before a computer run.

The RAM: The RAM is a 16X8 static TTL RAM. you can programe the RAM by means of the address and data

~~swith~~ switch register.



Instruction Register: It is a part of control unit. To fetch an instruction from the memory the computer does a memory read operation. This places the contents of the addressed memory location on the W bus.

Controller sequencer: Before each computer run, a $\overline{CLR}$ signal is sent to the program counter and a CLR signal to the instruction register. This resets the PC to 0000 and wipes out the last instruction.

A clock signal ~~ckk~~ CLK is sent to the buffer registers, this synchronizes the operation ~~to~~ ~~th~~ of the computer. The 12 bits that come out of the controller - sequencer form a word controling the reset of the computer

$$CON = C_P E_P I_m \overline{CE} \; I_I \overline{E_I} \; L_A E_A \; S_V E_V \; L_B$$

Accumulator: Accumulator (A) is a buffer register that stores intermediate answers during a computer run. ~~to~~

The Adder - Subtractor: SAP-I ~~wh~~ was a 2's complement of adder - subtractor. When S_V is low the sum is $A = A + B$ and when S_V is high the sum is $A = A + \overline{B}$

B register: The B register is another buffer register. It is used in arithmetic operations.

output Register: At the end of the computer run, the accumulator contains the answer to the problem

being solved when E_A is high and I_0 is low, the next positive clock loads the A word in the output register.

Binary Display: It is a row of the eight light emeting diodes (LEDs)

~~0000 0101~~

~~000~~

0001 0000
0010 0000
0011 0000
0000 0101
0011 0101

0011 0101
0000 0001
~~010~~
0011 0110
3 6

08.07.2017

10-2: Instruction set

LDA

Load the accumulator

ADD

Add the content of memory location "...H" to the accumulator content

Address	Data/Ins → 8bit
0H	
1H	LDA 9H →
2H	9 নং address
⋮	তার data কাছে
	যাবে Load করে
FH	data → 8 bit
	Ins → 4 bit

LDA 9H
ADD AH
Out

SUB

SUB the content of memory location "AH" to the accumulator content

HLT

HLT, OUT এর কোন address লাগবে না

CON = 12 bit

Mnemonics :

LDA
SUB

Example 10-1

Address	Mnemonics
0 H	LDA 9H
1 H	ADD AH
2 H	ADD BH
3 H	ADD CH
4 H	out
5 H	HLT
6 H	FFH
7 H	FFH
8 H	FFH
9H	10H
AH	20H
BH	05H
CH	FFH
DH	:
EH	FFH
FH	

Output = 36 H