

Quiz-1 [14.05.17] (Sunday)

Up to Chapter-9

इन्हीं Question के उत्तर नहीं।

11 May 2017

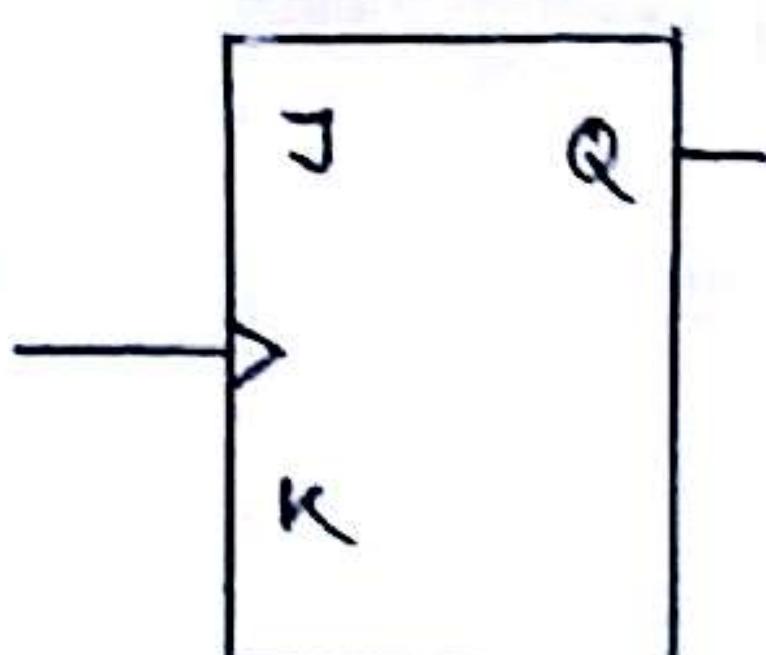
J-K flip flop

-T.I.T (Truth Table)

-	0	0	#id	0	0
.	0	1	Reset	0	0
.	0	1		1	0
-	1	0	Set	0	1
-	1	0		1	1
-	1	1	Toggle	0	1
-	1	1		1	0

Excitation Table

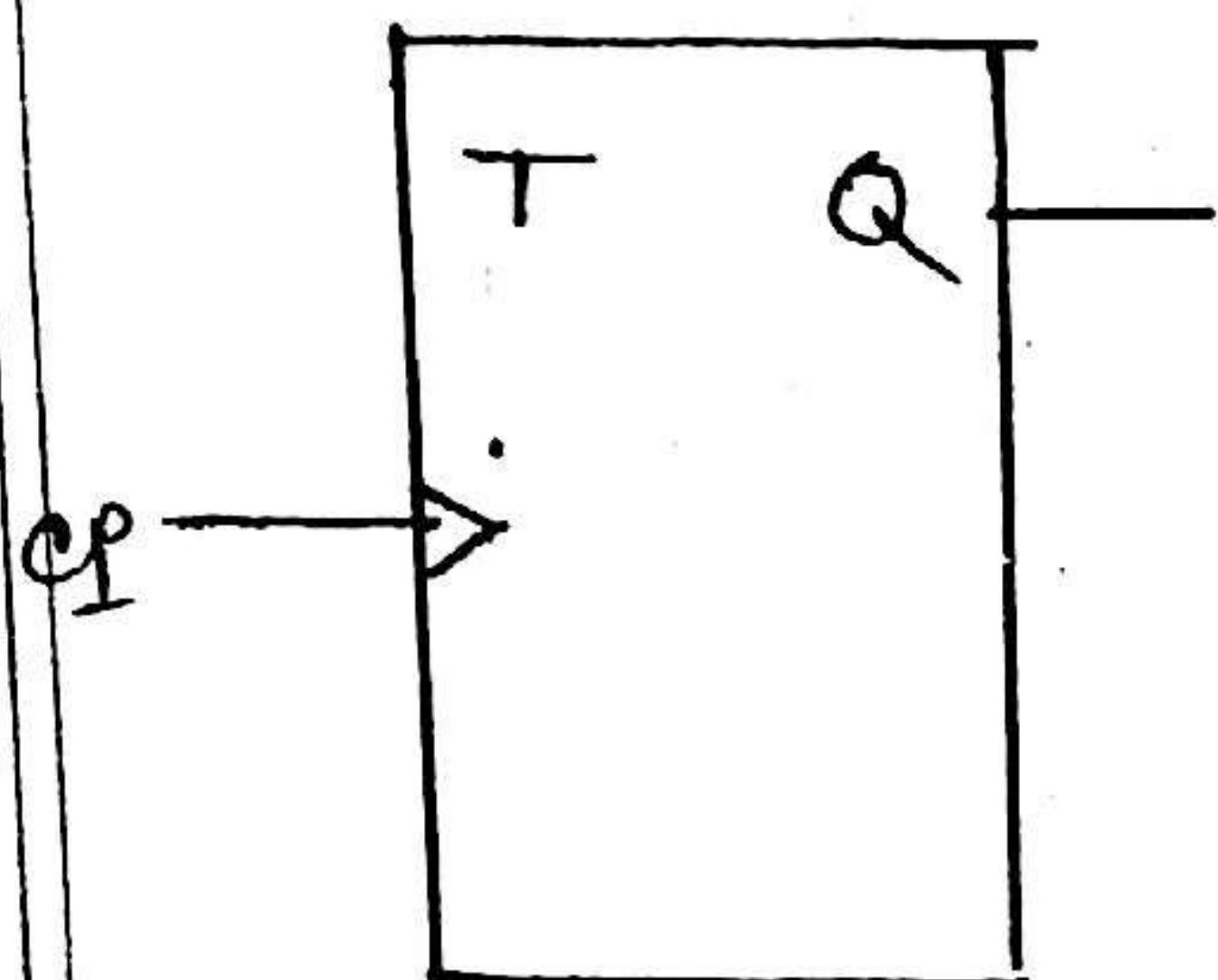
P.S	N.S	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0



T - flip flop

0 → Hold

1 → Toggle

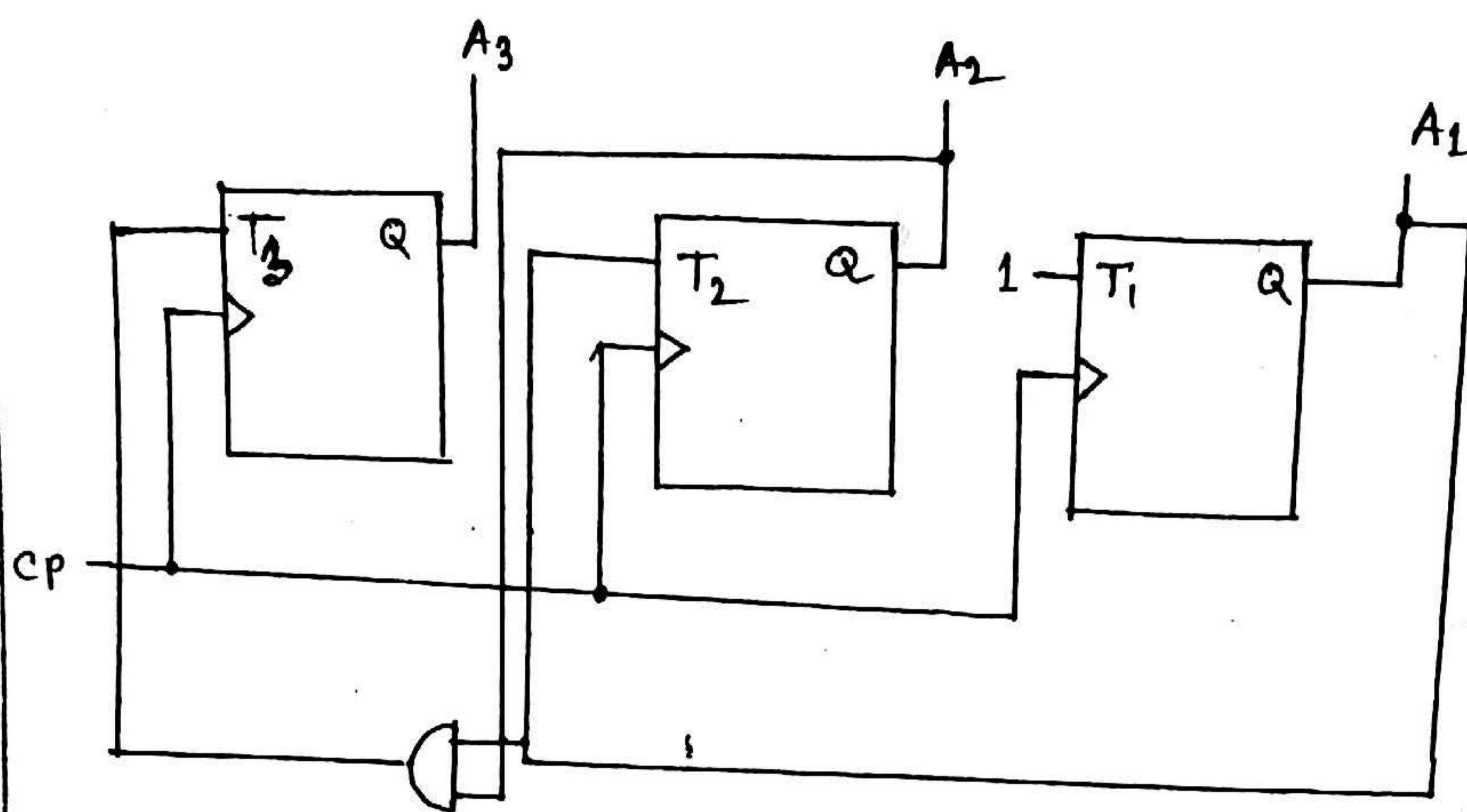


Excitation Table

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

3-bit binary Counter

R.S.			N.S.		
A ₃	A ₂	A ₁	A ₃	A ₂	A ₁
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0



$$T_1 = 1$$

$$T_2 = A_1$$

$$T_3 = A_2 A_1$$

Q. Design a 4-bit / 5-bit counter or specify
কোন flip flop (সিদ্ধান্ত) দিয়ে কাউন্ট করা হবে।

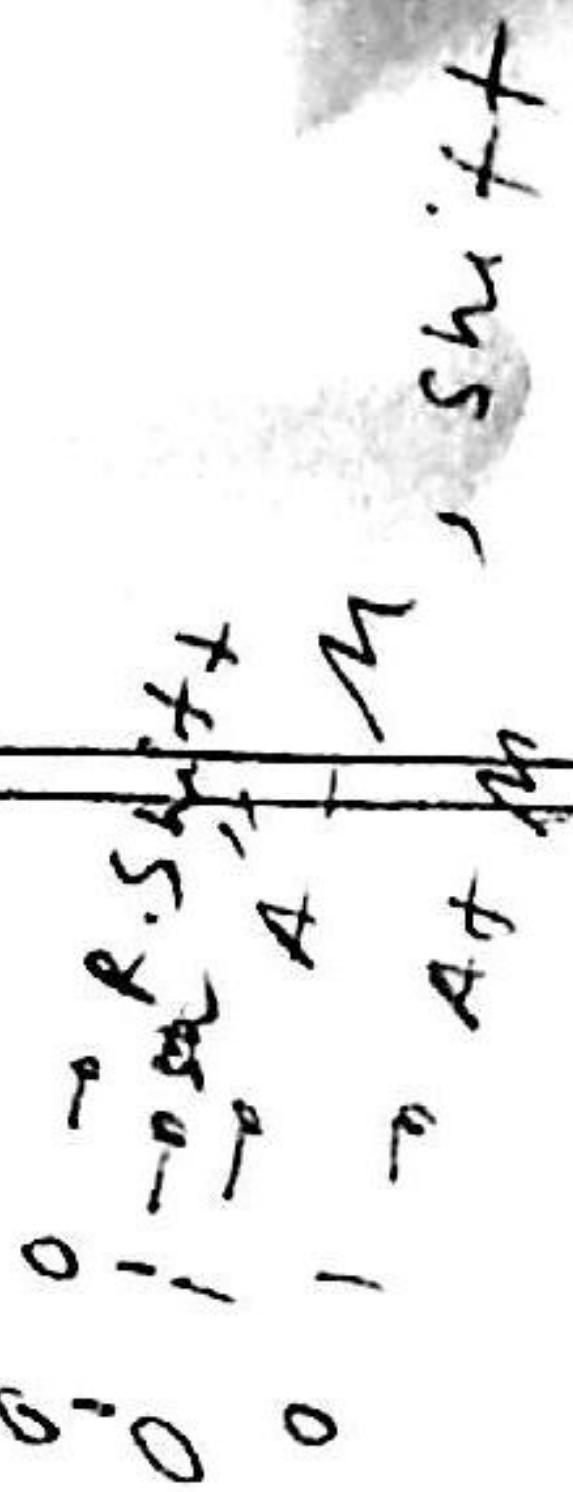
* Booth's "D-flip flop" এর ব্যবহার করা হবে।

D₁ with JK flip flop:

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	0	1	0	1	0	0	X				
0	1	0	0	1	1	0	X				
0	1	1	1	0	0	1	X				
1	0	0	1	0	1	X	0				
1	0	1	1	1	0	X	0				
1	1	0	1	1	1	X	0				
1	1	1	0	0	0	X	1				

Ans (1)

13 May 2017



Booth's Multiplication Algorithm

Booth's algorithm can be implemented by repeatedly adding one of two predetermined values A and s to a product p , then performing a rightward arithmetic shift on p .

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

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

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

⑪ S: fill the most significant bits with the value of $(-m)$ in two's complement notation. Fill the remaining $(y+1)$ bits with zeros.

⑫ P: fill the most significant x bits with zeros. To the right of this, append the value of n . Fill the least significant (right most) bit with a zero.

2] Determine the two least significant bits of P.

① If they are 01, find the value of $P+A$.
Ignore any overflow.

② If they are 10, find the value of $P+S$.
Ignore any overflow.

③ If they are 00, do nothing, use P directly in next step.

④ If they are 11, do nothing, use p directly
in next step.

③ Arithmetically shift the value obtained in
the 2nd step by a single place to the right.
Let p now equal this new value.

④ Repeat steps 2 and 3 until they have
been done j times.

⑤ Drop the least significant (right most) bit
from p . This is the product of m and n .

* Find $13 \times (-6)$ using Booth's algorithm with
 $m = 13$ and $n = -6$, and $x = 5$, $y = 5$

Sol²:

$$m = 01101$$

$$-m = 10011$$

$$\begin{array}{r} 01101 \\ 10011 \\ \hline n = 11010 \end{array}$$

$$A: 01101 \quad 00000 \quad 0$$

$$S: 10011 \quad 00000 \quad 0$$

$$P: 00000 \quad 11010 \quad 0$$

Step 1

$$\begin{array}{r} P: 00000 \quad 11010 \quad 0 \\ \rightarrow P: 00000 \quad 01101 \quad 0 \end{array} ; [right shift]$$

Step 2:

$$P: 00000 \quad 01101 \quad 0$$

$$P+S: 10011 \quad 01101 \quad 0$$

$$\rightarrow P: 11001 \quad 10110 \quad 1$$

Step 3:

$$P+A : \begin{array}{r} 00110 \\ 10110 \\ \hline \end{array} \quad 1$$

$$\rightarrow P : \begin{array}{r} 00011 \\ 01011 \\ \hline \end{array} \quad 0$$

Step 4:

$$P+S : \begin{array}{r} 00011 \\ 10011 \\ \hline \end{array} \quad 01011 \quad 0$$

$$\begin{array}{r} 10011 \\ \hline 00000 \end{array}$$

$$\begin{array}{r} 10110 \\ 01011 \\ \hline \end{array} \quad 0$$

$$\rightarrow P : \begin{array}{r} 11011 \\ 00101 \\ \hline \end{array} \quad 1$$

Step 5:

$$\rightarrow P : \begin{array}{r} 11101 \\ 10010 \\ \hline \end{array} \quad 1$$

Ans. 11101 10010

- 78

* $xm \rightarrow$ multiplication করার পথ মন্তব্য করা
 Booth's algorithm করা procedure 3 ব্যবহৃত হয়।

: 9

: 2 + 9

: 9 -

on ३x३ एन्ड यूट्यू- example निम्न design कर सकते, but
algorithm should be followed.

14 May 2017

5x5 bit Booth Multiplier

00	$\rightarrow 0$
01	$\rightarrow +y$
10	$\rightarrow -y$
11	$\rightarrow 0$

$$\begin{array}{c}
 X \\
 \hline
 01101 \quad (13) \\
 \end{array}
 \quad
 \begin{array}{c}
 Y \\
 \hline
 11010 \quad (-6) \\
 \end{array}
 \quad
 \begin{array}{c}
 -Y \\
 \hline
 00110 \\
 \end{array}$$

U	V	X	X_{-1}
00000	00000	01101	0 } 1st step
00110		01101	0 }
00110	00000	00110	1 }
(Right shift)	00011		
	00000		
11010			
11101	00000	00110	1 } 2nd step
(R.S.) 11110	10000	00011	0 }
00110			
00100	10000	00011	0 } 3rd step
(R.S.) 00010	01000	00001	1 }
00000			
00010	01000	00001	1 } 4th step
(R.S.) 00001	00100	00000	1 }
11010			
11011	00100	00000	1 } 5th step
(R.S.) 11101	10010	00000	0 }

* xm → 28th Jan 2021

1 Initialization :

$U \leftarrow 0$

$V \leftarrow 0$

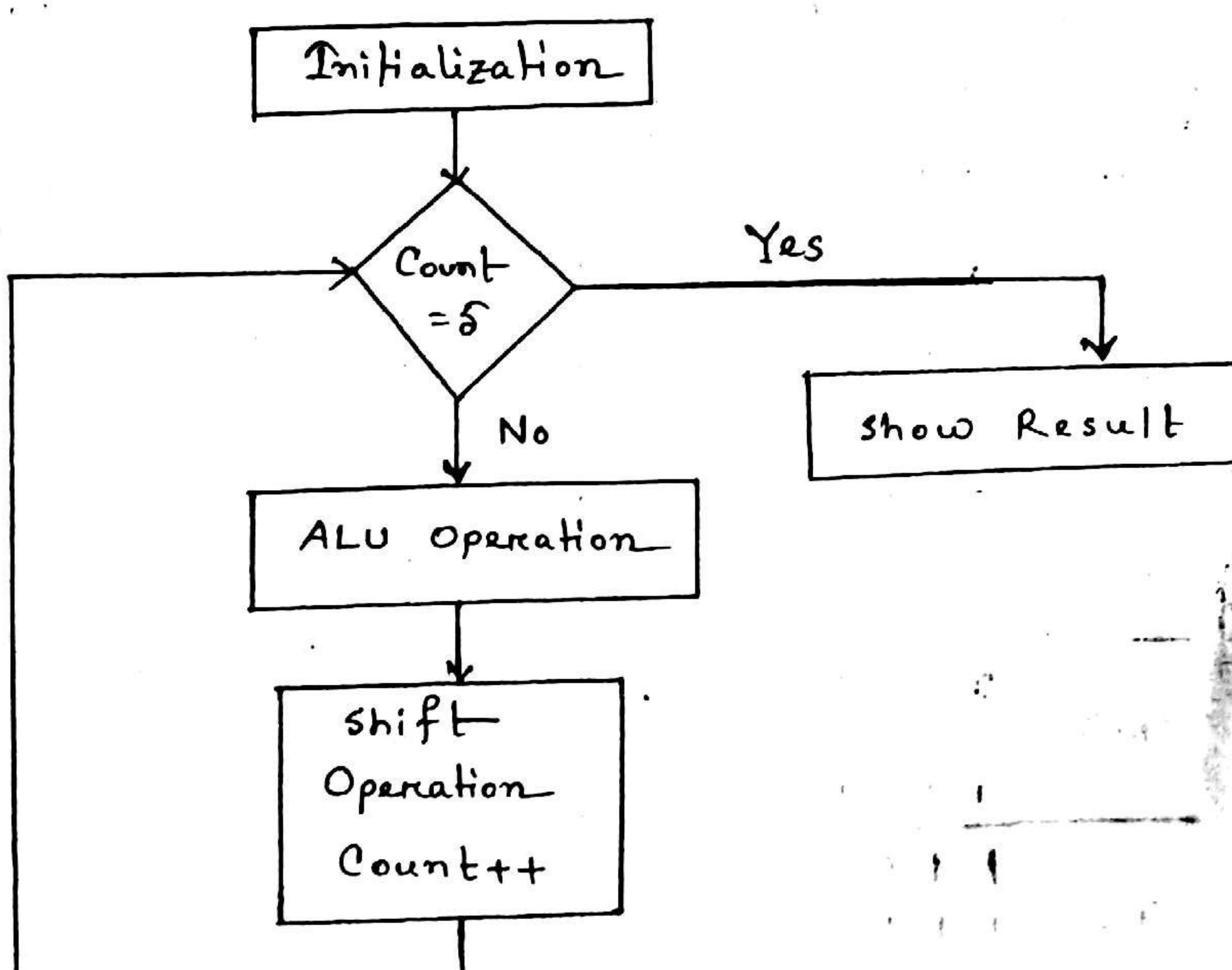
$X \leftarrow \text{Input}$

$Y \leftarrow \text{Input}$

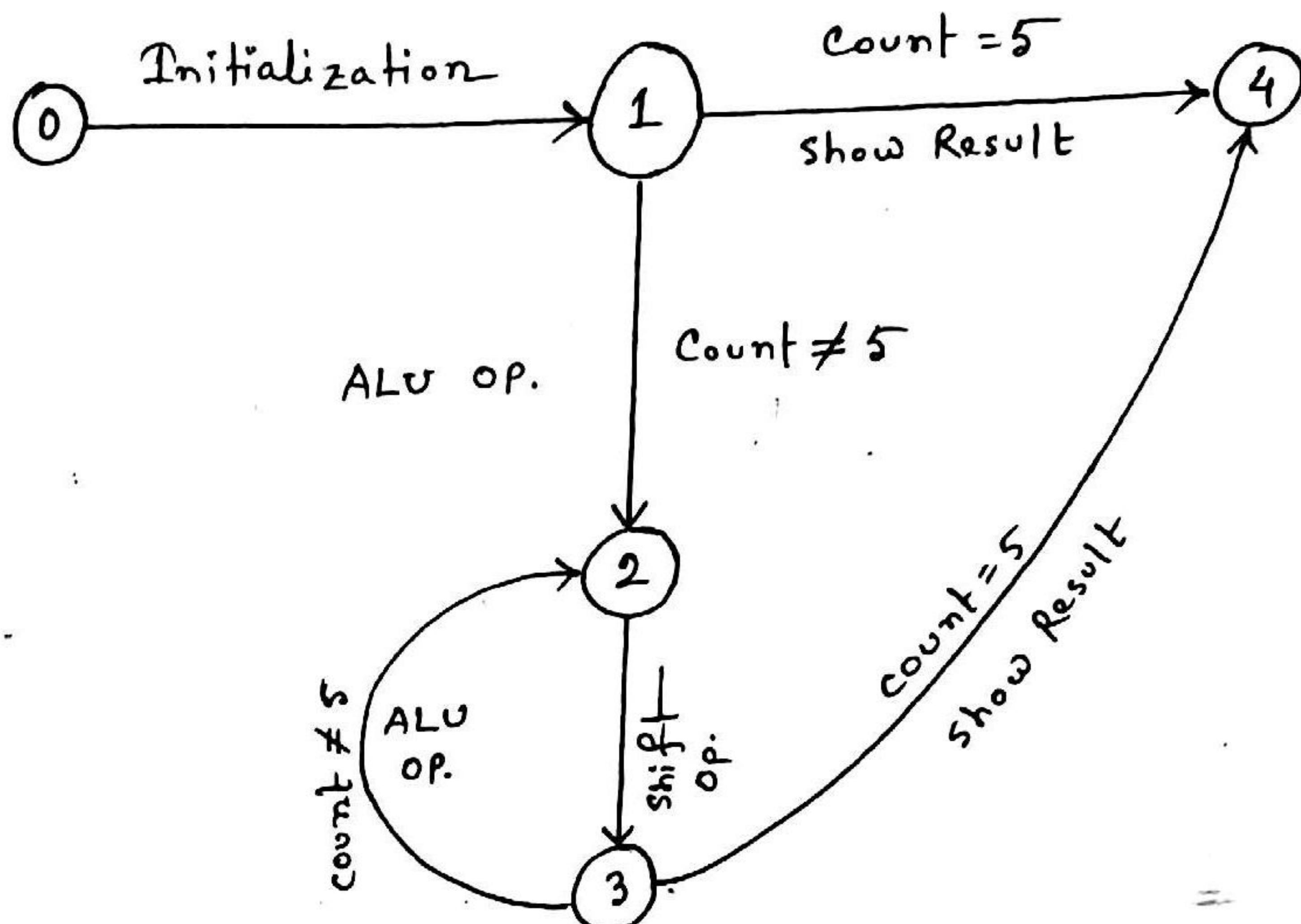
$X_{-1} \leftarrow 0$

Count $\leftarrow 0$

2 Flow Chart :



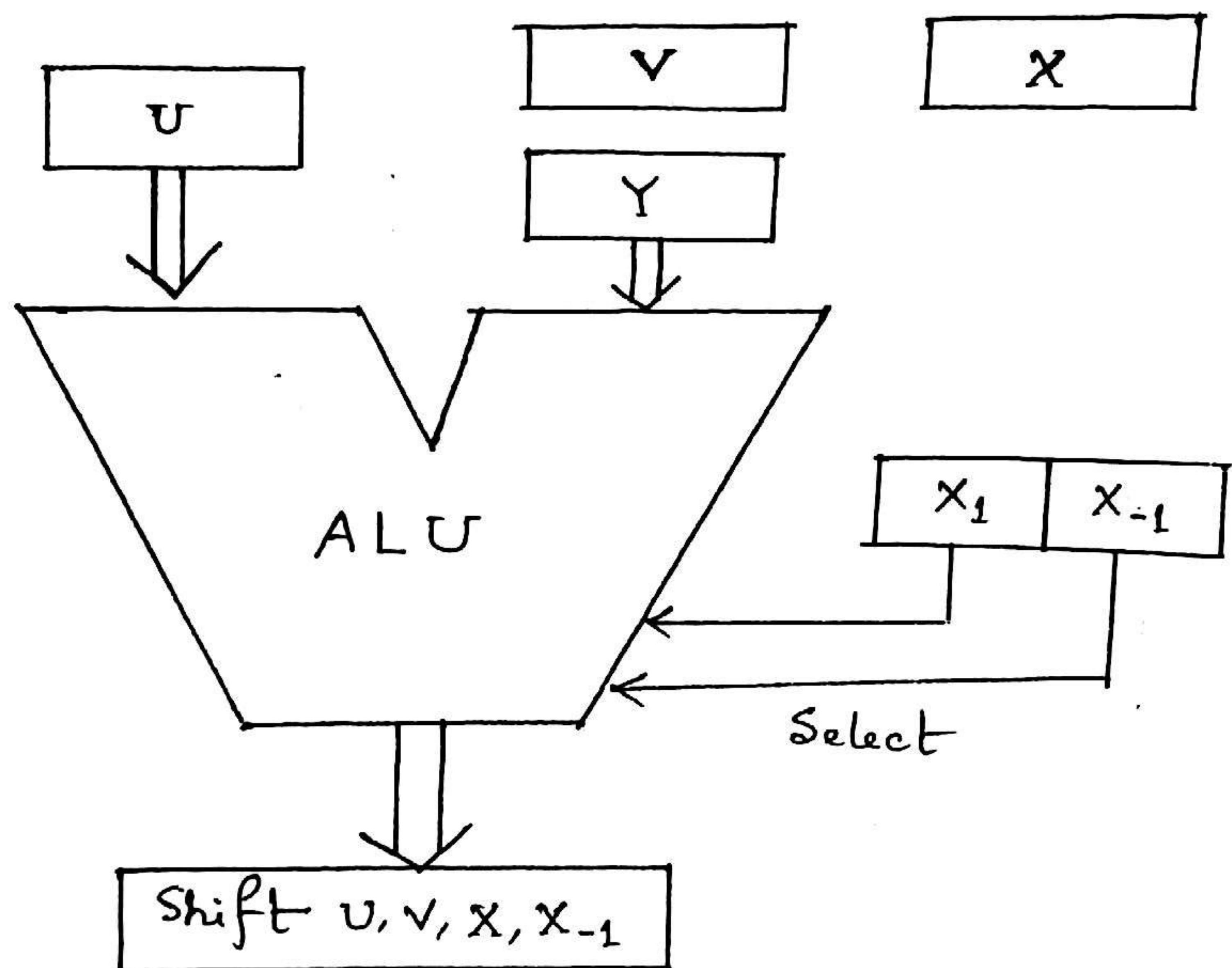
3 State Diagram :



ALU Op.

S_1	S_0	Operation
0	0	$U + 0$
0	1	$U + Y$
1	0	$U - Y$
1	1	$U + 0$

4 Architecture :



18 May 2017

10-3

Hardwired Control

Example - 1

The design is carried out in 5 consecutive steps:

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

① Statement of the problem

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

signbit
0 0101 → +5
1 0101 → -5

2 Equipment Configuration

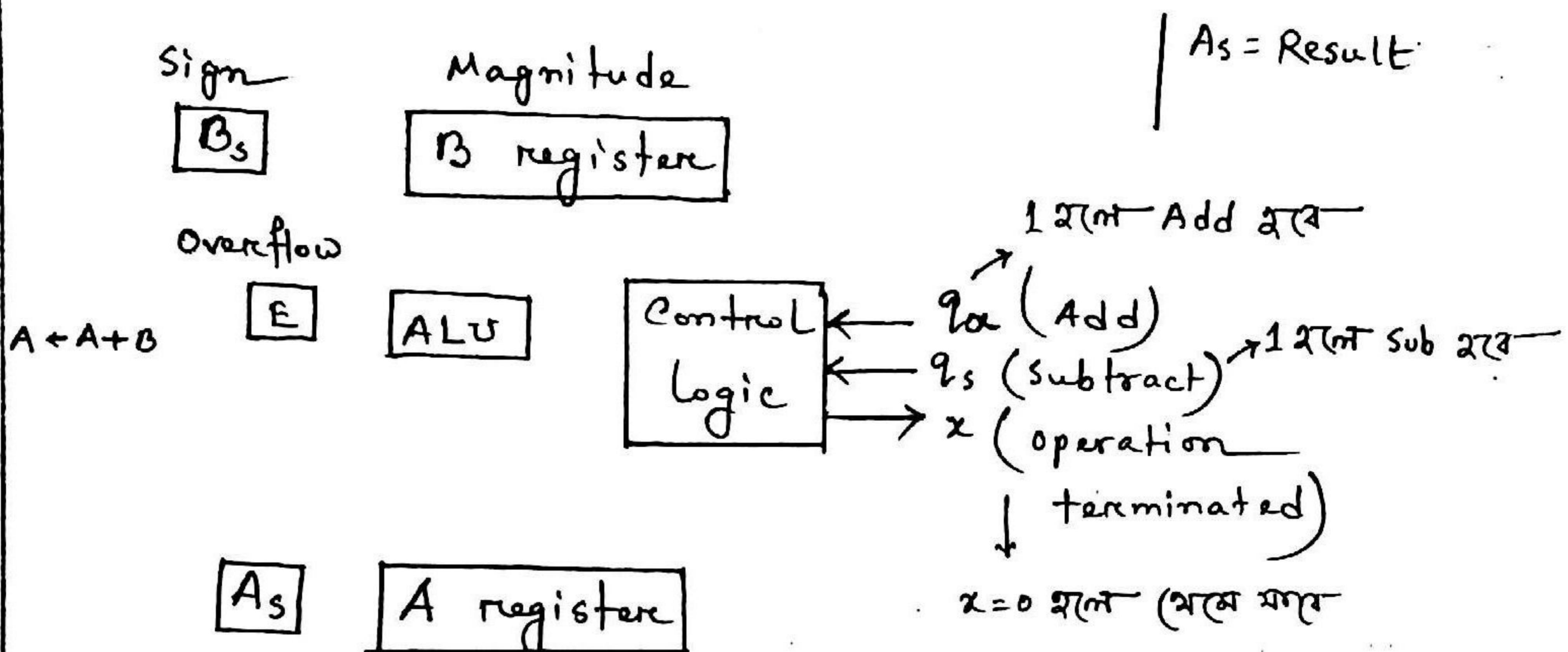


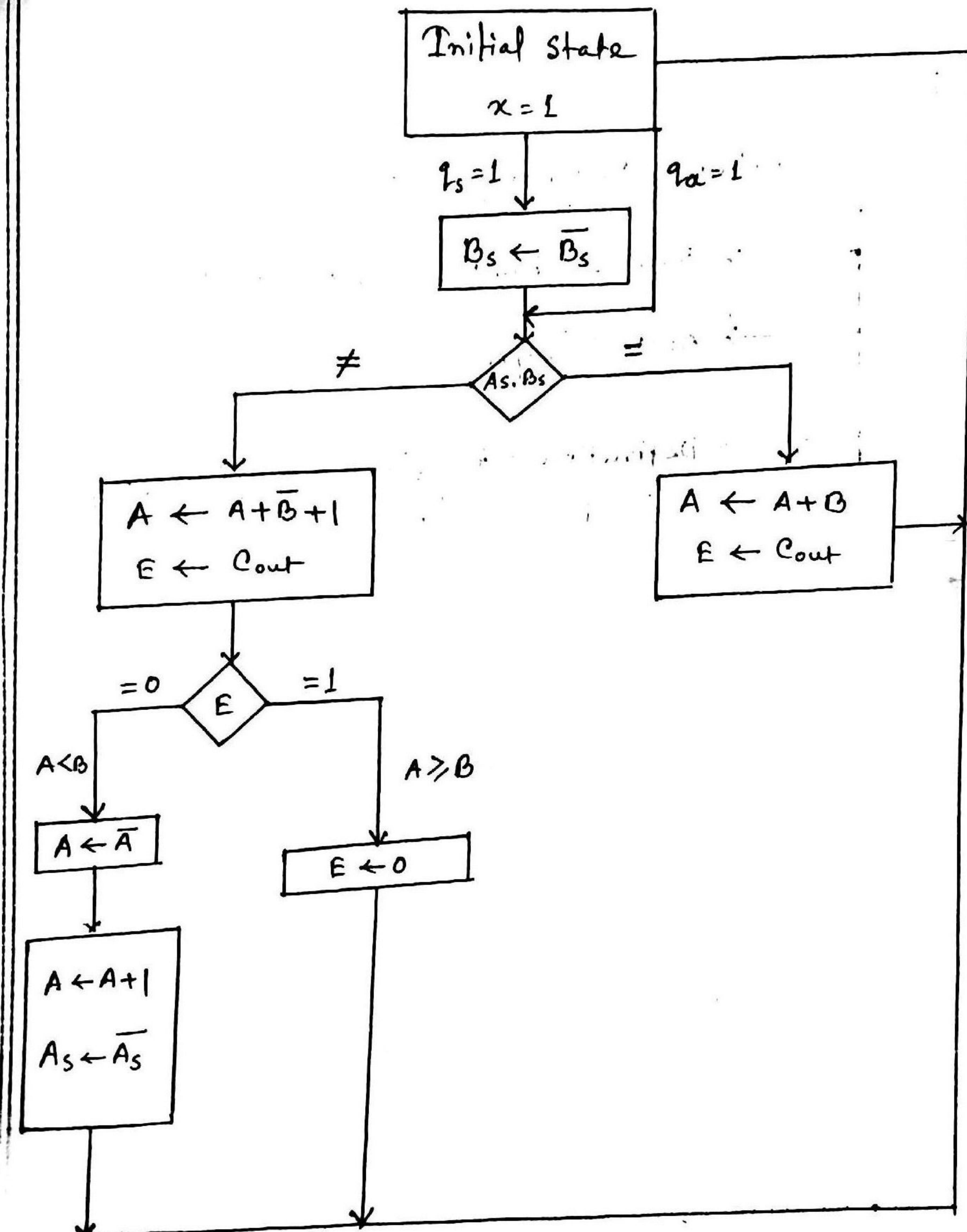
Fig 10-6 : Register Configuration for
the adder-subtractor

3 Derivation of the Algorithm:

$$\begin{array}{c}
 (\pm A) \pm (\pm B) \\
 \hline
 (\pm A) - (+B) &= (\pm A) + (-B) \\
 (\pm A) - (-B) &= (\pm A) + (+B) \\
 \hline
 (+A) + (+B) &= + (A+B) & \text{if } A \geq B & \text{if } A < B \\
 (+A) + (-B) &= + (A-B) & & = - (B-A) \text{ if } \\
 (-A) + (+B) &= - (A-B) & & = + (B-A) \text{ if } \\
 (-A) + (-B) &= - (A+B) & \xrightarrow{\text{same sign, sign omitted}}
 \end{array}$$

* B \leq
sign change \Rightarrow

Flow Chart



* Example (2020 Question CMW)

Sunday - Quiz-01

Ch-9

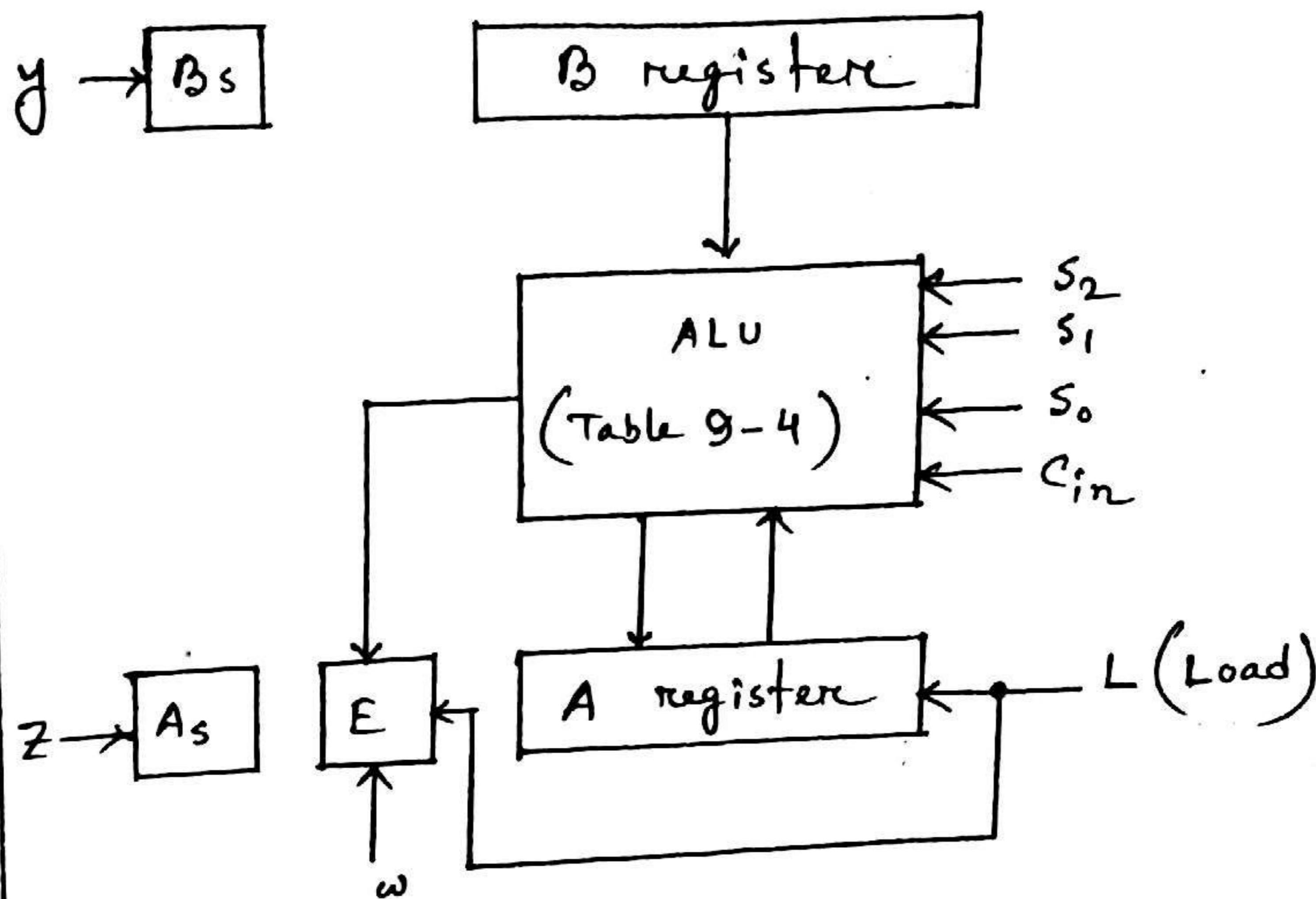
Logical Question (15 Marks) — Math

- PLA & Chapter-9, PLA square num 2 bit
- ALU - Design
- Definition (Register ? [from Lec])
Theory
RAM, ROM

25 May 2017

4 The data-processor part:

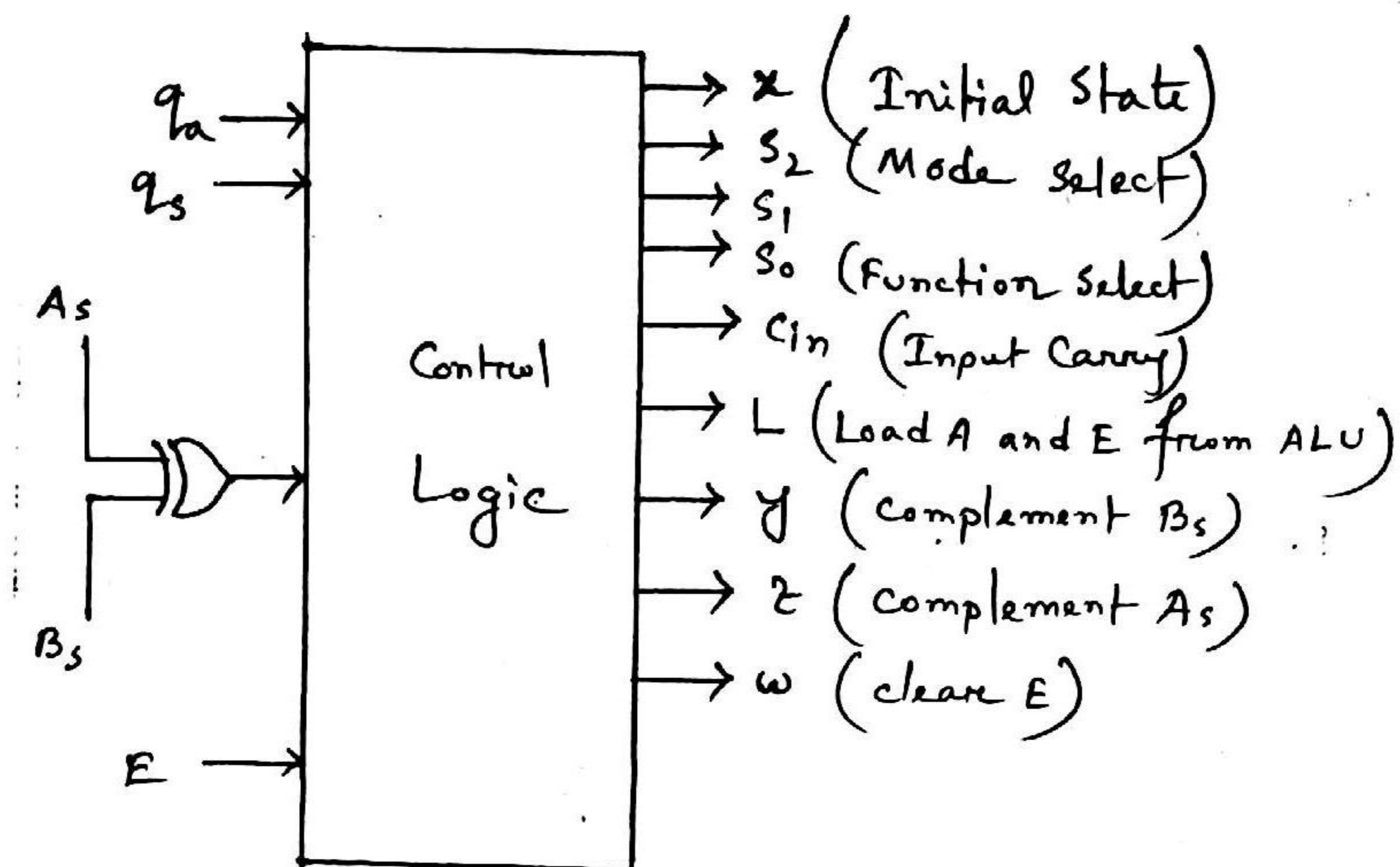
$A, B \Rightarrow$ magnitude



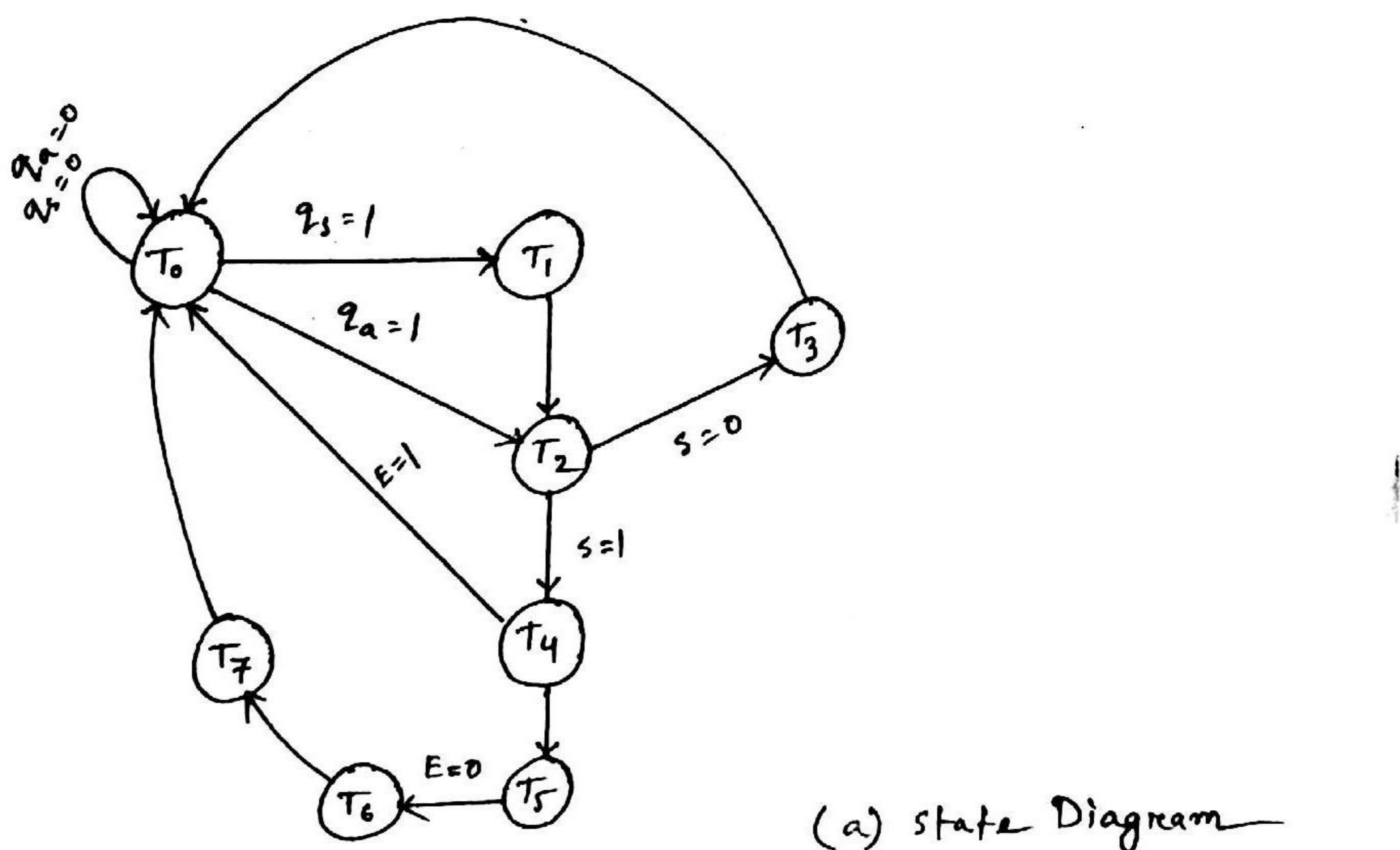
(a) Data processor registers and ALU

Fig 10-8: System block Diagram

magnitude



(b) Control block diagram



(a) state Diagram

Control outputs

	x	s_2	s_1	s_0	C_{in}	L	S	E	A
T_0 : Initial stage $x=1$	1	0	0	0	0	0	0	0	0
T_1 : $B_s \leftarrow \bar{B}_s$	0	0	0	0	0	0	1	0	0
T_2 : nothing	0	0	0	0	0	0	0	0	0
T_3 : $A \leftarrow A+B$, $E \leftarrow C_{out}$	0	0	0	1	0	1	0	0	0
T_4 : $A \leftarrow A+B+1$, $E \leftarrow C_{out}$	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 : $A \leftarrow \bar{A}$	0	1	1	1	0	1	0	0	0
T_7 : $A \leftarrow A+1$, $A_s \leftarrow \bar{A}_s$	0	0	0	0	1	1	0	1	0

(b) Sequence of register transfer ;

Fig : 10-9: Control Logic Diagram and sequence of micro-operations.

27 May 2017

5 Design of Hard-wired Control (Control Logic Design)

The control is designed by one flip-flop per state method

Table 10-1 : Boolean function for Control logic

Flip-flop input - function	Boolean function for output control
$D T_0 = q_s' q_a' T_0 + T_3 + ET_5 + T_7$	$x = T_0$
$D T_1 = q_s T_0$	$s_2 = T_6$
$D T_2 = q_a T_0 + T_1$	$s_1 = T_4 + T_6$
$D T_3 = s' T_2$	$s_0 = T_3 + T_6$
$D T_4 = s T_2$	$C_{in} = T_4 + T_7$
$D T_5 = T_4$	$L = T_3 + T_4 + T_6 + T_7$
$D T_6 = E' T_5$	$y = T_1$
$D T_7 = T_6$	$z = T_7$
	$w = T_5$

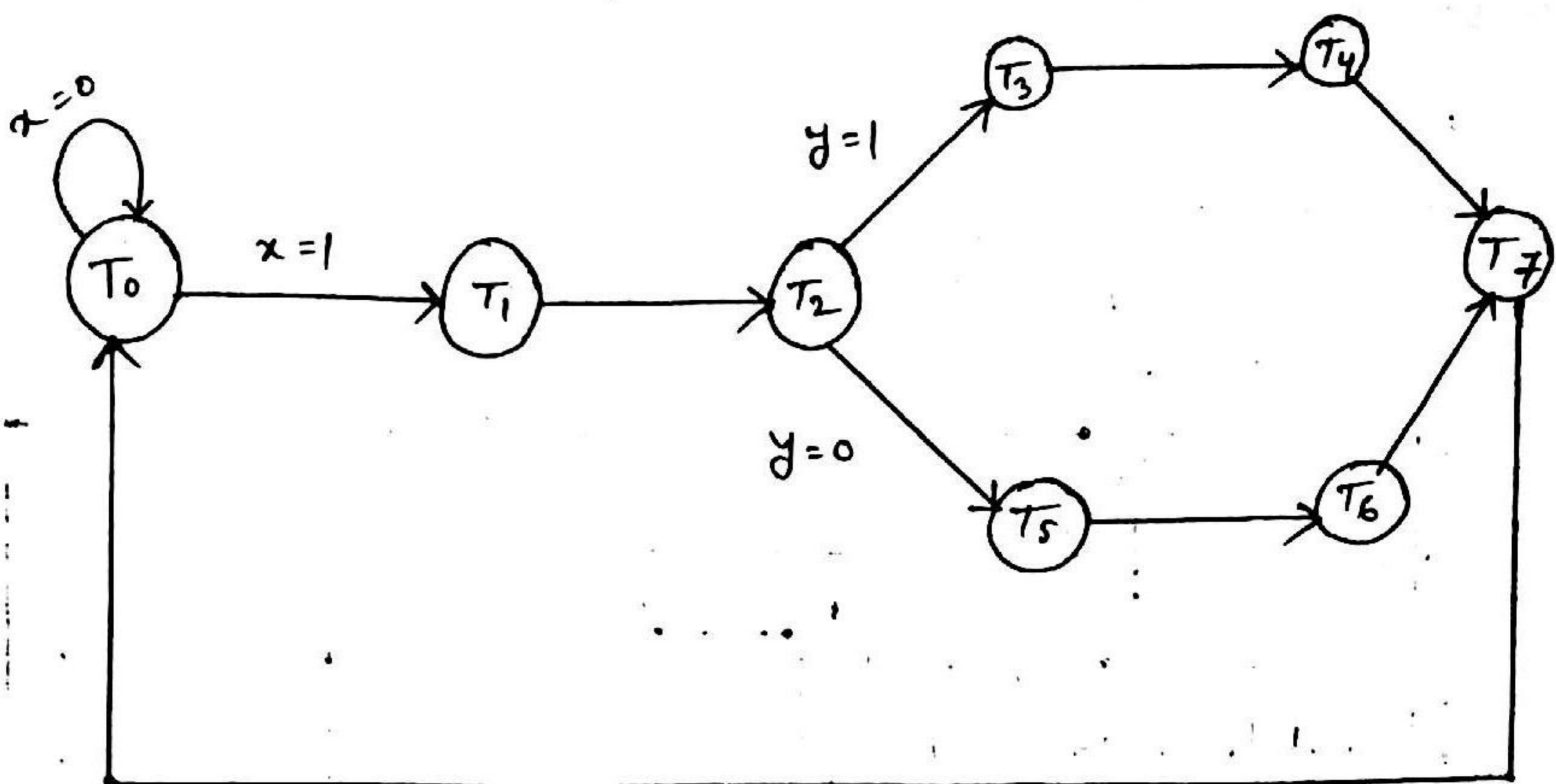


Fig : P10-27

(গুরু অসমা)

Q. Design the control using eight D-flip flops.

$$\Rightarrow DT_0 = x' T_0 + \bar{x}$$

$$DT_1 = x T_0$$

$$DT_2 = T_1$$

$$DT_3 = y T_2$$

$$DT_4 = T_3$$

$$DT_5 = y' T_2$$

$$DT_6 = T_5$$

$$DT_7 = T_4 + T_6$$

(fig 10.28)
পঞ্জি

28 May 2017

10-4 Microprogram Control

Hardware Configuration

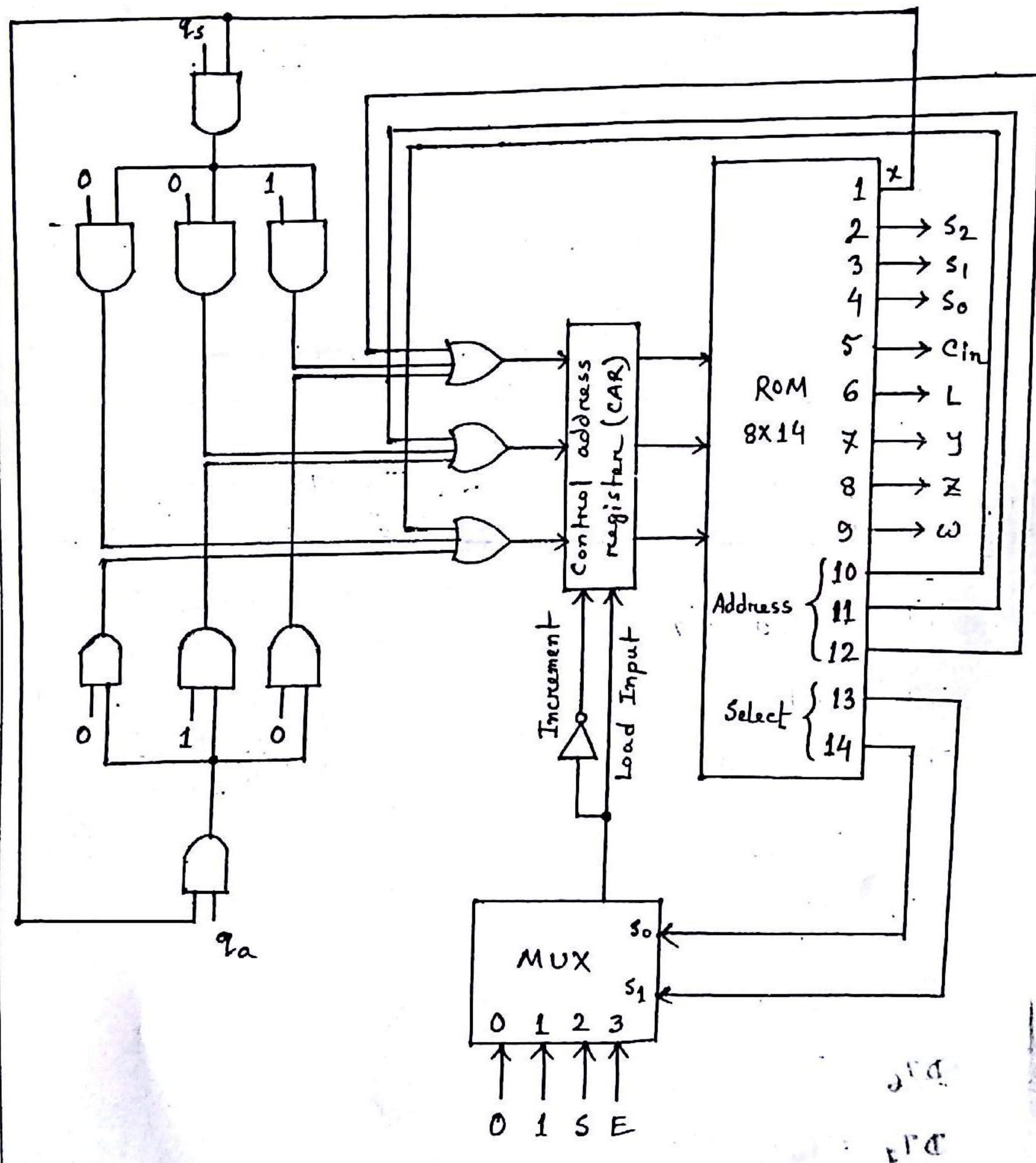


Fig = (10-10) Microprogram control
block diagram

ROM bits 13 14	MUX select function
0 0	Increment CAR
0 1	Load Input to CAR
1 0	Load Input to CAR if $S=1$, increment CAR if $S=0$
1 1	Load Input to CAR if $E=1$, increment CAR if $E=0$

CAR = Control Address Register

(with figure)

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)	Load 0 or, external address
1	$B_s \leftarrow \bar{B}_s$	$q_s=1$, start subtraction
2	If ($s=1$) then (go to 4)	$q_a=1$, start addition
3	$A \leftarrow A+B$, $E \leftarrow C_{out}$, go to 0	Add magnitudes and return
4	$A \leftarrow A+\bar{B}+1$, $E \leftarrow C_{out}$	Subtract magnitudes
5	If ($E=1$) then (go to 0), $E \leftarrow 0$	Operation terminated if $E=1$
6	$A \leftarrow \bar{A}$	$E=0$, complement A
7	$A \leftarrow A+1$, $A_s \leftarrow \bar{A}_s$, go to 0	Done, return to address 0

Table : 10-3 Binary Microprogram for
Control Memory

ROM Address	Control Word									Address	Select
	x	s ₂	s ₁	s ₀	Cin	L	y	z	w		
000	1	0	0	0	0	0	0	0	0	000	01
001	0	0	0	0	0	1	0	0	0	010	01
010	0	0	0	0	0	0	0	0	0	100	10
011	0	0	0	1	0	1	0	0	0	000	01
100	0	0	1	0	1	1	0	0	0	101	01
101	0	0	0	0	0	0	0	1	0	000	11
110	0	1	1	0	1	0	0	0	0	111	01
111	0	0	0	0	1	1	0	1	0	000	01

* \Rightarrow Table : 10-2 \Rightarrow ALU \Rightarrow selectore
— କେତେ ପରିମାଣ, Table - 10-3 କେତେ 2781 [7 Marks]



01 June 2017

10-6 : Hard-Wired Control

Example - 2

① Statement of the Problem

We wish to design an arithmetic circuit that multiplies two fixed point binary numbers in sign-magnitude representation

multiplicand :

10111

multiplier :

10011

1st multiplier bit = 1, copy multiplicand

10111

shift right to obtain 1st partial product (pp), 01.0111.

2nd multiplier bit = 1, copy multiplicand

10111

add multiplicand to previous pp

1000101

shift right to obtain 2nd pp

1000101

3rd multiplier bit = 0, shift right to obtain 3rd pp

01000101

4th " " = 0, " " " 4th pp

001000101

5th " " = 1, copy multiplicand

10111

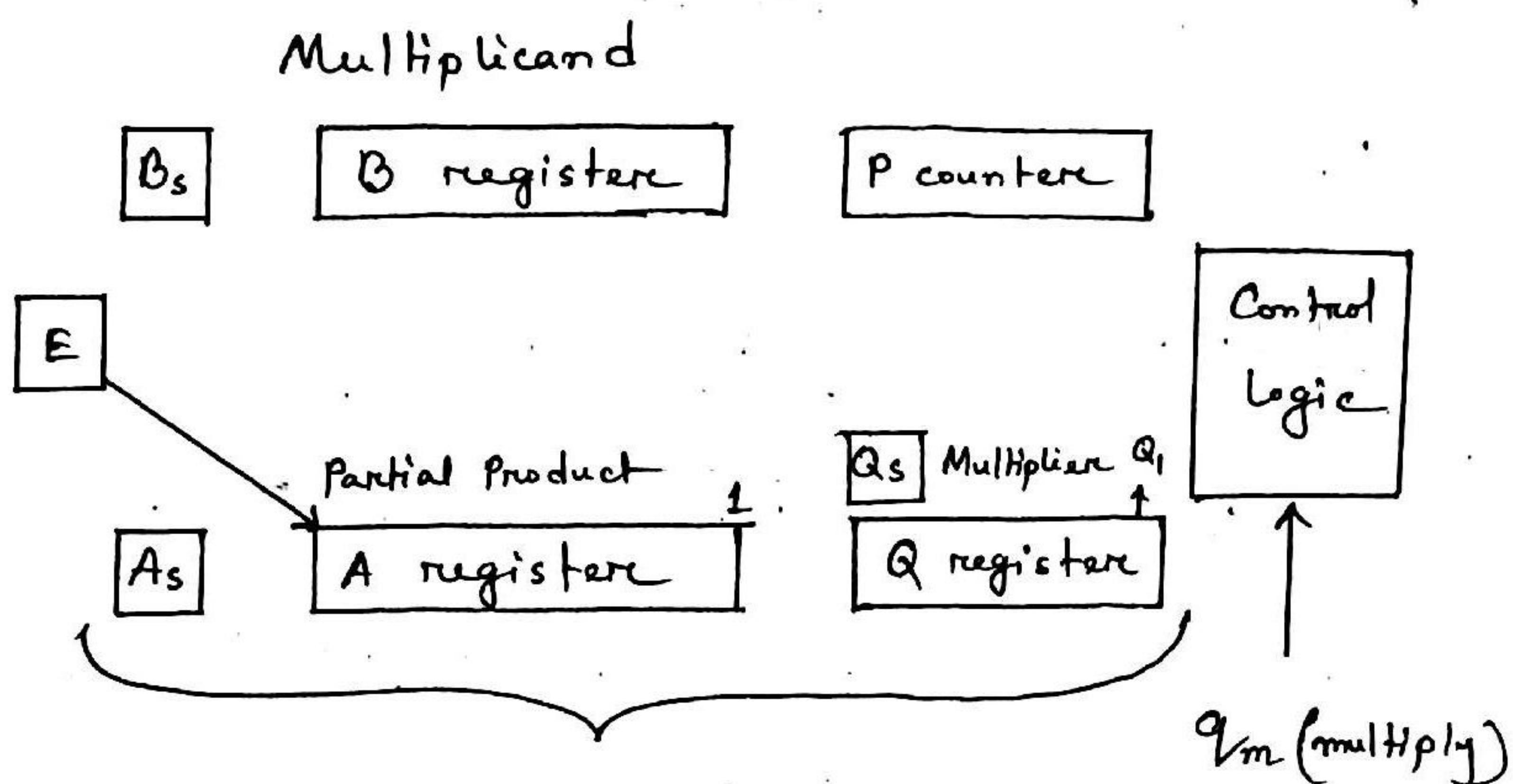
add multiplicand to previous pp

110110101

shift right to obtain 5th or final product

0110110101

2 Equipment Configuration :



* E ওর value SHIFT করে আসবে !

* E ওর value A ওর msB টে আসবে !

3 Derivation of Algorithm : (এই প্রক্রিয়াটি কী)

$B \leftarrow$ Multiplicand magnitude, $B_s \leftarrow$ sign

$Q \leftarrow$ Multiplier magnitude, $Q_s \leftarrow$ sign

$A \leftarrow A + B$ (Multiplicand + Partial Product)

$E \leftarrow$ Cout

$P \leftarrow k$, $k =$ number of bits in multiplier

shift Operation :

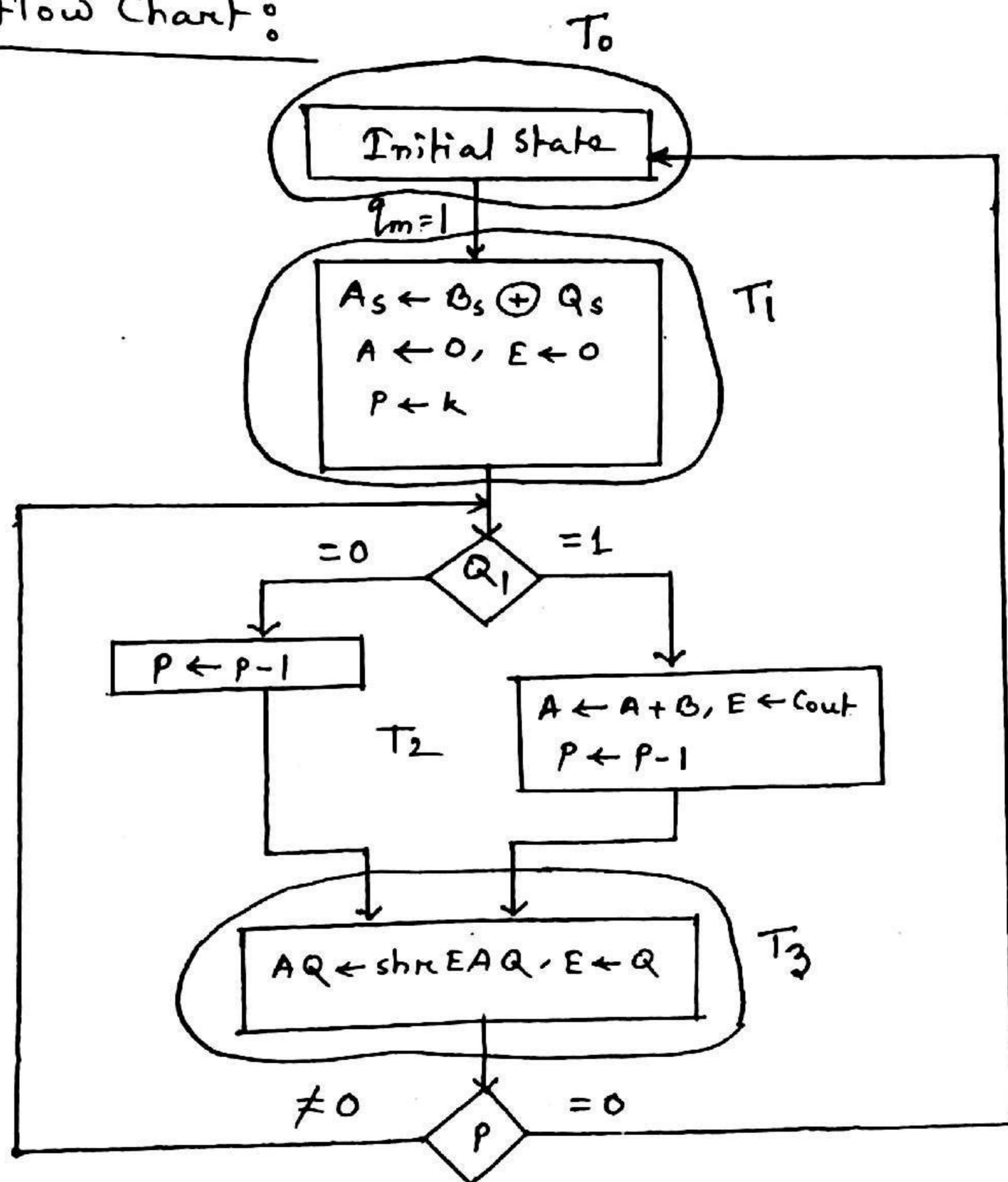
$AQ \leftarrow \text{shre} EAQ, E \leftarrow 0$

[Details \rightarrow Book]

$$\begin{array}{l} \text{++} = + \\ \text{--} = + \end{array} \quad 0$$

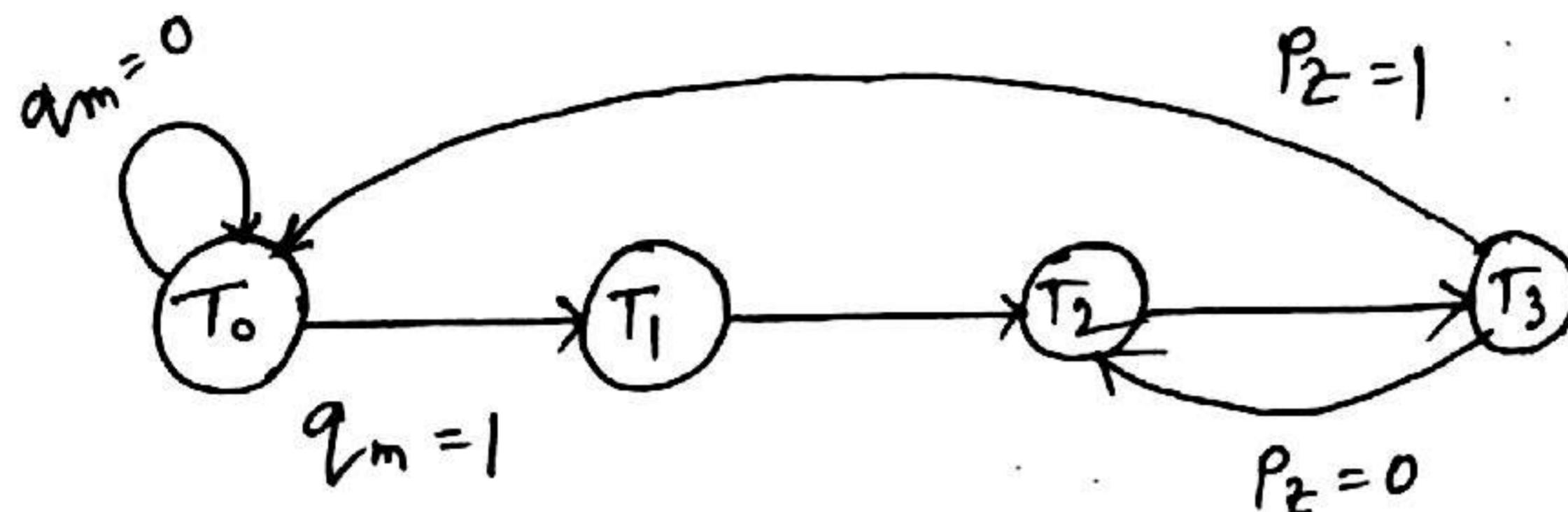
$$\begin{array}{l} + - \\ - + \end{array} = \begin{array}{l} - \\ - \end{array} \quad 1$$

4 Flow Chart:



03 June 2017

a) State Diagram



b) Sequence of register transfers

T₀: Initial state

T₁: A_s \leftarrow B_s \oplus Q_s, A \leftarrow 0, E \leftarrow 0, P \leftarrow k

Q₁ T₂: A \leftarrow A + B, E \leftarrow Cout

T₂: P \leftarrow P - 1

T₃: AQ \leftarrow shrc EAQ, E \leftarrow 0

Fig: 10-15: Control state Diagram and sequence
of micro-operations for multiplier.

Fig: 10-16

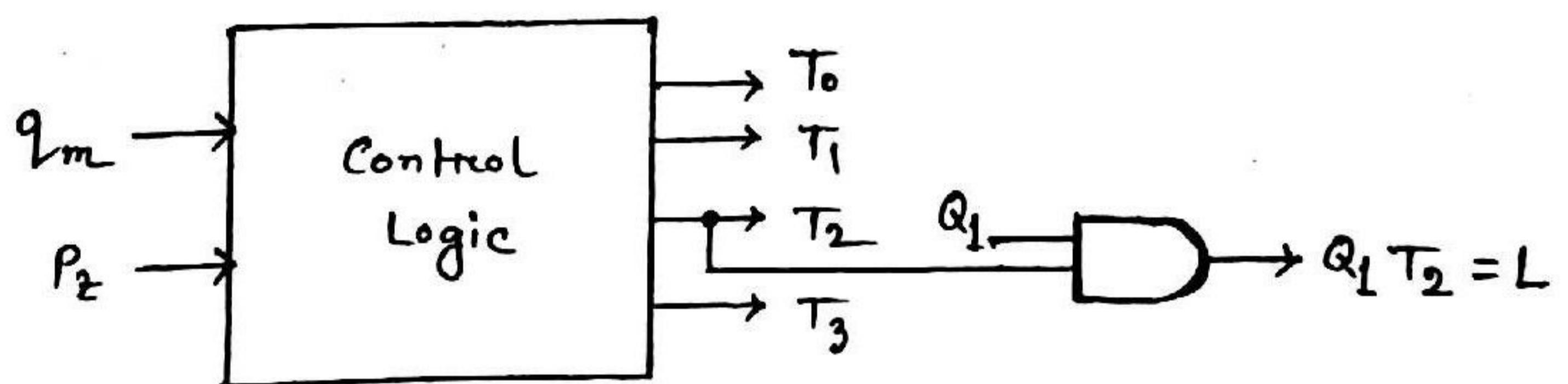
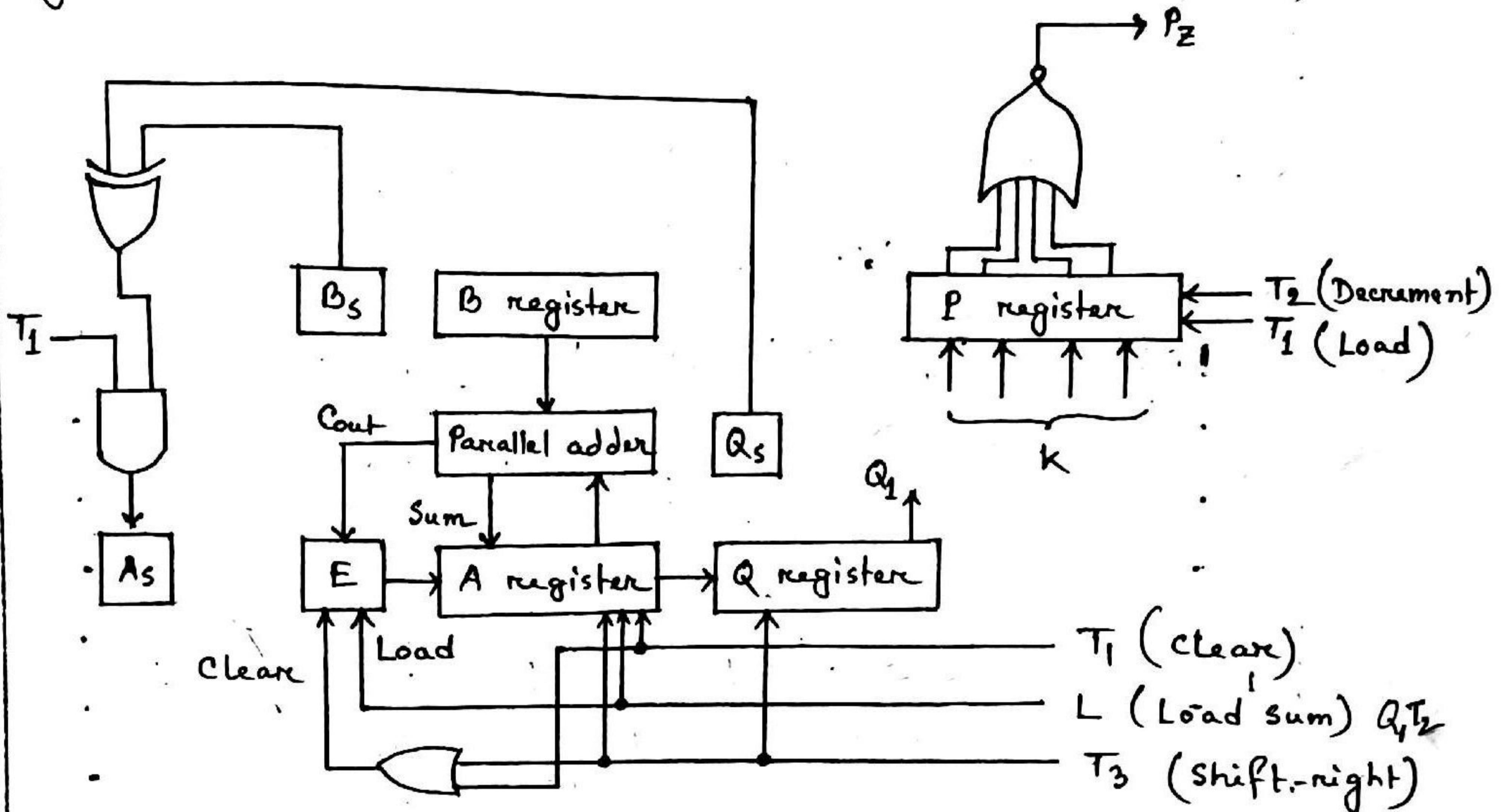


Figure: 10-16 Data processor for binary multiplier

04 Jun 2017

5 Design of Hard-wired Control

(a) Excitation Table

Present State		Inputs		Next State		Flip-flop inputs			
		q_m	P_2	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_1	0 0	1 X		0 1		0 X		1 X	
T_2	0 1	X X		1 0		1 X		X 1	
T_3	1 0	X X		1 1		X 0		1 X	
T_4	1 1	X 1		0 0		X 1		X 1	
T_5	1 1	X 0		1 0		X 0		X 1	

(2nd flip flop \rightarrow G_2 ; next 3 \rightarrow represent कठोर 2 bit नाम)

(b) Flip-flop input functions

$$JG_2 = T_1$$

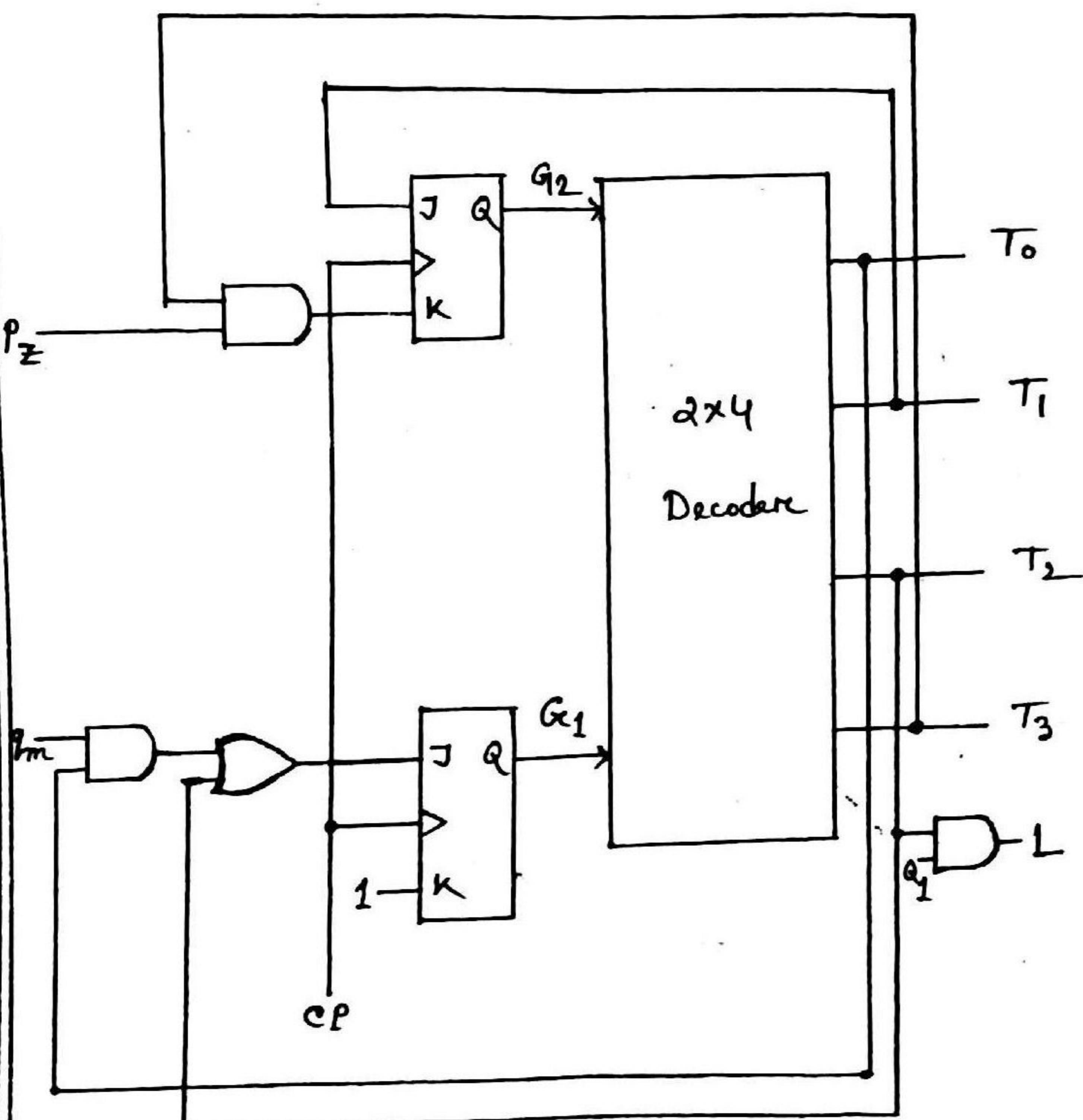
$$KG_2 = T_3 P_2$$

$$JG_1 = T_0 q_m + T_2$$

$$KG_1 = 1$$

State		J	K
P	N		
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

(c) Logic Diagram (Book) [10-17 (fig)]



$P_2 \rightarrow P$ register (or \bar{P})

$Q_m \rightarrow$ user input

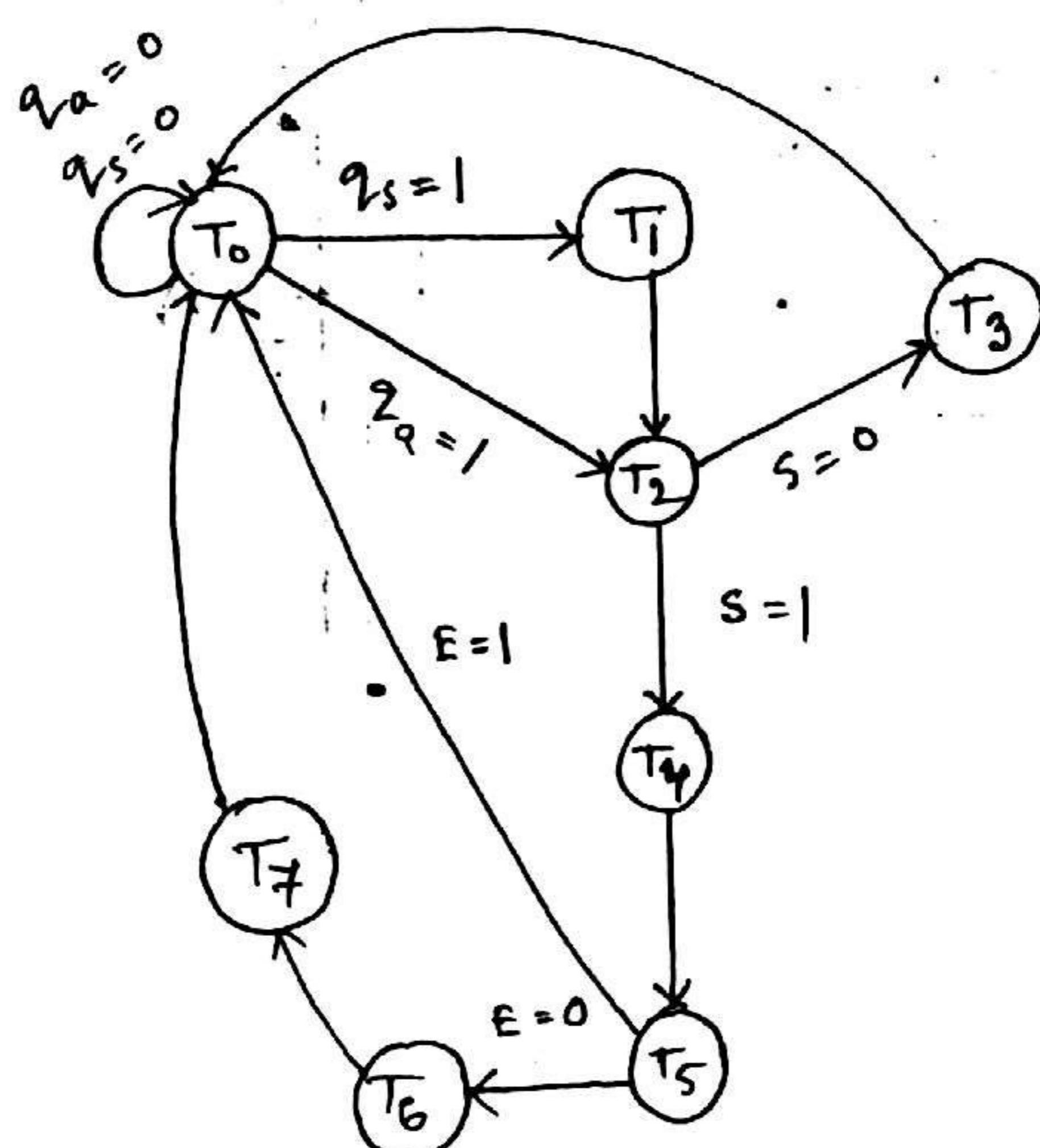
Fig : 10-17: Design of control for binary multiplier

* figure - p 10.28 & p 10.27 - JK flip flop \rightarrow

Solve \rightarrow 08 Jun 2017

08 Jun 2017

* Design the control specified in the following figure by the register and decoder method use 3 J-K flip flops.



B7

(a) Excitation Table

Present State	Input	Next State	flip flop inputs
$G_3 \ G_2 \ G_1$	$q_a \ q_s \ E \ S$	$G_3 \ G_2 \ G_1$	$JG_3 \ KG_3 \ JG_2 \ KG_2 \ JG_1 \ KG_1$
$T_0 \ 0 \ 0 \ 0$	$0 \ 0 \ X \ X$	$0 \ 0 \ 0$	$0 \ X \ 0 \ X \ 0 \ X$
$T_0 \ 0 \ 0 \ 0$	$X \ 1 \ X \ X$	$0 \ 0 \ 1$	$0 \ X \ 0 \ X \ 1 \ X$
$T_0 \ 0 \ 0 \ 0$	$1 \ X \ X \ X$	$0 \ 1 \ 0$	$0 \ X \ 1 \ X \ 0 \ X$
$T_1 \ 0 \ 0 \ 1$	$X \ X \ X \ X$	$0 \ 1 \ 0$	$0 \ X \ 1 \ X \ X \ 1$
$T_2 \ 0 \ 1 \ 0$	$X \ X \ X \ 0$	$0 \ 1 \ 1$	$0 \ X \ X \ 0 \ 1 \ X$
$T_2 \ 0 \ 1 \ 0$	$X \ X \ X \ 1$	$1 \ 0 \ 0$	$1 \ X \ X \ 1 \ 0 \ X$
$T_3 \ 0 \ 1 \ 1$	$X \ X \ X \ X$	$0 \ 0 \ 0$	$0 \ X \ X \ 1 \ X \ 1$
$T_4 \ 1 \ 0 \ 0$	$X \ X \ X \ X$	$1 \ 0 \ 1$	$X \ 0 \ 0 \ X \ 1 \ X$
$T_5 \ 1 \ 0 \ 1$	$X \ X \ 0 \ X$	$1 \ 1 \ 0$	$X \ 0 \ 1 \ X \ X \ 1$
$T_6 \ 1 \ 1 \ 0$	$X \ X \ X \ X$	$1 \ 1 \ 1$	$X \ 0 \ X \ 0 \ 1 \ X$
$T_7 \ 1 \ 1 \ 1$	$X \ X \ X \ X$	$0 \ 0 \ 0$	$X \ 1 \ X \ 1 \ X \ 1$
$T_5 \ 1 \ 0 \ 1$	$X \ X \ 1 \ X$	$0 \ 0 \ 0$	$X \ 1 \ 0 \ X \ X \ 1$

$$JG_3 = ST_2$$

$$KG_3 = T_7 + T_5 E$$

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

$$KG_2 = ST_2 + T_3 + T_7$$

$$KG_1 = 1$$

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

10 Jun 2017

PLA Control

Table 10-6 : state table for control circuit

Input		Output	
Present state	Inputs	Next state	Outputs
$G_2 \ G_1$	$q_m \ p_2 \ Q_1$	$G_2 \ G_1$	$T_0 \ T_1 \ T_2 \ L \ T_3$
T_0	0 0	0 X X	1 0 0 0 0
T_0	0 0	1 X X	1 0 0 0 0
T_1	0 1	X X X	0 1 0 0 0
T_2	1 0	X X 0	0 0 1 0 0
T_2	1 0	X X 1	0 0 1 1 0
T_3	1 1	X 0 X	0 0 0 0 1
T_3	1 1	X 1 X	0 0 0 0 1

* $T_2 \rightarrow T_3$ পুরো শব্দটা শব্দ, $Q_1 = 1, 0$ এবং -এন্ট-

কাবণ, L এবং -এন্ট- এ অন্তর কোথা থাকে।

* $T_3 \Rightarrow$ কোন state'র কাবণ কোন এবং determine কোন-

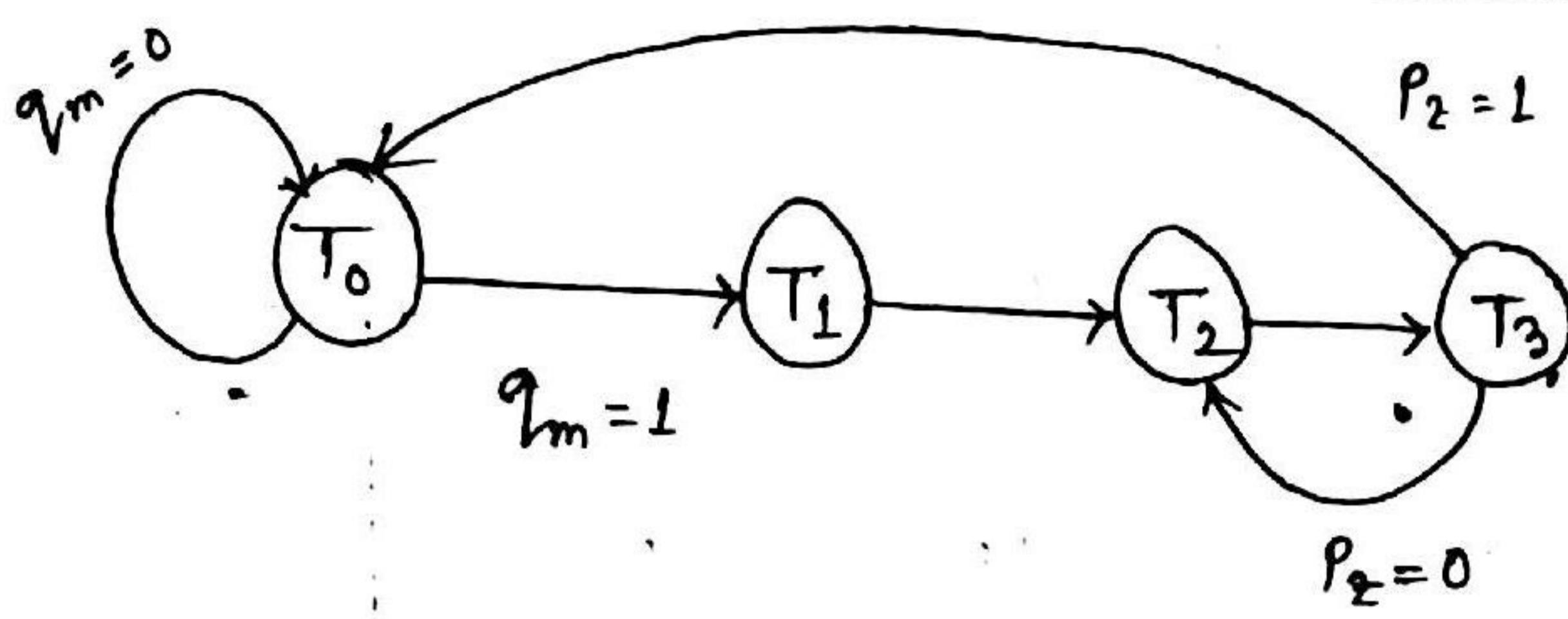
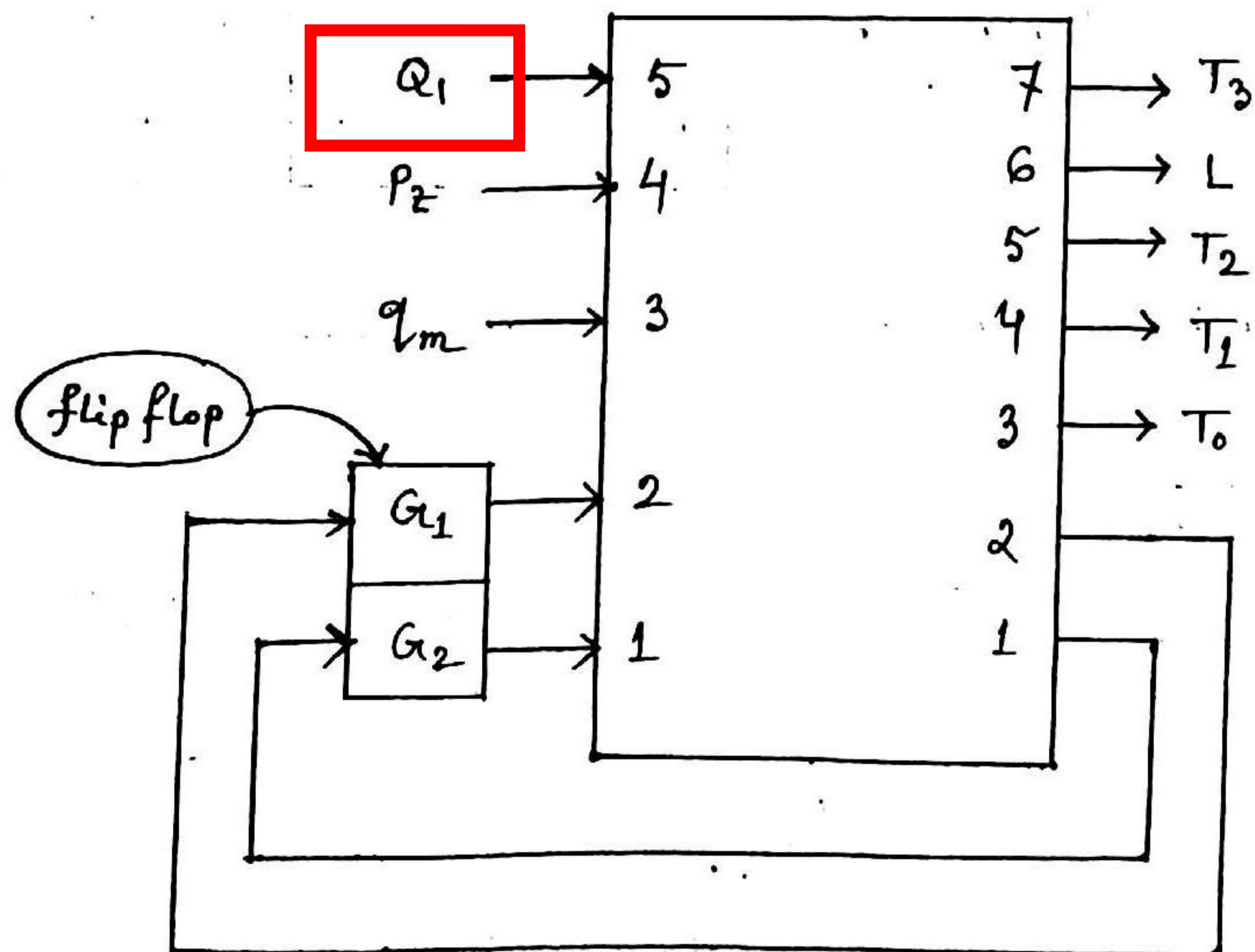


Fig 10-18 PLA control for binary multiplier.

(a) block diagram



* $Q_1 = 1$ \Rightarrow L, T_2 active

$Q_1 = 0$ \Rightarrow L, T_2 active

(b) PLA Program table

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

Question: Design the control logic PLA.

Exercise:

10-14, 10-22, 10-23, 10-25, 10-26, 10-27, 10-28

Quiz-02 [02.07.2017]

Syllabus \Rightarrow Chapter 10 ॥ १०/५ (from Lec after Quiz 1)

[Counter, Booth, Chapter-10] . . .

06 July 2017

SAP - I

Simple - As - Possible Computer

10 - 1 Architecture

Program Counter:

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

Input and MAR:

It includes the address and data switch registers. It allows you to send 4 address bits and 8 data bits to the RAM. Instructions and data words are written into the RAM before a computer run.

The RAM:

The RAM is a 16x8 static TTL RAM. You can program the RAM by means of the address and data switch registers.

Control Unit Data Processing Unit

