Information for Reviewer

1.

Truth Table

A	B	Equal	Operation
00	00	1	Multiply
00	01	0	Add
00	10	0	Add
00	11	0	Add
01	00	0	Subtract
01	01	1	Multiply
01	10	0	Subtract
01	11	0	Subtract
10	00	0	Subtract
10	01	0	Add
10	10	1	Multiply
10	11	0	Add
11	00	0	Subtract
11	01	0	Subtract
11	10	0	Add
11	11	1	Multiply

LOGICAL EQUATIONS:

A > B = A· B' = AB'

A < B = A'·B =A'B

A=B=AB + A'B'

Multiply (M):

Multiply : Y = A × B = AB

M=Equal×*A*×*B*

- Equal: This term ensures that the multiplication
- operation is performed only when the numbers are equal.
- *A*×*B*: This is the multiplication operation, where *A* and *B* are the 2-bit input values.

So, the Multiply operation (*M*) will be *A*×*B* only when the numbers are equal (Equal=1), and it will be 0 when the numbers are not equal.

Subtract(S):

Subtract when (A > B): Y = A – B = AB’

- **Equation:** $Y_{Subtract}=A\cdot B'$

If the first 2-bit fetched (*A*) has a higher magnitude than the second one (*B*), the two arguments are subtracted.

Addition (A):

Add when A < B): Y = A + B = A’B + AB’

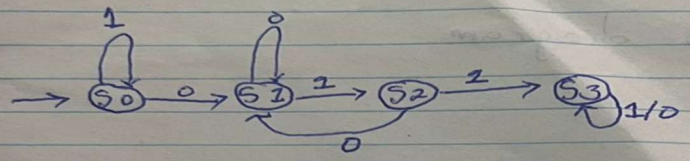
- **Equation:** $Y_{Add}=A'\cdot B+A\cdot B$

If the first 2-bit data fetched (*A*) has a lesser magnitude compared to the second one (*B*), both arguments are added.

`

2.

2a)



States;

state (S_0): is the start where ~~no~~ symbols have not been Read yet

State (S1): Represents the state where one symbol has be read.

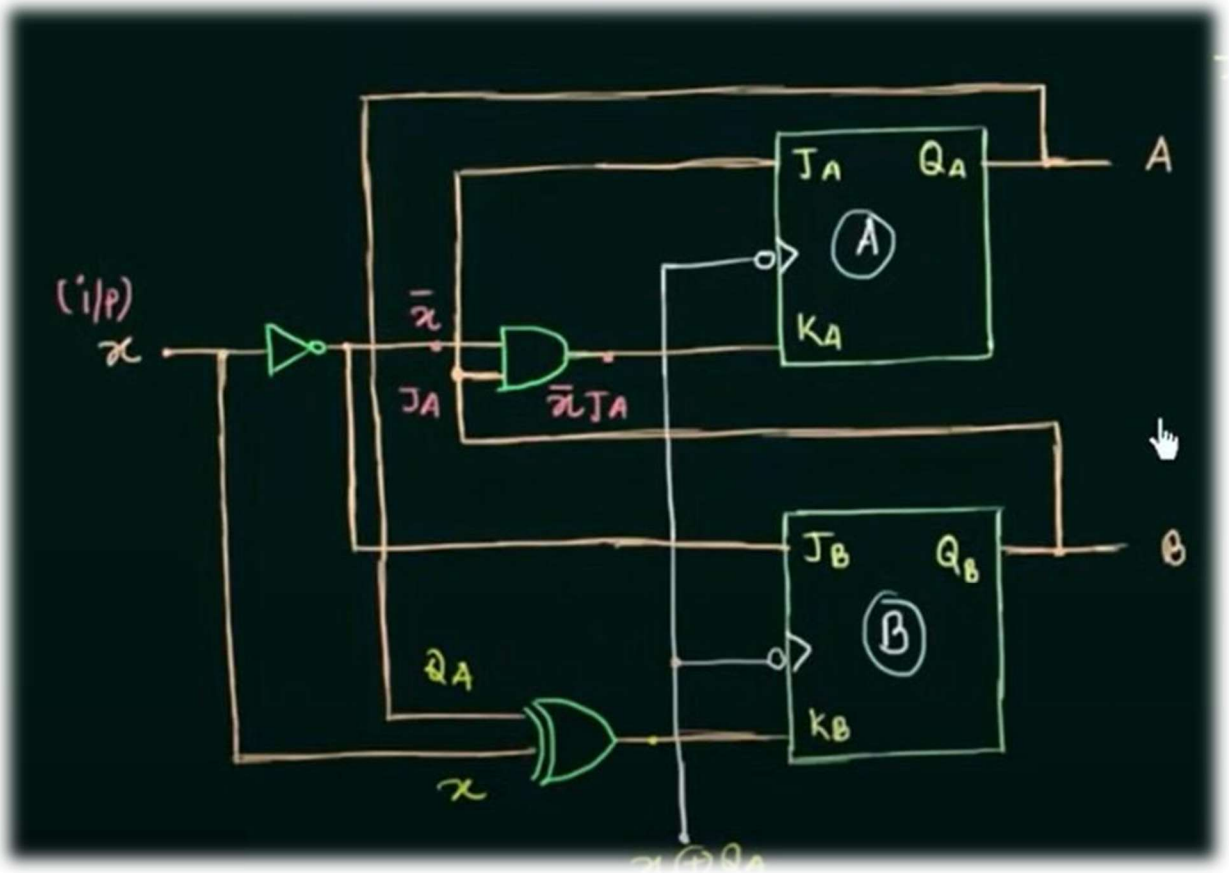
state (S_2): Represents the state where two symbols have been read.

State (S_3): Represents the state where the input string satisfies the equation.

b. State transition table

Current State	Input(0)	Input (1)	Next State
S0	S1	SO	S1
S1	S1	S2	S2
S2	S1	S3	S1
S3	S3	S3	S3

C.



State Table

Q_A	Q_B	x	J_A	K_A	J_B	K_B	Q_A^+	Q_B^-
0	0	0	0	0	1	0	0	1
0	0	1	0	0	0	1	0	0
0	1	0	1	1	1	0	1	1
0	1	1	1	0	0	1	1	0
1	0	0	0	0	1	1	1	1
1	0	1	0	0	0	0	1	0
1	1	0	1	1	1	1	0	0
1	1	1	1	0	0	0	1	1

EQUATIONS:

$J_A = Q_B$ $K_B = x(XOR)\ Q_A$

$K_A = x' J_A$

	Q_A	Q_B
$S_0 =$	0	0
$S_1 =$	0	1
$S_3 =$	1	1