FSM Design Concepts

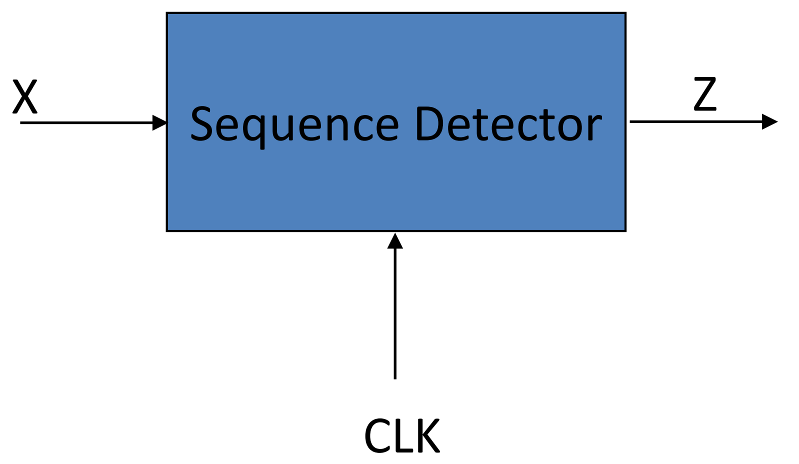
A [Super Expanded Version] Sequence Detector

Example

What’s a Sequence Detector, Yul?

- A sequence detector is a FSM that produces a logic 1 when a specific binary sequence has been observed
- When the FSM does not observe the sequence in the observed binary sequence, it emits a logic 0
- Let’s get started!

Macro View of the Sequence Detector



This sequence detector will be designed to recognize the pattern “101”. The behavior of the machine calls for the Z output to equal 1 whenever the programmed pattern is observed in the input bit stream X.

Example:

X = 0011011001010100

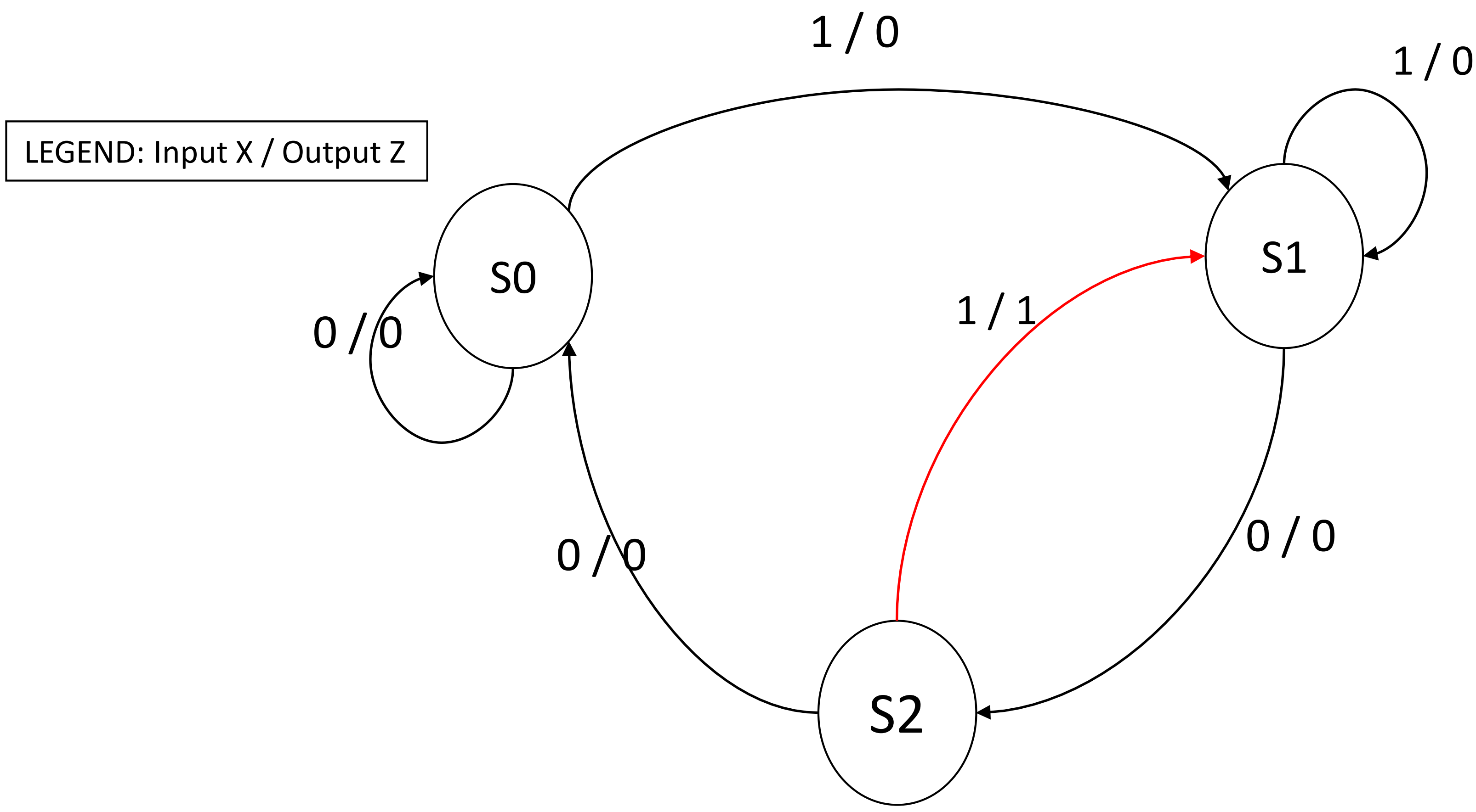
Z = 0000010000010100

Source: Fundamentals of Logic Design by Charles H. Roth

Design Strategy

- For the design of the sequence detector, we will select the Mealy machine model
- For this design, we will use the following process:
    1. Generate the state graph
    2. Create the state table
    3. Create the state transition table
    4. Generate the input expressions for the JKFF
    5. Realize the final logic design

Generate the State Graph



Create the state table

Present State	Next State		Z	
	X = 0	X = 1	X = 0	X = 1
S0	S0	S1	0	0
S1	S2	S1	0	0
S2	S0	S1	0	1

Create the State Transition Table

State Table

Present State	Next State		Z	
	X = 0	X = 1	X = 0	X = 1
S0	S0	S1	0	0
S1	S2	S1	0	0
S2	S0	S1	0	1

Let S0 = 00

S1 = 01

S2 = 10

State Transition Table

Present State	Next State		Z	
	X = 0	X = 1		
AB	A+B+	A+B+	X = 0	X = 1
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1

Let’s Examine Flip-Flop ‘A’

Generate the Input Expressions for the JKFF

The characteristic equation for the JKFF governs all of the actions required to populate the Ja and Ka K-Maps, respectively.

$$Q+ = JQ' + K'Q$$

For flip-flop A, we will populate the Ja and Ka K-maps. This will provide the input values required to match the behavior recorded in the State Transition Table.

$$A+ = JA' + K'A$$

Present State	Next State		Z	
	X = 0	X = 1		
AB	A+B+	A+B+	X = 0	X = 1
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1

When X=0 and AB = 00: A=0 A+ = 0
Applying the characteristic equation:

A+ =JA' + K'A, we get:
0 = J(1) + K'(0)
In order to satisfy the equation, the right-hand-side (RHS) of the equation must equal 0. That means, we need to "pick" the values to Ja and Ka that balance the equation. Therefore, for the position in the Ja and Ka K-Maps, position X =0 and AB = 00, Ja = 0 and Ka = 0

J_A-Map

AB \ X	0	1
00	0	
01		
11	X	X
10		

K_A-Map

AB \ X	0	1
00	0	
01		
11	X	X
10		

Generate the Input Expressions for the JKFF

The characteristic equation for the JKFF governs all of the actions required to populate the Ja and Ka K-Maps, respectively.

$Q+ = JQ' + K'Q$

For flip-flop A, we will populate the Ja and Ka K-maps. This will provide the input values required to match the behavior recorded in the State Transition Table.

$A+ = JA' + K'A$

Present State	Next State		Z	
	X = 0	X = 1		
AB	A+B+	A+B+	X = 0	X = 1
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1

When X=0 and AB = 01: A=0 A+ = 1
Applying the characteristic equation:

A+ =JA' + K'A, we get:
1 = J(1) + K'(0)
In order to satisfy the equation, the right-hand-side (RHS) of the equation must equal 0. That means, we need to "pick" the values to Ja and Ka that balance the equation. Therefore, for the position in the Ja and Ka K-Maps, position X =0 and AB = 01, Ja = 0 and Ka = 0

J_A-Map

AB \ X	0	1
00	0	
01	1	
11	X	X
10		

K_A-Map

AB \ X	0	1
00	0	
01	0	
11	X	X
10		

Generate the Input Expressions for the JKFF

The characteristic equation for the JKFF governs all of the actions required to populate the Ja and Ka K-Maps, respectively.

$$\boxed{Q+ = JQ' + K'Q}$$

For flip-flop A, we will populate the Ja and Ka K-maps. This will provide the input values required to match the behavior recorded in the State Transition Table.

$$\boxed{A+ = JA' + K'A}$$

Present State	Next State		Z	
	X = 0	X = 1		
AB	A+B+	A+B+	X = 0	X = 1
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1

J_A-Map

AB \ X	0	1
00	0	
01	1	
11	X	X
10	0	

K_A-Map

AB \ X	0	1
00	0	
01	0	
11	X	X
10	1	

When X=0 and AB = 10: A=1 A+ = 0
Applying the characteristic equation:

A+ =JA' + K'A, we get:
0 = J(0) + K'(1)
In order to satisfy the equation, the right-hand-side (RHS) of the equation must equal 0. That means, we need to "pick" the values to Ja and Ka that balance the equation. Therefore, for the position in the Ja and Ka K-Maps, position X =0 and AB = 10, Ja = 0 and Ka = 0

Generate the Input Expressions for the JKFF

The characteristic equation for the JKFF governs all of the actions required to populate the Ja and Ka K-Maps, respectively.

$Q+ = JQ' + K'Q$

For flip-flop A, we will populate the Ja and Ka K-maps. This will provide the input values required to match the behavior recorded in the State Transition Table.

$A+ = JA' + K'A$

Present State	Next State		Z	
	X = 0	X = 1		
AB	A+B+	A+B+	X = 0	X = 1
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1

When X=1 and AB = 00: A=0 A+ = 0
Applying the characteristic equation:

A+ =JA' + K'A, we get:
0 = J(1) + K'(0)
In order to satisfy the equation, the right-hand-side (RHS) of the equation must equal 0. That means, we need to "pick" the values to Ja and Ka that balance the equation. Therefore, for the position in the Ja and Ka K-Maps, position X =1 and AB = 00, Ja = 0 and Ka = 0

J_A-Map

AB \ X	0	1
00	0	0
01	1	
11	X	X
10	0	

K_A-Map

AB \ X	0	1
00	0	0
01	0	
11	X	X
10	1	

Generate the Input Expressions for the JKFF

Present State	Next State		Z	
	X = 0	X = 1		
AB	A+B+	A+B+	X = 0	X = 1
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1

When X=1 and AB = 01: A=0 A+ = 0
Applying the characteristic equation:

A+ =JA' + K'A, we get:
0 = J(1) + K'(0)
In order to satisfy the equation, the right-hand-side (RHS) of the equation must equal 0. That means, we need to "pick" the values to Ja and Ka that balance the equation. Therefore, for the position in the Ja and Ka K-Maps, position X =1 and AB = 01, Ja = 0 and Ka = 0

The characteristic equation for the JKFF governs all of the actions required to populate the Ja and Ka K-Maps, respectively.

$$\boxed{Q+ = JQ' + K'Q}$$

For flip-flop A, we will populate the Ja and Ka K-maps. This will provide the input values required to match the behavior recorded in the State Transition Table.

$$\boxed{A+ = JA' + K'A}$$

J_A-Map

AB \ X	0	1
00	0	0
01	1	0
11	X	X
10	0	

K_A-Map

AB \ X	0	1
00	0	0
01	0	0
11	X	X
10	1	

Generate the Input Expressions for the JKFF

The characteristic equation for the JKFF governs all of the actions required to populate the Ja and Ka K-Maps, respectively.

Q+ = JQ’ + K’Q

For flip-flop A, we will populate the Ja and Ka K-maps. This will provide the input values required to match the behavior recorded in the State Transition Table.

A+ = JA’ + K’A

Present State	Next State		Z	
	X = 0	X = 1		
AB	A+B+	A+B+	X = 0	X = 1
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1

When X=1 and AB = 10: A=1 A+ = 0
Applying the characteristic equation:

A+ =JA’ + K’A, we get:
0 = J(0) + K’(1)
In order to satisfy the equation, the right-hand-side (RHS) of the equation must equal 0. That means, we need to “pick” the values to Ja and Ka that balance the equation. Therefore, for the position in the Ja and Ka K-Maps, position X =1 and AB = 10, Ja = 0 and Ka = 0

J_A-Map

AB \ X	0	1
00	0	0
01	1	0
11	X	X
10	0	0

K_A-Map

AB \ X	0	1
00	0	0
01	0	0
11	X	X
10	1	1

Generate the Input Expressions for the JKFF

Present State	Next State		Z	
AB	X = 0 A+B+	X = 1 A+B+	X = 0	X = 1
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1

Now let’s capture the prime implicants and determine the input expressions for Ja and Ka, respectively.

J_A-Map

AB \ X	0	1
00	0	0
01	1	0
11	X	X
10	0	0

K_A-Map

AB \ X	0	1
00	0	0
01	0	0
11	X	X
10	1	1

Minimized expressions.

JA = X’B

KA = A

Let’s Examine Flip-Flop ‘B’

Generate the Input Expressions for the JKFF

The characteristic equation for the JKFF governs all of the actions required to populate the Jb and Kb K-Maps, respectively.

$$Q+ = JQ' + K'Q$$

Present State	Next State		Z	
	X = 0	X = 1		
AB	A+B+	A+B+	X = 0	X = 1
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1

When X=0 and AB = 00: B=0 B+ = 0
Applying the characteristic equation:

B+ =JB' + K'B, we get:
0 = J(1) + K'(0)
In order to satisfy the equation, the right-hand-side (RHS) of the equation must equal 0. That means, we need to "pick" the values to Jb and Kb that balance the equation. Therefore, for the position in the Jb and Kb K-Maps, position X =0 and AB = 00, Jb = 1 and Kb = 0

For flip-flop **B**, we will populate the Jb and Kb K-maps. This will provide the input values required to match the behavior recorded in the State Transition Table.

$$B+ = JB' + K'B$$

J_B-Map

AB \ X	0	1
00	1	
01		
11	X	X
10		

K_B-Map

AB \ X	0	1
00	0	
01		
11	X	X
10		

Generate the Input Expressions for the JKFF

The characteristic equation for the JKFF governs all of the actions required to populate the Jb and Kb K-Maps, respectively.

$$Q+ = JQ' + K'Q$$

Present State	Next State		Z	
	X = 0	X = 1		
AB	A+B+	A+B+	X = 0	X = 1
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1

When X=0 and AB = 01: B=1 B+ = 0
Applying the characteristic equation:

B+ =JB' + K'B, we get:
0 = J(0) + K'(1)
In order to satisfy the equation, the right-hand-side (RHS) of the equation must equal 0. That means, we need to "pick" the values to Jb and Kb that balance the equation. Therefore, for the position in the Jb and Kb K-Maps, position X =1 and AB = 01, Jb = 0 and Kb = 0

For flip-flop B, we will populate the Jb and Kb K-maps. This will provide the input values required to match the behavior recorded in the State Transition Table.

$$B+ = JB' + K'B$$

J_B-Map

AB \ X	0	1
00	1	
01	0	
11	X	X
10		

K_B-Map

AB \ X	0	1
00	0	
01	0	
11	X	X
10		

Generate the Input Expressions for the JKFF

The characteristic equation for the JKFF governs all of the actions required to populate the Jb and Kb K-Maps, respectively.

$$Q+ = JQ' + K'Q$$

Present State	Next State		Z	
	X = 0	X = 1		
AB	A+B+	A+B+	X = 0	X = 1
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1

When X=0 and AB = 10: B=0 B+ = 0
Applying the characteristic equation:

B+ =JB' + K'B, we get:
0 = J(1) + K'(0)
In order to satisfy the equation, the right-hand-side (RHS) of the equation must equal 0. That means, we need to "pick" the values to Jb and Kb that balance the equation. Therefore, for the position in the Jb and Kb K-Maps, position X =1 and AB = 10, Jb = 0 and Kb = 0

For flip-flop B, we will populate the Jb and Kb K-maps. This will provide the input values required to match the behavior recorded in the State Transition Table.

$$B+ = JB' + K'B$$

J_B-Map

AB \ X	0	1
00	1	
01	0	
11	X	X
10	1	

K_B-Map

AB \ X	0	1
00	0	
01	0	
11	X	X
10	1	

Generate the Input Expressions for the JKFF

The characteristic equation for the JKFF governs all of the actions required to populate the Jb and Kb K-Maps, respectively.

$$Q+ = JQ' + K'Q$$

Present State	Next State		Z	
	X = 0	X = 1	X = 0	X = 1
AB	A+B+	A+B+		
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1

When X=1 and AB = 00: B=0 B+ = 1
Applying the characteristic equation:

B+ =JB' + K'B, we get:
1 = J(1) + K'(0)
In order to satisfy the equation, the right-hand-side (RHS) of the equation must equal 0. That means, we need to "pick" the values to Jb and Kb that balance the equation. Therefore, for the position in the Jb and Kb K-Maps, position X =1 and AB = 00, Jb = 0 and Kb = 0

For flip-flop B, we will populate the Jb and Kb K-maps. This will provide the input values required to match the behavior recorded in the State Transition Table.

$$B+ = JB' + K'B$$

J_B-Map

AB \ X	0	1
00	1	1
01	0	
11	X	X
10	1	

K_B-Map

AB \ X	0	1
00	0	0
01	0	
11	X	X
10	1	

Generate the Input Expressions for the JKFF

The characteristic equation for the JKFF governs all of the actions required to populate the Jb and Kb K-Maps, respectively.

$$Q+ = JQ' + K'Q$$

For flip-flop B, we will populate the Jb and Kb K-maps. This will provide the input values required to match the behavior recorded in the State Transition Table.

$$B+ = JB' + K'B$$

Present State	Next State		Z	
	X = 0	X = 1		
AB	A+B+	A+B+	X = 0	X = 1
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1

When X=1 and AB = 01: B=1 B+ = 1
Applying the characteristic equation:

B+ =JB' + K'B, we get:
1 = J(0) + K'(1)
In order to satisfy the equation, the right-hand-side (RHS) of the equation must equal 0. That means, we need to "pick" the values to Jb and Kb that balance the equation. Therefore, for the position in the Jb and Kb K-Maps, position X =1 and AB = 01, Jb = 0 and Kb = 0

J_B-Map

AB \ X	0	1
00	1	1
01	0	0
11	X	X
10	1	

K_B-Map

AB \ X	0	1
00	0	0
01	0	0
11	X	X
10	1	

Generate the Input Expressions for the JKFF

The characteristic equation for the JKFF governs all of the actions required to populate the Jb and Kb K-Maps, respectively.

$$Q+ = JQ' + K'Q$$

Present State	Next State		Z	
	X = 0	X = 1		
AB	A+B+	A+B+	X = 0	X = 1
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1

When X=1 and AB = 10: B=0 B+ = 1
Applying the characteristic equation:

B+ =JB' + K'B, we get:
1 = J(1) + K'(0)
In order to satisfy the equation, the right-hand-side (RHS) of the equation must equal 0. That means, we need to "pick" the values to Jb and Kb that balance the equation. Therefore, for the position in the Jb and Kb K-Maps, position X =1 and AB = 10, Jb = 0 and Kb = 0

For flip-flop B, we will populate the Jb and Kb K-maps. This will provide the input values required to match the behavior recorded in the State Transition Table.

$$B+ = JB' + K'B$$

J_B-Map

AB \ X	0	1
00	1	1
01	0	0
11	X	X
10	1	1

K_B-Map

AB \ X	0	1
00	0	0
01	0	0
11	X	X
10	1	0

Generate the Input Expressions for the JKFF

Present State	Next State		Z	
	X = 0	X = 1		
AB	A+B+	A+B+	X = 0	X = 1
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1

Now let's capture the prime implicants and determine the input expressions for Jb and Kb, respectively.

J_B-Map

AB \ X	0	1
00	1	1
01	0	0
11	X	X
10	1	1

K_B-Map

AB \ X	0	1
00	0	0
01	0	0
11	X	X
10	1	0

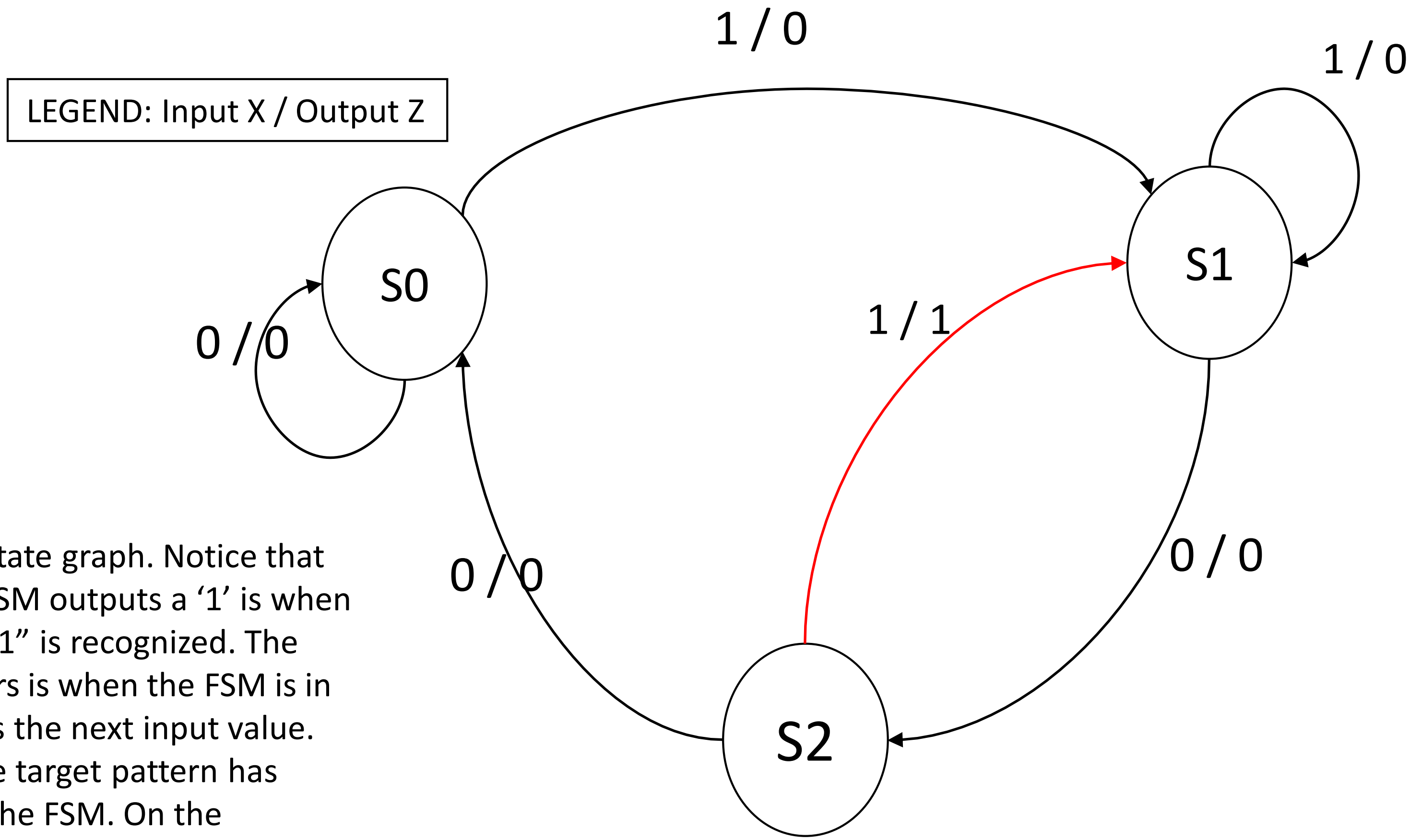
Minimized expressions.

JB = B'

KB = X'A

Let’s Examine the FSM Output ‘Z’

Generate the Input Expression for Z



Take a look at the state graph. Notice that the only time the FSM outputs a ‘1’ is when the full pattern “101” is recognized. The only time this occurs is when the FSM is in state S2 and X = 1 is the next input value. This means that the target pattern has been observed by the FSM. On the following slide, we will examine this transition by using the state transition table.

Generate the Input Expression for Z

Present State	Next State		Z	
	X = 0	X = 1		
AB	A+B+	A+B+	X = 0	X = 1
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1

In Pseudo Code:

When FSM is in State AB = 00 and X = 0 then Z = 0
When FSM is in State AB = 00 and X = 1 then Z = 0
When FSM is in State AB = 01 and X = 0 then Z = 0
When FSM is in State AB = 01 and X = 1 then Z = 0
When FSM is in State AB = 10 and X = 0 then Z = 0
When FSM is in State AB = 10 and X = 0 then Z = 1

Generate the Input Expression for Z

Take a look at the state transition table. Notice that the only time the FSM outputs a ‘1’ is when the full pattern “101” is recognized. The only time this occurs is when the FSM is in state S2 (i.e. AB=10) and X = 1 is the next input value. This means that the target pattern has been observed by the FSM. All other times Z = 0. Based on this observation, we can populate a K-Map for the ‘Z’ output using the information in the state transition table.

Present State	Next State		Z	
	X = 0	X = 1		
AB	A+B+	A+B+	X = 0	X = 1
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1

Z-Map

AB \ X	0	1
00	0	0
01	0	0
11	X	X
10	0	1

Z = XA

Final Minimal Expressions

Generate the Input Expressions for the JKFF

Present State	Next State		Z	
	X = 0	X = 1		
AB	A+B+	A+B+	X = 0	X = 1
00	00	01	0	0
01	10	01	0	0
10	00	01	0	1

J_A-Map

AB \ X	0	1
00	0	0
01	1	0
11	X	X
10	0	0

JA = X'B

K_A-Map

AB \ X	0	1
00	0	0
01	0	0
11	X	X
10	1	1

KA = A

J_B-Map

AB \ X	0	1
00	1	1
01	0	0
11	X	X
10	1	1

JB = B'

K_B-Map

AB \ X	0	1
00	0	0
01	0	0
11	X	X
10	1	0

KB = X'A

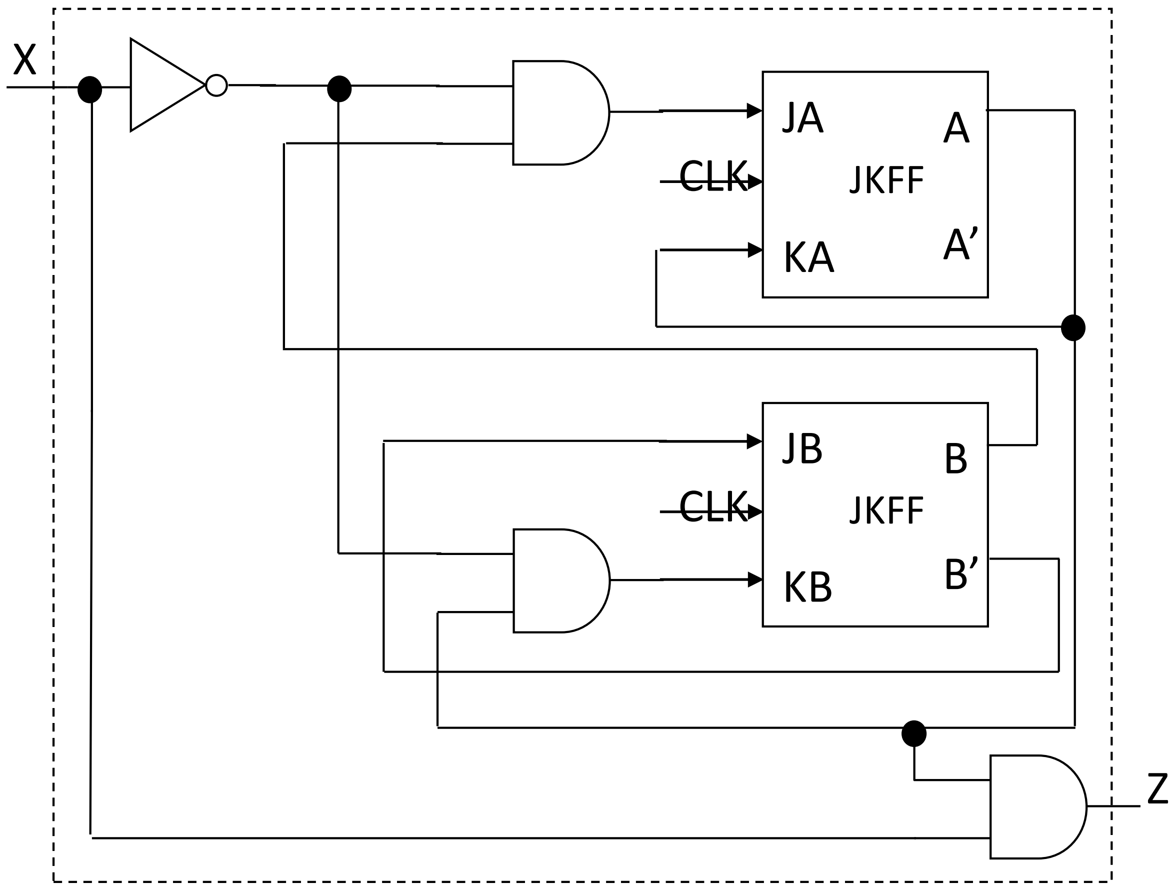
Z-Map

AB \ X	0	1
00	0	0
01	0	0
11	X	X
10	0	1

Z = XA

The Logic Circuit Rendering

Realize the final logic design



Sequence Detector based on the Mealy Machine model