Sequence Detector & Generator

by

Prof. Sujata Wakchaure

Sequence Detector

- In digital electronics a movement of data is commonly called as a **“Bit Stream”**
- Any one bit in a bit stream looks identical to many other bits
- It is imp for receiver to identify beginning & ending of a message
- This is a job of special bit sequence called **Flags**
- A flag is simply a bit sequence that serves as a marker in the bit stream
- To **detect a flag** in a bit stream a **Sequence Detector** is used

Sequence Detector (Contd..)

- The stream of bit has been feed as I/P, when clock is high & particular Pattern / Sequence is detected

- As soon as a sequence is detected, the O/P becomes high & then again becomes low

Sequence Detector (Contd..)

- Lets X= 0110010100……
- In this I/P Bit stream I want to detect 010
- We have 2 possibilities here
  1. When we consider overlapping
  2. When we do not consider the overlapping

0110010100

010 **O/P Y = 0**

0110010100

010 **Y = 00**

Sequence Detector (Contd..)

0110010100 **0110010100**

010 **Y = 000** **010** **Y = 000000**

– Now there is overlapping

0110$\boxed{\textbf{010}}$**100** **0110010100**

Y = 0000001

Here Pointer is not only at 1 But at 01

Final Output Y = 00000010010….

Sequence Detector (Contd..)

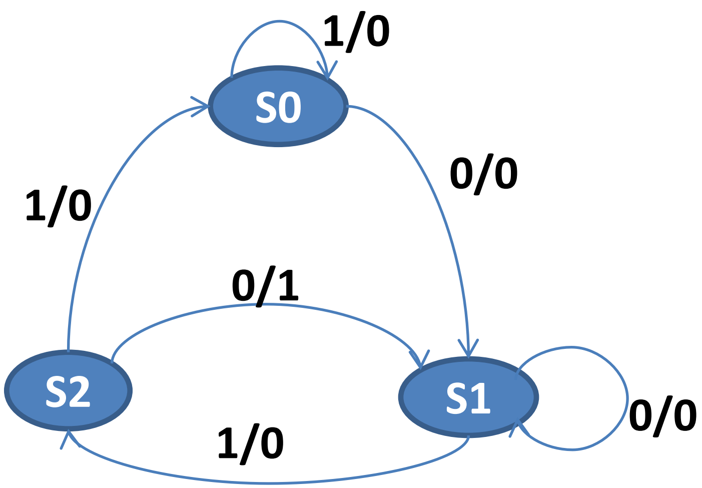
0110010100

- **Step 1:** Obtain the state diagram (Consider Mealy m/c)
  - Starting state of detection = S0
  - When pointer is at S0 = The 1st bit is **0**
  - When there is 0 = pointer will point to next state

S0= Reset

S1 = 0

S2= 01



Sequence Detector (Contd..)

- **0110010100**
  - Now Pointer is on **S0** =>
    - If I/P = **1** , O/P= 0 (Reset State)

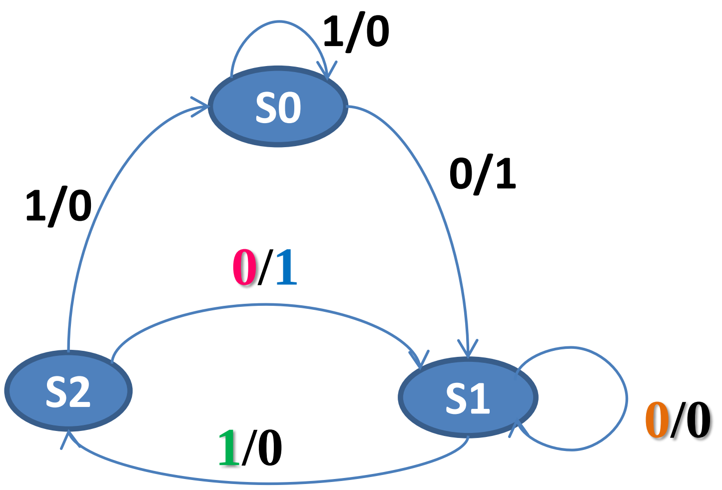
  - Now if Pointer is on **S1=>**
    - If Next Bit or I/P = **0** , O/P = 0 (Reset State)
    - If I/P = **1** then, got to next State =S2 , O/P = 0

  - Now if Pointer is on **S2=>**
    - If Next Bit or I/P = **0** , then go to S1, O/P = **1** (Reset State)
    - Because sequence **010** is detected here

Sequence Detector (Contd..)

0110 $\boxed{010}$ 100

- Currently pointer is at S1 with O/P = 1
- Next bit is 1, then again pointer will go to S2
- And the next bit is 0
- Then again O/P = 1 (010)



Sequence Detector Example

- Design a sequence detector to detect 3 or more consecutive 1s in a string of bits coming through an I/P line

- If X= 00111011110….

  Y= 0000**1**000**1****1**0 $\Longleftrightarrow$ (Overlapping)

  Y= 00001000100 $\Longleftrightarrow$ (Non - Overlapping)

Sequence Detector Example (Contd..)

- **Step 1:** State Diagram

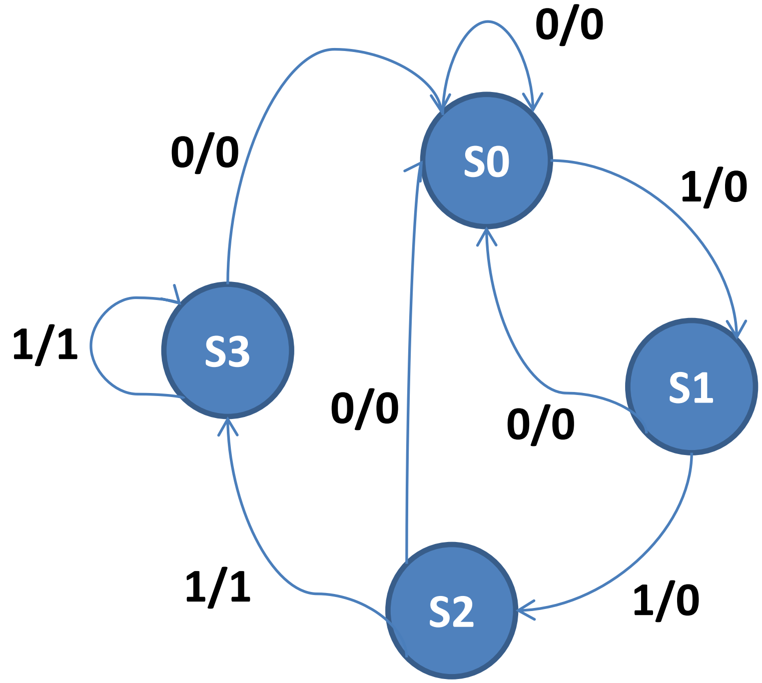
  S0=Reset

  S1= 1

States	Has	Awaits
S0= Reset	…….	111…..
S1=1	1	11……
S2=11	11	1
S3=111	111	…./0

Sequence Detector Example (Contd..)

- **Step 1:** State Diagram



Sequence Detector Example (Contd..)

- **Step 2:** State Assignment

  S0 = 00

  S1 = 01

  S2 = 10

  S3 = 11

Sequence Detector Example (Contd..)

- **Step 3:** State Table

Present State		Input	Next State		Output
QA	**QB**	**X**	**QA+**	**QB+**	**Y**
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	1
1	1	0	0	0	0
1	1	1	1	1	1

DA = QA+
DB = QB+

Sequence Detector Example (Contd..)

- **Step 4:** Minimized Expression

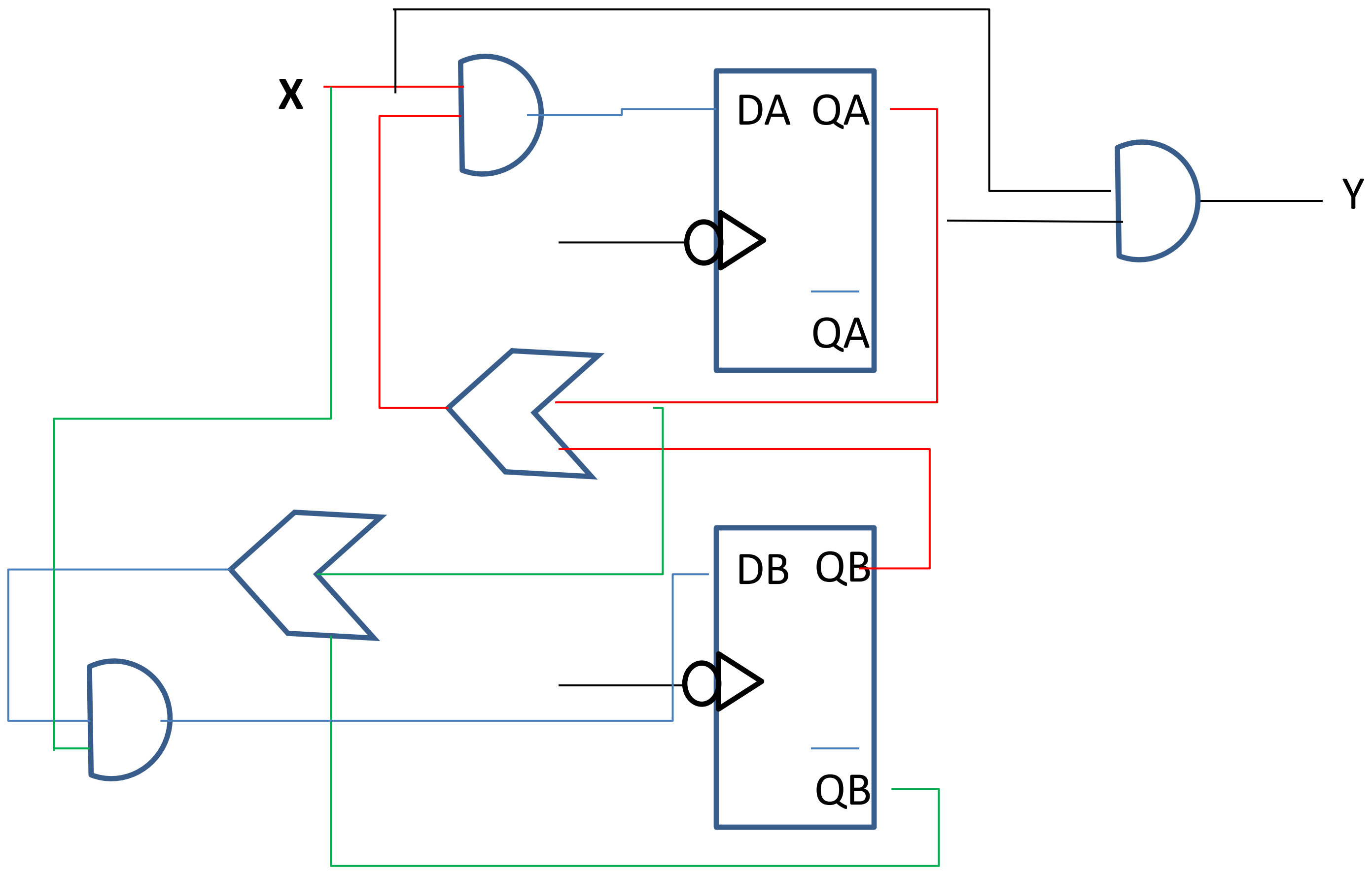
$$DA = QA+ = (QA + QB)X$$

$$DB = QB+ = (\overline{QB} + QA)X$$

$$Y = QAX$$

Sequence Detector Example (Contd..)

- **Step 5:** Implement Circuit



Sequence Generator

- It is a digital logic circuit whose purpose is to produce a prescribed sequence of output
- The sequence may be of indefinite length or a predetermined fixed length
- A Binary Counter is a special type of Sequence Generator
- Sequence Generators are useful in a wide variety of coding & control applications

Sequence Generator (Contd..)

- If you have **0101101**
- This is the sequence that you want to generate

- Then your O/P of sequence generator would be, it is going to continuously give out,

  0101101 0101101 0101101….

- So you have to decide the no. of Flip Flops

Sequence Generator (Contd..)

- **Step 1:** Number of Flip-Flops
  - J = No. of Zeros in sequence
  - K = No. of Ones in sequence

  N = Max (J,K)

  No. of FF (n) = N<= 2^{n-1}

- Now I/P sequence is **0101101**

  J=3 & K=4

  N = Max (J, K)

  N = Max (3, 4)

  N = 4

n = 4<= 2^{3-1} => if n=3

n=3

Sequence Generator (Contd..)

- **Step 2:** State Table

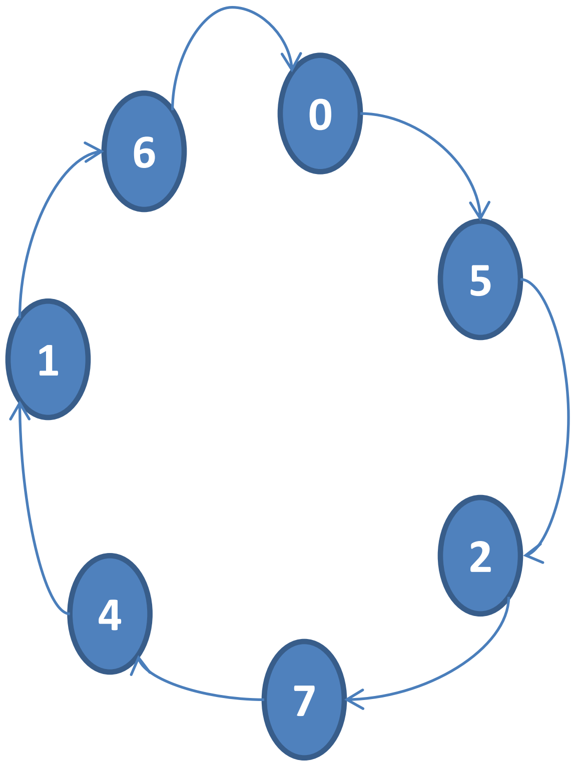
C	B	A	State
0	0	0	0
1	0	1	5
0	1	0	2
1	1	1	7
1	0	0	4
0	0	1	1
1	1	0	6

Not Used Case = 3

So N.S. of 3 = X

Sequence Generator (Contd..)

- **Step 3:** Sate Diagram



Sequence Generator (Contd..)

- **Step 4:** Write Excitation table of JK FF

Present State	Next State	Flip – Flop Outputs	
Qn	**Qn+1**	**J**	**K**
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Sequence Generator (Contd..)

- **Step 5:** Circuit Excitation table of JK FF

P. S.			N. S.			FF I/P					
C	B	A	C+	B+	A+	J_C	K_C	J_B	K_B	J_A	K_A
0	0	0	1	0	1	1	X	0	X	1	X
0	0	1	1	1	0	1	X	1	X	X	1
0	1	0	1	1	1	1	X	X	0	1	X
0	1	1	X	X	X	X	X	X	X	X	X
1	0	0	0	0	1	X	1	0	X	1	X
1	0	1	0	1	0	X	1	1	X	X	1
1	1	0	0	0	0	X	1	X	1	0	X
1	1	1	1	0	0	X	0	X	1	X	1

Sequence Generator (Contd..)

- Step 6: Derive Expression

$JC = \overline{C}$

$KC = \overline{A} + \overline{B}$

$JB = A$

$KB = C$

$JA = \overline{C} + \overline{B}$

$KA = A$

Sequence Generator (Contd..)