Assignment 3: CS220

Sequence Detector and 3-bit Odd Parity Bit Generator Using FSM

Abhishek Pardhi, Aayush Kumar, Jahnavi Kairamkonda

200026, 200008, 200482

B.Tech students

apardhi20@iitk.ac.in, akgarg20@iitk.ac.in, kjahnavi20@iitk.ac.in

Indian Institute of Technology Kanpur

Department of Computer Science & Engineering

February 17, 2022

Contents

1 Sequence Detector

1.1 Description

The sequence detector is made using ***Mealy FSM Machine***. Sequence to be detected is ***1010***. We've used **four states** to build the state machine: ***IDLE(2'b00)***, ***b(2'b10)***, ***c(2'b01)*** and ***d(2'b11)***. To make the sequence detector ***overlapping*** we took care of the transitions involved in making the state diagram. For the circuit, we've used two D Flip-Flops $\mathbf{D_x}$ and $\mathbf{D_y}$ to store the current state $\mathbf{XY}$ ($X = D_x$ & $Y = D_y$).

1.2 State Diagram

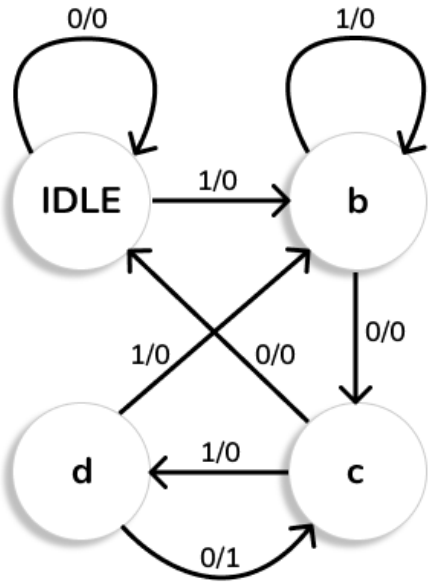


Figure 1: State diagram of sequence detector

1.3 State Table

PS	NS, O/P	
	x_1	
	0	1
00	(00,0)	(10,0)
01	(00,0)	(11,0)
10	(01,0)	(10,0)
11	(01,1)	(10,0)

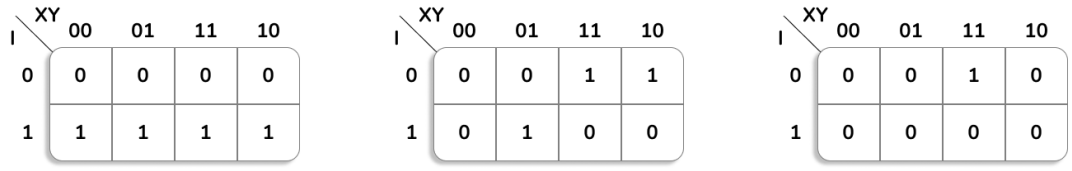
Table 1: State table

1.4 Excitation Table

P.S.	P.S.	I/P	N.S.	N.S.	FF's	FF's	O/P
X	Y		X'	Y'	D_x	D_Y	
0	0	0	0	0	0	0	0
0	0	1	1	0	1	0	0
0	1	0	0	0	0	0	0
0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	0
1	0	1	1	0	1	0	0
1	1	0	0	1	0	1	1
1	1	1	1	0	1	0	0

Table 2: Excitation table

1.5 K-map



(a) D_x (b) D_y (c) Z

Figure 2: *K-maps*

$\Rightarrow D_x = I$
$\Rightarrow D_y = I'XY' + IX'Y + I'XY = I'X + IX'Y$
$\Rightarrow Z = I'XY$

1.6 Circuit Diagram

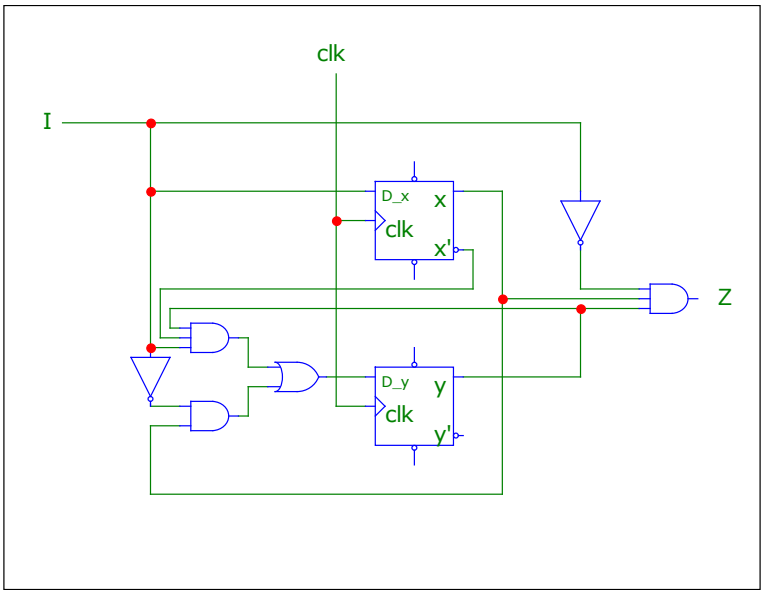


Figure 3: Circuit diagram of sequence detector

2 3-bit Odd Parity Bit Generator

2.1 Description

The odd parity bit generator was made using ***Moore FSM Machine***. We've used **three states** to build the state machine: ***IDLE(2'b00)***, ***a(2'b01)*** and ***b(2'b10)***. For the circuit, we've used one D Flip-Flop $\mathbf{D_x}$ to store the current state $\mathbf{X}$ ($X = D_x$). The state machine starts at the state *IDLE* and then goes to the state a if input is 1 or goes to b if input is 0 else it stays in state *IDLE* if input is a white space. After going to state a or b the state machine stays at the same state if the input is 0 or goes to the other state(from a to b or vice versa) else goes to state *IDLE* if the input is a white space. Since this is a Moore machine, it will give an output of 1 when it is in state b(2'b10) otherwise will give 0 as output in other states.

2.2 State Diagram

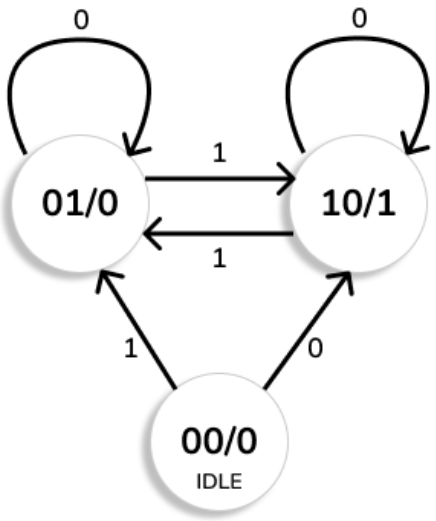


Figure 4: State diagram of odd parity bit generator

2.3 State Table

PS	NS, O/P	
	x_1	
	0	1
0	(0,0)	(1,1)
1	(1,1)	(0,0)

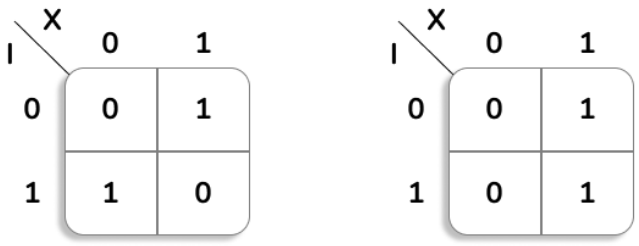
Table 3: State table

2.4 Excitation Table

P.S.	I/P	N.S.	FF's	O/P
X		X'	D_x	(at P.S.)
0	0	0	0	0
0	1	1	1	0
0	0	0	0	0
0	1	1	1	0

Table 4: Excitation table

2.5 K-map



(a) D_x (b) Z

Figure 5: *K-maps*

$\Rightarrow D_x = I'X + IX' = I \oplus X$
$\Rightarrow Z = X$

2.6 Circuit Diagram

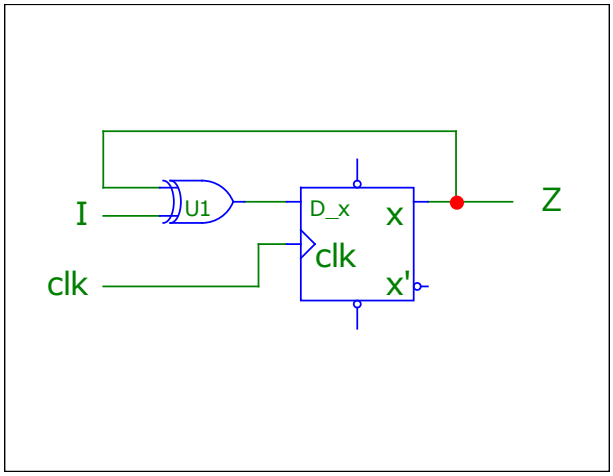


Figure 6: Circuit diagram of odd parity bit generator