COE328 Digital Systems

Lecture 15

Dr. Shazzat Hossain

Moore state : output depends on present states

1. State diagram
2. State Table
3. state-assigned table
4. logic function
5. Implementation.

Mealy → output depends on present states and current input

Mealy State Circuit

- Mealy Circuit: Outputs depend on the present state and inputs.
- Design steps for FSM:

Step#1: Determine how many states are possible

Step#2: Which transitions are possible from one state to another state

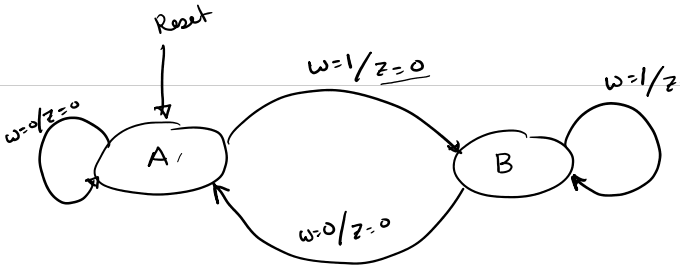
Step#3: Begin from a state (when power is turned on or reset is used)

Suppose that we wish to design a circuit that meets the following specification:

1. The circuit has one input, w, and one output, z.
2. All changes in the circuit occur on the positive edge of a clock signal.
3. The output z is equal to 1 if during two immediately preceding clock cycles the input w was equal to 1. Otherwise, the value of z is equal to 0.

Clock cycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w:	0	1	0	1	1	0	1	1	1	0	1
z:	0	0	0	0	0	1	0	0	1	1	0

Figure 8.2 Sequences of input and output signals.



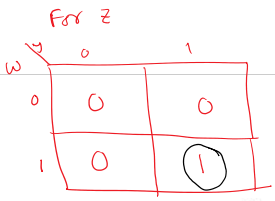
State Table

Present state	Next state $w=0$	Next state $w=1$	Output z: $w=0$	Output z: $w=1$
A	A	B	0	0
B	A	B	0	1

State-assigned table : A = 0 ; B = 1

Present state y	Next state Y: $w=0$	Next state Y: $w=1$	Output z: $w=0$	Output z: $w=1$
0	0	1	0	0

Logic Function For z



$z = wy$

	$w=0$	$w=1$	$w=0$	$w=1$
y	Y	Y		
0	0 ✓	1	0	0
1	0	1	0	1

Mealy State Model

The state diagram is shown in Fig 8.23. The state table shows that the output z depends on the present value of input w and the present state. The state table and state-assigned table are shown in the following figures.

We implement using D FF

$Y = D$

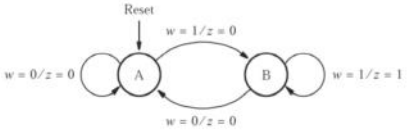


Figure 8.23 State diagram of an FSM that realizes the task in Figure 8.22.

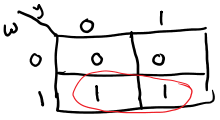
Present state	Next state		Output z	
	$w=0$	$w=1$	$w=0$	$w=1$
A	A	B	0	0
B	A	B	0	1

Figure 8.24 State table for the FSM in Figure 8.23.

Present state	Next state		Output	
	$w=0$	$w=1$	$w=0$	$w=1$
y	Y	Y	z	z
A 0	0	1	0	0
B 1	0	1	0	1

Figure 8.25 State-assigned table for the FSM in F

Logic Function



For Y

$Y = w$

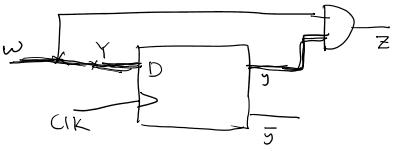
Logic Functions

Next state → $Y = w$, $z = wy$ ← Present state

output

input

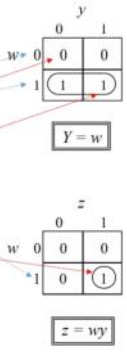
D



Logic Function

Present state	Next state		Output	
	$w=0$	$w=1$	$w=0$	$w=1$
y	Y	Y	z	z
A 0	0	1	0	0
B 1	0	1	0	1

Figure 8.25 State-assigned table for the FSM in Figure 8.24.



State-assigned Table

Present state y	Next state $w=0$ Y	T	$w=1$ Y	T	Output $w=0$	$w=1$
0	0	0	1	1	0	0
1	0	1	1	0	0	1

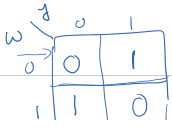
excitation Table

	P.S Q_n	N.S Q_{n+1}	V_p T
No change	0	0	0
Toggle →	0	1	1
Toggle →	1	0	1
No change	1	1	0

Formula sheet

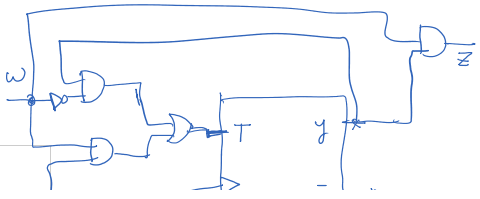
Logic Function

For T

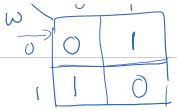


For z

$z = wy$



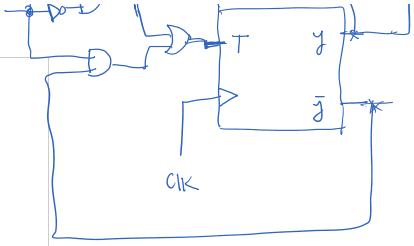
For T



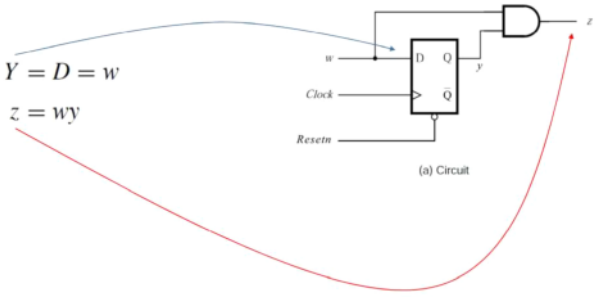
$T = \bar{w}y + w\bar{y}$

$= w \oplus y$

$z = wy$



Implementation



Design Using T-FF

Present State	Next State				Output, z	
	W=0		W=1		W=0	W=1
y	Y	T	Y	T		
0	0	0	1	1	0	0
1	0	1	1	0	0	1

0	1
1	0

0	0
0	1

$T = \bar{w}y + w\bar{y}$

z=wy

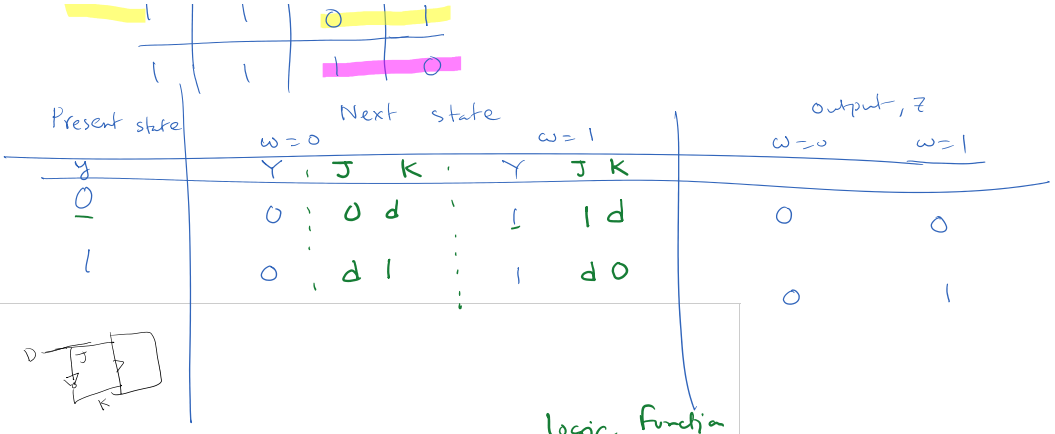
Characteristic Table

J	K	Q_n (P.S)	Q_{n+1} (N.S)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Excitation table

Q_n	Q_{n+1}	J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

↑ make sure that this table on your formula sheet



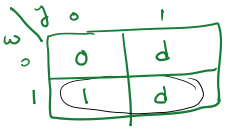
Design using J-K Flip-Flop

logic function

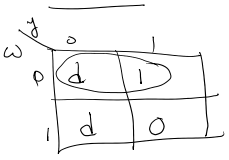
For J

For k

Present State	Next State						Output, z	
	W=0			W=1			W=0	W=1
y	Y	J	K	Y	J	K		
0	0	0	d	1	1	d	0	0
1	0	d	1	1	d	0	0	1



J = w



z = wy

K = $\bar{w}$

0	d
1	d

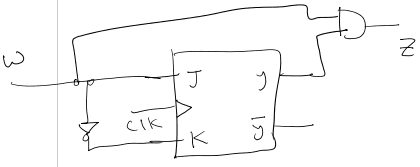
J = w

d	1
d	0

K = $\bar{w}$

0	0
0	1

z = wy



Design a Counter using the Sequential Circuit Approach

We will design a counter using the sequential circuit approach

The counting sequence is: 0-1-2-3-4-5-6-7-0

There exists an input signal w. The value of this signal is considered during each clock.

If w=0, the present count remains the same. If w=1, the count will be incremented.

Modulo-8 Counter

- State Diagram and State Table

Figure 8.60 gives a state diagram for the desired counter. There is a state associated with each count. In the diagram state *A* corresponds to count 0, state *B* to count 1, and so on. We show the transitions between the states needed to implement the counting sequence. Note that the output signals are specified as depending only on the state of the counter at a given time, which is the Moore model of sequential circuits.The state diagram may be represented in the state-table form as shown in Figure 8.61.

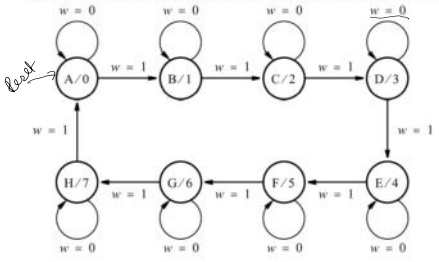


Figure 8.60 State diagram for the counter.

Present state	Next state $w=0$	Next state $w=1$	Output
A	A	B	0
B	B	C	1
C	C	D	2
D	D	E	3
E	E	F	4
F	F	G	5
G	G	H	6
H	H	A	7

← 3 Flip Flops

Figure 8.61 State table for the counter.

160

A → 000
B → 001

State Table and State-assigned Table

Three state variables are needed to represent the eight states. Let these variables, denoting the present state, be called y_2, y_1, and y_0. Let Y_2, Y_1, and Y_0 denote the corresponding next-state functions. The most convenient (and simplest) state assignment is to encode each state with the binary number that the counter should give as output in that state. Then the required output signals will be the same as the signals that represent the state variables. This leads to the state-assigned table in Figure 8.62.

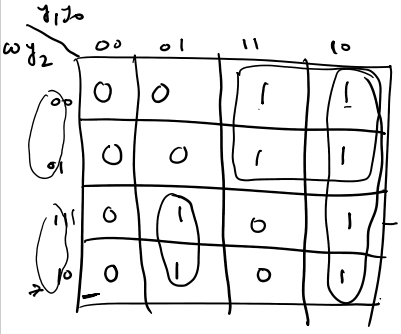
Present state	Next state $w=0$	Next state $w=1$	Output
A	A	B	0
B	B	C	1
C	C	D	2
D	D	E	3
E	E	F	4
F	F	G	5
G	G	H	6
H	H	A	7

Figure 8.61 State table for the counter.

	Present state $y_2y_1y_0$	Next state $w=0$ $Y_2Y_1Y_0$	Next state $w=1$ $Y_2Y_1Y_0$	Count $z_2z_1z_0$
A	000	000	001	000
B	001	001	010	001
C	010	010	011	010
D	011	011	100	011
E	100	100	101	100
F	101	101	110	101
G	110	110	111	110
H	111	111	000	111

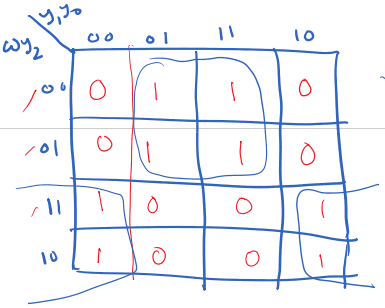
Figure 8.62 State-assigned table for the counter.

$y_0 = z_0$
$y_1 = z_1$
$y_2 = z_2$

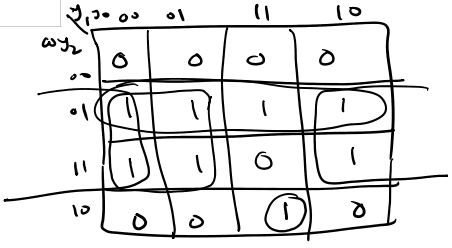


$Y_1 = \bar{w}y_1 + y_1\bar{y}_0 + w\bar{y}_1y_0$

logic Function For Y_0



$Y_0 = \bar{w}y_0 + w\bar{y}_0$



$Y_2 = \bar{w}y_2 + y_2\bar{y}_1 + y_2\bar{y}_0 + w\bar{y}_2y_1y_0$

Implementation using D Flip-Flops

When using D-type flip-flops to realize the finite state machine, each next-state function, Y_i, is connected to the D input of the flip-flop that implements the state variable y_i. The next-state functions are derived from the information in Figure 8.62. Using Karnaugh maps in Figure 8.63

	Present state $y_2y_1y_0$	Next state $w=0$ $Y_2Y_1Y_0$	Next state $w=1$ $Y_2Y_1Y_0$	Count $z_2z_1z_0$
A	000	000	001	000
B	001	001	010	001
C	010	010	011	010
D	011	011	100	011
E	100	100	101	100
F	101	101	110	101
G	110	110	111	110
H	111	111	000	111

wy_2 \ y_1y_0	00	01	11	10
00	0	*1	1	0
01	0	*1	1	0
11	1	0	0	1
10	1	0	0	1

$$Y_0 = \bar{w}y_0 + w\bar{y}_0$$

Figure 8.62 State-assigned table for the counter.

Logic Function

	Present state $y_2y_1y_0$	Next state $w=0$ $Y_2Y_1Y_0$	Next state $w=1$ $Y_2Y_1Y_0$	Count $z_2z_1z_0$
A	000	000	001	000
B	001	001	010	001
C	010	010	011	010
D	011	011	100	011
E	100	100	101	100
F	101	101	110	101
G	110	110	111	110
H	111	111	000	111

Figure 8.62 State-assigned table for the counter.

wy_2 \ y_1y_0	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	1	0	0	1
10	1	0	0	1

$$Y_0 = \bar{w}y_0 + w\bar{y}_0$$

y_1y_0 \	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	0	1	0	1
10	0	1	0	1

$$Y_1 = \bar{w}y_1 + y_1\bar{y}_0 + wy_0\bar{y}_1$$

wy_2 \ y_1y_0	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	1	1	0	1
10	0	0	1	0

$$Y_2 = \bar{w}y_2 + \bar{y}_0y_2 + \bar{y}_1y_2 + wy_0y_1\bar{y}_2$$

Implementation

$$D_0 = Y_0 = \overline{w}y_0 + w\overline{y}_0$$
$$D_1 = Y_1 = \overline{w}y_1 + y_1\overline{y}_0 + wy_0\overline{y}_1$$
$$D_2 = Y_2 = \overline{w}y_2 + \overline{y}_0y_2 + \overline{y}_1y_2 + wy_0y_1\overline{y}_2$$

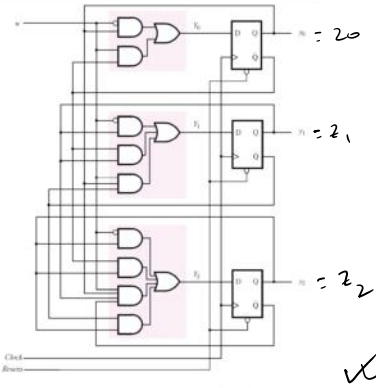


Figure 8.64 Circuit diagram for the counter implemented with D flip-flops.

Moore design

Implementation Using JK Flip-Flops

The final step in the design is to choose the type of flip-flops and derive the expressions that control the flip-flop inputs. The most straightforward choice is to use D-type flip-flops. We pursue this approach first. Then we show the alternative of using JK-type flip-flops. In either case the flip-flops must be edge triggered to ensure that only one transition takes place during a single clock cycle.

Present State	Next State D-FF	J K-FF	T-FF
0	0	0 d	0
0	1	1 d	1
1	1	d 0	0
1	0	d 1	1

J_2K_2 J_1K_1 Next state J_0K_0 $w=0$

Present state $y_2\ y_1\ y_0$	$Y_2\ Y_1\ Y_0$	$J_2\ K_2$	$J_1\ K_1$	$J_0\ K_0$
0 0 0	0 0 0	0 d	0 d	0 d
0 0 1	0 0 1	0 d	0 d	d 0
0 1 0	0 1 0	0 d	d 0	0 d
0 1 1	0 1 1	0 d	d 0	d 0
1 0 0	1 0 0	d 0	0 d	0 d
1 0 1	1 0 1	d 0	0 d	d 0
1 1 0	1 1 0	d 0	d 0	0 d
1 1 1	1 1 1	d 0	d 0	d 0

State-assigned Table $w=1$

$Y_2\ Y_1\ Y_0$	$J_2\ K_2$	$J_1\ K_1$	J_0K_0	output
0 0 1	0 d	0 d	1 d	
0 1 0	0 d	1 d	d 1	
0 1 1	0 d	d 0	1 d	
1 0 0	1 d	d 1	d 1	
1 0 1	d 0	0 d	1 d	
1 1 0	d 0	1 d	d 1	
1 1 1	d 0	d 0	1 d	
0 0 0	d 1	d 1	d 1	

State-assigned Table

Once the table in Figure 8.65 has been derived, it provides a truth table with inputs y_2, y_1, y_0, and w, and outputs J_2, K_2, J_1, K_1, J_0, and K_0. We can then derive expressions for

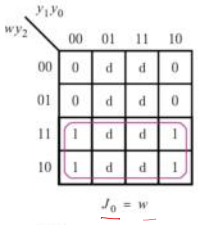
Present State	Next State D-FF	J K-FF	T-FF
0	0	0 d	0
0	1	1 d	1
1	1	d 0	0
1	0	d 1	1

	Present state $y_2y_1y_0$	Flip-flop inputs (Next State) $w = 0$ $Y_2Y_1Y_0$	J_2K_2	J_1K_1	J_0K_0	$w = 1$ $Y_2Y_1Y_0$	J_2K_2	J_1K_1	J_0K_0	Count $z_2z_1z_0$
A	000	000	0*d*	0*d*	0*d*	001	0*d*	0*d*	1*d*	000
B	001	001	0*d*	0*d*	*d*0	010	0*d*	1*d*	*d*1	001
C	010	010	0*d*	*d*0	0*d*	011	0*d*	*d*0	1*d*	010
D	011	011	0*d*	*d*0	*d*0	100	1*d*	*d*1	*d*1	011
E	100	100	*d*0	0*d*	0*d*	101	*d*0	0*d*	1*d*	100
F	101	101	*d*0	0*d*	*d*0	110	*d*0	1*d*	*d*1	101
G	110	110	*d*0	*d*0	0*d*	111	*d*0	*d*0	1*d*	110
H	111	111	*d*0	*d*0	*d*0	000	*d*1	*d*1	*d*1	111

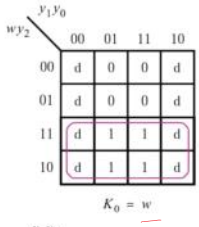
Figure 8.65 Excitation table for the counter with JK flip-flops.

169

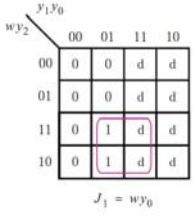
Logic Function



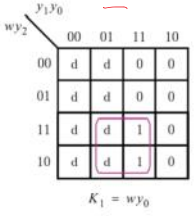
$J_0 = w$



$K_0 = w$



$J_1 = wy_0$



$K_1 = wy_0$

Logic Function

wy_2 \ y_1y_0	00	01	11	10
00	0	0	0	0
01	d	d	d	d
11	d	d	d	d
10	0	0	1	0

$J_2 = wy_0y_1$

wy_2 \ y_1y_0	00	01	11	10
00	d	d	d	d
01	0	0	0	0
11	0	0	1	0
10	d	d	d	d

$K_2 = wy_0y_1$

Figure 8.66 Karnaugh maps for JK flip-flops in the counter.

Implementation

$J_0 = K_0 = w$
$J_1 = K_1 = wy_0$
$J_2 = K_2 = wy_0y_1$

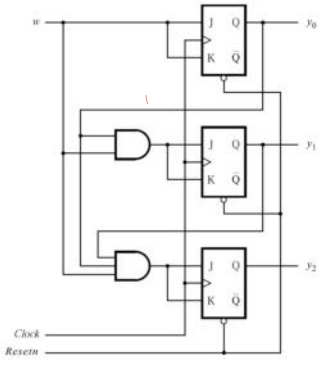


Figure 8.67 Circuit diagram using JK flip-flops.

Implement using T Flip-Flops

	Present state $y_2 y_1 y_0$	Next state $w=0$ $Y_2 Y_1 Y_0$	Next state $w=1$ $Y_2 Y_1 Y_0$	Count $z_2 z_1 z_0$
A	000	000	001	000
B	001	001	010	001
C	010	010	011	010
D	011	011	100	011
E	100	100	101	100
F	101	101	110	101
G	110	110	111	110
H	111	111	000	111

Present State	Next State D-FF	J K-FF	T-FF
0	0	0 d	0
0	1	1 d	1
1	1	d 0	0
1	0	d 1	1

- Convert the state assigned table to T-Type FFs
- Design the circuit, use the K-Map