JK Flip flops Implementation using D flip flops (Synchronous Counter)

Satheesh K Simhachalam*

Contents

***Abstract*—This manual explains steps to implement JK flip flop synchronous counters with D flip flops using finite state machine.**

1 The Assignment

1. **Problem Statement:** A synchronus counter using two J-K flip flops that goes through the sequence of states: $Q_1Q_2 = 00 \rightarrow 10 \rightarrow 01 \rightarrow 11 \rightarrow 00$ .. is required as mentioned in Fig. 1. We need to implement this sequencing logic in ide, assembly and avr-gcc using D-flip flops. The *state transitioning* decoder and *display* decoder are part of *combinational* logic, while the *delay* is part of *sequential* logic.

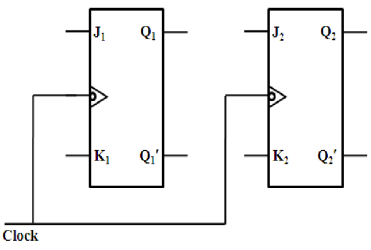

Fig. 1: Synchronous counter with JK flip flops

*The author is a PhD scholar with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: ee22resch11010@iith.ac.in.

2. The required J and K inputs for 2 J-K flip flops are as mentioned in the Table 2.

Present state		Next state		Flip flop Inputs			
X (Q_1)	W (Q_2)	B (Q_1^+)	A (Q_2^+)	J_1	K_1	J_2	K_2
0	0	1	0	1	X	0	X
1	0	0	1	X	1	1	X
0	1	1	1	1	X	X	0
1	1	0	0	X	1	X	1

TABLE 2: JK flip flop State Transition Table

3. From the columns of J_1,K_1, J_2,K_2, we can say

$$J_1 = 1 \tag{1.1}$$

$$K_1 = 1 \tag{1.2}$$

$$J_2 = Q_1 \tag{1.3}$$

$$K_2 = Q_1 \tag{1.4}$$

2 Realization of JK flip flop using D flip flop

1. Now, we have to design to realize JK flip flop using a D flip flop. D flip flop is primarily meant to provide delay as the output of this flip flop is same as the input.

   The first thing that needs to be done for this conversion is to draw the truth table for both the flip flops as shown in Table 1. The next step is to create the equivalent K-Maps for the required outputs.

2. The K-Map for required input- output relation is as shown in Fig 2
3. Obtain the state transition equations for **D** from Fig. 2

$$D = JQ' + K'Q \tag{2.1}$$

J	K	Q	Q^+	D
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

TABLE 1: JK vs D Transition Table

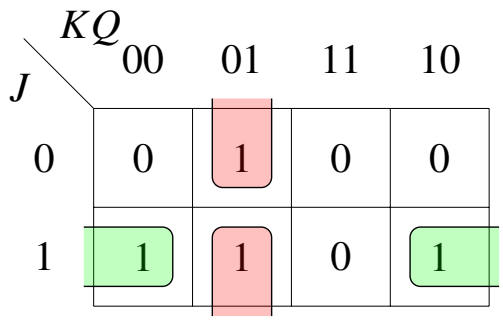

Fig. 2: K-map for D.

3 Finite State Machine

1. Fig. 1 shows a *finite state machine* (FSM) diagram for the counter in Fig. 1.

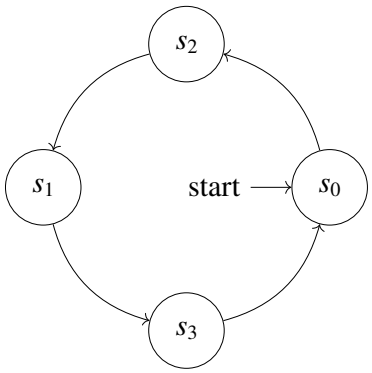

Fig. 1: FSM for the counter

2. From 2.1, D inputs for D flip flops are derived using the following equations

$$Q_1{}^+ = D_1 = J_1Q_1{}' + K_1{}'Q_1 \tag{3.1}$$

Substituting values for J_1, K_1 from 1.1 and 1.2,

$$Q_1{}^+ = D_1 = 1Q_1{}' + 0Q_1 \tag{3.2}$$

$$Q_1{}^+ = D_1 = Q_1{}' \tag{3.3}$$

Similarly,

$$Q_2{}^+ = D_2 = J_2Q_2{}' + K_2{}'Q_2 \tag{3.4}$$

Substituting values for J_2, K_2 from 1.3 and 1.4,

$$Q_2{}^+ = D_2 = Q_1Q_2{}' + Q_1{}'Q_2 \tag{3.5}$$

$$Q_2{}^+ = D_2 = Q_1Q_2{}' + Q_1{}'Q_2 \tag{3.6}$$

3. The *state transition table* for the FSM is Table 3 where the present state is denoted by the variables $W(Q_2), X(Q_1)$ and the next state by $A(Q_2{}^+), B(Q_1{}^+)$.

X	W	B	A
0	0	1	0
1	0	0	1
0	1	1	1
1	1	0	0

TABLE 3: State Transition Table for D flip flops

The associated equations are:

$$A = XW' + WX' \tag{3.7}$$

$$B = X' \tag{3.8}$$

4 Implementation Details

1. Connect the Arduino, 7447, 7474 as per the details mentioned in Table 4 and Fig 4
2. The code for IDE implementation is avaialble at the following github link

https://github.com/satheeshsimha/fwc−2/blob/main/ide/assignment/codes/assignment1D_ide.cpp

3. The code for assembly implementation is avaialble at the following github link

https://github.com/satheeshsimha/fwc−2/blob/main/assembly/assignment/codes/assignment1.asm

4. The code for avr-gcc implementation is avaialble at the following github link

https://github.com/satheeshsimha/fwc−2/blob/main/avr−gcc/assignment/codes/main.c

	INPUT		OUTPUT		CLOCK		5V				GND		
	W	X	A	B									
Arduino	D8	D9	D2	D3	D13								
7474	5	9	2	12	CLK1	CLK2	1	4	10	13	7		
7447			7	1			16				8	2	6

TABLE 4: Connection Diagram

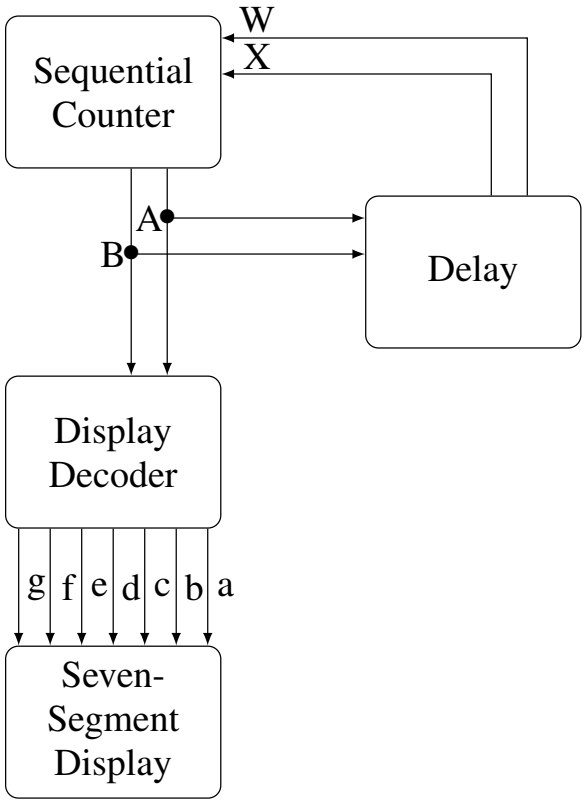

Fig. 4: The Synchronous counter with LED Display