Digital Systems Design with VHDL II

Dr. Dalila FARES
Assistant professor
M'Hamed Bougara University, Boumerdes.
Institue of Electrical and Electronic Engineering,IGEE.
Email: d.fares@univ-boumerdes.dz

May 2024

Caption

Table of contents

I- Chapter 3: Synchronous Counters *7*

A. Objectives.....*7*

B. Basic Concept.....*8*

C. Procedures to design a synchronous counter.....*10*

D. Design of 3-bit Gray code counter.....*10*

1. Step 1: State Diagram.....10
2. Step 2: Next-State Table.....10
3. Step 3: Flip-Flop Transition Table.....11
4. Step 4: Karnaugh Maps.....12
5. Step 5: Logic Expressions for Flip-Flop Inputs.....13
6. Step 6: Counter Implementation.....14

E. VHDL modeling of Synchronous counters.....*14*

F. Solved Problem.....*15*

G. Solved Problem.....*16*

Chapter 3: Synchronous Counters

Objectives	7
Basic Concept	8
Procedures to design a synchronous counter	10
Design of 3-bit Gray code counter.	10
VHDL modeling of Synchronous counters	14
Solved Problem	15
Solved Problem	16

A. Objectives

1. Define synchronous counters and explain their operational principles.
2. Identify the key components and characteristics of synchronous counters.
3. Explain the differences in behavior between synchronous counters and ripple counters.
4. Design synchronous counters for specific counting tasks, considering clocking requirements and sequence generation.
5. Write the VHDL code describing the behavior of synchronous counters.

B. Basic Concept

Fundamental

In **Synchronous Counter**, the external clock signal is connected to the clock input of **EVERY** individual **flip-flop** within the counter so that all of the flip-flops are clocked together **simultaneously** (in parallel) at the same time giving a **fixed time relationship**. In other words, changes in the output occur in **"synchronisation"** with the clock signal [3][Bibliography 1].

The result of the synchronisation is that all the individual output bits changing state at exactly the same time in response to the common clock signal with **no ripple effect** and therefore, **no propagation delay**.

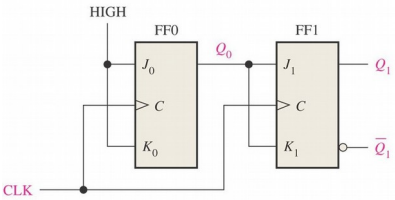


Image 1 2-Bit Synchronous Counter using JK Flip-Flops

- **J-K flip-flops** are used to illustrate most synchronous counters.

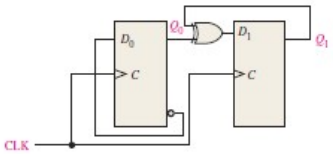


Image 2 2-Bit Synchronous Counter using D Flip-Flops

- **D flip-flops** can also be used but generally require more logic because of having no direct toggle or no-change states.

Combinational logic must be provided (calculated) attached to the flip-flop inputs to produce the desired sequence.

Moreover, in this mode, it is possible to design a counter that can scan through **any sequence** and possibly have more than **2-different directions** of counting.

Note

Due to its **parallel arrangement**, a synchronous counter can operate at a much higher frequency; the circuitry however is more complex compared to a ripple counter [1][Bibliography 1].[Bibliography 1]

C. Procedures to design a synchronous counter

Method

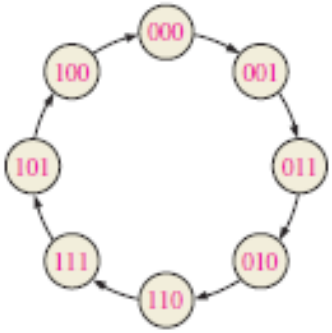
1. Specify the counter sequence and draw a state diagram.
2. Derive a next-state table from the state diagram.
3. Develop a transition table showing the flip-flop inputs required for each transition. The transition table is always the same for a given type of flip-flop.
4. Transfer the J and K states from the transition table to Karnaugh maps. There is a Karnaugh map for each input of each flip-flop.
5. Group the Karnaugh map cells to generate and derive the logic expression for each flip-flop input.
6. Implement the expressions with combinational logic, and combine with the flip-flops to create the counter.

D. Design of 3-bit Gray code counter.

1. Step 1: State Diagram

A **state diagram** shows the **progression of states** through which the counter advances when it is clocked.

For the Gray code counter circuit, it has no inputs other than **the clock** and no outputs other than the outputs taken off each flip-flop in the counter.



State diagram for a 3-bit Gray code counter

2. Step 2: Next-State Table

The **next-state table** lists each state of the counter (**present state**) along with the corresponding next state.

The **next state** is the state that the counter goes to from its present state upon application of a clock pulse.

Method

The next-state table is derived from the state diagram and is shown in the following Table for the **3-bit Gray code counter**. Q0 is the least significant bit.

Present State			Next State		
Q2	Q1	Q0	Q2	Q1	Q0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0

Next-state table of the 3-bit Gray code counter

3. Step 3: Flip-Flop Transition Table

All possible output transitions are listed in the transition table by showing the Q output of the flip-flop going from present states to next states.

QN is the present state of the flip-flop (before a clock pulse) and QN + 1 is the next state (after a clock pulse). For each output transition, the J and K inputs that will cause the transition to occur are listed. An X indicates a "don't care" (the input can

be either a 1 or a 0).

Method : Filling the Transition Table

To design the counter, the transition table is applied to each of the flip-flops in the counter, based on the next-state table.

- For the present state 000, Q0 goes from a present state of 0 to a next state of 1. To make this happen, J0 must be a 1 and you don't care what K0 is (J0 = 1, K0 = X), as you can see in the transition table.
- Next, Q1 is 0 in the present state and remains a 0 in the next state. For this transition, J1 = 0 and K1 = X.
- Finally, Q2 is 0 in the present state and remains a 0 in the next state. Therefore, J2 = 0 and K2 = X. This analysis is repeated for each present state in the table.

Output Transitions			Flip-Flop inputs	
Q_N		Q_{N+1}	J	K
0	---→	0	0	X
0	---→	1	1	X
1	---→	0	X	1
1	---→	1	X	0

Q_N: Present state, Q_{N+1}: Next state, X: don't care

Transition Table of a JK Flip Flop

4. Step 4: Karnaugh Maps

Karnaugh maps can be used to determine the logic required for the J and K inputs of each flip-flop in the counter.

There is a Karnaugh map for the J input and a Karnaugh map for the K input of each flip-flop.

Method

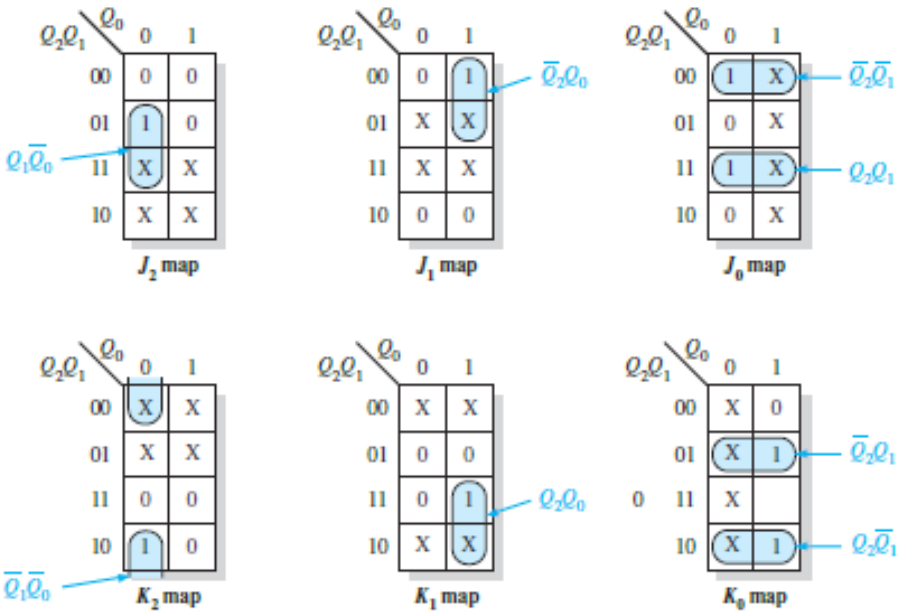
Each cell in a Karnaugh map represents one of the present states in the counter sequence .

From the J and K states in the transition table, 1, 0, or X is entered into each present-state cell on the maps depending on the transition of the Q output for a particular

flip-flop.

5. Step 5: Logic Expressions for Flip-Flop Inputs

From the Karnaugh maps of the present-state J and K inputs, the following expressions for the J and K inputs of each flip-flop can be obtained:



The completed Karnaugh maps for present-state J and K inputs

$$J_0 = Q_2Q_1 + \bar{Q}_2\bar{Q}_1$$

$$K_0 = Q_2\bar{Q}_1 + \bar{Q}_2Q_1$$

$$J_1 = \bar{Q}_2Q_0$$

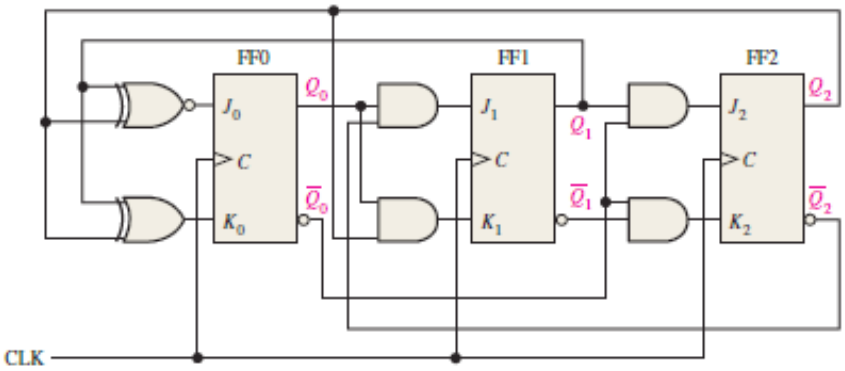
$$K_1 = Q_2Q_0$$

$$J_2 = Q_1\bar{Q}_0$$

$$K_2 = \bar{Q}_1\bar{Q}_0$$

6. Step 6: Counter Implementation

The final step is to implement the combinational logic from the expressions for the J and K inputs and connect the flip-flops to form the complete 3-bit Gray code counter as shown below:



Three-bit Gray code counter.

E. VHDL modeling of Synchronous counters

The behavioral VHDL code to synthesize a Mod-8 UP synchronous counter with positive-edge triggering and active high clear input [3][Bibliography 3]:

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3
4  entity Mod_8_Up_Synchronous_Counter is
5      Port ( clk : in STD_LOGIC;
6             reset : in STD_LOGIC;
7             count_out : out STD_LOGIC_VECTOR(2 downto 0));
8  end Mod_8_Up_Synchronous_Counter;
9
10 architecture Behavioral of Mod_8_Up_Synchronous_Counter is
11     signal counter : STD_LOGIC_VECTOR(2 downto 0);
12 begin
13     process(clk, reset)
14     begin
15         if reset = '1' then
16             counter <= "000";
17         elsif rising_edge(clk) then
18             if counter = "111" then
19                 counter <= "000";
20             else
21                 counter <= counter + 1;
22             end if;
23         end if;
24     end process;
25
26     count_out <= counter;
27 end Behavioral;
28
```

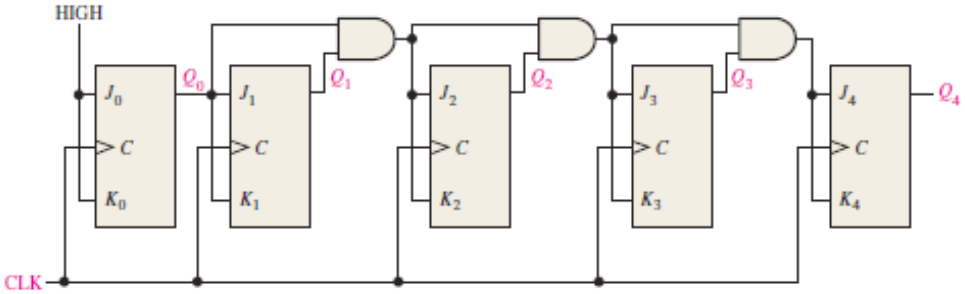
A structural VHDL code to synthesize a Mod-8 UP synchronous counter using D-FF with positive-edge triggering and active high clear input:

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
```

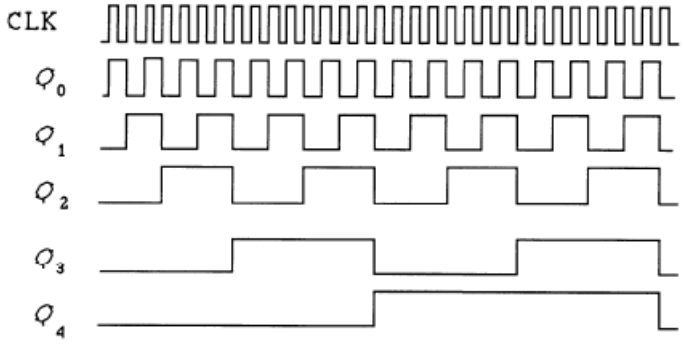
```
3
4   entity Mod8UpSyncCounter is
5       Port (
6           Clock : in STD_LOGIC;
7           Clear : in STD_LOGIC;
8           Count : out STD_LOGIC_VECTOR(2 downto 0)
9       );
10  end Mod8UpSyncCounter;
11
12  architecture Structural of Mod8UpSyncCounter is
13      component DFlipFlop is
14          Port (
15              D, Clock, Clear : in STD_LOGIC;
16              Q : out STD_LOGIC
17          );
18      end component;
19
20      signal Q0, Q1, Q2 : STD_LOGIC;
21  begin
22      FF0: DFlipFlop port map(D => '1', Clock => Clock, Clear => Clear,
    Q => Q0);
23      FF1: DFlipFlop port map(D => Q0, Clock => Clock, Clear => Clear,
    Q => Q1);
24      FF2: DFlipFlop port map(D => Q1, Clock => Clock, Clear => Clear,
    Q => Q2);
25
26      Count <= Q2 & Q1 & Q0;
27  end Structural;
28
```

F. Solved Problem

Show the complete timing diagram for the 5-stage synchronous binary counter in the illustrated figure. Verify that the waveforms of the Q outputs represent the proper binary number after each clock pulse.



Solution:

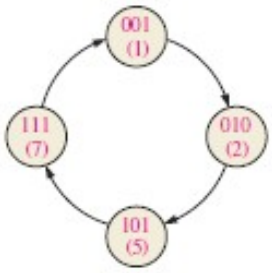


G. Solved Problem

Design a counter with the irregular binary count sequence (1, 2, 5, 7).

Solution

Step 1: The state diagram



Step 2: The next-state table

Next-state table.

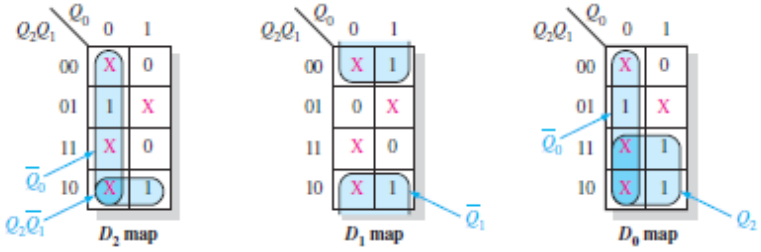
Present State			Next State		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	1	0	1	0
0	1	0	1	0	1
1	0	1	1	1	1
1	1	1	0	0	1

Step 3: The transition table for the D flip-flop

Transition table for a D flip-flop.

Output Transitions			Flip-Flop Input
Q_N		Q_{N+1}	D
0	⟶	0	0
0	⟶	1	1
1	⟶	0	0
1	⟶	1	1

Step 4: The D inputs are plotted on the present-state Karnaugh maps.



Step 5: The expressions

$$D_0 = \bar{Q}_0 + Q_2$$

$$D_1 = \bar{Q}_1$$

$$D_2 = \bar{Q}_0 + Q_2\bar{Q}_1$$

Step 6: The implementation of the counter