

Tut 10

EE 2000
2020/21 Sem B

1. Using a synchronous binary counters as shown in Figure 1, design and draw a counter to generate the following repeating sequences 2 to 9 repeatedly for a free running clock.

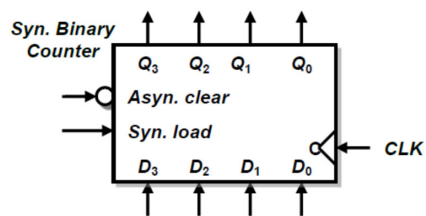


Figure 1.

Q0	Q1	Q2	Q3
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

PS		Inputs		NS		FF-A		FF-B		Outputs	
A	B	E	x	A	B	JA	KA	JB	KB	A	B
0	0	0	0	0	0	0	X	0	X	0	0
0	0	0	1	0	0	0	X	0	X	0	0
0	0	1	0	1	1	1	X	1	X	0	0
0	0	1	1	0	1	0	X	1	X	0	0
0	1	0	0	0	1	0	X	x	0	0	1
0	1	0	1	0	1	0	X	x	0	0	1
0	1	1	0	0	0	0	X	x	1	0	1
0	1	1	1	1	0	1	X	x	1	0	1
1	0	0	0	1	0	x	0	0	X	1	0
1	0	0	1	1	0	x	0	0	X	1	0
1	0	1	0	0	1	x	1	1	X	1	0
1	0	1	1	1	1	x	0	1	X	1	0
1	1	0	0	1	1	x	0	x	0	1	1
1	1	0	1	1	1	x	0	x	0	1	1
1	1	1	0	1	0	x	0	x	1	1	1
1	1	1	1	0	0	x	1	x	1	1	1

Q(t)	Q(t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

JK Excitation Table.

EX

AB	00	01	11	10
00				1
01			1	
11	X	X	X	X
10	X	X	X	X

$JA = (Bx + B'x')E$

EX

AB	00	01	11	10
00	X	X	X	X
01	X	X	X	X
11			1	
10				1

$KA = (Bx + B'x')E$

EX

AB	00	01	11	10
00			1	1
01	X	X	X	X
11	X	X	X	X
10			1	1

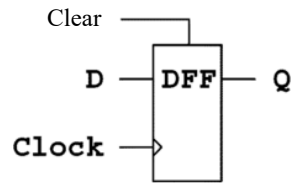
$JB = E$

EX

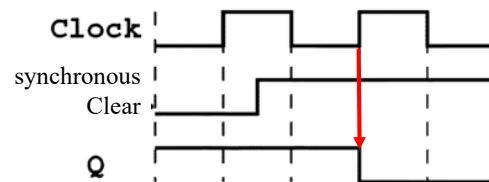
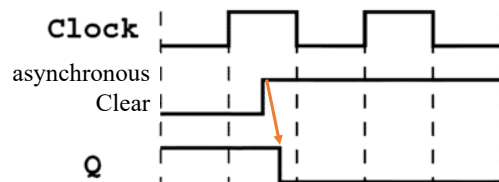
AB	00	01	11	10
00	X	X	X	X
01			1	1
11			1	1
10	X	X	X	X

$KB = E$

3. Write the VHDL code for a D-FF with (i) asynchronous Preset and Clear, and (ii) synchronous Preset and Clear.



(a)



D-FF asynchronous Preset and Clear

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

entity DFF_Asyn_Pre_Clr is
    port(
        D : in std_logic;
        Clk : in std_logic;
        Pre: in std_logic;
        Clr: in std_logic;
        Q : out std_logic
    );
end DFF_Asyn_Pre_Clr;

architecture Behavioral of DFF_Asyn_Pre_Clr is
begin
    process(Clk,Pre,Clr)
    begin
        if(Pre='1') then
            Q <= '1';
        elsif (Clr = '1') then
            Q <= '0';
        elsif(rising_edge(Clk)) then
            Q <= D;
        end if;
    end process;
end Behavioral;
```

D-FF synchronous Preset and Clear.

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

entity DFF_Syn_Pre_Clr is
    port(
        D : in std_logic;
        Clk : in std_logic;
        Pre: in std_logic;
        Clr: in std_logic;
        Q : out std_logic
    );
end DFF_Syn_Pre_Clr;

architecture Behavioral of DFF_Syn_Pre_Clr is
begin
    process(Clk,Pre,Clr)
    begin
        if(rising_edge(Clk)) then
            if(Pre = '1') then
                Q <= '1';
            elsif (Clr = '1') then
                Q <= '0';
            else
                Q <= D;
            end if;
        end if;
    end process;
end Behavioral;
```