**Counter**

Q1. Following circuit is a counter

1. Complete the timing diagram of the counter starting from Q2Q1Q0=000.

1

Q

Q

Q

T

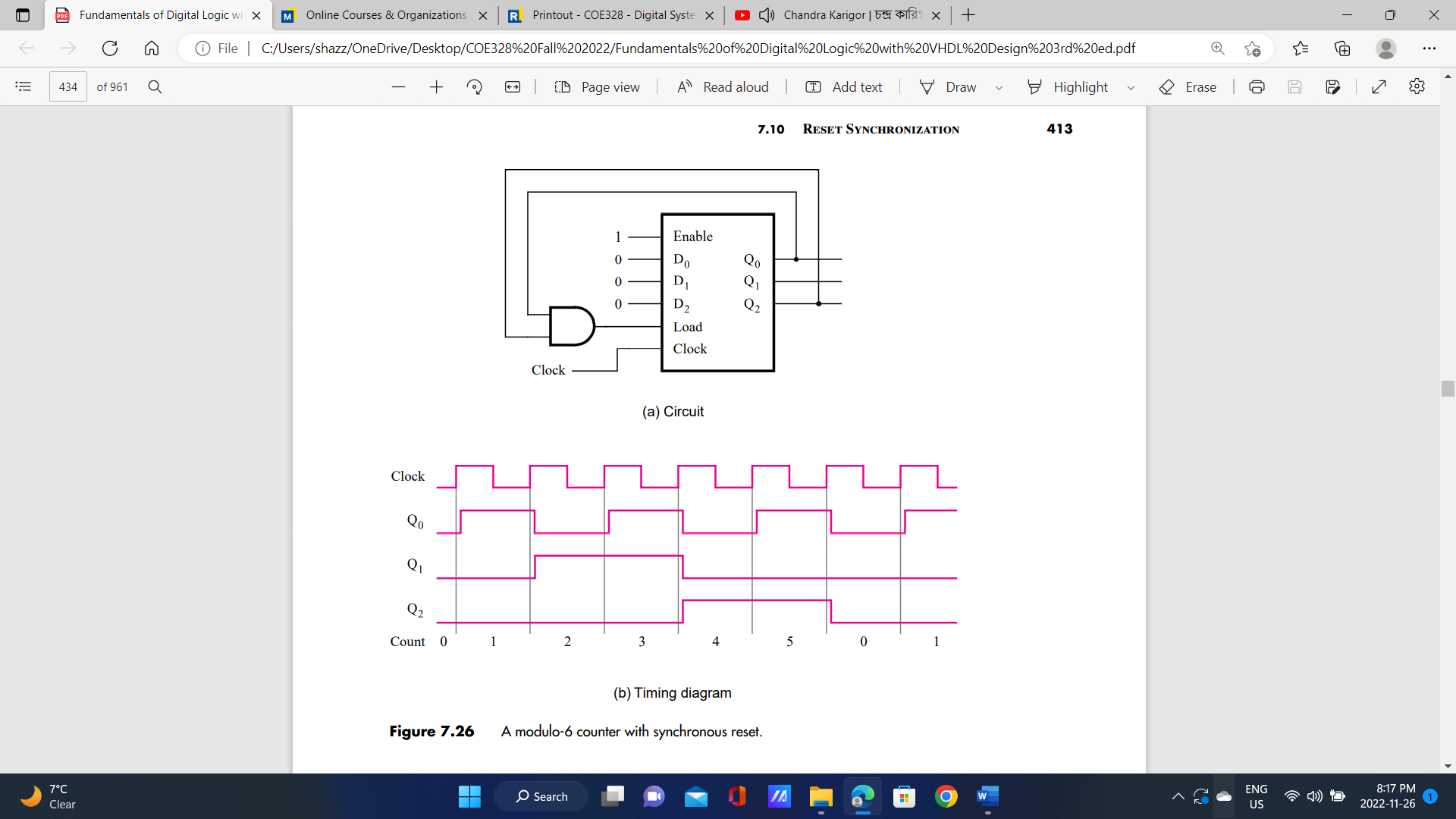
J

T

K

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. Change the design of the counter to a modulo-10 up counter.
2. Design a modulo-6 counter with synchronous reset



1. Design a modulo-6 counter with asynchronous reset.

Diagram

Description automatically generated

Q2. Show how a JK flip-flop can be constructed using a T flip-flop and other logic gates.

A picture containing graphical user interface

Description automatically generated

**VHDL Code**

Q1. Consider the following VHDL code:

LIBRARY ieee;

USE iee.std\_loic\_1164.all;

ENTITY fsm IS

PORT (Clock, Resetn, w : IN STD\_LOGIC;

z : OUT STD\_LOGIC );

END fsm;

ARCHITECTURE Behavior OF fsm IS

TYPE State\_type IS (A, B, C);

SIGNAL y: State\_type;

BEGIN

PROCESS(Resetn, Clock)

BEGIN

IF Resetn =’0’ THEN

y<=A;

ELSEIF (Clock’EVENT AND Clock = ‘1’) THEN

CASE y IS

WHEN A=>

IF w=’0’ THEN y<=A;

ELSE y<=B;

END IF;

WHEN B=>

IF w=’0’ THEN y<=A;

ELSE y<=C;

END IF;

WHEN C=>

IF w=’0’ THEN y<=A;

ELSE y<=C;

END IF;

END CASE;

ENDIF;

END PROCESS;

z<=’1’ WHEN y=C ELSE ‘0’;

END Behavior;

1. Describe the behavior of the given FSM using a state diagram.
2. Assume that an 8×4-bit EPROM is available, provide a programmable implementation of the FSM in part (a):
3. Fill in the content of the EPROM in the table below by setting up the address of the EPROM as follows: a2=w, a1ao=y1y2 (present sate), and explain what the contents of the EPROM represents.
4. Draw a schematic diagram for the implementation.

Q2.

LIBRARY ieee;

USE iee.std\_loic\_1164.all;

ENTITY fsm IS

PORT (Clock, Resetn, w : IN STD\_LOGIC;

z : OUT STD\_LOGIC );

END fsm;

ARCHITECTURE Behavior OF fsm IS

TYPE State\_type IS (A, B, C);

SIGNAL y: State\_type;

BEGIN

PROCESS(Resetn, Clock)

BEGIN

IF Resetn =’0’ THEN y<=A;

ELSEIF (Clock’EVENT AND Clock = ‘1’) THEN

CASE y IS

WHEN A=>

IF w=’0’ THEN y<=B;

ELSE y<=C;

END IF;

WHEN B=>

IF w=’0’ THEN y<=A;

ELSE y<=C;

END IF;

WHEN C=>

IF w=’0’ THEN y<=B;

ELSE y<=C;

END IF;

END CASE;

ENDIF;

END PROCESS;

PROCESS (y, w)

BEGIN

CASE y IS

WHEN A=>

z<=0

WHEN B=>

z<=NOT w;

WHEN C=>

z<=w

END CASE;

ENDIF;

END PROCESS;

END Behavior;

1. Describe the behavior of the given FSM using a state diagram.
2. Assume that an 84-bit EPROM is available, provide a programmable implementation of the FSM in part (a):
3. Fill in the content of the EPROM in the table below by setting up the address of the EPROM as follows: a2=w, a1ao=y­1y2 (present sate), and explain what the contents of the EPROM represents.
4. Draw a schematic diagram for the implementation.
5. Implement the FSM using D flip-flops.