Switch De-Bouncing

# CODE

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity Debounce is

Port ( Sig : in STD\_LOGIC;

Clk : in STD\_LOGIC;

Debout : out STD\_LOGIC);

end Debounce;

architecture Behavioral of Debounce is

signal Q0, Q1, Q2 : STD\_LOGIC :='0';

begin

process (Clk) is

begin

if rising\_edge(Clk) then

Q0 <= Sig;

Q1 <= Q0;

Q2 <= Q1;

end if;

end process;

Debout <= Q0 and Q1 and Q2;

end Behavioral;

TESTBENCH

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY temp\_tb IS

END temp\_tb;

ARCHITECTURE behavior OF temp\_tb IS

-- Component Declaration for the Unit Under Test (UUT)

COMPONENT Debounce

PORT(

Sig : IN std\_logic;

Clk : IN std\_logic;

Debout : OUT std\_logic

);

END COMPONENT;

--Inputs

signal Sig : std\_logic := '0';

signal Clk : std\_logic := '0';

--Outputs

signal Debout : std\_logic;

-- Clock period definitions

--constant Clk\_period : time := 10 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: Debounce PORT MAP (

Sig => Sig,

Clk => Clk,

Debout => Debout

);

-- Clock process definitions

-- Clk\_process :process

-- begin

-- Clk <= '0';

-- wait for Clk\_period/2;

-- Clk <= '1';

-- wait for Clk\_period/2;

-- end process;

--

-- Stimulus process

stim\_proc: process

begin

-- hold reset state for 100 ns.

Sig<='1';

Clk<='1';

wait for 5 ns;

Clk<='0';

wait for 5 ns;

Clk<='1';

wait for 5 ns;

Clk<='0';

wait for 5 ns;

Clk<='1';

wait for 5 ns;

Clk<='0';

wait for 5 ns;

Sig<='0';

Clk<='1';

wait for 5 ns;

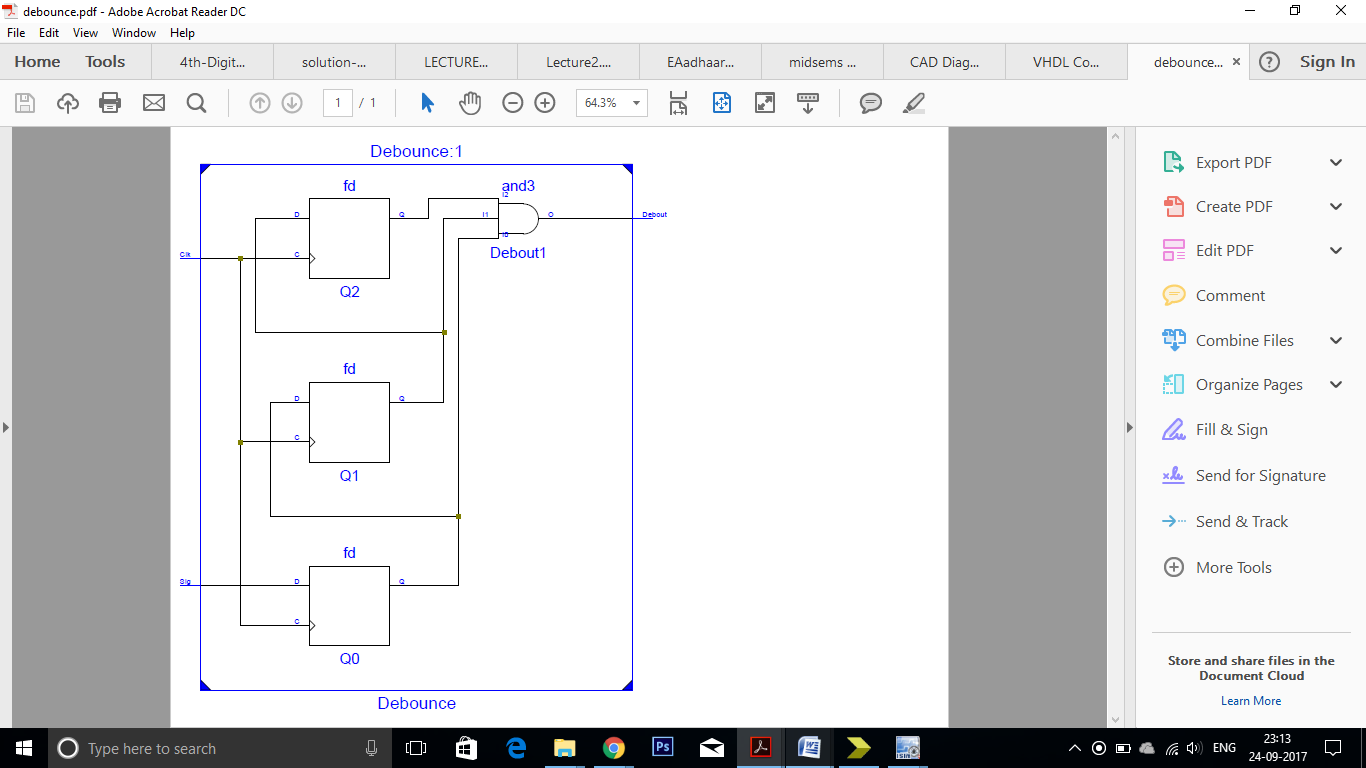
Clk<='0';

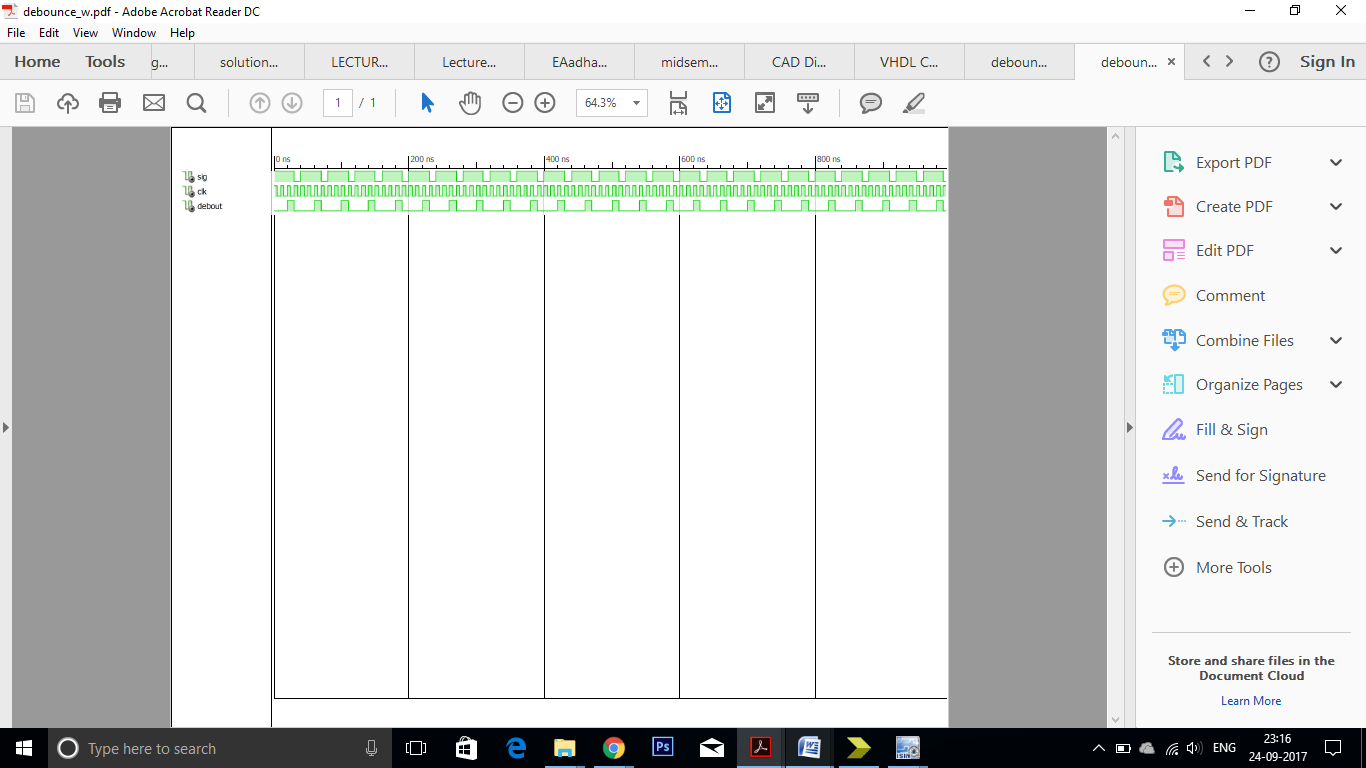
wait for 5 ns;

end process;

END;

OUTPUT**:**

****



Latch With Enable Signal

# CODE:

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity dlatch is

Port ( D : in STD\_LOGIC;

E : in STD\_LOGIC;

Q : inout STD\_LOGIC;

Qnot : inout STD\_LOGIC);

end dlatch;

architecture Behavioral of dlatch is

signal X1,X2 : STD\_LOGIC;

begin

X1<= E and (not D);

X2<= E and D;

Q<= X1 nor Qnot;

Qnot<= X2 nor Q;

end Behavioral;

# TESTBENCH

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY latch IS

END latch;

ARCHITECTURE behavior OF latch IS

-- Component Declaration for the Unit Under Test (UUT)

COMPONENT dlatch

PORT(

D : IN std\_logic;

E : IN std\_logic;

Q : INOUT std\_logic;

Qnot : INOUT std\_logic

);

END COMPONENT;

--Inputs

signal D : std\_logic := '0';

signal E : std\_logic := '0';

--BiDirs

signal Q : std\_logic;

signal Qnot : std\_logic;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: dlatch PORT MAP (

D => D,

E => E,

Q => Q,

Qnot => Qnot

);

-- Stimulus process

stim\_proc:process

begin

wait for 50 ns;

E <= not E;

wait for 50 ns;

end process;

input\_proc:process

begin

D <= '0';

wait for 50 ns;

D <= '1';

wait for 50 ns;

end process;

END;

# OUTPUTS

