Zayin Conde
Degital design and signal processing
Assignment 2
SDU

Question 2.1:
Write the VHDL code for a simple positive edge triggered D- flipflop. Use std_logic signals. Document the functionality of the circuit with relevant simulations. Describe the timing relations for the flipflop.

Add a Reset and Preset functionality to the flipflop. Document the new functionality with relevant simulations. Decide if Reset/Preset is asynchronous or synchronous and explain the difference in the VHDL model, as well as the timing relations.

Design code for synchronus timing

```
33
34  entity DFlipFlop is
35      Port ( D : in STD_LOGIC;
36             CLK : in STD_LOGIC;
37             Reset : in STD_LOGIC;
38             Preset : in STD_LOGIC;
39             Output : out STD_LOGIC);
40  end DFlipFlop;
41
42  architecture Behavioral of DFlipFlop is
43  begin
44      process(CLK) --need to create a "process" block which
45      --allows me to run sequential statements like "if", "for", "case" and "loop"
46     begin
47          -- can use rising_edge(CLK) or CLK'event and CLK ='1'.
48          --second option is a bit more manual and
49          --first option is apart of the IEEE.STD_LOGIC_1164.ALL toolbox
50          if rising_edge(CLK) then  --- this is synchronus cause it depends on the risin
51              if Reset = '0' then
52                  Output <= '0'; -- synchronous reset
53              elsif Preset = '1' then
54                  Output <= '1';
55              else
56                  Output <= D;
57              end if;
58          end if;
59     end process;
60  end Behavioral;
```

The test bench which triggers the cases (D, Preset and Reset) independently

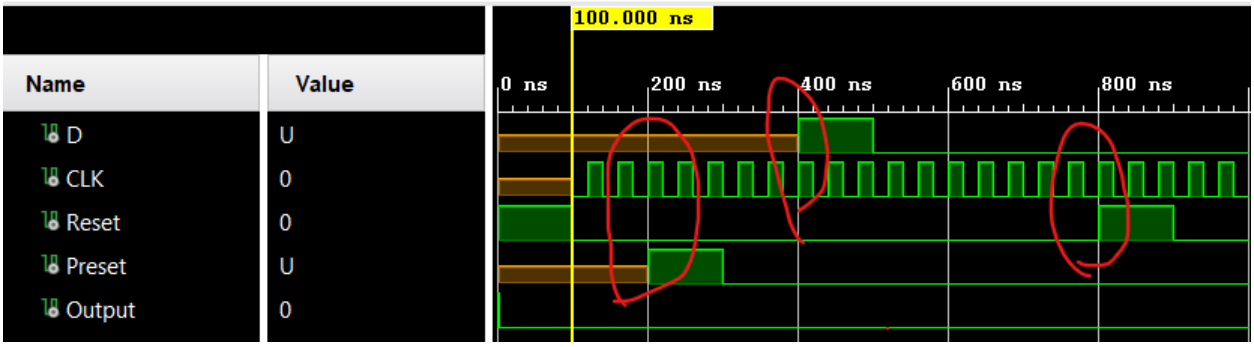
Zayin Conde
Degital design and signal processing
Assignment 2
SDU

```
49      --define signals
50      Signal D : STD_logic;
51      Signal CLK : STD_LOGIC;
52      Signal Reset : STD_LOGIC;
53      Signal Preset : STD_LOGIC;
54      Signal Output : STD_LOGIC;
55
56      begin
57          -- start uut
58          uut : DFlipFLop
59              Port Map (D => D, CLK => CLK, Reset => Reset, Preset => Preset, Output =>
60              -- start pårocess for loops and clk
61              process
62              begin
63                  wait for 100ns;
64                  while true loop
65                  CLK <= '0';
66                  wait for 20ns;
67                  CLK <= '1';
68                  wait for 20ns;
69                  end loop;
70              end process;

72              -- begin process for testing different case
73              process
74              begin
75                  -- reset test
76
77                  Reset <= '1';
78                  wait for 100ns;
79                  Reset <= '0';
80                  wait for 100ns;
81
82                  -- preset test
83                  Preset <= '1';
84                  wait for 100ns;
85                  Preset <= '0';
86                  wait for 100ns;
87
88                  -- D test
89                  D <= '1';
90                  wait for 100ns;
91                  D <= '0';
92                  wait for 100ns;
93                  wait for 200ns;
94              end process;
95      end Behavioral;
96
```

Zayin Conde
Degital design and signal processing
Assignment 2
SDU

When running the simulation I can see that in the synchronous setup that D, Preset and Reset trigger independently and on the rising_edge.



Design code should be changed for the asychronus setup so that variables will be triggered independently of the clock.

Sample code below for design file

```
-- below is asychronus d flip flop cause you can reset and preset whenever independant
--begin
--    process(CLK, RESET, PRESET)
--    begin
--        if RESET = '1' then
--            Q <= '0';  -- Asynchronous Reset
--        elsif PRESET = '1' then
--            Q <= '1';  -- Asynchronous Preset
--        elsif rising_edge(CLK) then
--            Q <= D;    -- Transfer Data Input to Output
--        end if;
--    end process;
--end Behavioral;
```

Qustion 2.2:
Write the VHDL code for an 8 bit up/down counter with Reset. Use std_logic types on all signals. Document the functionality of the circuit with relevant simulations and give a qualified guess of the counters expected maximum clock frequency.

Design code

Zayin Conde
Degital design and signal processing
Assignment 2
SDU

```
34  entity CounterEightBit is
35      Port ( CLK : in STD_LOGIC;
36             Reset : in STD_LOGIC;
37             Sig : in STD_LOGIC;
38             Counter : out STD_LOGIC_VECTOR (7 downto 0));
39  end CounterEightBit;
40
41  architecture Behavioral of CounterEightBit is
42      --uncomment numeric stdall for unsigned values, needed for inner signals
43      Signal Count : unsigned(7 downto 0) := (others => '0'); --inner counter
44      begin
45          process(CLK)
46          begin
47              if rising_edge(CLK)then
48                  if Reset = '1' then
49                      Count <= (others => '0'); -- reset counter
50                  elsif Sig = '1' then
51                      Count <= Count + 1; --increase 1
52                  end if;
53              end if;
54          end process;
55  -- set inner var to output counter
56  Counter <= STD_LOGIC_VECTOR(Count);
57  end Behavioral;
```

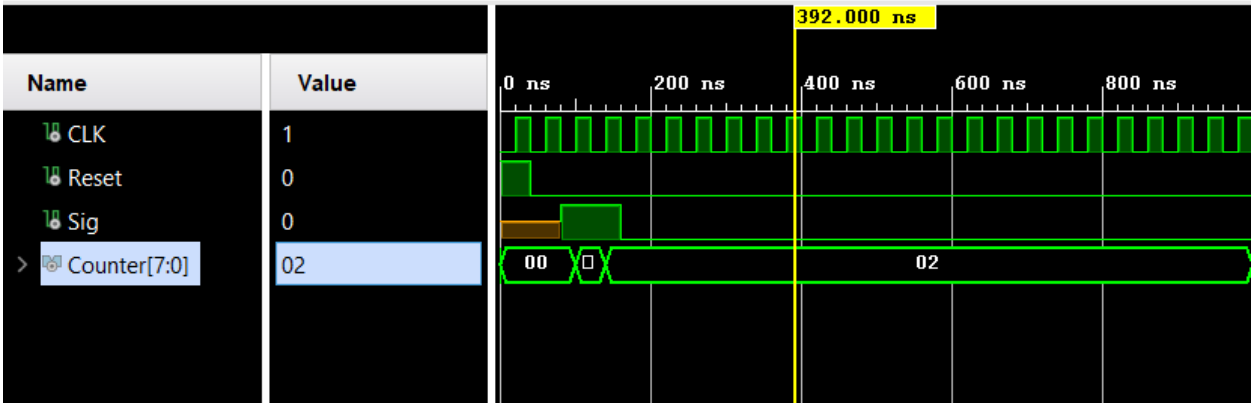
Test bench code

```
38      architecture Behavioral of tb_CounterEightBit is
39      -- import component from design file
40      component CounterEightBit
41          Port ( CLK : in STD_LOGIC;
42                 Reset : in STD_LOGIC;
43                 Sig : in STD_LOGIC;
44                 Counter : out STD_LOGIC_VECTOR (7 downto 0));
45      end component;
46      -- create signals
47      Signal CLK : STD_LOGIC;
48      Signal Reset : STD_LOGIC;
49      Signal Sig : STD_LOGIC;
50      Signal Counter : STD_LOGIC_VECTOR (7 downto 0);
51
52      begin
53          uut : CounterEightBit
54              Port Map ( CLK => CLK, Reset => Reset, Sig => Sig, Counter => Counter);
55              process
56              begin
57                  while true loop
58                      CLK <= '0';
59                      wait for 20ns;
60                      CLK <= '1';
61                      wait for 20ns;
62                  end loop;
63              end process;
```

Zayin Conde
Degital design and signal processing
Assignment 2
SDU

```
64
65          --testing process
66          process
67          begin
68              --test reset
69              Reset <= '1';
70              wait for 40ns;
71              Reset <= '0';
72              wait for 40ns;
73
74              --increment counter
75              Sig <= '1';
76              wait for 80ns;
77              Sig <= '0';
78              wait for 80ns;
79              wait;
80          end process;
81      end Behavioral;
82
```

Simulation signals:



Didn’t manage to finish the other 2 exercises before deadline.