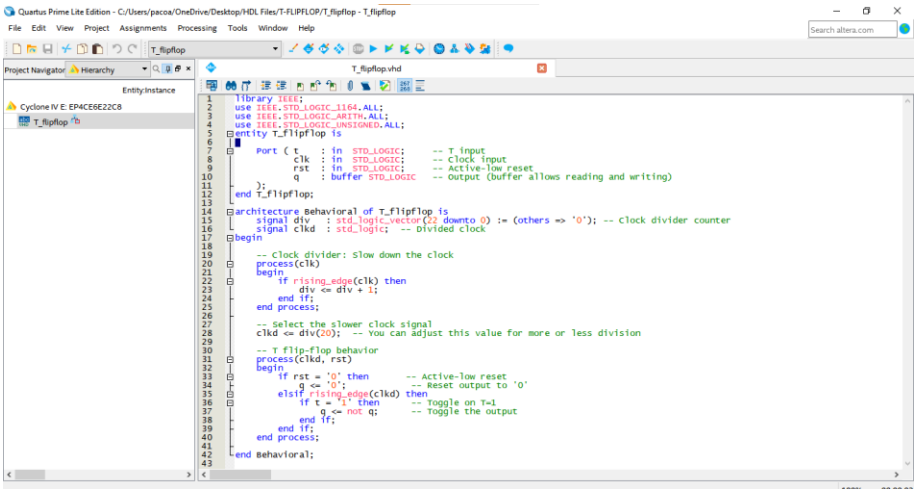
PACO, ARMIN R.
BSCpE – 3A

VHDL CODE FOR T FLIP FLOP:



CODE:

Clear	T	Clock	Qn+1	$\overline{Qn+1}$
1	0	0	0	$\overline{Qn}$
0	1	⎍	$\overline{Qn}$	Qn

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

entity T_flipflop is
    Port ( t    : in  STD_LOGIC;     -- T input
           clk  : in  STD_LOGIC;     -- Clock input
           rst  : in  STD_LOGIC;     -- Active-low reset
           q    : buffer STD_LOGIC   -- Output (buffer allows reading and writing)
    );
end T_flipflop;

architecture Behavioral of T_flipflop is
    signal div   : std_logic_vector(22 downto 0) := (others => '0'); -- Clock divider counter
    signal clkd  : std_logic;  -- Divided clock
begin

    -- Clock divider: Slow down the clock
    process(clk)
    begin
        if rising_edge(clk) then
            div <= div + 1;
        end if;
    end process;

    -- Select the slower clock signal
    clkd <= div(20);  -- You can adjust this value for more or less division

    -- T flip-flop behavior
    process(clkd, rst)
    begin
        if rst = '0' then       -- Active-low reset
```

```
        q <= '0';             -- Reset output to '0'
      elsif rising_edge(clkd) then
        if t = '1' then       -- Toggle on T=1
          q <= not q;       -- Toggle the output
        end if;
      end if;
    end process;

end Behavioral;
```