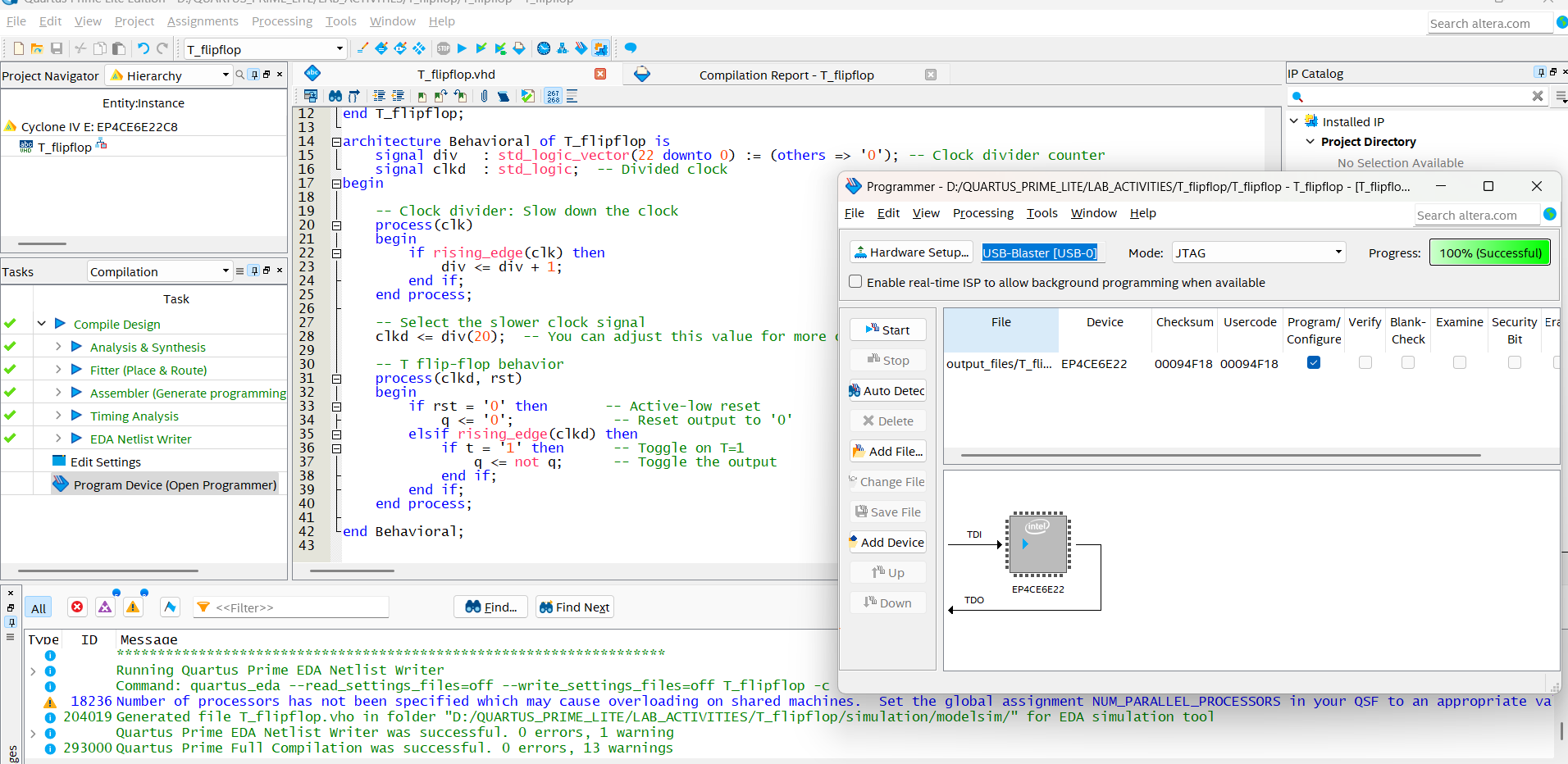
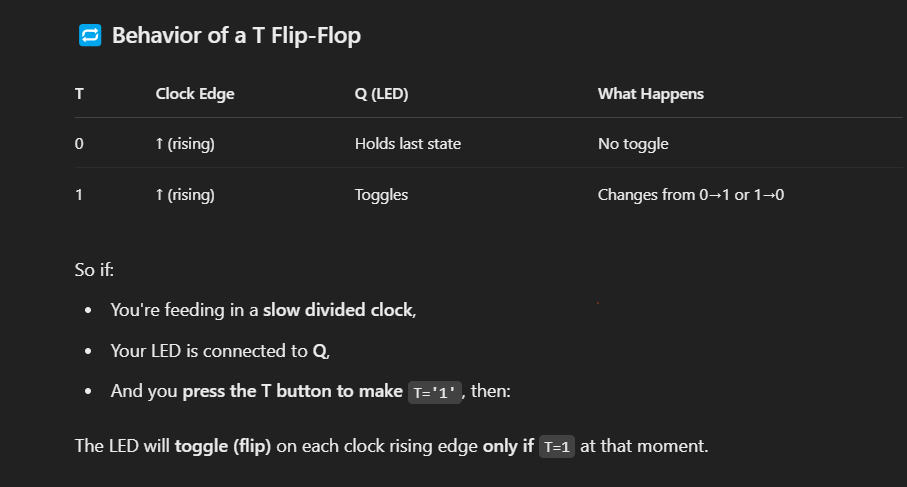
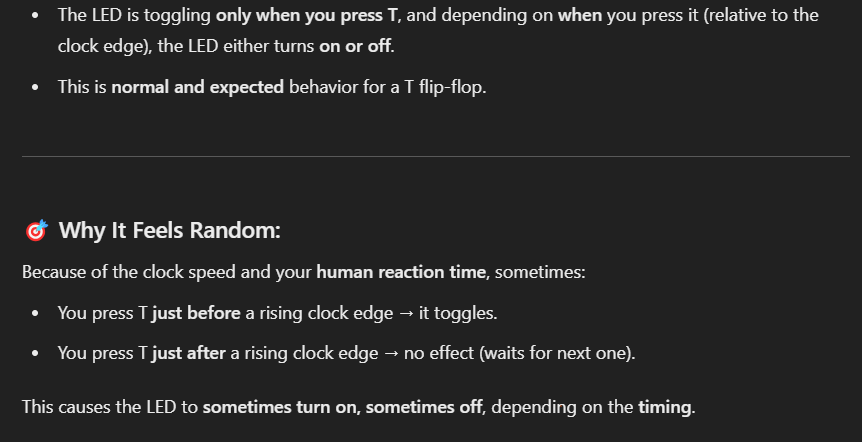
Name: Richard Raymond J. Canda

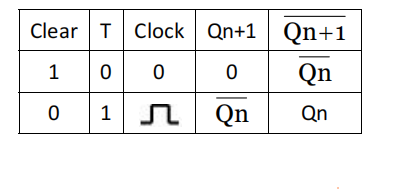
C.Y.S.: BSCpE - 3A







**VHDL CODE FOR T FLIP FLOP:**



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;