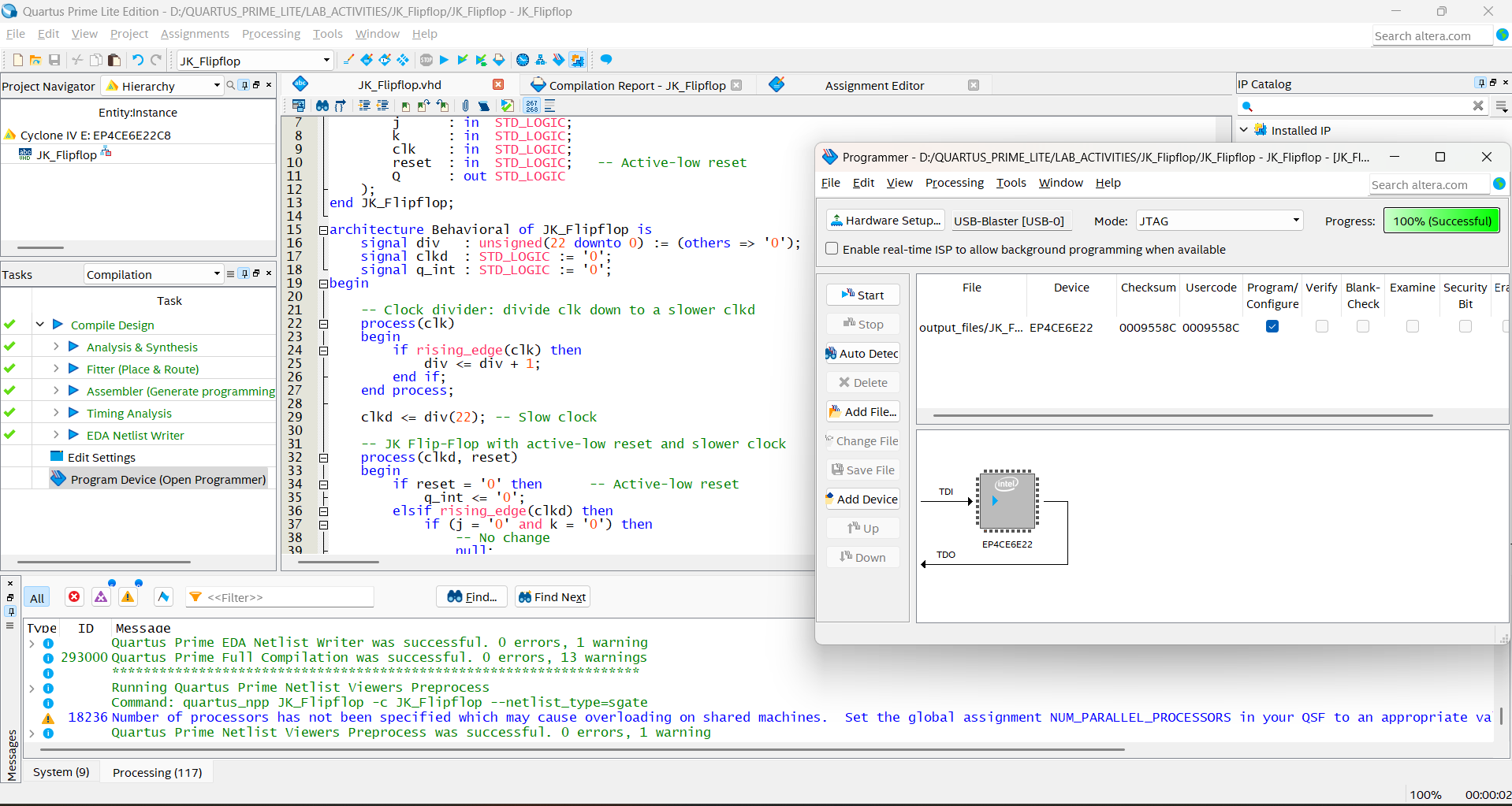
Name: Richard Raymond J. Canda

C.Y.S.: BSCpE - 3A



**VHDL CODE FOR JK FLIP FLOP WITH ASYNCHRONOUS RESET: (Master Slave JK Flip‐Flop)**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.NUMERIC\_STD.ALL;

entity JK\_Flipflop is

Port (

j : in STD\_LOGIC;

k : in STD\_LOGIC;

clk : in STD\_LOGIC;

reset : in STD\_LOGIC; -- Active-low reset

Q : out STD\_LOGIC

);

end JK\_Flipflop;

architecture Behavioral of JK\_Flipflop is

signal div : unsigned(22 downto 0) := (others => '0');

signal clkd : STD\_LOGIC := '0';

signal q\_int : STD\_LOGIC := '0';

begin

-- Clock divider: divide clk down to a slower clkd

process(clk)

begin

if rising\_edge(clk) then

div <= div + 1;

end if;

end process;

clkd <= div(22); -- Slow clock

-- JK Flip-Flop with active-low reset and slower clock

process(clkd, reset)

begin

if reset = '0' then -- Active-low reset

q\_int <= '0';

elsif rising\_edge(clkd) then

if (j = '0' and k = '0') then

-- No change

null;

elsif (j = '0' and k = '1') then

q\_int <= '0'; -- Reset

elsif (j = '1' and k = '0') then

q\_int <= '1'; -- Set

elsif (j = '1' and k = '1') then

q\_int <= not q\_int; -- Toggle

end if;

end if;

end process;

Q <= q\_int;

end Behavioral;