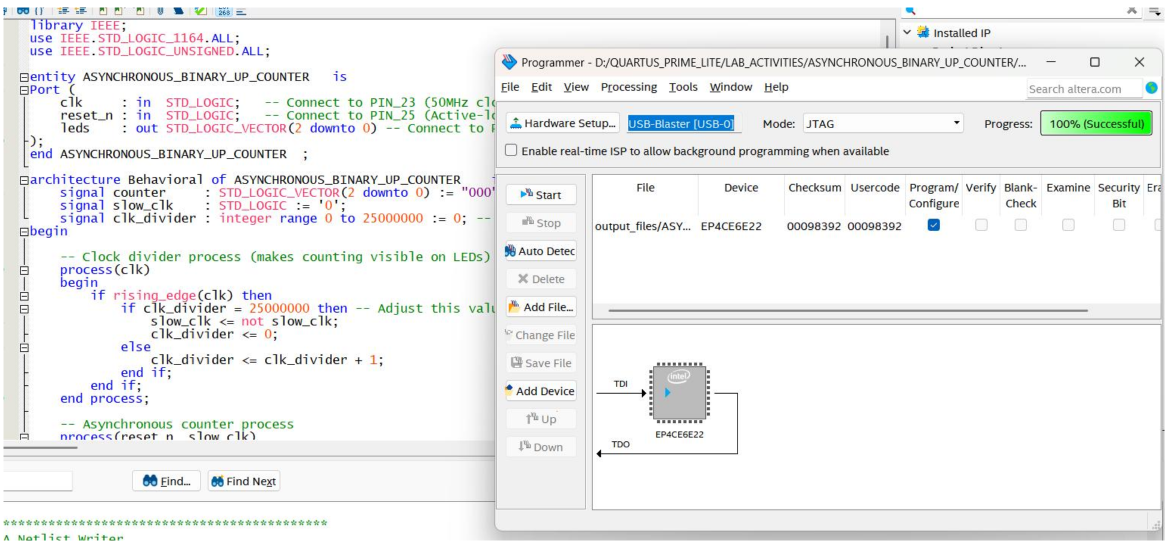
Kris M. Vallozo
BSCpE- 3A

ASYNCHRONOUS BINARY UP COUNTER

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

entity ASYNCHRONOUS_BINARY_UP_COUNTER        is
Port (
   clk     : in  STD_LOGIC;   -- Connect to PIN_23 (50MHz clock)
   reset_n : in  STD_LOGIC;   -- Connect to PIN_25 (Active-low reset button)
   leds    : out STD_LOGIC_VECTOR(2 downto 0) -- Connect to Pins 87,86,85
);
end ASYNCHRONOUS_BINARY_UP_COUNTER          ;

architecture Behavioral of ASYNCHRONOUS_BINARY_UP_COUNTER           is
   signal counter     : STD_LOGIC_VECTOR(2 downto 0) := "000";
   signal slow_clk    : STD_LOGIC := '0';
   signal clk_divider : integer range 0 to 25000000 := 0; -- For 1Hz @ 50MHz
begin

   -- Clock divider process (makes counting visible on LEDs)
   process(clk)
   begin
      if rising_edge(clk) then
         if clk_divider = 25000000 then -- Adjust this value for speed
            slow_clk <= not slow_clk;
            clk_divider <= 0;
         else
            clk_divider <= clk_divider + 1;
         end if;
      end if;
   end process;

   -- Asynchronous counter process
   process(reset_n, slow_clk)
   begin
      if reset_n = '0' then  -- Active-low reset
         counter <= "000";
      else
         -- First stage (LSB)
         if rising_edge(slow_clk) then
            counter(0) <= not counter(0);
         end if;

         -- Second stage
         if falling_edge(counter(0)) then
```

```vhdl
                counter(1) <= not counter(1);
            end if;

            -- Third stage (MSB)
            if falling_edge(counter(1)) then
                counter(2) <= not counter(2);
            end if;
        end if;
    end process;

    -- Active-high LED outputs (invert if your board needs active-low)
    leds <= counter;

end Behavioral;
```



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

entity ASYNCHRONOUS_BINARY_UP_COUNTER   is
Port (
      clk     : in  STD_LOGIC;    -- Connect to PIN_23 (50MHz clo
      reset_n : in  STD_LOGIC;    -- Connect to PIN_25 (Active-lo
      leds    : out STD_LOGIC_VECTOR(2 downto 0) -- Connect to P
);
end ASYNCHRONOUS_BINARY_UP_COUNTER  ;

architecture Behavioral of ASYNCHRONOUS_BINARY_UP_COUNTER
    signal counter     : STD_LOGIC_VECTOR(2 downto 0) := "000'
    signal slow_clk    : STD_LOGIC := '0';
    signal clk_divider : integer range 0 to 25000000 := 0; --
begin

    -- Clock divider process (makes counting visible on LEDs)
    process(clk)
    begin
        if rising_edge(clk) then
            if clk_divider = 25000000 then -- Adjust this valu
                slow_clk <= not slow_clk;
                clk_divider <= 0;
            else
                clk_divider <= clk_divider + 1;
            end if;
        end if;
    end process;

    -- Asynchronous counter process
    process(reset_n, slow_clk)
```

Programmer - D:/QUARTUS_PRIME_LITE/LAB_ACTIVITIES/ASYNCHRONOUS_BINARY_UP_COUNTER/...

Hardware Setup... USB-Blaster [USB-0]   Mode: JTAG   Progress: 100% (Successful)

☐ Enable real-time ISP to allow background programming when available

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Era
output_files/ASY...	EP4CE6E22	00098392	00098392	☑	☐	☐	☐	☐	☐