



PyroElectro.com - PyroEDU

Introduction To FPGA And CPLD – Lesson 9: Handheld LED POV

FORMULAS

The following formulas and information are meant to go with the online lesson found here:

http://www.pyroelectro.com/edu/fpga/handheld_pov/

LESSON9 VHDL CODE

In lesson 9 we built a hand-held LED POV when waved in front of a video/still camera created a neat effect of displaying a message. The code for building this POV was also surprisingly simple, requiring only a ROM and a counter. Here is the code for this lesson:

```
Lesson9.vhd

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity lesson9 is
  port(
    RESET: in std_logic;
    CLOCK: in std_logic;
    LED:   out std_logic_vector(7 downto 0)
  );
end lesson9;

architecture rtl of lesson9 is

  type num_array is array(0 to 31) of std_logic_vector(7 downto 0);

  constant led_message : num_array := (
0=> "00000000",
1=> "11111111",--*****
2=> "00001001",-- * *
3=> "00001111",-- ****
4=> "00000000",--
5=> "00001111",-- ****
6=> "11111000",--*****
7=> "00001111",-- ****
8=> "00000000",--
9=> "11111111",--*****
10=> "00000100",-- *
11=> "00000010",-- *
12=> "00000000",--
13=> "11111111",--*****
14=> "10000001",--* *
15=> "11111111",--*****
16=> "00000000",
17=> "00000000",
18=> "00000000",
19=> "00000000",
20=> "00000000",
21=> "00000000",
22=> "00000000",
23=> "00000000",
24=> "00000000",
25=> "00000000",
26=> "00000000",
27=> "00000000",
28=> "00000000",
29=> "00000000",
30=> "00000000",
```

```

31=> "00000000"
);

signal led_output : std_logic_vector(7 downto 0);
signal clock_div2 : std_logic;

begin

    CLOCK_DIVIDER_0 : process(RESET, CLOCK)
    begin
        if RESET = '1' then
            clock_div2 <= '0';
        elsif rising_edge(CLOCK) then
            clock_div2 <= not clock_div2;
        end if;
    end process CLOCK_DIVIDER_0;

    LETTER_COUNT_0 : process(RESET, clock_div2)
    variable letter_count : integer range 0 to 31;
    begin
        if RESET = '1' then
            letter_count := 0;
            led_output <= "00000000";
        elsif rising_edge(clock_div2) then
            if letter_count < 30 then
                letter_count := letter_count + 1;
            else
                letter_count := 0;
            end if;

            led_output <= led_message(letter_count);

        end if;
    end process LETTER_COUNT_0;

    LED <= led_output;

end rt;

```

The input frequency from the clock generation circuit was double what we needed, so the first process in the code was a very simple clock divider. Then afterward, we have the timer-counter process that counts over and over telling the ROM which 8 bits to output to the row of LEDs.

ADDITIONAL INFORMATION

If you have any questions about the formulas or information found in this document, please feel free to head on over to the forums and ask us some questions!

<http://www.pyroelectro.com/forums/viewforum.php?f=26>