Clock Divider

# Oded Yosef

# Code:

library ieee;

use ieee.std\_logic\_1164.all;

Entity ClockDivider is

port(CLOCK\_50 : in std\_logic;

SW : in std\_logic\_vector(17 downto 16);

LEDR : out std\_logic\_vector(0 downto 0);

GPIO : out std\_logic\_vector(0 downto 0));

end ClockDivider;

architecture behave of ClockDivider is

shared VARIABLE count : integer RANGE 0 to 50000000 :=0;

signal Hz1\_50 : std\_logic;

signal Hz1\_25 : std\_logic;

signal Hz10\_50 : std\_logic;

signal Hz10\_75 : std\_logic;

begin

process(CLOCK\_50)

begin

if CLOCK\_50'event and CLOCK\_50='1' then

if(SW="00") then

LEDR(0)<=Hz1\_50;

GPIO(0)<=Hz1\_50;

count:=count+1;

if (count=25000000) then

Hz1\_50<=not Hz1\_50;

count:=0;

end if;

elsif(SW="01") then

LEDR(0)<=Hz1\_25;

GPIO(0)<=Hz1\_25;

count:=count+1;

if (count=37500000) then

Hz1\_25<='1';

elsif(count=50000000) then

Hz1\_25<='0';

count:=0;

end if;

elsif(SW="10") then

LEDR(0)<=Hz10\_50;

GPIO(0)<=Hz10\_50;

count:=count+1;

if (count=5000000) then

Hz10\_50<=not Hz10\_50;

count:=0;

end if;

elsif(SW="11") then

LEDR(0)<=Hz10\_75;

GPIO(0)<=Hz10\_75;

count:=count+1;

if (count=1250000) then

Hz10\_75<='1';

elsif(count=5000000) then

Hz10\_75<='0';

count:=0;

end if;

end if;

end if;

end process;

end behave;

# RTL View

