**1.Summary of the Verilog code functionality**

By executing this verilog code,

* FPGA Processor is programmed by 12 MHz Clock signal.
* Blue Led connected to the port pin 40 is made to blink(supplied Active High and Low using Pulse width modulated signal).
* Other Red and Green Led are connected to Active LOW signal which are made not to blink.
* Current 4mA is supplied to all LEDs through RGB LED Driver.

Table 1 explains the functionality of each statement in verilog code.

**Table 1:Description of Verilog Code**

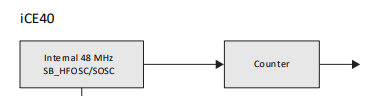
|  |  |
| --- | --- |
| **Verilog Code** | **Explanation** |
| module top (output wire led\_red , output wire led\_blue , output wire led\_green , input wire hw\_clk, output wire testwire ); | Module Declaration with output and input ports |
| wire int\_osc ;  reg [27:0] frequency\_counter\_i; | Declaration of intermediate signals |
| assign testwire = frequency\_counter\_i[5]; | 6th bit of the register output frequency\_counter\_i[5] is connected to testwire |
| SB\_HFOSC #(  .CLKHF\_DIV ("0b10"))  u\_SB\_HFOSC ( .CLKHFPU(1'b1),  .CLKHFEN(1'b1),  .CLKHF(int\_osc)); | High Frequency Oscillator with output divider    CLKHF\_DIV =0b10 ;Divider setting for 12MHZ clock signal  CLKHFPU=1'b1; Power up signal Active HIGH  CLKHFEN=1'b1; Clock Enable signal Active HIGH  CLKHF=int\_osc; Internal Oscillator Output is connected to the wire int\_osc |
| always @(posedge int\_osc) begin  frequency\_counter\_i <= frequency\_counter\_i + 1'b1;  end | During positive edge of the clock signal int\_osc, frequency counter value is incremented by one. |
| SB\_RGBA\_DRV RGB\_DRIVER  (.RGBLEDEN(1'b1 ),  .RGB0PWM (1'b0),  .RGB1PWM (1'b0),  .RGB2PWM (1'b1),  .CURREN (1'b1 ),  .RGB0 (led\_red ),  .RGB1 (led\_green ),  .RGB2 (led\_blue ));  defparam RGB\_DRIVER.RGB0\_CURRENT = "0b000001";  defparam RGB\_DRIVER.RGB1\_CURRENT = "0b000001";  defparam RGB\_DRIVER.RGB2\_CURRENT = "0b000001"; | RGB LED Driver primitive Instantiation    RGBLEDEN=1'b1 ;Enable control for RGB LED is Active HIGH  RGB0PWM =1'b0 ;PWM signal for RGB\_PAD0 is Active LOW for Red LED.  RGB1PWM =1'b0 ;PWM signal for RGB\_PAD1 is Active LOW for Green LED.  RGB2PWM =1'b1 ;PWM signal for RGB\_PAD2 is Active HIGH for Blue LED.  CURREN =1'b1 ; Power up signal Active HIGH  RGB0=led\_red ; RGB LED Driver output RGB0 is connected to red led.  RGB1=led\_green ; RGB LED Driver output RGB1 is connected to green led.  RGB0=led\_blue ; RGB LED Driver output RGB2 is connected to blue led.  0b000001 means 4mA current is supplied to red led.  0b000001 means 4mA current is supplied to green led.  0b000001 means 4mA current is supplied to blue led. |
| endmodule | Ending the Module. |

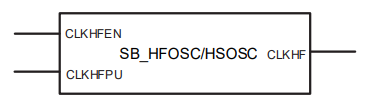
**2.Description of internal logic and oscillator**

**Two on-chip oscillators:** An ultra-low power 10 kHz oscillator is provided for Always-On applications and background polling that allow higher power processors to remain in power-down or sleep mode, conserving overall power consumption. A low power 48 MHz oscillator with output divider is provided for sensor management and pre-processing functions. These oscillators are intended for general clocking of internal logic and state machines.

* SB\_LFOSC – Low Frequency Oscillator
* SB\_HFOSC – High Frequency Oscillator with output divider

SB\_LFOSC runs at 10 kHz and SB\_HFOSC runs at maximum 48 MHz with output divider by 1, 2, 4 or 8. SB\_LFOSC and SB\_HFOSC provide internal clock sources to user designs. These clocks can directly route to the global clock network or to local fabric.



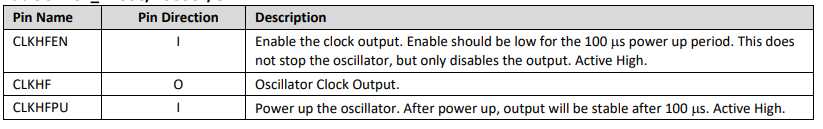


**Figure 1. Block Diagram of Internal Oscillator**

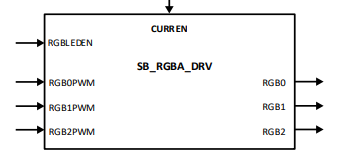
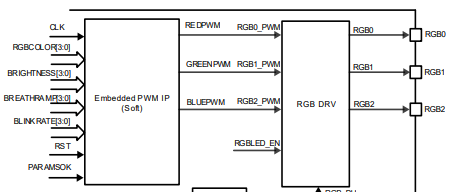
SB\_HFOSC provide internal clock sources to user designs. These clocks can directly route to the global clock network or to local fabric.

Parameter CLKHF\_DIV = "0b00" (default), "0b01", "0b10", "0b11" (Clock divider selection. 0b00 = 48 MHz, 0b01 = 24 MHz, 0b10 = 12 MHz, 0b11 = 6 MHz).

**Table 2.Description of Each Pin in Internal Oscillator**



**3.Functionality of the RGB LED driver and its relationship to the outputs**

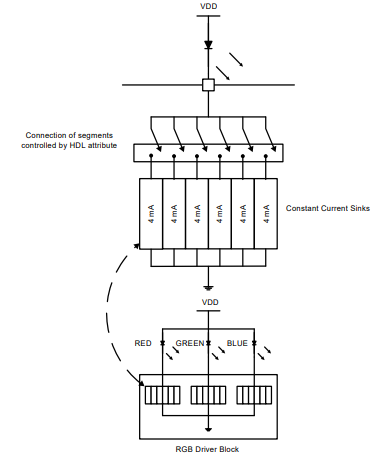


**Figure 2. Block Diagram of RGB LED driver**

The RGB LED driver block provides an open-drain driver for the LED DIODE with constant current from 4 mA to 24 mA in 4 mA stepas shown in figure 4. full current mode or from 2 mA to 12 mA in 2 mA step in half current mode with up to +/-10% accuracy. Each of the steps is controlled by an HDL attribute.



**Figure 3. PWM Signal for LED**



**Figure 4. Supply Current to RGB LED**

**4.Document the pin mapping and explain the significance of each connection in context with the Verilog code and board hardware.**

|  |  |
| --- | --- |
| ***PIN MAPPING* FROM THE *PCF* FILE** | **IN CONTEXT WITH THE *VERILOG CODE* AND *BOARD HARDWARE.*** |
| set\_io led\_red 39 | RGB LED Driver output RGB0 is connected to GPIO pin number 39 where red led is connected. |
| set\_io led\_blue 40 | RGB LED Driver output RGB1 is connected to GPIO pin number 40 where blue led is connected. |
| set\_io led\_green 41 | RGB LED Driver output RGB2 is connected to GPIO pin number 41 where green led is connected. |
| set\_io hw\_clk 20 | Hardware Oscillator is connected to GPIO pin 20 |
| set\_io testwire 17 | GPIO pin 17 is named as testwire where 6th bit of the register output frequency\_counter\_i[5] is connected. |

**5. Integration steps and observations while working with the FPGA Mini board**

*a.* Connect the board to the computer using USB-C and ensuring FTDI connection.

b. Commands for building and flashing the Verilog code:

* + Run 'make clean' to clear any previous builds
  + Run 'make build' to compile the design
  + Run 'sudo make flash' to program the FPGA board

1. Submit the final document along with the working Verilog and PCF files.

*Verilog File*

module top (output wire led\_red , output wire led\_blue , output wire led\_green , input wire hw\_clk, output wire testwire );

wire int\_osc ;

reg [27:0] frequency\_counter\_i;

assign testwire = frequency\_counter\_i[5];

//-----------------------------------------------------------------------------------------------------------------------------------

// Internal Oscillator

//----------------------------------------------------------------------------------------------------------------------------------

SB\_HFOSC #(.CLKHF\_DIV ("0b10")) u\_SB\_HFOSC ( .CLKHFPU(1'b1), .CLKHFEN(1'b1), .CLKHF(int\_osc));

//-----------------------------------------------------------------------------------------------------------------------------------

// Counter

//------------------------------------------------------------------------------------------------------------------------------------

always @(posedge int\_osc) begin

frequency\_counter\_i <= frequency\_counter\_i + 1'b1;

end

//------------------------------------------------------------------------------------------------------------------------------------

// Instantiate RGB primitive

//-------------------------------------------------------------------------------------------------------------------------------------

SB\_RGBA\_DRV RGB\_DRIVER (.RGBLEDEN(1'b1 ), .RGB0PWM (1'b0),.RGB1PWM (1'b0), .RGB2PWM (1'b1), .CURREN (1'b1 ), .RGB0 (led\_red ), .RGB1 (led\_green ),.RGB2 (led\_blue ));

defparam RGB\_DRIVER.RGB0\_CURRENT = "0b000001";

defparam RGB\_DRIVER.RGB1\_CURRENT = "0b000001";

defparam RGB\_DRIVER.RGB2\_CURRENT = "0b000001";

endmodule

*PCF file*

set\_io led\_red 39

set\_io led\_blue 40

set\_io led\_green 41

set\_io hw\_clk 20

set\_io testwire 17