**VSDSquadron FPGA Mini Board - LED Control Implementation**

**1. Understanding the Verilog Code**

**Module Overview**

The top module is responsible for controlling an RGB LED using an internal oscillator. It includes:

* **LED Outputs:** led\_red, led\_blue, led\_green (for RGB control).
* **Clock Input:** hw\_clk (hardware oscillator, though unused in the current code).
* **Test Output:** testwire (debug signal driven by the frequency counter).

**Internal Components**

1. **Internal Oscillator (SB\_HFOSC)**
   * Configured with CLKHF\_DIV = "0b10" (divides clock frequency).
   * Generates a clock signal (int\_osc) that drives the counter.
2. **Frequency Counter**
   * frequency\_counter\_i (28-bit register) increments on every rising edge of int\_osc.
   * testwire is assigned frequency\_counter\_i[5], generating a test signal.
3. **RGB LED Driver (SB\_RGBA\_DRV)**
   * Controls the hardware LED outputs (led\_red, led\_green, led\_blue).
   * **Current settings:**
     + RGB0\_CURRENT = "0b000001" (Red)
     + RGB1\_CURRENT = "0b000001" (Green)
     + RGB2\_CURRENT = "0b000001" (Blue)
   * **LED Behavior:**
     + Red and Green LEDs are off (RGB0PWM = 0, RGB1PWM = 0).
     + Blue LED is on (RGB2PWM = 1).

**2. Creating the PCF File (Pin Mapping)**

The **Pin Constraint File (PCF)** assigns FPGA pins to module signals.

|  |  |  |
| --- | --- | --- |
| **Signal** | **Pin** | **Function** |
| led\_red | **39** | Controls Red LED |
| led\_blue | **40** | Controls blue LED |
| led\_green | **41** | Controls green LED |
| hw\_clk | **20** | |  | | --- | | External clock input |  |  | | --- | |  | |
| testwire | **17** | Debug/test signal |

**Verifying Pin Assignments**

* The **PCF file** matches the **VSDSquadron FPGA Mini board** specifications.
* These mappings ensure the LEDs and test signal are correctly routed to the board's pins.

**3. Integrating with the VSDSquadron FPGA Mini Board**

**Board Setup**

1. Connect the FPGA board to a computer via **USB-C**.
2. Ensure FTDI drivers are installed (for serial communication).

**Building & Flashing Steps**

Run the following commands in the project directory:

make clean # Clears previous builds

make build # Compiles the Verilog design

sudo make flash # Programs the FPGA

* If successful, the **blue LED should turn on**, confirming correct operation.

**4. Final Summary**

**Key Observations**

* **Verilog Code:** Generates an internal clock and drives an RGB LED.
* **PCF File:** Correctly assigns FPGA pins to module signals.
* **Board Programming:** The **blue LED turns on**, validating the design.

**Challenges & Solutions**

|  |  |
| --- | --- |
| **Challenge** | **Solution** |
| FPGA not detected during flashing | Ensured correct USB connection and FTDI setup |