VSDSquadron FPGA Mini Board Project

**Step 1: What the Verilog Code Does**

**Overview**

The Verilog file [top.v](https://github.com/thesourcerer8/VSDSquadron_FM/blob/main/led_blue/top.v) is a simple program that makes an RGB LED light up in different colors. It does this by using a clock signal inside the FPGA to control when the light changes.

**What’s Inside?**

* **Inputs:**
  + hw\_clk: This is the hardware clock input. It comes from an onboard oscillator (SB\_HFOSC) that keeps everything running at a steady pace, like a heartbeat.
* **Outputs:**
  + led\_red, led\_blue, led\_green: These are the three-color channels of the RGB LED. The FPGA controls these signals to change the LED’s color.
  + testwire: This is a test signal, usually used for debugging or checking if the FPGA is running properly.

**How It Works**

* **Oscillator (SB\_HFOSC)**: Generates a clock signal that controls the timing of the LED blinks.
* **Counter Logic**: Uses the clock signal to determine when to switch LED colors.
* **RGB LED Driver**: Manages the brightness and color of the LED by turning the red, blue, and green signals on or off.

**Summary**

Basically, the FPGA takes a clock signal, processes it, and then makes the RGB LED blink in a specific pattern.

**Step 2: Understanding the PCF File**

The [VSDSquadronFM.pcf](https://github.com/thesourcerer8/VSDSquadron_FM/blob/main/led_blue/VSDSquadronFM.pcf) file tells the FPGA which physical pins to use.

**Pin Assignments**

|  |  |  |
| --- | --- | --- |
| **Signal** | **Pin** | **Description** |
| led\_red | 39 | Controls the red part of the RGB LED. |
| led\_blue | 40 | Controls the blue part of the RGB LED. |
| led\_green | 41 | Controls the green part of the RGB LED. |
| hw\_clk | 20 | The main clock input for timing. |
| testwire | 17 | A test signal used for debugging. |

These numbers match the physical pins on the FPGA board where each component is connected.

**Step 3: Hooking Up the FPGA Mini Board**

**What’s on the Board?**

* **FPGA Chip**: The main brain of the board.
* **USB-to-SPI Communication**: Helps the computer talk to the FPGA.
* **32 GPIO Pins**: These are extra pins you can use to connect stuff.
* **4MB Flash Storage**: Stores the FPGA’s program.
* **RGB LED**: Blinks different colors!

**Setting It Up**

1. Plug the board into your computer with a USB-C cable.
2. Open a terminal and type:

lsusb

If everything is working, you should see Future Technology Devices International in the list.

**Uploading the Code**

1. **Clear old files:**

make clean

1. **Compile the program:**

make build

1. **Send it to the FPGA:**

sudo make flash

1. Look at the RGB LED—it should be blinking!

**Step 4: Wrapping Up**

**What We Learned**

* The Verilog code controls the RGB LED with a clock.
* The PCF file tells the FPGA which pins to use.
* We successfully programmed the FPGA and made the LED blink.

**Troubleshooting**

|  |  |
| --- | --- |
| Problem | Solution |
| Board not recognized | Unplug and plug it back in, then run lsusb |
| Flashing failed | Try sudo make flash again |

**What’s Next?**

* Try changing the LED blink pattern.
* Add a button to change the LED color when pressed.