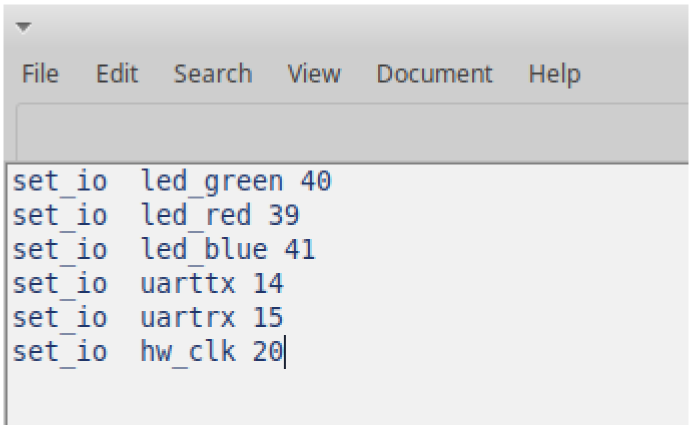
Task 2

Step 1:

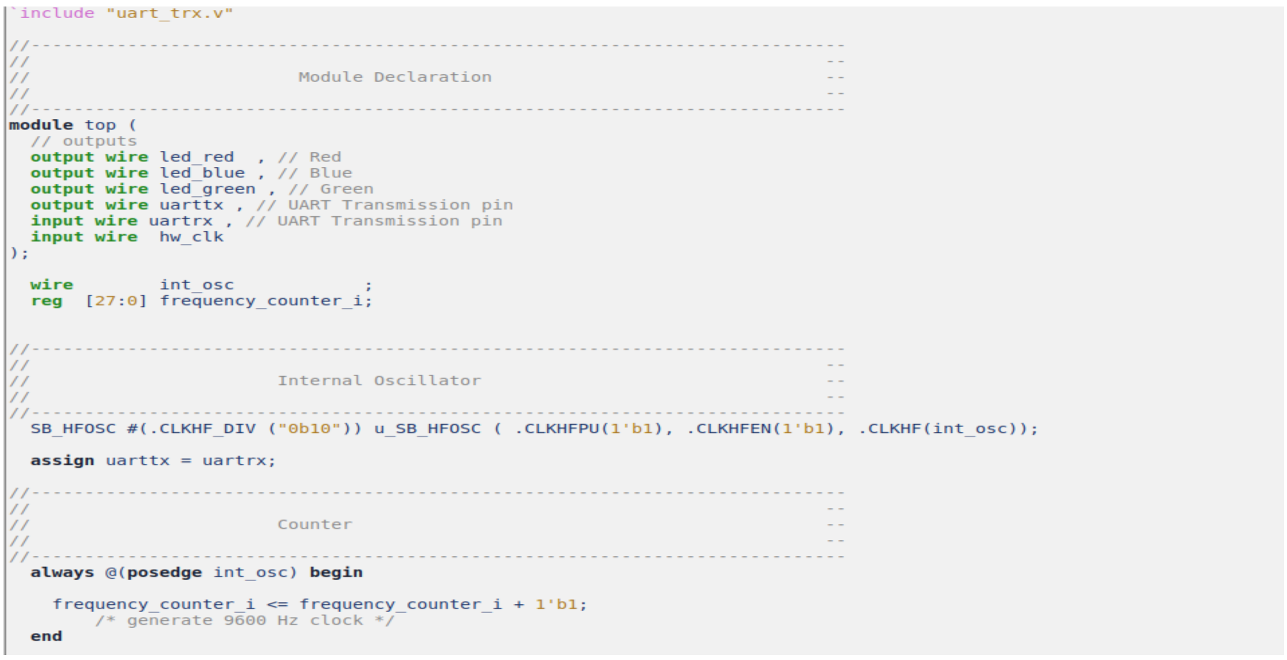
analysing the .v logic

UART (Universal Asynchronous Receiver/Transmitter) loopback is a testing and debugging technique used to verify the functionality of a UART module without needing an external device. In loopback mode, the transmitted (TX) data is internally connected to the received (RX) data, allowing data sent by the UART to be received by itself.

A UART loopback mechanism is a test or diagnostic mode where data, which is transmitted to the TX (transmit) pin is directly routed back to the RX (receive) pin of the same module. This allows the system to verify that the TX and RX lines function correctly without the need of an external device.



```
set_io  led_green 40
set_io  led_red 39
set_io  led_blue 41
set_io  uarttx 14
set_io  uartrx 15
set_io  hw_clk 20
```

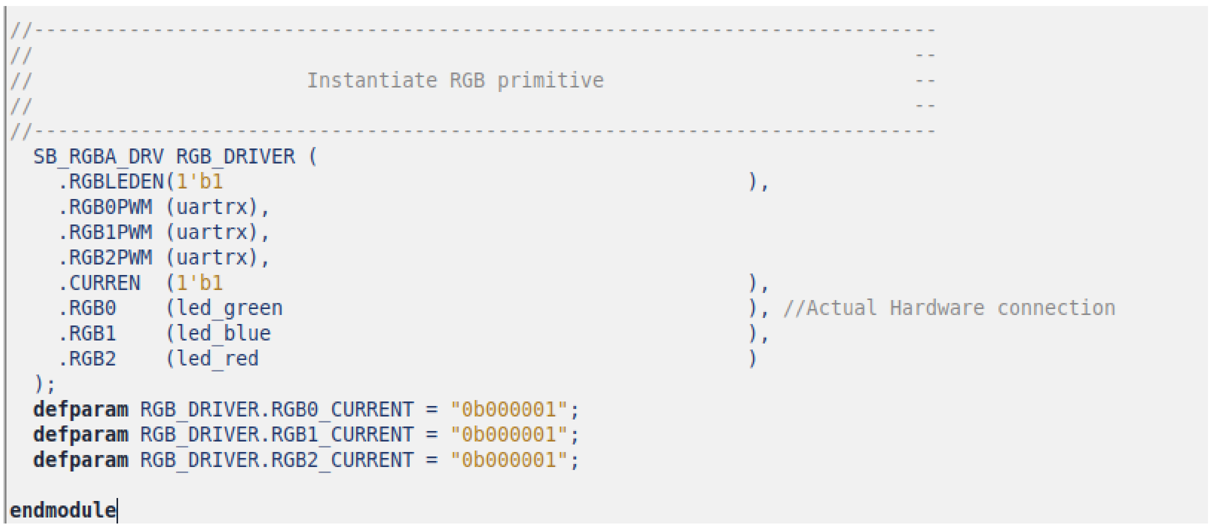


```
`include "uart_trx.v"

//----------------------------------------------------------------------------
//                                                                          --
//                         Module Declaration                               --
//                                                                          --
//----------------------------------------------------------------------------
module top (
  // outputs
  output wire led_red  , // Red
  output wire led_blue , // Blue
  output wire led_green , // Green
  output wire uarttx , // UART Transmission pin
  input wire uartrx , // UART Transmission pin
  input wire  hw_clk
);

  wire        int_osc            ;
  reg  [27:0] frequency_counter_i;


//----------------------------------------------------------------------------
//                                                                          --
//                       Internal Oscillator                                --
//                                                                          --
//----------------------------------------------------------------------------
  SB_HFOSC #(.CLKHF_DIV ("0b10")) u_SB_HFOSC ( .CLKHFPU(1'b1), .CLKHFEN(1'b1), .CLKHF(int_osc));

  assign uarttx = uartrx;

//----------------------------------------------------------------------------
//                                                                          --
//                       Counter                                            --
//                                                                          --
//----------------------------------------------------------------------------
  always @(posedge int_osc) begin

    frequency_counter_i <= frequency_counter_i + 1'b1;
        /* generate 9600 Hz clock */
  end
```

This is the top.verilog file ,

- 3 RGB LED outputs (led red, led blue, led green)

- UART transmit/receive pins (UARTTX, UARTRX) {UARTTX is output, UARTRX is input}
- Clock input (hw_clk)
- The internal oscillator implements a high frequency oscillation
- The 28-bit frequency counter is used for timing generation
- Also, the transmitted signal is equal to the received signal
- The UART loopback gives direct connection btw the tx and rx pins



```
//----------------------------------------------------------------------------
//                                                                          --
//                       Instantiate RGB primitive                          --
//                                                                          --
//----------------------------------------------------------------------------
  SB_RGBA_DRV RGB_DRIVER (
    .RGBLEDEN(1'b1                                            ),
    .RGB0PWM (uartrx),
    .RGB1PWM (uartrx),
    .RGB2PWM (uartrx),
    .CURREN  (1'b1                                            ),
    .RGB0    (led_green                                       ), //Actual Hardware connection
    .RGB1    (led_blue                                        ),
    .RGB2    (led_red                                         )
  );
  defparam RGB_DRIVER.RGB0_CURRENT = "0b000001";
  defparam RGB_DRIVER.RGB1_CURRENT = "0b000001";
  defparam RGB_DRIVER.RGB2_CURRENT = "0b000001";

endmodule
```

The RGB LED Driver controls all 3 RGB channels and maps the UART input to the LED intensity.

The basic understanding of operation,

The received UART data appears on the UARTRX pin and this data is looped back through UARTTX

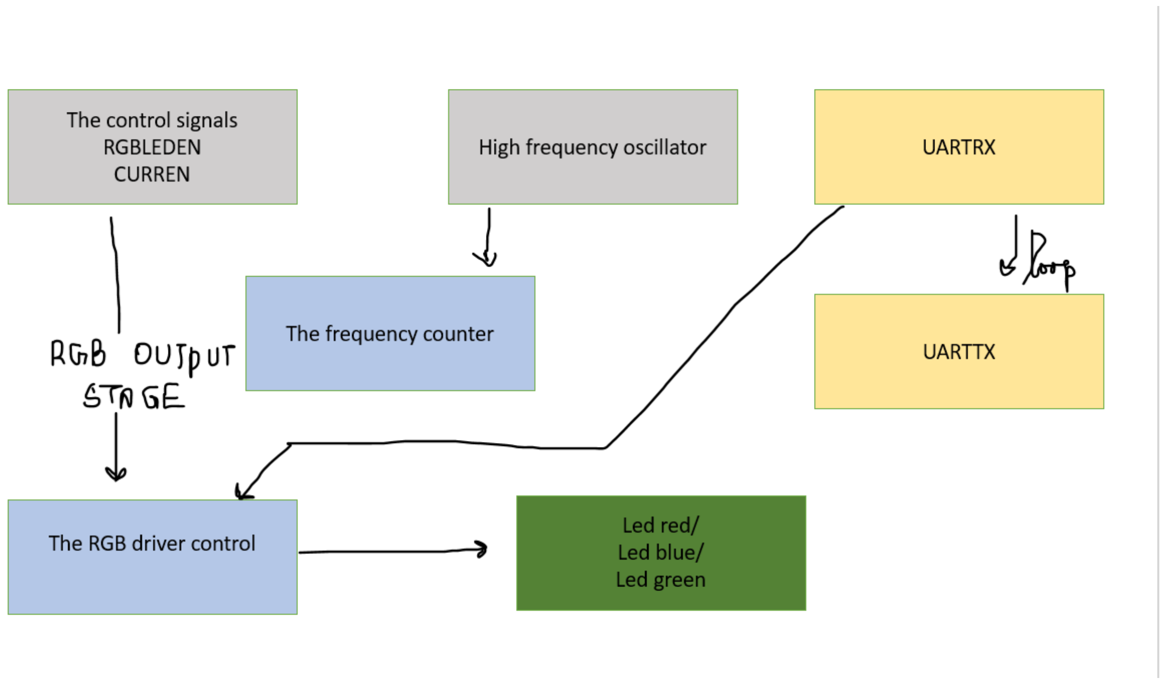
The same piece of data drives the RGB channels simultaneously.

The RGB driver converts UART signal to PWM output

Also, the current limiting is set to 0b000001 for each channel

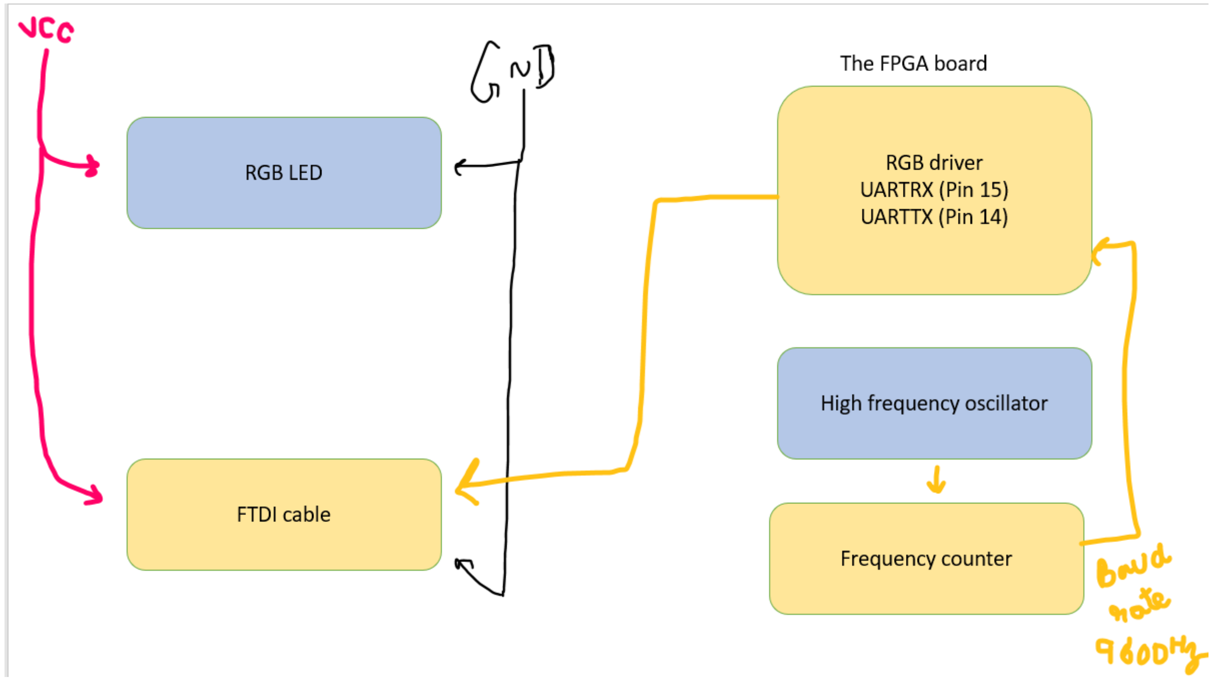
In FPGA design, toggling inputs are signals that switch between **high (1) and low (0)** states, typically driven by a **clock signal** or an external stimulus. (RGBLEDEN, CURREN)

STEP 2

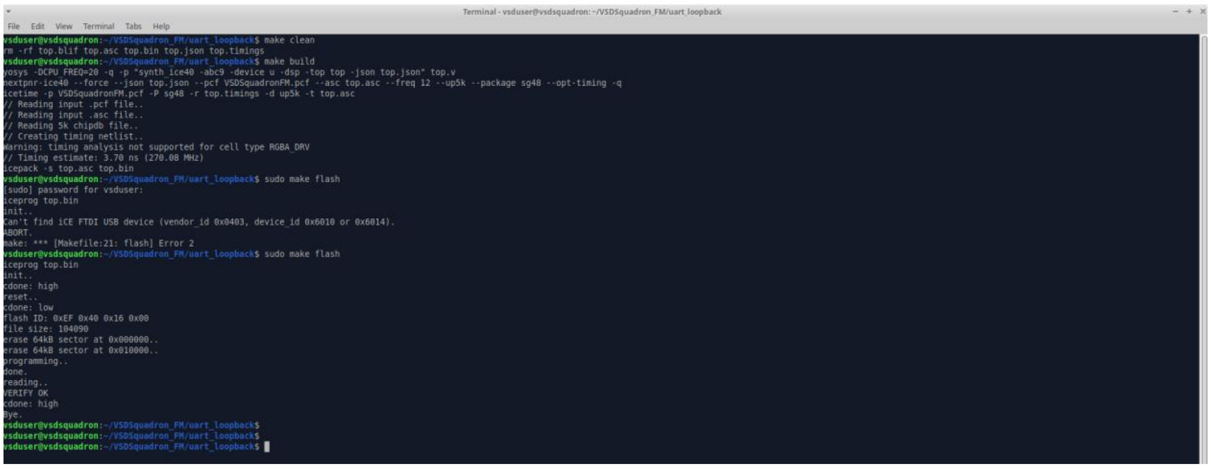


Block diagram: the architecture

Circuit diagram



Step 3

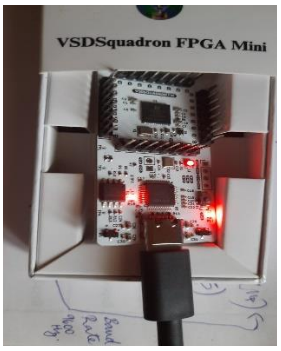


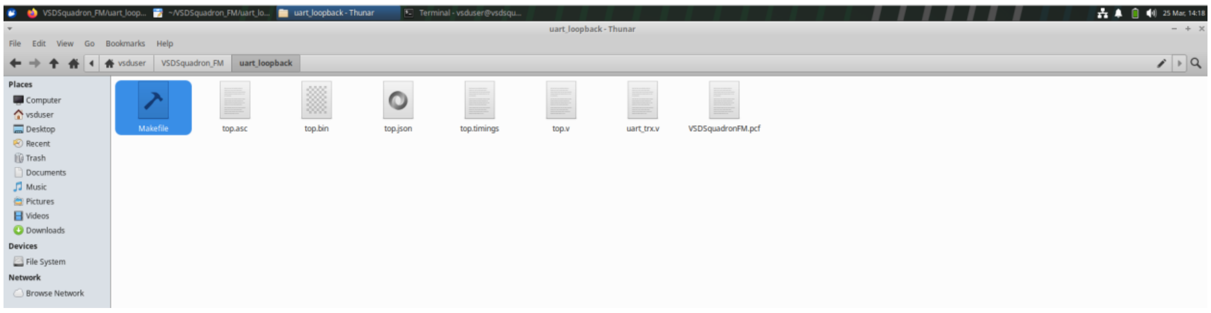
```
vsduser@vsdsquadron:~/VSDSquadron_FM/uart_loopback$ make clean
rm -rf top.blif top.asc top.bin top.json top.timings
vsduser@vsdsquadron:~/VSDSquadron_FM/uart_loopback$ make build
yosys -DCPU_FREQ=20 -q -p "synth_ice40 -abc9 -device u -dsp -top top -json top.json" top.v
nextpnr-ice40 --force --json top.json --pcf VSDSquadronFM.pcf --asc top.asc --freq 12 --up5k --package sg48 --opt-timing -q
icetime -p VSDSquadronFM.pcf -P sg48 -r top.timings -d up5k -t top.asc
// Reading input .pcf file..
// Reading input .asc file..
// Reading 5k chipdb file..
// Creating timing netlist..
Warning: timing analysis not supported for cell type RGBA_DRV
// Timing estimate: 3.70 ns (270.08 MHz)
icepack -s top.asc top.bin
vsduser@vsdsquadron:~/VSDSquadron_FM/uart_loopback$ sudo make flash
[sudo] password for vsduser:
iceprog top.bin
init..
Can't find iCE FTDI USB device (vendor_id 0x0403, device_id 0x6010 or 0x6014).
ABORT.
make: *** [Makefile:21: flash] Error 2
vsduser@vsdsquadron:~/VSDSquadron_FM/uart_loopback$ sudo make flash
iceprog top.bin
init..
cdone: high
reset..
cdone: low
flash ID: 0xEF 0x40 0x16 0x00
file size: 104090
erase 64kB sector at 0x000000..
erase 64kB sector at 0x010000..
programming..
done.
reading..
VERIFY OK
cdone: high
Bye.
vsduser@vsdsquadron:~/VSDSquadron_FM/uart_loopback$
vsduser@vsdsquadron:~/VSDSquadron_FM/uart_loopback$
vsduser@vsdsquadron:~/VSDSquadron_FM/uart_loopback$
```

Terminal - vsduser@vsdsquadron: ~/VSDSquadron_FM/uart_loopback
File Edit View Terminal Tabs Help

These are the commands I used for running the uart loopback

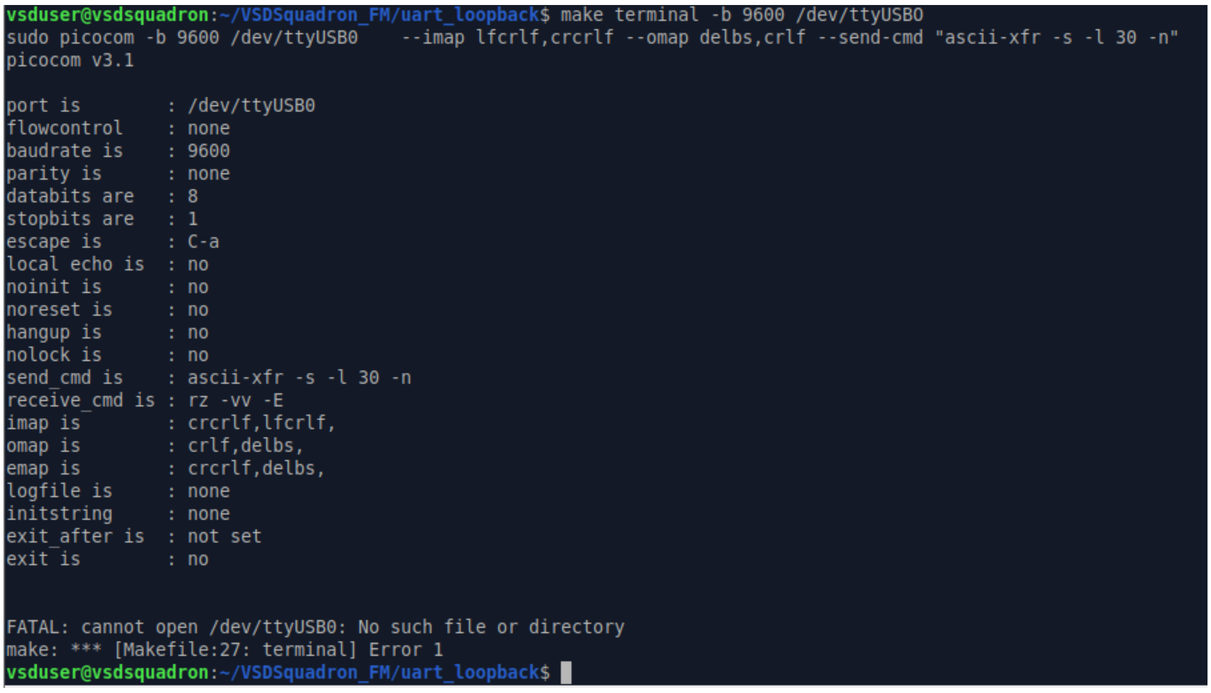
Make clean, make build, sudo make flash.

The output was, 



These files were formed after the flash

Step 4:



```
vsduser@vsdsquadron:~/VSDSquadron_FM/uart_loopback$ make terminal -b 9600 /dev/ttyUSB0
sudo picocom -b 9600 /dev/ttyUSB0    --imap lfcrlf,crcrlf --omap delbs,crlf --send-cmd "ascii-xfr -s -l 30 -n"
picocom v3.1

port is        : /dev/ttyUSB0
flowcontrol    : none
baudrate is    : 9600
parity is      : none
databits are   : 8
stopbits are   : 1
escape is      : C-a
local echo is  : no
noinit is      : no
noreset is     : no
hangup is      : no
nolock is      : no
send_cmd is    : ascii-xfr -s -l 30 -n
receive_cmd is : rz -vv -E
imap is        : crcrlf,lfcrlf,
omap is        : crlf,delbs,
emap is        : crcrlf,delbs,
logfile is     : none
initstring     : none
exit_after is  : not set
exit is        : no

FATAL: cannot open /dev/ttyUSB0: No such file or directory
make: *** [Makefile:27: terminal] Error 1
vsduser@vsdsquadron:~/VSDSquadron_FM/uart_loopback$
```

I used, Make terminal -b 9600 /dev/ttyUSB0 as shown above

That was a mistake I had made,

Instead I used,

 sudo picocom -b 9600 /dev/ttyUSB0 –echo

This worked and is shown in the video attached to the main branch.

In the video (**Screen Recording 2025-03-26 132146.mp4**), the input is the output according to the Verilog file.

So, When I typed a, I got aa

Also, use

Ctrl a+q to quit the terminal

---- Archana Bhat---