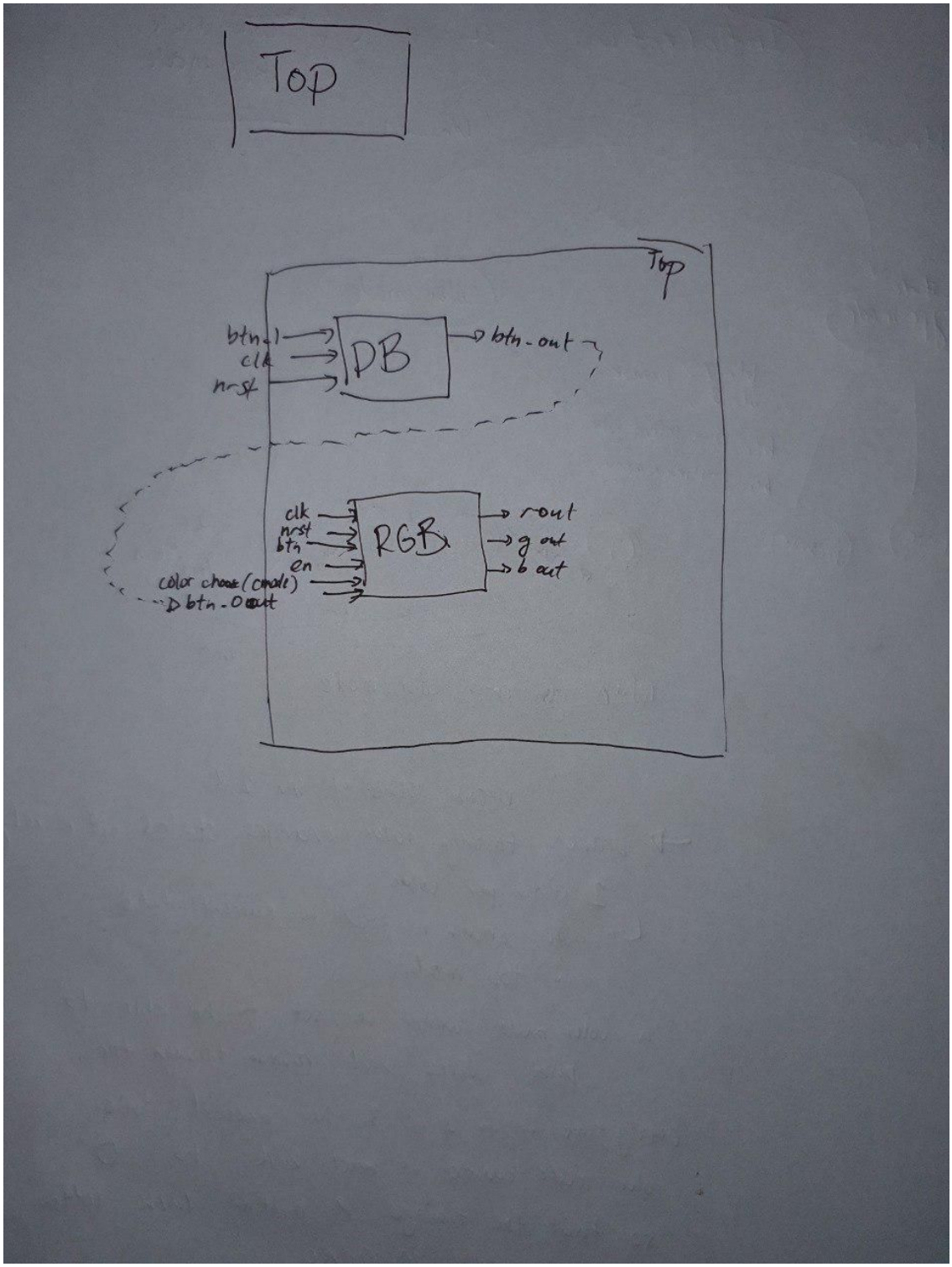
Russtin Tolentino
2015-08760

CoE 111

The MP consists of 3 main modules: top, RGB, debounce.

Top module primarily controls the LCD and what is shown on the screen. The RGB module is primarily responsible for the colors shows in the LED of the FPGA. The debounce module fixes the button presses on the FPGA so that the top module and RGB module can read these presses properly.

The btn pressed output of the debounce module is what is fed into the RGB module and the Top module so that these presses can be translated properly into the right instructions.




For Initialization, pages 33  and 47-48 were referenced primarily in being able to write the code for it.
The functions commands were followed based on the diagram.

For State 1, an initialization counter was made. It needed to exceed at least 15ms before being able to proceed to state 2. As shown in the code,

Data ‘0011’ was sent after the counter reached 15ms + 100ns delay to set function command.

State 3 needed to wait for 4.1ms before being able to send data ‘0011’(State 4) to set function set command. Hence, a delay of 4.1ms + 100ns was used to ensure proper sending of data.

Throughout initialization, delays of 500 ns and 100 ns were used to ensure proper sending of data.
State 5 needed to wait for more than 100 microseconds before data ‘0011’(State 6) was sent again for function set command.

State 7 used 500ns + 100ns to send data ‘0010’ to set interface to 4-bits

State 8 used the same delay (500ns + 100ns) to send ‘0010’
An interterval of 0000 was sent after

For State 9, ‘1000’ was sent after the delay (Display Off)

0000 interval
State 10 sent ‘0001’ after the delay to clear display
State 11 sent ‘0110’ For entry mode set
State 12 sent ‘1111’ for Display ON

CODE:

```
                        if(init_counter == 1600000) begin
                            enable <= 1;
                            data_lines <= 4'b0011; //State 2: Send Data
(000011)
                        end

                            else if(init_counter == 1600050+500) begin
                            enable <= 1;
                        end

                            else if(init_counter == 1600100+1000) begin
                            enable <= 0;
                           // data_lines <= 4'b0000;
                        end
```

```
                        else if(init_counter == 1600150+1500) begin
                        enable <= 0;
                    end             //State 3: delay 4.1ms + 100ns
//State 4: Send Data (000011)

                        else if (init_counter == 2300000+2000) begin
                        enable <= 1;
                        data_lines <= 4'b0011;          //2
                    end

                        else if (init_counter == 2300050+2500) begin
                        //enable <= 1;
                    end

                        else if (init_counter == 2300100+3000) begin
                        enable <= 0;
                        //data_lines <= 4'b0000;
                    end

                        else if (init_counter == 2300150+3500) begin
                      //  enable <= 0;
                    end            //State 5: delay 100us + 100ns
//State 6: Send Data (000011)

                        else if (init_counter == 2320000+4000) begin
                        enable <= 1;
                        data_lines <= 4'b0011;          //3
                    end

                        else if (init_counter == 2320050+4500) begin
                        enable <= 1;
                    end

                        else if (init_counter == 2320150+5500) begin
                        enable <= 0;
                      //  data_lines <= 4'b0000;
```

```
                        end

                            else if (init_counter == 2320200+6000) begin
                            enable <= 0;
                        end  + 100ns //State 7: Send Data (000010)

                            else if (init_counter == 2320260+6500+5000)
begin
                            enable <= 1;
                            data_lines <= 4'b0010;
                        end

                            else if (init_counter == 2320410+8000+5000)
begin
                            enable <= 0;
                          //  data_lines <= 4'b0000;
                        end  + 100ns //State 8: Send Data (000010)

                            else if (init_counter == 2320600+9000+10000)
begin
                            enable <= 1;
                            data_lines <= 4'b0010;
                        end

                            else if (init_counter == 2320700+10000+10000)
begin
                            enable <= 0;
                          //  data_lines <= 4'b0000;
                        end  + 100ns //State 9: Send Data (000000)

                            else if (init_counter == 2320800+11000+15000)
begin
                            enable <= 1;
                            data_lines <= 4'b1000;
                        end  + 100ns
```

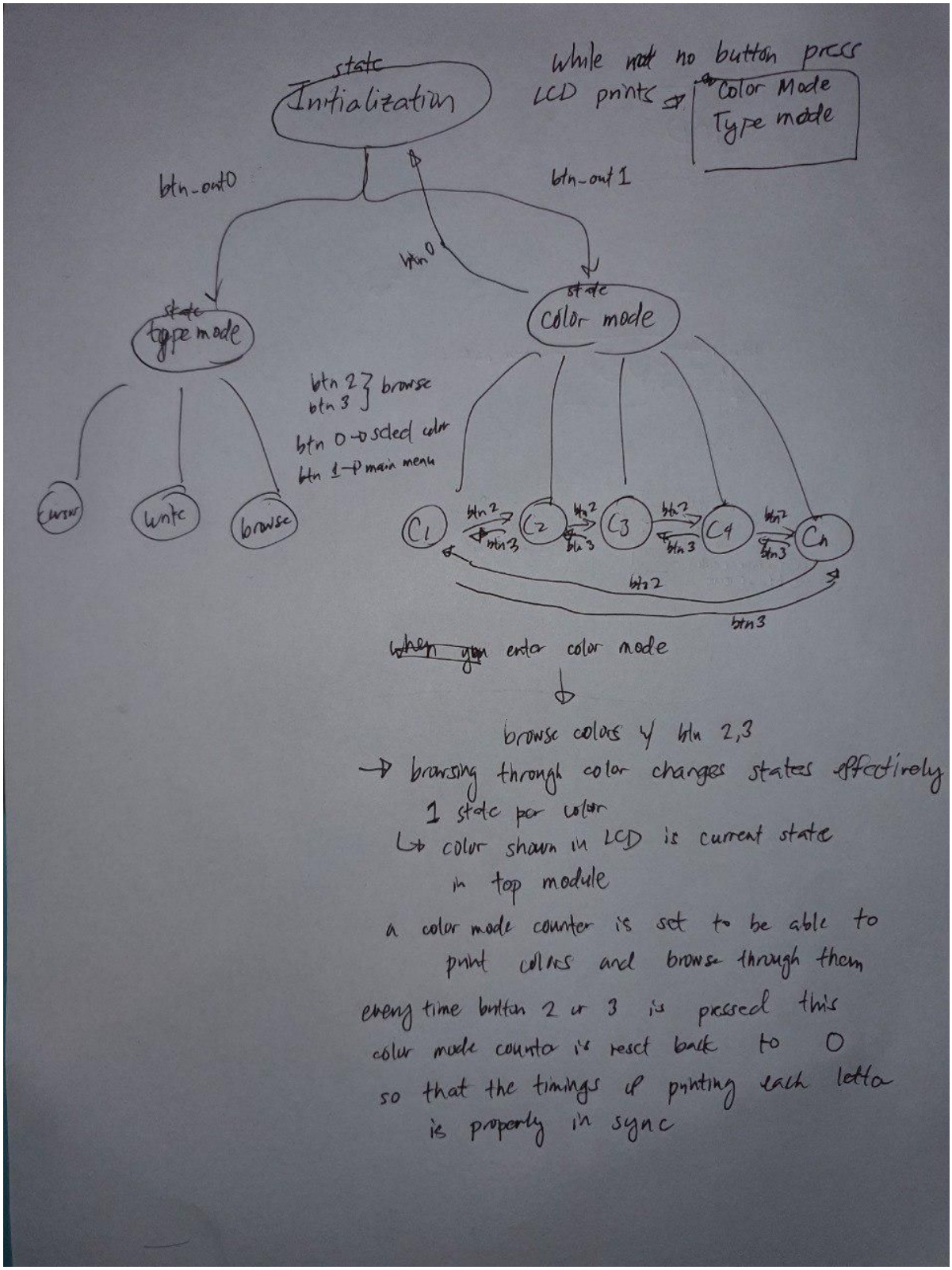
```
                            else if (init_counter == 2320900+12000+15000)
begin
                                enable <= 0;

                            end

                                else if (init_counter == 2321000+13000+20000)
begin
                                enable <= 1;
                            end

                                else if (init_counter == 2321100+14000+20000)
begin
                                enable <= 0;

                            end

                                else if (init_counter == 2321200+15000+25000)
begin
                                enable <= 1;
                                data_lines <= 4'b1000;
                            end

                                else if (init_counter == 2321300+16000+25000)
begin
                                enable <= 0;
                             //   data_lines <= 4'b0000;
                            end  + 100ns //State 10: Send Data (0001)

                                else if (init_counter == 2321400+17000+30000)
begin
                                enable <= 1;
                                data_lines <= 4'b0000;
                            end

                                else if (init_counter == 2321500+18000+30000)
begin
```

```
                            enable <= 0;
                        end
                        else if (init_counter == 2321600+19000+35000)
begin
                            enable <= 1;
                            data_lines <= 4'b0001;
                        end

                            else if (init_counter == 2321700+20000+35000)
begin
                            enable <= 0;

                        end  + 100ns //State 11: Send Data (0110)

                            else if (init_counter ==
2321800+21000+40000+1500000) begin
                            enable <= 1;
                            data_lines <= 4'b0000;
                        end

                            else if (init_counter ==
2321900+22000+40000+1500000) begin
                            enable <= 0;
                        end

                            else if (init_counter ==
2322000+23000+45000+1500000) begin
                            enable <= 1;
                            data_lines <= 4'b0110;
                        end

                            else if (init_counter ==
2322100+24000+45000+1500000) begin
                            enable <= 0;

                        end
```

```
                            else if (init_counter ==
2322200+25000+50000+1500000) begin
                            enable <= 1;
                            data_lines <= 4'b0000;
                        end   + 100ns //State 12: Send Data (1111)

                            else if (init_counter ==
2322300+27000+50000+1500000) begin
                            enable <= 0;
                          //  data_lines <= 4'b0000;
                        end

                            else if (init_counter ==
2322400+28000+55000+1500000) begin
                            enable <= 1;
                            data_lines <= 4'b1111; //send 1111
                        end

                            else if (init_counter ==
2322400+30000+55000+1500000) begin
                            enable <= 0;
                            rs <= 1;           //added
                           // data_lines <= 4'b0000;
                        end
```

Rs is set to 0 for LCD instructions, 1 for receiving data

Color mode:


state
Initialization
while not no button press
LCD prints
Color Mode
Type mode
btn_out0
btn_out1
btn 0
state
type mode
state
color mode
btn 2 } browse
btn 3
btn 0 → select color
btn 1 → main menu
Erase
write
browse
C1
C2
C3
C4
Cn
btn 2
btn 3
when you enter color mode
browse colors w/ btn 2,3
→ browsing through color changes states effectively
1 state per color
↳ color shown in LCD is current state
in top module
a color mode counter is set to be able to
print colors and browse through them
every time button 2 or 3 is pressed this
color mode counter is reset back to 0
so that the timings of printing each letter
is properly in sync

Type mode:

Type mode enters either state cursor, write or browse
It has its own counter, type mode counter to know which state is selected.

Additional notes,
Sir, I asked for help from my classmates in making this MP especially Andy Concejero (2016-0995). I had difficulty in conceptualizing everything and my classmates helped me
Anyway,
Thank you very much for the sem sir!! Thanks for being very understanding and approachable.