Digital Clock Report

EE23BTECH11203 : Adarsh A
EE23BTECH11210 : Machineni Dhyana Teja

1 Introduction

This report presents the implementation of a ”Digital clock” (with some features (timer, day and year)) on Vaman Board using Verilog HDL.

1.1 Apparatus

- Vaman Board
- 16 x 2 LCD Display
- 10k Potentiometer
- Breadboard
- Jumper wires

1.2 Vaman board Pinouts

The PU64 package, describes the pin definitions of this board used in this project

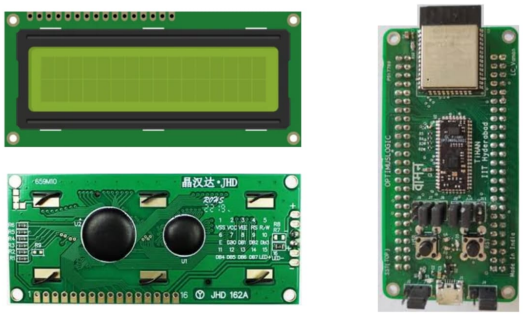
Refer : Pinouts

1.3 16x2 LCD Display

This LCD is comprised of 16 pins and the brief overview of the functionality of each pin is discussed below.

- VSS (Ground): This pin is connected to ground (0V) and serves as the reference for all other voltages on the LCD.
- VDD (Power Supply): This pin is connected to the positive power supply (+5V from the Vaman) and provides power to the LCD.

- VO (Contrast Adjustment): This pin is used to adjust the contrast of the display. By varying the voltage (for which the potentiometer is connected) applied to this pin, you can control the contrast of the characters displayed on the LCD.
- RS (Register Select): This pin determines whether data sent to the LCD is interpreted as a command (RS=0) or as character data (RS=1).
- RW (Read/Write): This pin determines the direction of data transfer between the microcontroller and the LCD. When RW is high, the microcontroller reads from the LCD; when RW is low, the microcontroller writes to the LCD.
- E (Enable): This pin is used to enable the LCD. When a high-to-low pulse is applied to this pin, the LCD reads the data present on its data pins.
- DB0-DB7 (Data Pins): These eight pins are used to send data and commands to the LCD. When writing data to the LCD, these pins carry the ASCII code of the characters to be displayed. When sending commands, these pins carry control signals to configure the LCD (e.g., clear display, set cursor position).
- A (Backlight Anode): This pin is connected to the anode of the backlight LED(s). Applying a voltage to this pin turns on the backlight and controls its brightness.
- K (Backlight Cathode): This pin is connected to the cathode of the backlight LED(s). Connecting this pin to ground completes the circuit and turns on the backlight.



Vaman board and LCD Display

2 Verilog Code

2.1 The Design (”des.v”)

```
module des(input switch, input dd, input timer, input start, input stop,
    output reg LCD_RS, output reg LCD_E, output reg[7:4] DATA, output
    redled, output blueled);

wire clk;

qlal4s3b_cell_macro u_qlal4s3b_cell_macro (
.Sys_Clk0 (clk)
);

reg clk1 = 0;
reg [26:0] x=0;
integer i = 1;
integer c = 3;
integer sec = 0;
reg [5:0] min;
reg [4:0] hr_24;
reg [3:0] hr_12;
integer o;
reg [25:0] count=0;
reg [3:0] Datas [1:77];
reg [3:0] b = 0;
integer sect;
integer seco;
integer mint;
integer mino;
integer hrt;
integer hro;
reg led1;
reg led2;
integer k = 0;
integer w = 0;
reg [5:0] st;
reg [5:0] mt;
reg [5:0] ht;
reg [5:0] st_i;
reg [5:0] mt_i;
reg [5:0] ht_i;
integer year;
integer htt;
integer hto;
integer mtt;
integer mto;
integer stt;
integer sto;
integer day;
integer j = 0;
integer yth,yh,yt,yo;

always @(posedge clk) begin

    if (j == 0) begin
        //Initialising Year
```

```
                year = 2024;
                j = j + 1;
            end

            x = x + 1;

            if(x > 6000000) begin
                clk1 = !clk1; //Clock with period 1sec
                x = 0;
            end

            yth = year / 1000;
            yh  = (year % 1000) / 100;
            yt  = ((year % 1000) % 100) / 10;
            yo  = ((year % 1000) % 100) % 10;

        end
        always @(posedge clk1) begin

            if (k == 0) begin
                min = 59; //Initialising Clock time and Day
                hr_24 = 23;
                hr_12 = (hr_24 <= 12) ? hr_24 :  (hr_24 - 12);
                day = 5;
                k = k + 1;
            end

            else begin

                if (sec < 59) begin
                    sec = sec + 1;
                end

                else begin
                    sec = 0;
                end

                if (sec == 0 & min < 59) begin
                    min = min + 1;
                end

                else if (sec > 0) begin
                    min = min;
                end

                else begin
                    min = 0;
                end

                if (sec == 0 & min == 0 & hr_24 <= 23) begin
                    hr_24 = hr_24 + 1;
                end

                else begin
                    hr_24 = hr_24;
                end

                if (hr_24 > 23) begin
                    hr_24 = 0;
```

```
                    day = day + 1;

                    if (day > 6) begin
                        day = 0;
                    end
                end

                if (sec == 0 & min == 0 & hr_24 <= 12) begin
                    hr_12 = hr_12 + 1;
                end

                else begin
                    hr_12 = hr_12;
                end

                if (hr_12 > 12) begin
                    hr_12 = 1;
                end

            end

            seco = sec % 10;
            sect = (sec - seco) / 10;

            mino = min % 10;
            mint = (min - mino) / 10;

            if (switch == 1) begin

                hro = hr_12 % 10;
                hrt = (hr_12 - hro) / 10;
                led2 = (hr_24 <= 12) ? 0 : 1; // Blue led blinks when PM
            end

            else begin

                hro = hr_24 % 10;
                hrt = (hr_24 - hro) / 10;
                led2 = 0;
            end
        end


        always @(posedge clk1) begin

            if (w == 0) begin
                ht_i = 0; //Initialising Timer
                mt_i = 0;
                st_i = 30;
                mt = mt_i;
                ht = ht_i;
                st = st_i;
                w = w + 1;
            end

            else if (start == 1 & w != 2) begin

                if (stop != 1 & st >= 0) begin
                    if (st > 0)
```

```
                        st = st - 1;
                    else
                        st = 59;
                end


                if (stop != 1 & st == 59 & mt >= 0) begin
                    if (mt > 0)
                        mt = mt - 1;
                    else
                        mt = 59;
                end

                if (stop != 1 & st == 59 & mt == 59 & ht > 0) begin
                    ht = ht - 1;
                end

                if (st == 0 & mt == 0 & ht == 0) begin
                    if (stop != 1)
                        led1 = 1;
                    else
                        led1 = 0;
                        w = 2;
                end

                if (stop == 1) begin
                    st = st_i;
                    mt = mt_i;
                    ht = ht_i;
                end
            end

            else begin
                led1 = 0;
            end

            if (st == 0 & mt == 0 & ht == 0) begin
                if (stop != 1) begin
                    led1 = 1;  //Red led blinks when timer hits zero
                end
                else begin
                    st = st_i;
                    mt = mt_i;
                    ht = ht_i;
                    w = 0;
                end
            end

            sto = st % 10;
            stt = (st - sto) / 10;

            mto = mt % 10;
            mtt = (mt - mto) / 10;

            hto = ht % 10;
            htt = (ht - hto) / 10;


        end
```

```
229
230         assign redled = led1;
231         assign blueled = led2;
232
233         always @(posedge clk) begin
234             Datas[1]    =  4'h3;      //-- initializing controller--
235             Datas[2]    =  4'h3;
236             Datas[3]    =  4'h3;      //-- set to 4-bit input mode --
237             Datas[4]    =  4'h2;
238             Datas[5]    =  4'h2;      //--2 line, 5x7 matrix  --
239             Datas[6]    =  4'h8;
240             Datas[7]    =  4'h0;      //--turn cursor off (0x0E to enable) --
241             Datas[8]    =  4'hC;
242             Datas[9]    =  4'h0;      //-- cursor direction = right --
243             Datas[10]   =  4'h6;
244             Datas[11]   =  4'h0;      //--  start with clear display  --
245             Datas[12]   =  4'h1;
246             Datas[13]   =  4'h8;      //starting from line 1
247             Datas[14]   =  4'h0;
248             Datas[15]   =  1'b1;
249
250             if (timer == 1 & dd != 1) begin
251
252                 Datas[16]   =  4'h2;
253                 Datas[17]   =  4'h0;
254                 Datas[18]   =  4'h2;
255                 Datas[19]   =  4'h0;
256                 Datas[20]   =  4'h2;
257                 Datas[21]   =  4'h0;
258                 Datas[22]   =  4'h2;
259                 Datas[23]   =  4'h0;
260                 Datas[24]   =  4'h2;
261                 Datas[25]   =  4'h0;
262                 Datas[26]   =  4'h2;
263                 Datas[27]   =  4'h3;
264                 Datas[28]   =  4'h5;
265                 Datas[29]   =  4'h4;
266                 Datas[30]   =  4'h4;
267                 Datas[31]   =  4'h9;
268                 Datas[32]   =  4'h4;
269                 Datas[33]   =  4'hD;
270                 Datas[34]   =  4'h4;
271                 Datas[35]   =  4'h5;
272                 Datas[36]   =  4'h5;
273                 Datas[37]   =  4'h2;
274                 Datas[38]   =  4'h2;
275                 Datas[39]   =  4'h0;
276                 Datas[40]   =  4'h2;
277                 Datas[41]   =  4'h0;
278                 Datas[42]   =  4'h2;
279                 Datas[43]   =  4'h0;
280             end
281
282             else if (dd == 1) begin
283
284                 Datas[16]   =  4'h4;
285                 Datas[17]   =  4'h4;
286                 Datas[18]   =  4'h6;
287                 Datas[19]   =  4'h1;
```

```
                Datas[20]  =  4'h7;
                Datas[21]  =  4'h9;
                Datas[22]  =  4'h2;
                Datas[23]  =  4'h0;
                Datas[24]  =  4'h3;
                Datas[25]  =  4'hA;
                Datas[26]  =  4'h2;
                Datas[27]  =  4'h0;

                case(day)
                1 : begin
                    Datas[28]  =  4'h4;//4d 6f 6e 64 61 79
                    Datas[29]  =  4'hD;
                    Datas[30]  =  4'h6;
                    Datas[31]  =  4'hF;
                    Datas[32]  =  4'h6;
                    Datas[33]  =  4'hE;
                    Datas[34]  =  4'h6;
                    Datas[35]  =  4'h4;
                    Datas[36]  =  4'h6;
                    Datas[37]  =  4'h1;
                    Datas[38]  =  4'h7;
                    Datas[39]  =  4'h9;
                    Datas[40]  =  4'h2;
                    Datas[41]  =  4'h0;
                    Datas[42]  =  4'h2;
                    Datas[43]  =  4'h0;

                end

                0 : begin
                    Datas[28]  =  4'h5;//53 75 6e 64 61 79
                    Datas[29]  =  4'h3;
                    Datas[30]  =  4'h7;
                    Datas[31]  =  4'h5;
                    Datas[32]  =  4'h6;
                    Datas[33]  =  4'hE;
                    Datas[34]  =  4'h6;
                    Datas[35]  =  4'h4;
                    Datas[36]  =  4'h6;
                    Datas[37]  =  4'h1;
                    Datas[38]  =  4'h7;
                    Datas[39]  =  4'h9;
                    Datas[40]  =  4'h2;
                    Datas[41]  =  4'h0;
                    Datas[42]  =  4'h2;
                    Datas[43]  =  4'h0;

                end

                2 : begin
                    Datas[28]  =  4'h5;//54 75 65 73 64 61 79
                    Datas[29]  =  4'h4;
                    Datas[30]  =  4'h7;
                    Datas[31]  =  4'h5;
                    Datas[32]  =  4'h6;
                    Datas[33]  =  4'h5;
                    Datas[34]  =  4'h7;
                    Datas[35]  =  4'h3;
```

```
                    Datas[36]  =  4'h6;
                    Datas[37]  =  4'h4;
                    Datas[38]  =  4'h6;
                    Datas[39]  =  4'h1;
                    Datas[40]  =  4'h7;
                    Datas[41]  =  4'h9;
                    Datas[42]  =  4'h2;
                    Datas[43]  =  4'h0;
                end

                3 : begin
                    Datas[28]  =  4'h5;//57 65 64 2d 64 61 79
                    Datas[29]  =  4'h7;
                    Datas[30]  =  4'h6;
                    Datas[31]  =  4'h5;
                    Datas[32]  =  4'h6;
                    Datas[33]  =  4'h4;
                    Datas[34]  =  4'h2;
                    Datas[35]  =  4'hD;
                    Datas[36]  =  4'h6;
                    Datas[37]  =  4'h4;
                    Datas[38]  =  4'h6;
                    Datas[39]  =  4'h1;
                    Datas[40]  =  4'h7;
                    Datas[41]  =  4'h9;
                    Datas[42]  =  4'h2;
                    Datas[43]  =  4'h0;
                end

                4 : begin
                    Datas[28]  =  4'h5;//54 68 75 72 73 64 61 79
                    Datas[29]  =  4'h4;
                    Datas[30]  =  4'h6;
                    Datas[31]  =  4'h8;
                    Datas[32]  =  4'h7;
                    Datas[33]  =  4'h5;
                    Datas[34]  =  4'h7;
                    Datas[35]  =  4'h2;
                    Datas[36]  =  4'h7;
                    Datas[37]  =  4'h3;
                    Datas[38]  =  4'h6;
                    Datas[39]  =  4'h4;
                    Datas[40]  =  4'h6;
                    Datas[41]  =  4'h1;
                    Datas[42]  =  4'h7;
                    Datas[43]  =  4'h9;
                end

                5 : begin
                    Datas[28]  =  4'h4;//46 72 69 64 61 79
                    Datas[29]  =  4'h6;
                    Datas[30]  =  4'h7;
                    Datas[31]  =  4'h2;
                    Datas[32]  =  4'h6;
                    Datas[33]  =  4'h9;
                    Datas[34]  =  4'h6;
                    Datas[35]  =  4'h4;
                    Datas[36]  =  4'h6;
                    Datas[37]  =  4'h1;
```

```
406                     Datas[38]  =  4'h7;
407                     Datas[39]  =  4'h9;
408                     Datas[40]  =  4'h2;
409                     Datas[41]  =  4'h0;
410                     Datas[42]  =  4'h2;
411                     Datas[43]  =  4'h0;
412                 end
413
414                 6 : begin
415                     Datas[28]  =  4'h5;//53 61 74 75 72 64 61 79
416                     Datas[29]  =  4'h3;
417                     Datas[30]  =  4'h6;
418                     Datas[31]  =  4'h1;
419                     Datas[32]  =  4'h7;
420                     Datas[33]  =  4'h4;
421                     Datas[34]  =  4'h7;
422                     Datas[35]  =  4'h5;
423                     Datas[36]  =  4'h7;
424                     Datas[37]  =  4'h2;
425                     Datas[38]  =  4'h6;
426                     Datas[39]  =  4'h4;
427                     Datas[40]  =  4'h6;
428                     Datas[41]  =  4'h1;
429                     Datas[42]  =  4'h7;
430                     Datas[43]  =  4'h9;
431                 end
432                 endcase
433             end
434
435             else begin
436
437                 Datas[16]  =  4'h2;
438                 Datas[17]  =  4'h0;
439                 Datas[18]  =  4'h2;
440                 Datas[19]  =  4'h0;
441                 Datas[20]  =  c;
442                 Datas[21]  =  hrt;
443                 Datas[22]  =  c;
444                 Datas[23]  =  hro;
445                 Datas[24]  =  4'h2;
446                 Datas[25]  =  4'h0;
447                 Datas[26]  =  4'h3;
448                 Datas[27]  =  4'hA;
449                 Datas[28]  =  4'h2;
450                 Datas[29]  =  4'h0;
451                 Datas[30]  =  c;
452                 Datas[31]  =  mint;
453                 Datas[32]  =  c;
454                 Datas[33]  =  mino;
455                 Datas[34]  =  4'h2;
456                 Datas[35]  =  4'h0;
457                 Datas[36]  =  4'h3;
458                 Datas[37]  =  4'hA;
459                 Datas[38]  =  4'h2;
460                 Datas[39]  =  4'h0;
461                 Datas[40]  =  c;
462                 Datas[41]  =  sect;
463                 Datas[42]  =  c;
464                 Datas[43]  =  seco;
```

```
            end

            Datas[44]  =  4'hC; //next line
            Datas[45]  =  4'h0;

            if (switch == 1 & timer != 1 & dd != 1) begin

                Datas[46]  =  4'h4;
                Datas[47]  =  4'h3;
                Datas[48]  =  4'h4;
                Datas[49]  =  4'hC;
                Datas[50]  =  4'h4;
                Datas[51]  =  4'hF;
                Datas[52]  =  4'h4;
                Datas[53]  =  4'h3;
                Datas[54]  =  4'h4;
                Datas[55]  =  4'hB;
                Datas[56]  =  4'h2;
                Datas[57]  =  4'h0;
                Datas[58]  =  c;
                Datas[59]  =  1;
                Datas[60]  =  c;
                Datas[61]  =  2;
                Datas[62]  =  4'h6;
                Datas[63]  =  4'h8;
                Datas[64]  =  4'h2;
                Datas[65]  =  4'h0;

                if (hr_24 > 12) begin
                    Datas[66]  =  4'h5;
                    Datas[67]  =  4'h0;
                end
                else begin
                    Datas[66]  =  4'h4;
                    Datas[67]  =  4'h1;
                end
                Datas[68]  =  4'h4;
                Datas[69]  =  4'hD;
                Datas[70]  =  4'h2;
                Datas[71]  =  4'h0;
                Datas[72]  =  4'h4;
                Datas[73]  =  4'h9;
                Datas[74]  =  4'h5;
                Datas[75]  =  4'h3;
                Datas[76]  =  4'h5;
                Datas[77]  =  4'h4;
            end

            else if (timer == 1 & dd != 1) begin

                Datas[46]  =  4'h2;
                Datas[47]  =  4'h0;
                Datas[48]  =  4'h2;
                Datas[49]  =  4'h0;
                Datas[50]  =  c;
                Datas[51]  =  htt;
                Datas[52]  =  c;
                Datas[53]  =  hto;
                Datas[54]  =  4'h2;
```

```
                Datas[55]  =  4'h0;
                Datas[56]  =  4'h3;
                Datas[57]  =  4'hA;
                Datas[58]  =  4'h2;
                Datas[59]  =  4'h0;
                Datas[60]  =  c;
                Datas[61]  =  mtt;
                Datas[62]  =  c;
                Datas[63]  =  mto;
                Datas[64]  =  4'h2;
                Datas[65]  =  4'h0;
                Datas[66]  =  4'h3;
                Datas[67]  =  4'hA;
                Datas[68]  =  4'h2;
                Datas[69]  =  4'h0;
                Datas[70]  =  c;
                Datas[71]  =  stt;
                Datas[72]  =  c;
                Datas[73]  =  sto;
                Datas[74]  =  4'h2;
                Datas[75]  =  4'h0;
                Datas[76]  =  4'h2;
                Datas[77]  =  4'h0;
            end

            else if (dd == 1) begin

                Datas[46]  =  4'h5;//59 65 61 72 20 3a 20
                Datas[47]  =  4'h9;
                Datas[48]  =  4'h6;
                Datas[49]  =  4'h5;
                Datas[50]  =  4'h6;
                Datas[51]  =  4'h1;
                Datas[52]  =  4'h7;
                Datas[53]  =  4'h2;
                Datas[54]  =  4'h2;
                Datas[55]  =  4'h0;
                Datas[56]  =  4'h3;
                Datas[57]  =  4'hA;
                Datas[58]  =  4'h2;
                Datas[59]  =  4'h0;
                Datas[60]  =  c;
                Datas[61]  =  yth;
                Datas[62]  =  c;
                Datas[63]  =  yh;
                Datas[64]  =  c;
                Datas[65]  =  yt;
                Datas[66]  =  c;
                Datas[67]  =  yo;
                Datas[68]  =  4'h2;
                Datas[69]  =  4'h0;
                Datas[70]  =  4'h2;
                Datas[71]  =  4'h0;
                Datas[72]  =  4'h2;
                Datas[73]  =  4'h0;
                Datas[74]  =  4'h2;
                Datas[75]  =  4'h0;
                Datas[76]  =  4'h2;
                Datas[77]  =  4'h0;
```

```
583             end
584
585             else begin
586
587                 Datas[46]  =  4'h2;
588                 Datas[47]  =  4'h0;
589                 Datas[48]  =  4'h4;
590                 Datas[49]  =  4'h3;
591                 Datas[50]  =  4'h4;
592                 Datas[51]  =  4'hC;
593                 Datas[52]  =  4'h4;
594                 Datas[53]  =  4'hF;
595                 Datas[54]  =  4'h4;
596                 Datas[55]  =  4'h3;
597                 Datas[56]  =  4'h4;
598                 Datas[57]  =  4'hB;
599                 Datas[58]  =  4'h2;
600                 Datas[59]  =  4'h0;
601                 Datas[60]  =  c;
602                 Datas[61]  =  2;
603                 Datas[62]  =  c;
604                 Datas[63]  =  4;
605                 Datas[64]  =  4'h6;
606                 Datas[65]  =  4'h8;
607                 Datas[66]  =  4'h7;
608                 Datas[67]  =  4'h2;
609                 Datas[68]  =  4'h2;
610                 Datas[69]  =  4'h0;
611                 Datas[70]  =  4'h4;
612                 Datas[71]  =  4'h9;
613                 Datas[72]  =  4'h5;
614                 Datas[73]  =  4'h3;
615                 Datas[74]  =  4'h5;
616                 Datas[75]  =  4'h4;
617                 Datas[76]  =  4'h2;
618                 Datas[77]  =  4'h0;
619             end
620         end
621
622         always @(posedge clk) begin
623
624             if (i <= 14) begin
625
626                 LCD_RS <= 1'b0;
627                 DATA = Datas[i];
628                 LCD_E <= 1'b1;
629
630                 if (count == 800) begin //waiting 40us
631                     LCD_E <= 1'b0;
632                     count <= 0;
633                     i <= i + 1;
634                 end
635                 else
636                     count <= count + 1;
637
638             end
639             if (i==15) begin
640                 if (count==60000) begin //waiting 3ms
641                     count <= 0;
```

```
                    i <= i + 1;
                end
                else
                    count <= count + 1;
            end
            if (i > 15 & i <= 43) begin

                LCD_RS <= 1'b1;
                DATA = Datas[i];
                LCD_E <= 1'b1;

                if (count == 800) begin //waiting 40us
                    LCD_E <= 1'b0;
                    count <= 0;
                    i <= i + 1;
                end
                else
                    count <= count + 1;
            end

            if (i >= 44 & i <= 45) begin
                LCD_RS <= 1'b0;
                DATA = Datas[i];
                LCD_E <= 1'b1;

                if (count == 800)begin //waiting 40us
                    LCD_E <= 1'b0;
                    count <= 0;
                    i <= i + 1;
                end

                else
                    count <= count + 1;

            end

            if (i > 45 & i <= 77) begin
                LCD_RS <= 1'b1;
                DATA = Datas[i];
                LCD_E <= 1'b1;

            if (count == 800)begin //waiting 40us
                LCD_E <= 1'b0;
                count <= 0;
                i <= i + 1;
            end
            else
                    count <= count + 1;
            end

            if (i > 77) begin
                i <= 13;
            end

        end
        endmodule
```

2.2 Assigning the pins ("port.pcf")

```
1  //LCD_E:IO_9 ,LCD_RS:IO_10 ,DATA(7):IO_11 ,DATA(6):IO_12 ,DATA(5):
       IO_13 ,DATA(4):IO_14, switch : IO_21, timer : IO_30, start : IO_31,
        stop : IO_29, dd : IO_26
2  set_io LCD_E 60
3  set_io LCD_RS 59
4  set_io DATA(7) 57
5  set_io DATA(6) 56
6  set_io DATA(5) 55
7  set_io DATA(4) 54
8  set_io redled 34
9  set_io blueled 38
10 set_io switch 39
11 set_io timer 25
12 set_io start 23
13 set_io stop 26
14 set_io dd 30
```

Refer : Pinouts for the pin definitions

3 Code Overview

This code summarizes the function of a clock (counter) with timer and day feature in it. It also describes on how to display it on a LCD.

3.1 Clock and Features

- The inputs and outputs are taken as mentioned in the top module of the code.
- A predefined clock (clk) is used with frequency 20 MHz

```
qlal4s3b_cell_macro u_qlal4s3b_cell_macro (.Sys_Clk0 (clk));
```

- A register clock (clk1) is defined by negating it with some delay (x) to get the period as 1 sec.
- The registers for hour, minute and seconds with their corresponding ten's and one's place integers are defined as follows in the code.
- The clock time is initialized in the always block at line 72 of the code.
- "if-else" statements are used to define the functions of a counter (clock is a counter with specific modulo values).
- The "$led2$" is set to high, when the "hr_{24}" becomes greater than 12, which essentially means that the led glows when it is PM and turns off when AM.

- Integers like hrt, hro, $mint$, $mino$, $sect$, $seco$ are defined to store the corresponding ten's and one's place digits of hour, minute and seconds respectively (both 12 and 24hr formats). These are required to display them on the LCD.
- The code also describes about the day and year feature, where we initialize them at lines 52 and 75 respectively.
- When the hr_{24} becomes zero, we increment the day register by 1, so that switch-case statements can be used to describe the day to be printed.
- This code also describes the implementation of a timer feature.
- The time for the timer is initialized at line 157 of the code.
- Similar to clock, "if-else" statements are used to define the functions of a down counter (which is the timer itself).
- Input "$mode$" is used to switch between clock and timer mode. When mode is high, it is switched to timer mode and, clock otherwise.
- The timer starts de-counting when input "$start$" is high and is paused otherwise.
- The "$led1$" is set to high when the timer hits zero.
- The input "$stop$" is used to reset the timer to the initialized one.
- Similar to clock, integers are defined for the ten's and one's places of the timer to display the hours, minutes and seconds on the LCD.

3.2 Displaying it on LCD

The next part of the code describes on how to print the characters on a LCD.

- A register array "`Datas[]`" is defined to store the instructions and datas to be given to the LCD.
- The commands from "`Datas[1]`" to "`Datas[14]`" and from "`Datas[44]`" to "`Datas[45]`" are the instruction data lines given to the LCD by setting the "Register select (RS)" to 0. (Look how its done at line 624 of the code)
- Similarly, the commands from "`Datas[16]`" to "`Datas[43]`" and from "`Datas[46]`" to "`Datas[77]`" are the character data lines given to the LCD by setting the "Register select (RS)" to 1. (Refer line 649 of the code)
- The commands are given in "Hex" format. Refer : Here to covert to Hex.
- The always block at line 622, is to give delays to each data line of the LCD for its proper functioning at every period of the "clk".

4 Flashing it on to Vaman

To flash the code on Vaman FPGA board, we run a set of commands on the terminal of Ubuntu 20.04.

- Creation of .bin file with our "`des.v`" and "`port.pcf`" files

```
ql_symbiflow -compile -d ql-eos-s3 -P PU64 -v des.v -t des -p port.pcf -dump binary
```

- Dumping this des.bin onto Vaman board.

```
python3 ~/TinyFPGA-Programmer-Application/tinyfpga-programmer-gui.py
 --port/dev/ttyACM0 --appfpga des.bin --mode fpga --reset
```

5 Conclusion

In conclusion, the successful implementation of a digital clock on the Vaman FPGA board using Verilog has demonstrated the practical application of digital design concepts in real-world embedded systems.

The project not only provided valuable hands-on experience in FPGA-based digital design and Verilog programming but also highlighted the versatility and potential of FPGA technology in various embedded applications.

6 Codes and Recordings

Refer : Here , for the recordings and the codes for this project.