LCD DE2115

Integrantes:

- Cazares Cruz Jeremy Sajid
- Bucio Barrera Oscar Daniel
- Perez Ortiz Saúl
- Acosta Cortes Gerardo

Introducción:

El código proporcionado representa la implementación en Verilog de un proyecto en Quartus para controlar un LCD mediante una FPGA. El objetivo principal es mostrar un mensaje específico en el LCD utilizando una estructura modular y un controlador dedicado.

```
always@(posedge CLK or negedge RST_N)
   begin
      if (!RST_N) begin
         LUT_INDEX<=0;
         mLCD_ST<=0;
         mDLY<=0;
         mLCD_Start<=0;
         mLCD_DATA<=0;
         mLCD_RS<=0;
      end else begin
         if (LUT_INDEX<LUT_SIZE) begin
            case (mLCD_ST)
               0:
                  begin
                     mLCD_DATA<=LUT_DATA[7:0];
                     mLCD_RS<=LUT_DATA[8];
                     mLCD_Start<=1;
                     mLCD_ST<=1;
```

Desarrollo:

Bloque de Declaración del Módulo LCD

```
module LCD(
    CLK,
    RST_N,
    MP,

    LCD_ON,
    LCD_BLON,
    LCD_RW,
    LCD_EN,
    LCD_RS,
    LCD_DATA
);
    input       CLK,RST_N;
```

Este bloque define el módulo LCD con sus puertos de entrada (CLK, RST_N, MP) y puertos de salida (LCD_ON, LCD_BLON, LCD_RW, LCD_EN, LCD_RS, LCD_DATA). CLK es la señal de reloj, RST_N es la señal de reinicio asincrónico activa baja, MP es la entrada que contiene el mensaje que se mostrará en el LCD, y los demás son puertos de salida relacionados con el control y la interfaz del LCD.

Declaración de Variables y Parámetros

```
input       CLK,RST_N;

/////Agregado///////
input       [255:0]MP;
//imprimir 32 ascii - 16 por linea
////////////////////


LCD Module 16X2
output      [7:0]LCD_DATA;
output      LCD_ON;
output      LCD_BLON;
output      LCD_RW;
output      LCD_EN;
output      LCD_RS;

Internal Wires/Registers
reg         [5:0]LUT_INDEX;
reg         [8:0]LUT_DATA;
reg         [7:0]LUT_DATA1;
reg         [5:0]mLCD_ST;
reg         [17:0]mDLY;
reg         mLCD_Start;
reg         [7:0]mLCD_DATA;
reg         mLCD_RS;
wire        mLCD_Done;
parameter   LCD_INTIAL=0;
parameter   LCD_LINE1=5;
parameter   LCD_CH_LINE=LCD_LINE1+16;
parameter   LCD_LINE2=LCD_LINE1+16+1;
parameter   LUT_SIZE=LCD_LINE1+32+1;
```

En este bloque se declaran variables y parámetros internos necesarios para el funcionamiento del módulo. MP es un vector de 256 bits utilizado para almacenar el

mensaje que se mostrará en el LCD. LUT_INDEX y LUT_DATA se utilizan en la lógica interna para determinar qué datos enviar al LCD. Otros registros y parámetros también se declaran aquí.

Bloque de Control de Estado

```
always                                                  //se
   begin
      case (LUT_INDEX)
Initial
          LCD_INTIAL+0:LUT_DATA1<=8'h00;
          LCD_INTIAL+1:LUT_DATA1<=8'h00;
          LCD_INTIAL+2:LUT_DATA1<=8'h00;
          LCD_INTIAL+3:LUT_DATA1<=8'h00;
          LCD_INTIAL+4:LUT_DATA1<=8'h00;
Line 1
          LCD_LINE1+0:LUT_DATA1<={MP[255:248]};
          LCD_LINE1+1:LUT_DATA1<={MP[247:240]};
          LCD_LINE1+2:LUT_DATA1<={MP[239:232]};//
          LCD_LINE1+3:LUT_DATA1<={MP[231:224]};
          LCD_LINE1+4:LUT_DATA1<={MP[223:216]};
          LCD_LINE1+5:LUT_DATA1<={MP[215:208]};
          LCD_LINE1+6:LUT_DATA1<={MP[207:200]};
          LCD_LINE1+7:LUT_DATA1<={MP[199:192]};
          LCD_LINE1+8:LUT_DATA1<={MP[191:184]};
          LCD_LINE1+9:LUT_DATA1<={MP[183:176]};
          LCD_LINE1+10:LUT_DATA1<={MP[175:168]};
          LCD_LINE1+11:LUT_DATA1<={MP[167:160]};
          LCD_LINE1+12:LUT_DATA1<={MP[159:152]};
          LCD_LINE1+13:LUT_DATA1<={MP[151:144]};
          LCD_LINE1+14:LUT_DATA1<={MP[143:136]};
          LCD_LINE1+15:LUT_DATA1<={MP[135:128]};
Change Line
```

Este bloque maneja la lógica de control de estado del módulo, determinando en qué estado se encuentra y cómo proceder en consecuencia. Utiliza la variable LUT_INDEX para controlar el flujo del programa.

tiempos.

Modulo Prueba LCD

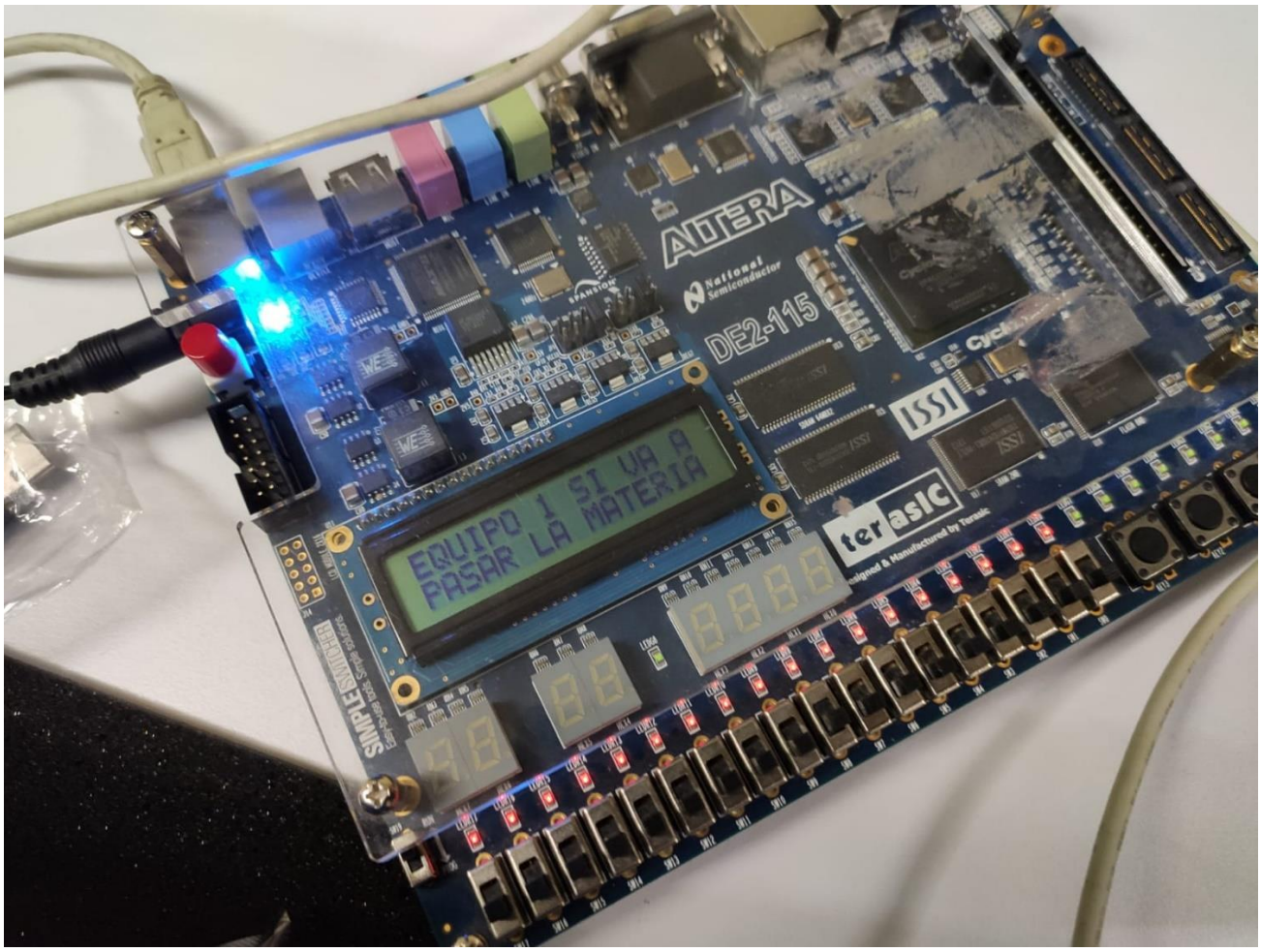
En este bloque de código le damos los valores al mensaje que vamos a mostrar en el LCD, le damos valores de código ASCCI en hexadecimal

```
wire     [255:0] mensaje_w;
//linea 1
assign       mensaje_w[255:248] = 8'h45;
assign       mensaje_w[247:240] = 8'h51;
assign       mensaje_w[231:224] = 8'h49;
assign       mensaje_w[223:216] = 8'h50;
assign       mensaje_w[215:208] = 8'h4F;
assign       mensaje_w[207:200] = 8'h20;
assign       mensaje_w[199:192] = 8'h31;
assign       mensaje_w[191:184] = 8'h20;
assign       mensaje_w[183:176] = 8'h53;
assign       mensaje_w[175:168] = 8'h49;
assign       mensaje_w[167:160] = 8'h20;
assign       mensaje_w[159:152] = 8'h56;
assign       mensaje_w[151:144] = 8'h41;
assign       mensaje_w[143:136] = 8'h20;
assign       mensaje_w[135:128] = 8'h41;
//linea 2
assign       mensaje_w[127:120] = 8'h50;
assign       mensaje_w[119:112] = 8'h41;
assign       mensaje_w[111:104] = 8'h53;
assign       mensaje_w[103:96] = 8'h41;
assign       mensaje_w[95:88] = 8'h52;
assign       mensaje_w[87:80] = 8'h20;
assign       mensaje_w[79:72] = 8'h4C;
assign       mensaje_w[71:64] = 8'h41;
assign       mensaje_w[63:56] = 8'h20;
assign       mensaje_w[55:48] = 8'h4D;
assign       mensaje_w[47:40] = 8'h41;
assign       mensaje_w[39:32] = 8'h54;
assign       mensaje_w[31:24] = 8'h45;
assign       mensaje_w[23:16] = 8'h52;
assign       mensaje_w[15:8] = 8'h49;
assign       mensaje_w[7:0] = 8'h41;
```

Asignamos pines:

in clk_i	Input	PIN_Y2
out lcd_blon	Output	PIN_L6
out lcd_data[7]	Output	PIN_M5
out lcd_data[6]	Output	PIN_M3
out lcd_data[5]	Output	PIN_K2
out lcd_data[4]	Output	PIN_K1
out lcd_data[3]	Output	PIN_K7
out lcd_data[2]	Output	PIN_L2
out lcd_data[1]	Output	PIN_L1
out lcd_data[0]	Output	PIN_L3
out lcd_en	Output	PIN_L4
out lcd_on	Output	PIN_L5
out lcd_rs	Output	PIN_M2
out lcd_rw	Output	PIN_M1
in rst_ni	Input	PIN_M23

Resultados



Conclusiones

El bloque PruebaLCD se integra de manera efectiva en la estructura general del diseño. Se encarga de establecer el mensaje que se mostrará en el LCD mediante la inicialización de la memoria mensaje_w y su posterior transmisión al módulo LCD. La instancia de lcd_u0 facilita la conexión de señales, asegurando una correcta comunicación entre los componentes del sistema.