

# 驱动 1602 液晶显示

任务要求：

- 1、弄清楚液晶 1602 驱动的原理，理解并写在实训报告中
- 2、在 DE2 板上的液晶上显示你的名字和学校名字。

建议步骤：

- 1、建立所需其他辅助模块，包括分频等；
- 2、建立 LCD 控制模块（用纯硬件实现，也可以先实现 NIOS II，再用 C 实现）；

通过 1602 时序资料我们知道 LCD\_EN 最小允许脉冲宽度周期为 900 纳秒（无最大值）、数据传输允许的最小时钟周期为 1000 纳秒，由于我们 DE2 实验板为 50MHZ 因此我们这里分频时钟周期大于 1 微秒就可，这里分频为 4HZ。本实验在 LCD1602 中间第一行，第二行分别显示某个字符，并且显示完后，以一定的速率向左循环移动。

```
module lcd1602(clk,rst,lcd_e,lcd_rw,lcd_rs,data,lcd_on,lcd_blon);
```

```
input clk,rst;
```

```
output lcd_e,lcd_rw,lcd_rs,lcd_blon,lcd_on;
```

```
output [7:0] data;
```

```
reg lcd_rw,lcd_rs;
```

```
reg [7:0] data;
```

```
reg [9:0] state;
```

```
reg [5:0] address;
```

```
reg [22:0] cnt;
```

```
reg clk500hz;
```

```
parameter IDLE =10'b00000000000;
```

```
parameter CLEAR =10'b00000000001; //清屏
```

```
parameter RETURNCURSOR =10'b0000000010; //归home位
```

```
parameter SETMODE =10'b0000000111;
```

```
//输入方式设置，读写数据后ram地址增/减 1；画面动/不动
```

```
parameter SWITCHMODE =10'b0000001111;
```

//显示状态设置，显示开/关；光标开/关；闪烁开/关

parameter SHIFT =10'b0000011100;

//光标画面滚动 画面/光标平移一位；左/右平移一位

parameter SETFUNCTION =10'b0000111100;

//工作方式设置 1: 8/1: 4 位数据接口；两行/一行显示；5x10/5x7 点阵

parameter SETCGRAM =10'b0001000000; //设置CGRAM

parameter SETDDRAM1 =10'b0010000001; //设置DDRAM

parameter SETDDRAM2 =10'b0010000010; //设置DDRAM

parameter READFLAG =10'b0100000000; //读状态

parameter WRITERAM1 =10'b1000000001; //写RAM

parameter WRITERAM2 =10'b1000000010; //写RAM

parameter READRAM =10'b1100000000; //读RAM

assign lcd\_on=1'b1;

assign lcd\_blon=1'b1;

always@(posedge clk) //由 50MHZ时钟分频得到 4HZ时钟

begin if(cnt==19'h7a120) //19'h7a120=19'd500000

begin cnt<=0;clk500hz<=~clk500hz; //500000\*20ns=1000000us 取反即为 2ms=1/500 s

end

else cnt<=cnt 1;

end

assign lcd\_e=clk500hz;

function [7:0] ddram; //

input [5:0] n;

begin

case(n)

//下面修改成你所需内容

6'b000\_000:ddram=8'b0100\_0011;//C

6'b000\_001:ddram=8'b0100\_1000;//H

6'b000\_010:ddram=8'b0100\_0101;//E

6'b000\_011:ddram=8'b0100\_1110;//N

6'b000\_100:ddram=8'b0010\_0000;//

6'b000\_101:ddram=8'b0100\_1010;//J

6'b000\_110:ddram=8'b0101\_0101;//U

6'b000\_111:ddram=8'b0100\_1110;//N

```

6'b001_000:ddram=8'b0100_1010;//J
6'b001_001:ddram=8'b0100_1001;//I
6'b001_010:ddram=8'b0100_0001;//A
6'b001_011:ddram=8'b0100_1110;//N
6'b001_100:ddram=8'b0111_0111;//G
6'b001_101:ddram=8'b0010_0000;//
6'b001_110:ddram=8'b0101_1000;//X
6'b001_111:ddram=8'b0100_1001;//I
6'b010_000:ddram=8'b0010_0000;//
6'b010_001:ddram=8'b0101_0010;//R
6'b010_010:ddram=8'b0100_0101;//E
6'b010_100:ddram=8'b0100_1110;//N

endcase
end
endfunction
always@(posedge clk500hz or negedge rst)
if(!rst) begin
state<=IDLE;
address<=6'b0000000;
data<=8'b00000000;
lcd_rs<=0;
lcd_rw<=0;
end
else begin
case(state)
IDLE: begin state<=CLEAR;
data<=8'bzzzz_zzzz;
end

CLEAR:begin lcd_rs<=0;
lcd_rw<=0;data<=8'h01;
state<=SETFUNCTION; //清屏
end
SETFUNCTION :begin lcd_rs<=0;lcd_rw<=0;
data<=8'h3c;
state<=SWITCHMODE;
end
SWITCHMODE: begin lcd_rs<=0;lcd_rw<=0; //显示状态开关设置 0C
data<=8'h0c; //开光标、开闪烁
state<=SETMODE;
end
SETMODE: begin lcd_rs<=0;lcd_rw<=0;
data<=8'h06;//采取增量方式，光标不移动

```

```

        state<=SHIFT;
    end
SHIFT:    begin lcd_rs<=0;lcd_rw<=0;

        data<=8'h18;

        state<=SETDDRAM1;
    end

SETDDRAM1 :begin lcd_rs<=0;lcd_rw<=0;data<=8'b10000100; //显示数据存储器地址 80 4
            state<=WRITERAM1;end
SETDDRAM2 :begin lcd_rs<=0;lcd_rw<=0;data<=8'b11000010; //显示数据存储器地址 80
40 2
            state<=WRITERAM2;end

WRITERAM1 :begin
    if(address<=6'b000_111 )
    begin
        lcd_rs<=1;
        lcd_rw<=0;
        data<=ddram(address);
        address<=address 1;
        state<=WRITERAM1;
    end
    else
    begin
        lcd_rs<=0;
        lcd_rw<=0;
        state<=SETDDRAM2;
    end
    end
WRITERAM2 :begin
    if(address<=6'b010_100 )
    begin
        lcd_rs<=1;
        lcd_rw<=0;
        data<=ddram(address);
        address<=address 1;
        state<=WRITERAM2;
    end
    else
    begin

```

```
    lcd_rs<=0;
    lcd_rw<=0;
    state<=SHIFT;
    address<=6'b000000;
end
end
endcase
end

endmodule
```