



特别说明

此资料来自百度文库（<http://wenku.baidu.com/>）

您目前所看到的文档是使用的抱米花百度文库下载器所生成

此文档原地址来自

<http://wenku.baidu.com/view/fd27528483d049649b665832.html>

感谢您的支持

抱米花

<http://blog.sina.com.cn/lotusbaob>

FPGA 控制 LCD 1602 调试笔记

初始化步骤:

- 1、0x38 设置为 16*2 显示， 5*7 点阵， 8 位数据接口
- 2、0x0C 说明 这里 0c 表示的是开显示，不显示光标，光标不显示，完整描述如下：

指令码								功能
0	0	0	0	1	D	C	B	D=1 开显示； D=0 关显示 C=1 显示光标； C=0 不显示光标 B=1 光标闪烁； B=0 光标不显示

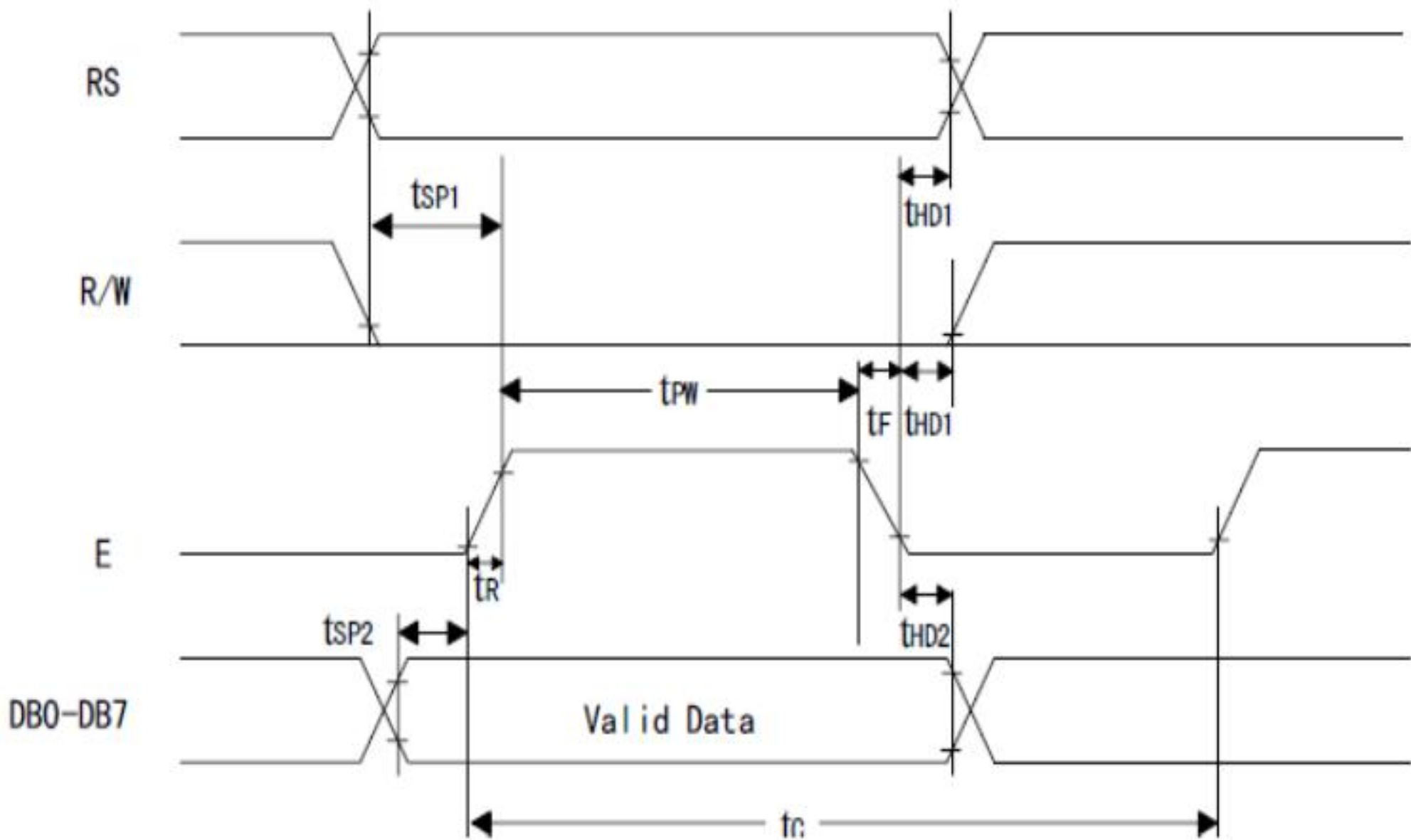
- 3、0x01 清屏幕
- 4、0x06

表示读或者写之后，地址指针加 1，光标加 1

0	0	0	0	0	1	N	S	N=1 当读或写一个字符后地址指针加一，且光标加一 N=0 当读或写一个字符后地址指针减一，且光标减一 S=1 当写一个字符，整屏显示左移（N=1）
---	---	---	---	---	---	---	---	--

- 5、0x80 位置寄存器定位于第一行的最左边

时序图:



时序参数	符号	极限值			单位	测试条件
		最小值	典型值	最大值		
E 信号周期	t_C	400	-	-	ns	引脚 E
E 脉冲宽度	t_{PW}	150	-	-	ns	
E 上升沿/下降沿时间	t_R, t_F	-	-	25	ns	
地址建立时间	t_{SP1}	30	-	-	ns	引脚 E、RS、R/W
地址保持时间	t_{HD1}	10	-	-	ns	
数据建立时间(读操作)	t_D	-	-	100	ns	引脚 DB0~DB7
数据保持时间(读操作)	t_{HD2}	20	-	-	ns	
数据建立时间(写操作)	t_{SP2}	40	-	-	ns	
数据保持时间(写操作)	t_{HD2}	10	-	-	ns	

在本例中采用 50 兆分频到 1k 的时钟，也就是 E 的保持时间是 1ms，达到以上的时序要求。

在 Altera 的 DEO 上验证通过，显示效果如下：



完整代码如下：

（本实例中，显示大写的 LOVE!,第二行显示 www.txsky.net）

```
module lcm(
CLOCK_50,
LCD_DATA,
LCD_RW,
LCD_RS,
LCD_EN,
LCD_BLON
);
input CLOCK_50;
output [7:0]LCD_DATA;
output LCD_RW;
output LCD_RS;
output LCD_EN;
output LCD_BLON;

reg [7:0]LCD_DATA;
reg LCD_RW ;
reg LCD_RS;
reg LCD_EN;
reg clk_1k=1'b0;

reg [20:0]counter=0;
reg [10:0]counter1=0;

assign LCD_BLON=1;

always@(posedge CLOCK_50)

    if(counter==25000)
        begin
            clk_1k<=~clk_1k;
            counter<=0;
        end
    else counter<=counter+1;

always@(posedge clk_1k)//
begin
    if(counter1<1023)
        counter1<=counter1+1;
```

```
case (counter1)
400:begin
    LCD_DATA<=8'b00111000; //0x38 设置显示模式
    LCD_RW<=1'b0;
    LCD_RS<=1'b0;
    end
401:LCD_EN<=1'b1;

410:                                //0x0c 开显示 关光标
    begin
        LCD_DATA<=8'b00001100;
        LCD_RW<=1'b0;
        LCD_RS<=1'b0;
        end
411: LCD_EN<=1'b1;

420:    begin                                //0x01 清屏幕
        LCD_DATA<=8'b00000001;
        LCD_RW<=1'b0;
        LCD_RS<=1'b0;
        end
421: LCD_EN<=1'b1;

430:begin                                //0x06 读或者写后 自动加 1
        LCD_DATA<=8'b00000110;
        LCD_RW<=1'b0;
        LCD_RS<=1'b0;
        end
431: LCD_EN<=1'b1;

440:begin                                //0xc0 定位到第二行
        LCD_DATA<=8'b11000000;
        LCD_RW<=1'b0;
        LCD_RS<=1'b0;
        end
441: begin
        LCD_EN<=1'b1;
        ack<=1;
        end

450: begin                                //write data  W
        LCD_DATA<=8'h57;
```

```
        LCD_RW<=1'b0;
        LCD_RS<=1'b1;
        end
451: LCD_EN<=1'b1;

460: begin                //write data  W
        LCD_DATA<=8'h57;
        LCD_RW<=1'b0;
        LCD_RS<=1'b1;
        end
461: LCD_EN<=1'b1;
470:   begin                //write data  W
        LCD_DATA<=8'h57;
        LCD_RW<=1'b0;
        LCD_RS<=1'b1;
        end
471: LCD_EN<=1'b1;
480: begin                //write data  .
        LCD_DATA<=8'h2E;
        LCD_RW<=1'b0;
        LCD_RS<=1'b1;
        end
481: LCD_EN<=1'b1;

490: begin                //write data T
        LCD_DATA<=8'h54;
        LCD_RW<=1'b0;
        LCD_RS<=1'b1;
        end
491: LCD_EN<=1'b1;

500:begin                //write data X
        LCD_DATA<=8'h58;
        LCD_RW<=1'b0;
        LCD_RS<=1'b1;
        end
501: LCD_EN<=1'b1;

510:begin                //write data S
        LCD_DATA<=8'h53;
        LCD_RW<=1'b0;
        LCD_RS<=1'b1;
        end
511: LCD_EN<=1'b1;
```

```
520: begin                //write data K
    LCD_DATA<=8'h4B;
    LCD_RW<=1'b0;
    LCD_RS<=1'b1;
    end
521: LCD_EN<=1'b1;

530: begin                //write data Y
    LCD_DATA<=8'h59;
    LCD_RW<=1'b0;
    LCD_RS<=1'b1;
    end
531: LCD_EN<=1'b1;
540: begin                //write data .
    LCD_DATA<=8'h2E;
    LCD_RW<=1'b0;
    LCD_RS<=1'b1;
    end
541: LCD_EN<=1'b1;

550: begin                //write data N
    LCD_DATA<=8'h4E;
    LCD_RW<=1'b0;
    LCD_RS<=1'b1;
    end
551: LCD_EN<=1'b1;

560: begin                //write data E
    LCD_DATA<=8'h45;
    LCD_RW<=1'b0;
    LCD_RS<=1'b1;
    end
561: LCD_EN<=1'b1;

570: begin                //write data T
    LCD_DATA<=8'h54;
    LCD_RW<=1'b0;
    LCD_RS<=1'b1;
    end
571: LCD_EN<=1'b1;

580: begin                //定位到第一行
    LCD_DATA<=8'h80;
    LCD_RW<=1'b0;
```

```

        LCD_RS<=1'b0;
    end
581: LCD_EN<=1'b1;

590: begin                //write data L
    LCD_DATA<=8'h4C;
    LCD_RW<=1'b0;
    LCD_RS<=1'b1;
    end
591: LCD_EN<=1'b1;
600: begin                //write data O
    LCD_DATA<=8'h4F;
    LCD_RW<=1'b0;
    LCD_RS<=1'b1;
    end
601: LCD_EN<=1'b1;
610: begin                //write data V
    LCD_DATA<=8'h56;
    LCD_RW<=1'b0;
    LCD_RS<=1'b1;
    end
611: LCD_EN<=1'b1;
620: begin                //write data E
    LCD_DATA<=8'h45;
    LCD_RW<=1'b0;
    LCD_RS<=1'b1;
    end
621: LCD_EN<=1'b1;
630: begin                //write data !
    LCD_DATA<=8'h21;
    LCD_RW<=1'b0;
    LCD_RS<=1'b1;
    end
631: LCD_EN<=1'b1;
default: LCD_EN<=1'b0;
endcase

end
endmodule

```