

# 实验四

## 基于 FPGA 的 LCD 显示系统设计

### 实验目的：

在熟悉Quartus基本操作、Verilog编程语言的基本规则、SignalTap的基本应用后，学习LCD显示系统的设计方法。

### 预备知识：

1. DE2-115 的基本使用方法；
2. Verilog 语言基础知识；

### 实验环境：

1. 实验人数 50 人，每 2 人一组，每组两台电脑
2. 电脑 50 台

### 实验内容：

利用试验箱LCD显示器（CFAH1602B-TMC-JP），学习LCD设计方法，用verilog语言编程，驱动LCD固定显示以下规定字符：

第一行： **FPGA DESIGNER**

第二行： **个人姓名的拼音**

### 实验原理：

1. LCD 显示器基本结构

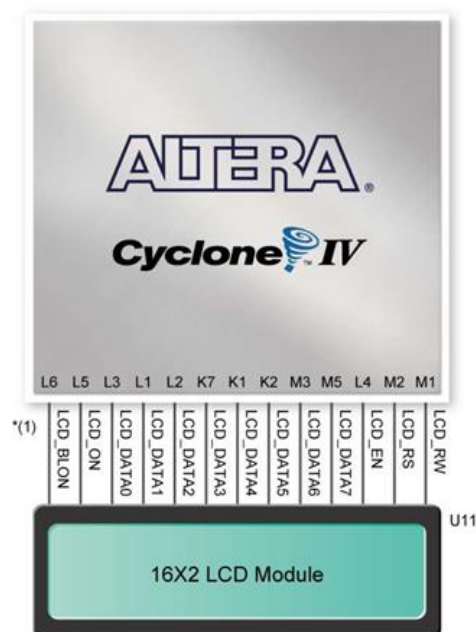


图1. Cyclone IV E FPGA芯片和LCD模块间连接示意图

表1 LCD 模块引脚配置说明

信号名	FPGA 引脚号	说明	I/O 标准
LCD_DATA[7]	PIN_M5	LCD Data[7]	3.3V
LCD_DATA[6]	PIN_M3	LCD Data[6]	3.3V
LCD_DATA[5]	PIN_K2	LCD Data[5]	3.3V
LCD_DATA[4]	PIN_K1	LCD Data[4]	3.3V
LCD_DATA[3]	PIN_K7	LCD Data[3]	3.3V
LCD_DATA[2]	PIN_L2	LCD Data[2]	3.3V
LCD_DATA[1]	PIN_L1	LCD Data[1]	3.3V
LCD_DATA[0]	PIN_L3	LCD Data[0]	3.3V
LCD_EN	PIN_L4	启用 LCD	3.3V
LCD_RW	PIN_M1	LCD 读/写选择, 0 = 写, 1 =读	3.3V
LCD_RS	PIN_M2	LCD 命令/数据选择, 0 = 命令 d, 1 =数据	3.3V
LCD_ON	PIN_L5	LCD 电源开/关	3.3V
LCD_BLON	PIN_L6	LCD 背光开/关	3.3V

在DE2-115 中使用的LCD 模块并不含背光单元，故而LCD\_BLON 信号在用户工程中的设定是无效的。

1602LCD显示的内容为16×2，即可以显示两行，每行16个字符，目前市面上字符液晶大多数是基于HD44780和SPLC780D液晶芯片，控制原理是完全相同的，因此基于HD44780和SPLC780D写的控制程序可以很方便地应用于市面上大部分的字符型液晶。本实验室所使用的是型号为CFAH1602B-TMC-JP的液晶，而其内部采用的正是HD44780液晶芯片。下图为LCD的原理框图。

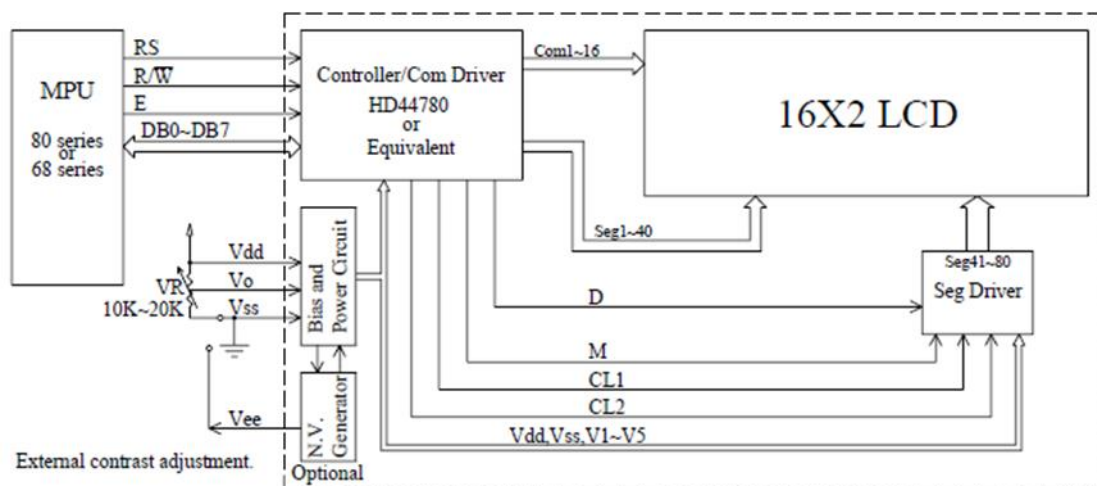


图2. LCD原理框图

从上图可以看出，LCD显示模块内嵌了一个HD44780的控制器，控制器有两个8位的寄存器，指令寄存器（IR）和数据寄存器（DR）。

指令寄存器IR只能由MPU执行写入操作，其存储的内容包括：

- 1) 指令码（例如显示器清零、光标移位等）；
- 2) 地址信息（DDRAM和CGRAM）

当地址信息被写入到IR中时，数据寄存器DR执行数据缓存操作，暂存的数据为准备写入DDRAM、CGRAM的数据，或者是从DDRAM、CGRAM读出的

数据。RS信号为寄存器选择信号，可以选择DR或IR。

## 2. LCD 显示器缓存分类

1602LCD中，除了DDRAM和CGRAM之外，还有一个CGROM，这几个存储器的主要区别描述如下：

- 1) **CGROM**：存储了一些标准的字符的字模编码，是液晶屏出厂时固化在控制芯片中的，用户不能改变其中的存储内容，只能读取调用，包含有标准的ASCII码、日文字符和希腊文字符。
- 2) **CGRAM**：留给用户，用以存储用户自己设计的字模编码。（8个字节，{0000\_X000~0000\_X111}）
- 3) **DDRAM**：是和屏幕显示区域有对应关系的一组存储器，其功能有点中转的性质。（80个字节）

为了便于理解，可以如下打一比方：CGROM和CGRAM中存储的字模信息相当于厨房中的食品，CGROM是厨房中现成的熟食，CGRAM是用户自行制作的菜肴，这些食品都要通过托盘DDRAM转移一下，才能送到餐桌上食用；类似的字模编码都要先被读取到对应的DDRAM中，经如上中转以后，屏幕的相应位置才显示出字符。

## 3. 基本操作时序

1602液晶的基本的操作分为以下四种：

- 状态字读操作：输入 RS=低、RW=高、EP=高； 输出：DB0~7读出为状态字；
- 数据读出操作：输入 RS=高、RW=高、EP=高； 输出：DB0~7读出为数据；
- 指令写入操作：输入 RS=低、RW=低、EP=上升沿； 输出：无；
- 数据写入操作：输入 RS=高、RW=低、EP=上升沿； 输出：无。

各类操作的时序图如下：

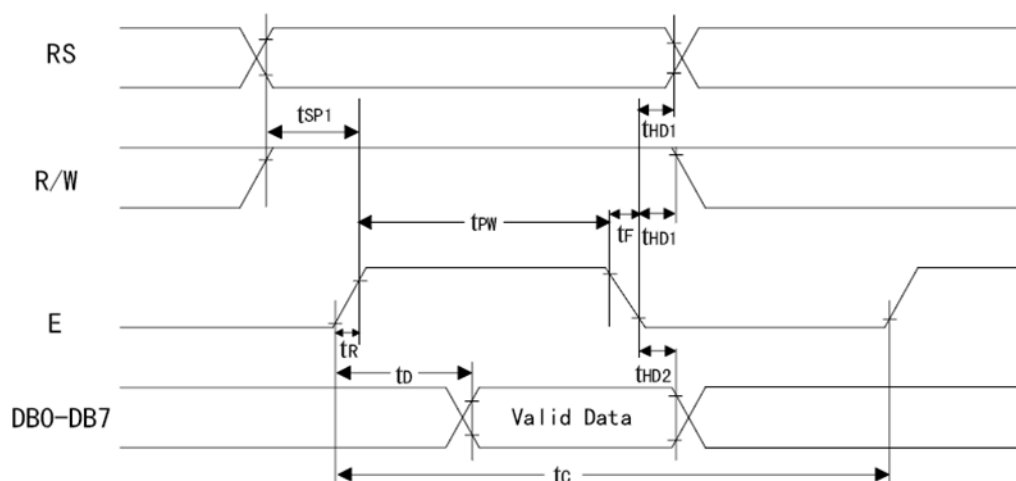


图3. 读操作的时序图

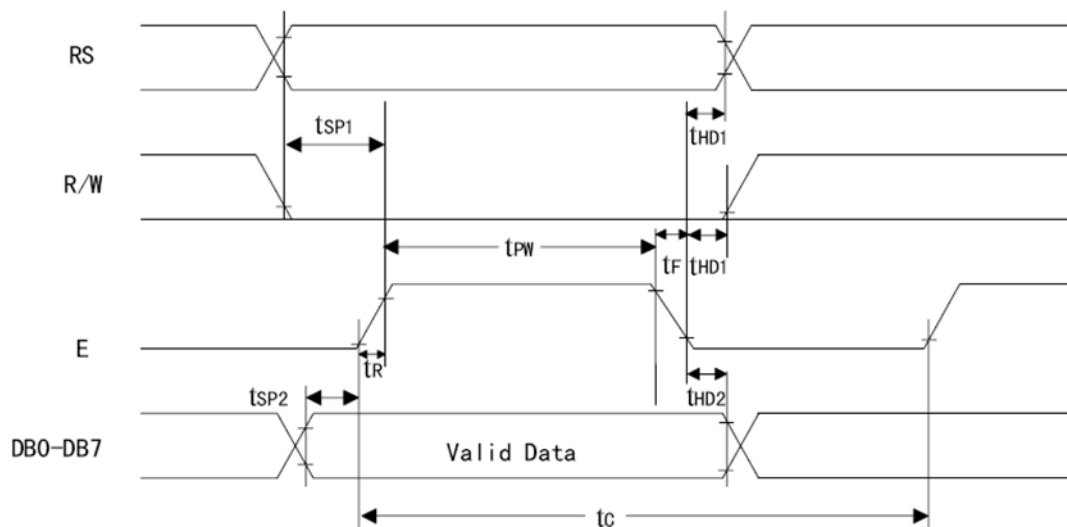


图4. 写操作的时序图

各类操作时序时间参数如下表：

表2. 时序时间参数

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	

#### 4. LCD 的指令

1602 LCD的控制器内置有80个byte的显存，而1602 LCD只有两行X 16个字符的显示区域，所以显存中有些地址是无法对应上LCD屏的，下图为显存地址对应图：

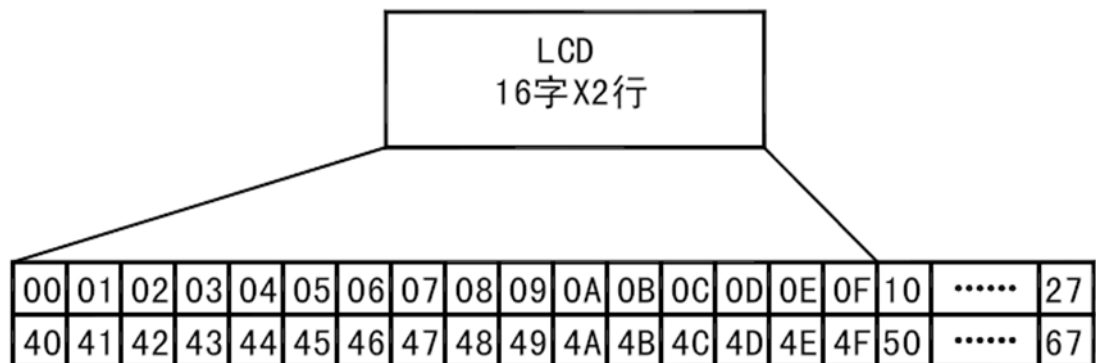


图5. 显存地址对应图

要显示字符时要先输入显示字符地址，也就是告诉模块在哪里显示字符，上

图是1602LCD的内部显示地址。例如第二行第一个字符的地址是40H，那么是否直接写入40H就可以将光标定位在第二行第一个字符的位置呢？这样不行，因为写入显示地址时要求最高位D7恒定为高电平1所以实际写入的数据应该是01000000B（40H）+10000000B(80H)=11000000B(C0H)。因此，第一行地址就必须加80H，而第二行地址就必须加C0H。

表3. 1602LCD控制指令

序号	指令	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0	E-Cycle
1	清显示	0	0	0	0	0	0	0	0	0	1	1.64ms
2	光标返回	0	0	0	0	0	0	0	0	1	*	1.64ms
3	置输入模式	0	0	0	0	0	0	0	1	I/D	S	40us
4	显示开/关控制	0	0	0	0	0	0	1	D	C	B	40us
5	光标或字符移位	0	0	0	0	0	1	S/C	R/L	*	*	40us
6	置功能	0	0	0	0	1	DL	N	F	*	*	40us
7	置字符发生存储器地址	0	0	0	1	字符发生存储器地址						40us
8	置数据存储器地址	0	0	1	显示数据存储器地址							40us
9	读忙标志或地址	0	1	BF	计数器地址							40us
10	写数到CGRAM或DDRAM	1	0	要写的数字内容								40us
11	从CGRAM或DDRAM读数	1	1	读出的数字内容								40us

- 指令1——清显示，指令码01H，光标复位到地址00H位置。
- 指令2——光标返回，光标返回到地址00H。
- 指令3——置输入模式，I/D：光标移动方向，高电平右移，低电平左移；S：屏幕上所有文字是否左移或右移，高电平表示有效，低电平无效。
- 指令4——显示开/关控制，D：控制整体显示的开/关，高电平为开显示，低电平为关显示；C：控制光标的开与关，高电平表示有光标，低电平表示无光标；B：控制光标是否闪烁，高电平闪烁，低电平不闪烁。
- 指令5——光标或字符移位，S/C：高电平时移动显示的文字，低电平时移动光标。
- 指令6——功能设置命令，DL：高电平时为8位总线，低电平时为4位总线；N：低电平时为单行显示，高电平时双行显示；F：低电平时显示5x7的点阵字符，高电平时显示5x10的点阵字符。
- 指令7——字符发生器RAM地址设置。
- 指令8——DDRAM地址设置。
- 指令9——读忙信号和光标地址，BF：为忙标志位，高电平表示忙，此时模块不能接收命令或者数据，如果为低电平表示不忙。
- 指令10——写数据。

➤ 指令11——读数据。

## 5. CGROM 字符映射表

Upper 4 bit Lower 4 bit		LLLL	LLH	LLHL	LLHH	LHLL	LHLH	LHHL	LHHH	HLLL	HLLH	HLHL	HLHH	HHLL	HHLH	HHHL	HHHH
LLLL	CG RAM (1)			0	a	P	'	P					一	2	3	e	p
LLH	(2)			!	1	A	D	a	a				二	ア	チ	△	あ
LLHL	(3)			"	2	B	R	b	r				〒	イ	リ	×	β
LLHH	(4)			#	3	C	S	c	s				┘	ウ	チ	ε	ω
LHLL	(5)			\$	4	D	T	t	t				√	エ	ト	ト	μ
LHLH	(6)			%	5	E	L	e	u				＝	オ	★	∫	0
LHHL	(7)			&	6	F	V	v	v				ヲ	カ	ニ	3	ρ
LHHH	(8)			'	7	G	W	w	w				フ	キ	ア	ウ	π
HLLL	(1)			(	8	H	X	x	x				ノ	ク	ホ	リ	ノ
HLLH	(2)			)	9	I	Y	y	y				ウ	ケ	ル	ル	μ
HLHL	(3)			*	0	J	Z	z	z				エ	コ	ノ	ル	π
HLHH	(4)			+	1	K	C	k	c				★	サ	ヒ	ロ	°
HHLL	(5)			.	<	L	#	1	1				ホ	セ	フ	フ	π
HHLH	(6)			—	=	M	J	m	j				ユ	ズ	ノ	ホ	+
HHHL	(7)			.	>	N	ノ	n	ノ				ヨ	セ	ホ	ノ	
HHHH	(8)			/	7	O	ノ	o	ノ				ウ	ウ	マ	°	π

6. 液晶显示流程

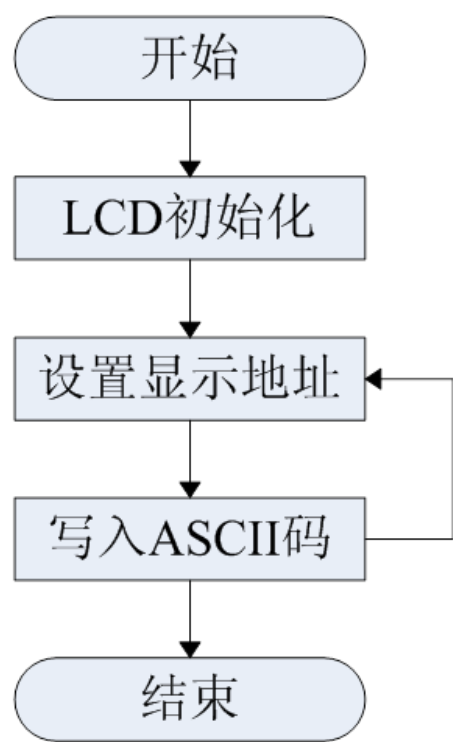


图 6. 液晶显示流程图

7. LCD 设计顶层框图

LCD 设计顶层框图如图 7 所示。

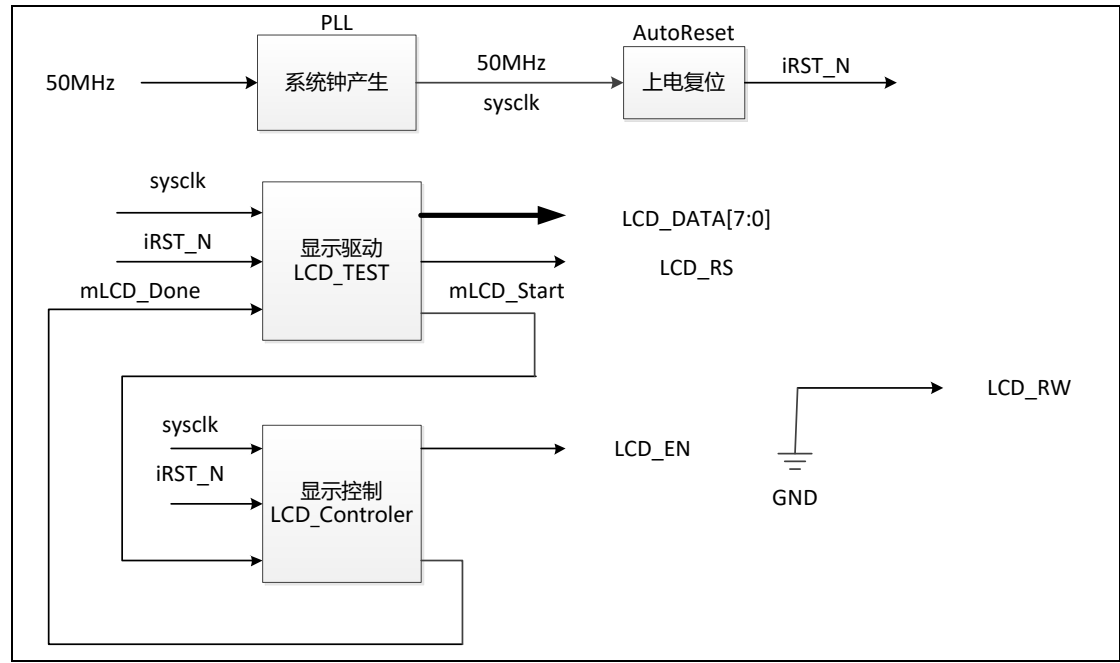


图7 LCD显示顶层结构

各模块功能描述如下：

- 系统钟产生：利用 PLL 产生 50MHz 时钟；
- 上电复位：产生全局上电复位脉冲；
- 显示驱动 (LCD\_TEST.v)：完成显示内容设置及驱动功能；
- 显示控制 (LCD\_Controller.v)：完成显示器时序控制功能；

## 时钟设计实验步骤：

1. 启动系统生成器, 生成 project;
2. 构建模块并连接;
3. 确定显示内容
4. 编译;
5. 下载;
6. 利用 signalTap 调试、验证关键功能。

## 实验报告：

1. 简述实验步骤;
2. 在实验原理部分简单描述各模块的流程图

具体要求：

- 1) 画出 LCD\_TEST.v 程序段中的状态机转换图；

状态机由状态寄存器和组合逻辑电路构成，能够根据控制信号按照预先设定的状态进行状态转移，是协调相关信号动作、完成特定操作的控制中心。状态机简称为 FSM (Finite State Machine)，主要分为 2 大类：

- 第一类，若输出只和状态有关而与输入无关，则称为 Moore 状态机；
- 第二类，输出不仅和状态有关而且和输入有关系，则称为 Mealy 状态机；

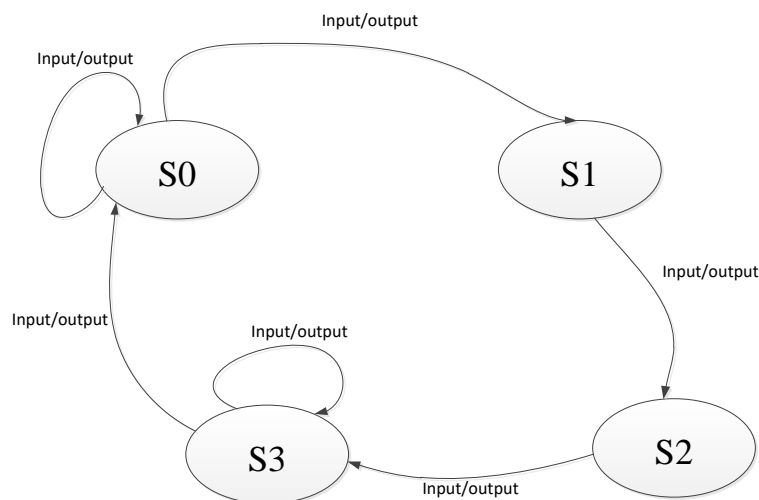
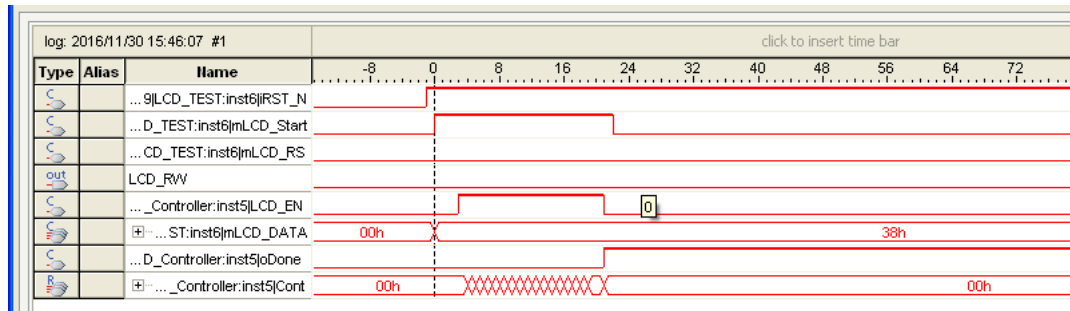




图 8 状态机流程

- 2) 对代码中注释有“描述此行代码含义”的行进行简单的分析；（可对应表 3. 1602LCD 控制指令分析）
- 3) 画出 LCD\_Controller.v 程序段中状态机转换图。
3. 利用 SignalTap 分析重点时序；  
时序分析图如下图，在报告中对时序进行解释。



4. 提交设计顶层文件的截图；
5. 提交显示器的照片



6. 对本次实验做出心得体会。