**一、概念介绍**

1.1、什么是LCD

LCD （ Liquid Crystal Display 的简称）液晶显示器。

1888年,澳大利亚科学家莱尼茨尔，合成了一种奇怪的有机化合物，它有两个熔点.把它的固态晶体加热到145℃时，便熔成液体，只不过是浑浊的，而一切纯净物质熔化时却是透明的。如果继续加热到175℃时，它似乎再次熔化，变成清澈透明的液体。后来，德国物理学家列曼把处于“中间地带”的浑浊液体叫做晶体。它好比是既不象马，又不象驴的骡子,所以有人称它为有机界的骡子。

液晶在正常情况下，其分子排列很有秩序，显得清澈透明，一旦加上直流电场后，分子的排列被打乱，部分液晶分子进行旋转，而旋转会影响透光，颜色加深，因而能显示数字和图象。

LCD 的构造是在两片平行的[玻璃](https://baike.baidu.com/item/%E7%8E%BB%E7%92%83/287)基板当中放置液晶盒，下基板玻璃上设置TFT（薄膜晶体管），上基板玻璃上设置彩色滤光片。在LCD下方存在背光源，而按照背光源的不同，LCD可以分为CCFL(冷阴极荧光灯管)背光液晶屏和LED背光液晶屏。

1.3、LCD液晶屏分类：

LCD液晶屏按显示技术来分：点阵屏，断码屏 ，字符屏，彩屏

显示类型来分：正显 负显 全透 半透 放射

按材质来分：TN(扭曲向列型) STN(超扭曲向列型) TFT(薄膜晶体管) IPS等

我们主要了解一下TFT和IPS：

TFT是薄膜晶体管的缩写。TFT式显示屏是各类笔记本电脑和台式机上的主流显示设备，也普遍应用于中高端彩屏手机中，该类显示屏上的每个液晶像素点都是由集成在像素点后面的薄膜晶体管来驱动，因此TFT式显示屏也是一类[有源矩阵](https://baike.baidu.com/item/%E6%9C%89%E6%BA%90%E7%9F%A9%E9%98%B5)[液晶](https://baike.baidu.com/item/%E6%B6%B2%E6%99%B6)显示设备。

IPS屏幕就是基于TFT的一种技术，其实质还是TFT屏幕，俗称“Super TFT”。IPS面板最大的特点就是它的两极都在同一个面上，而不像其它液晶模式的电极是在上下两面。

**二、LCD硬件原理**

2.1 lcd控制器支持三种接口：

1) 传统的RGB-LCD接口

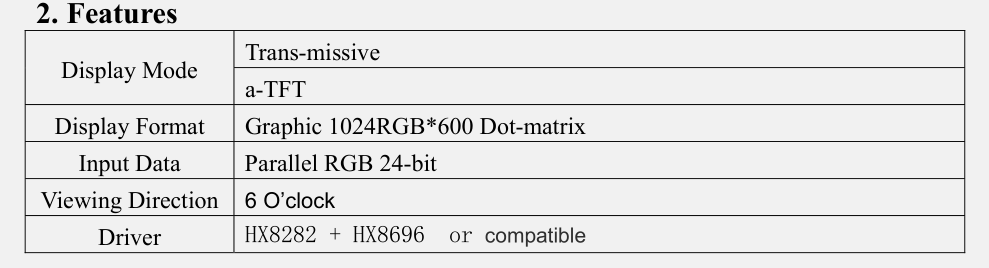
2) I80接口

3) YUV接口

RGB-LCD与I80的区别在什么地方？

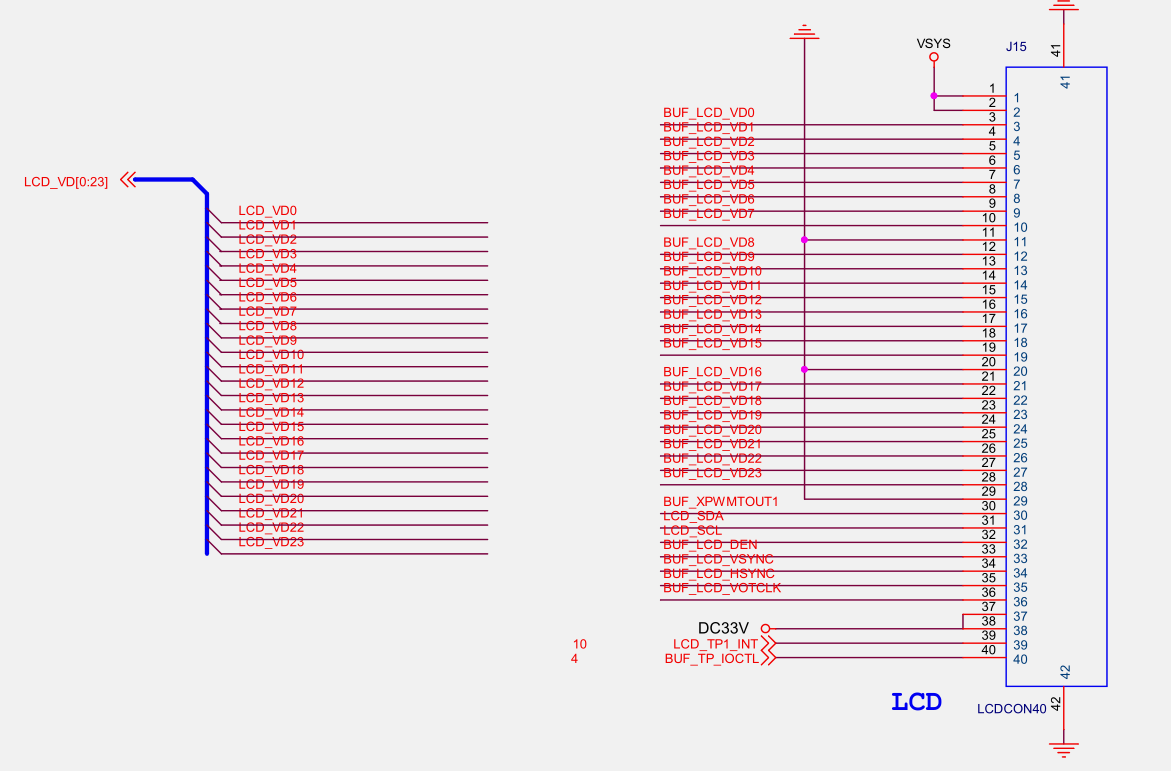
RGB-LCD把显示的数据放在了自己的系统内存中，然后通过dma的方式，将数据传送给lcd控制器，然后lcd控制器自己把数据发给lcd去显示。但I80没有自己的系统内存，因为这种模式的lcd有自己的显存。RGB-LCD模式的速度要比I80模式快。

查看手册JT70DT5012-ZH：



我们可以看到当前我们这个屏幕支持的是RGB接口。

2.2 看原理图：弄清楚exynos4412和lcd外设接口对接



BUF\_LCD\_VOTCLK:产生lcd的时钟信号

BUF\_LCD\_DEN:数据使能引脚

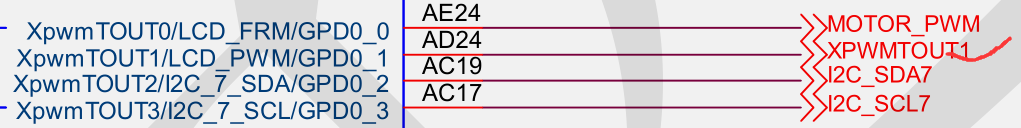
BUF\_LCD\_VD0 - BUF\_LCD\_CD23:数据引脚

BUF\_LCD\_HSYNC:水平方向同步信号

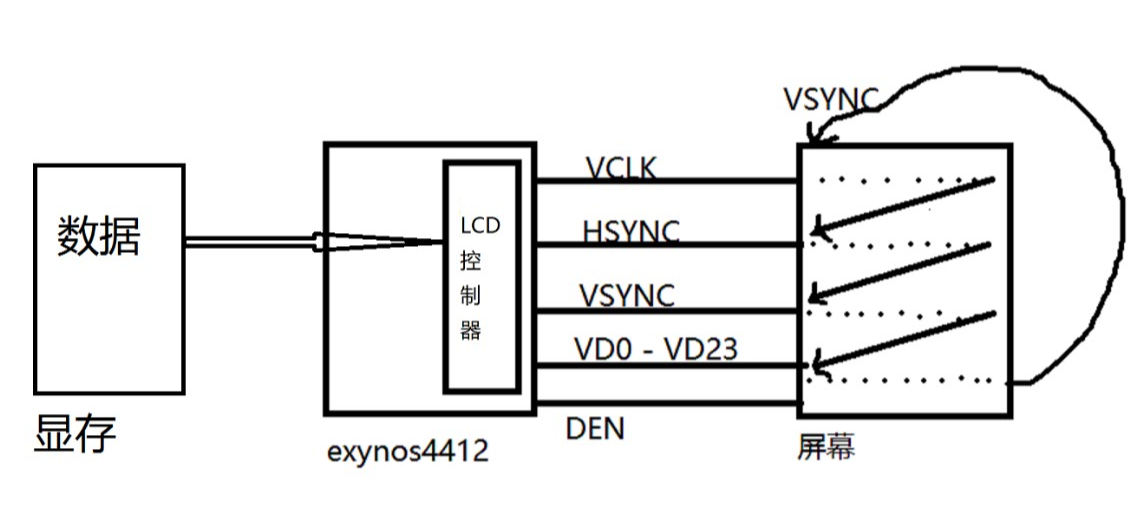
BUF\_LCD\_VSYNC:垂直方向同步信号

BUF\_XPWMTOUY1: 通过pwm控制背光灯

背光灯核心板原理图如下：



2.3 数据如何传输原理



屏幕需要的数据是存放在显存上的，每个像素点的信息都是由LCD控制器从显存中读取数据，交给VD0-VD23数据线进而显示到屏幕上的。在显示像素数据时速度不能太快也不能太慢，所以这个速率要由时钟VCLK来控制，LCD控制器每产生一次VCLK就会将一个像素点显示到屏幕上。当像素点显示到一行的最后一个位置时，LCD控制器会发出一个HSYNC水平方向同步信号将像素点重新定位到下一行的起始位置。当像素点将屏幕最后一行信息也显示完成后，LCD会发出一个VSYNC垂直方向同步信号重新将像素点定位到第一行起始位置。但是要注意DEN引脚被使能才会将像素点数据和颜色显示到屏幕上，如果DEN没有被使能则对应像素点就是黑色。

显存：soc在内存中挑选一段内存，然后通过配置将lcd控制器和这一段内存（显存）连接起来构成一个映射关系，

一旦建立映射关系之后，lcd控制器就会自动从显存中读取像素的数据传输给lcd驱动器。

总结：我们只需要把要显示的图像的像素数据丢到显存中，硬件自动响应（在屏幕上就能自动看到显示的图像）。

**三、LCD寄存器配置**

3.1、基本引脚功能设置

GPF0CON:控制水平同步信号，垂直同步信号，数据使能位，时钟引脚，VD0-VD3

GPF1CON:控制数据VD4-VD11

GPF2CON:控制数据VD12-VD19

GPF3CON:控制数据VD20-VD23

3.2、背光灯控制

P1767

Using the display controller data, you can select one of the above data paths by setting LCDBLK\_CFG Register (0x1001\_0210). For more information, refer to the "System Others" manual

三星手册12章

LCDBLK\_CFG：设置FIMD(完全交互式移动显示设备)为bypass模式

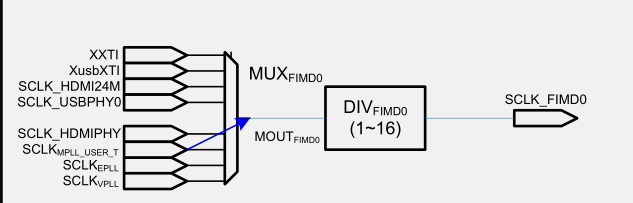
LCDBLK\_CFG2：使能PWM输出

GPD0CON：1 << 1 设置输出模式

GPD0DAT：控制背光灯亮灭

3.3、时钟源配置

P457



CLK\_SRC\_LCD0 选择SCLK\_MPLL\_USER为时钟源——MPLL 800MHZ

CLK\_DIV\_LCD 配置LCD时钟分频值，这里不分频

3.4、LCD时钟配置

VIDCON0：14 << 6 用于设置VCLK的值

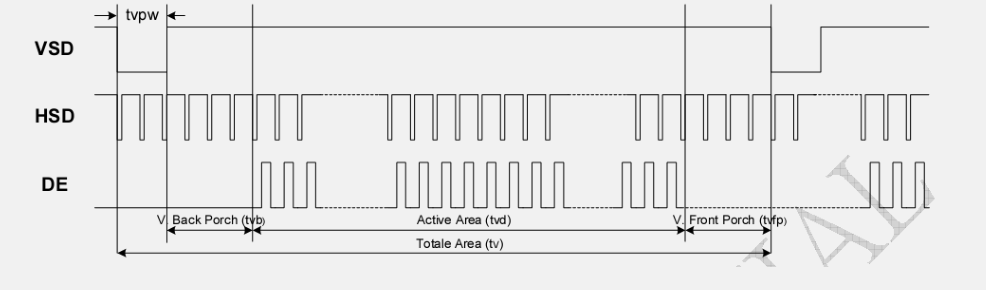
VCLK = FIMD\_SCLK/(CLKVAL+1)

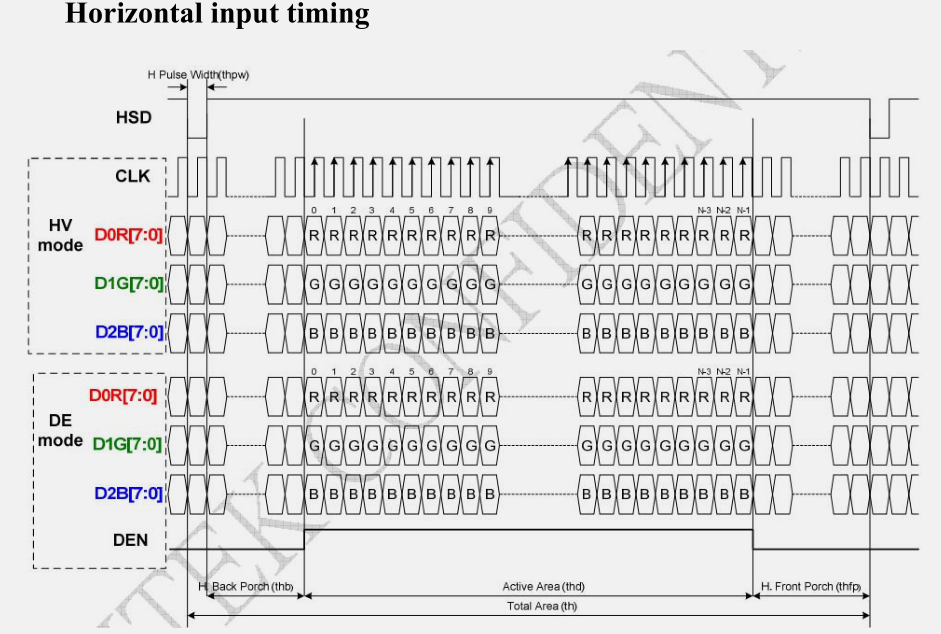
液晶屏手册11页，DCLK典型值为51.2，最大值为67.2

3.5、极性配置

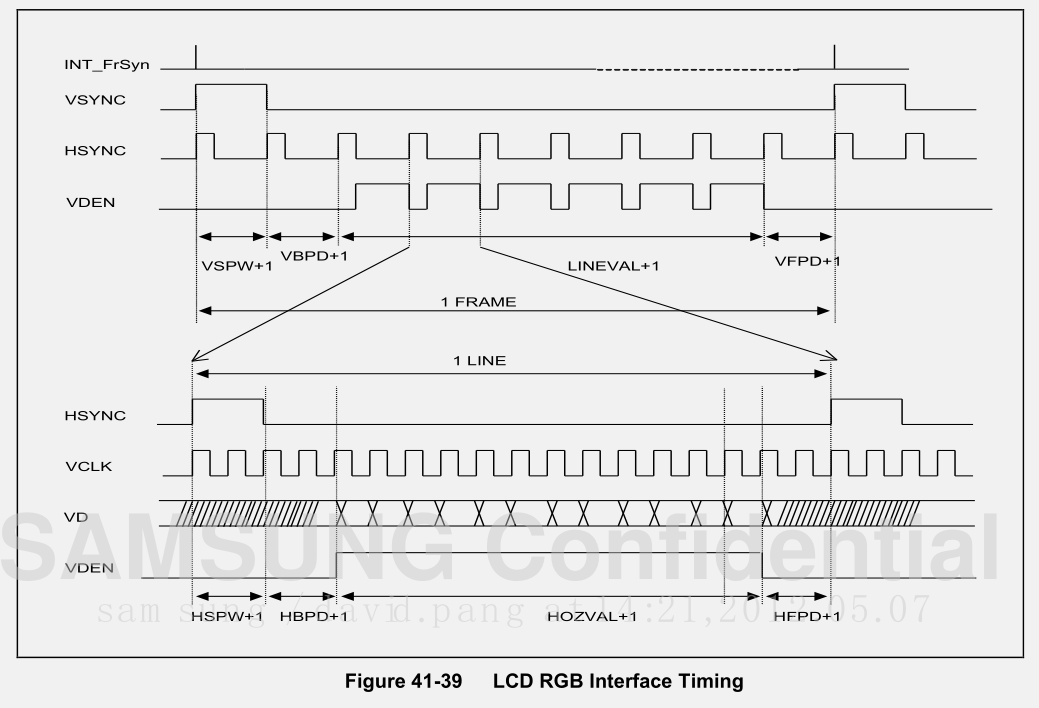
VIDCON1：

液晶屏手册如下：

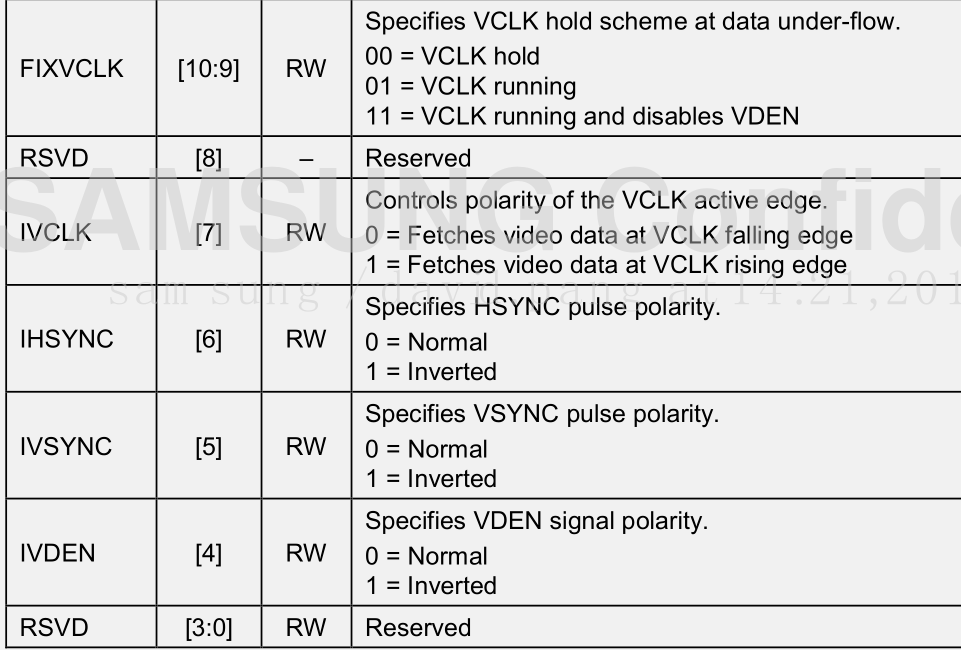




LCD控制器手册如下：



lcd控制器的管脚极性必须和lcd屏幕的参数接近或者吻合才能匹配。



LCD控制器的IVDEN、IVSYNC、IHSYNC等极性如果和LCD屏幕的极性相同则设置为normal，如果相反则设置为inverted(反向)

VDEN:两者都是高电平有效，所以设置为0

VSYNC：LCD控制器是高电平有效，LCD屏幕是低电平有效，所以设置为1

HSYNC：LCD控制器是高电平有效，LCD屏幕是低电平有效，所以设置为1

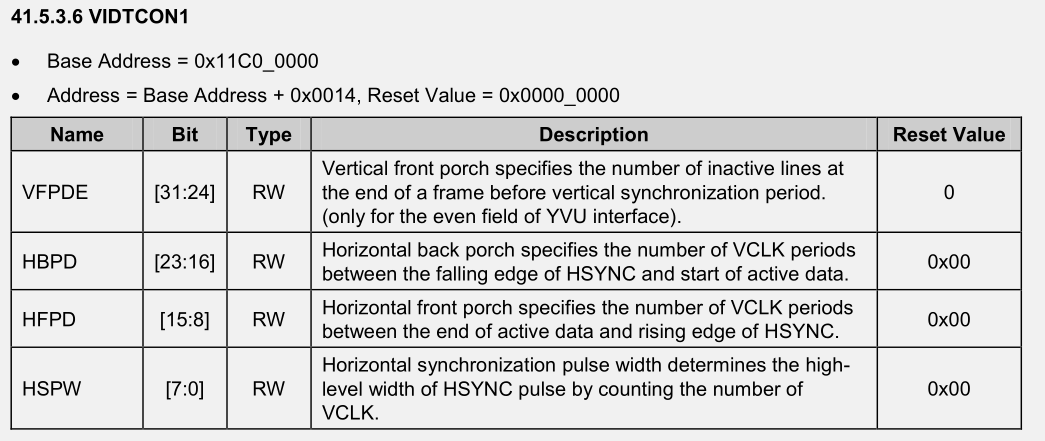
VCLK：时序图上看不出来上升沿还是下降沿有效，所以要自己试。我们当前是下降沿有效

FIXVCLK：保持VCLK有效

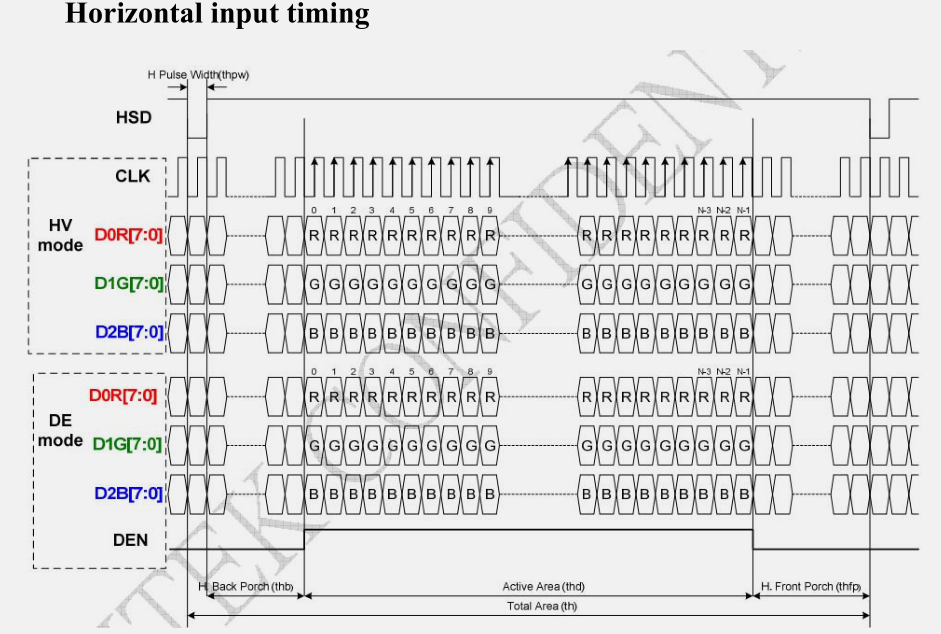
VIDCON2：[15:14]位有提示，NOTE: This bit should be set to 1，其他位不需要操作

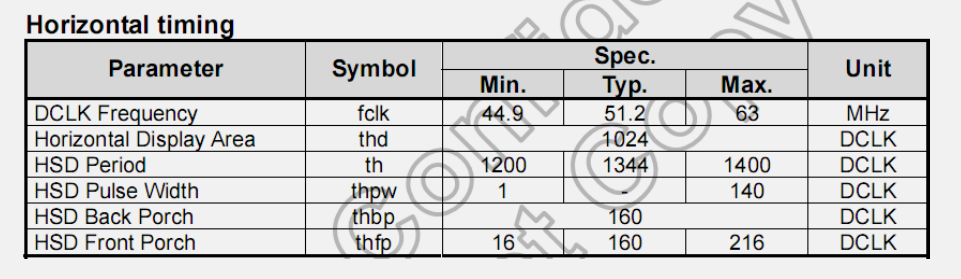
3.6、时序配置

LCD控制器手册：



LCD手册：





HSPW：水平同步脉冲宽度

这个值的设置首先查看上图中的thpw，然后在LCD手册中查找thpw的规定值，发现这个值在1到140之间。

HFPD：水平同步信号前端

查看LCD手册找到thfp，这个值的使用方法:最小值16，典型值160，最大值216

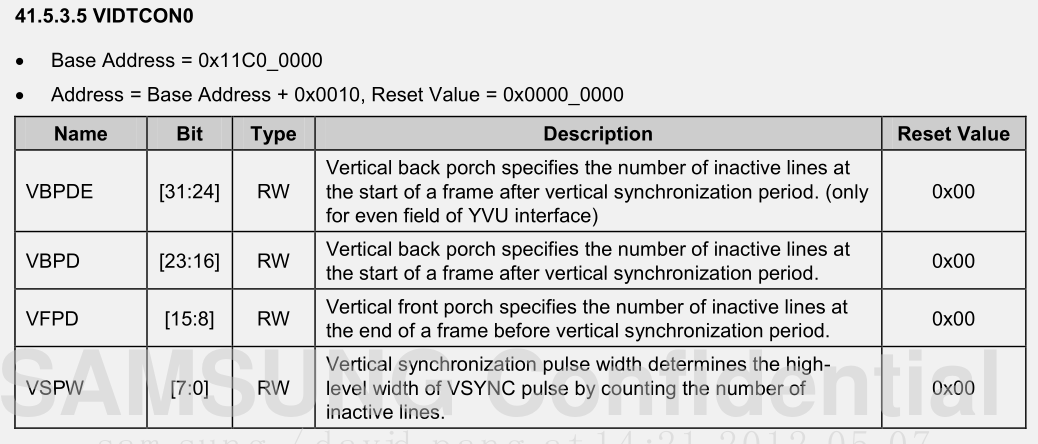
HBPD：水平同步信号后端

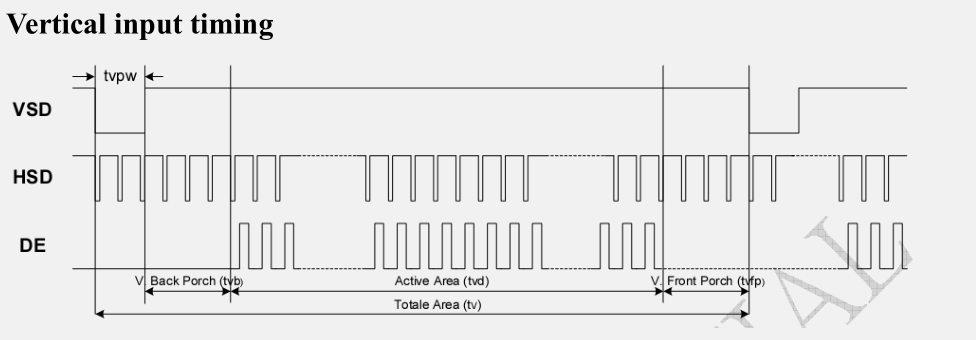
查看LCD手册找到

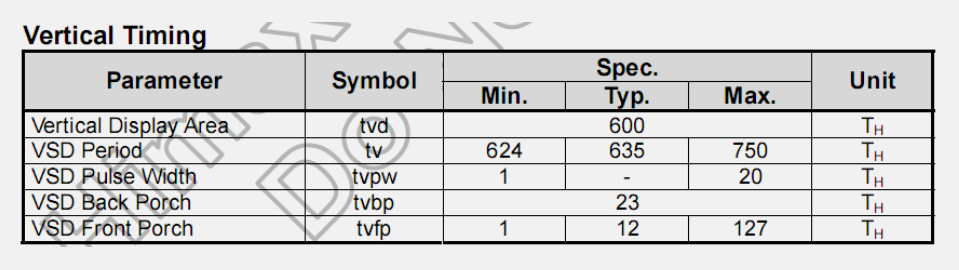
th = thpw + thb + thd + thfp

1344 40 1024 160 所以当前thb大概值为120

——————————————————————————————————————————————————







VSPW：垂直同步脉冲宽度

上图中的tvpw中1到20选一个值

VFPD：垂直同步脉冲前端

上图中的tvfp，最小值1，典型值12，最大值127

VBPD：垂直同步脉冲后端

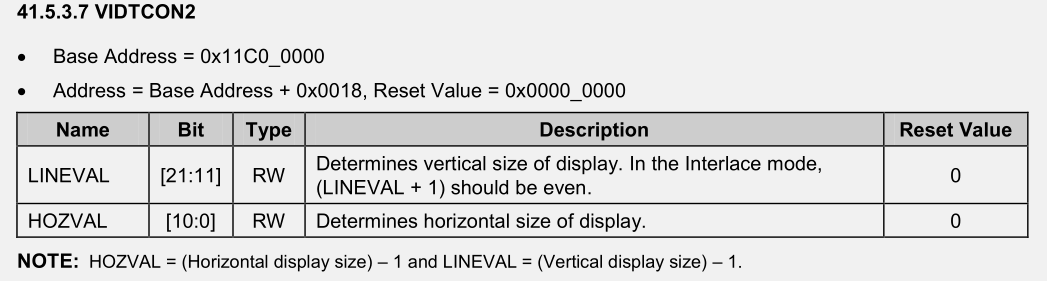
上图中的tvb，但是需要计算这个值

tv = tvpw + tvb + tvd + tvfp

635 9 600 12

算出tvb大概值为14

—————————————————————————————————————————————



控制屏幕的分辨率

—————————————————————————————————————————————

WINCON0：

假如现在我们显示的是一张16bits（R:5：G6：B5）bpp, non-pallette的位图。那一个pixel就占用了两个bytes，假设Lcd显示屏的输出像素是1024\*600，这样每一帧实际内存占用大小就是1024\*600\*2个字节。

[2:5] 控制每个像素点的色深，这里我们选择0101 = 16 BPP (non-palletized, R:5-G:6-B:5)

[9] Selects DMA Burst Maximum Length

[16] 使能半字交换控制位——因为上面设置的是16BPP，每个像素占用2字节

SHADOWCON0：1 << 0 使能第一个窗口

WIN0MAP：指定颜色值

VIDOSD0A：控制屏幕左上角的x、y轴坐标

VIDOSD0B：控制屏幕右下角的x、y轴坐标

VIDOSD0C：指定显示区域的大小 （注意单位）

VIDW0nADD0Bn：W00表示windows0，ADD0表示显存的起始地址

VIDW0nADD1Bn：ADD1表示显存终止地址

VBASEL = VBASEU + (PAGEWIDTH + OFFSIZE) \*(LINEVAL + 1)如果支持横向滚动则需要使用OFFSIZE

VIDW0nADD2：控制x轴方向上的字节数(1024 \* 16 / 8)，以及是否支持横向滚动

最后不要忘记添加：不然不会正常出现图像

writel(readl(vidcon0) | 3,vidcon0);

writel(readl(wincon0) | 1,wincon0);