freescale平台1920x720@60 分离屏模组客户项目

存在的问题：

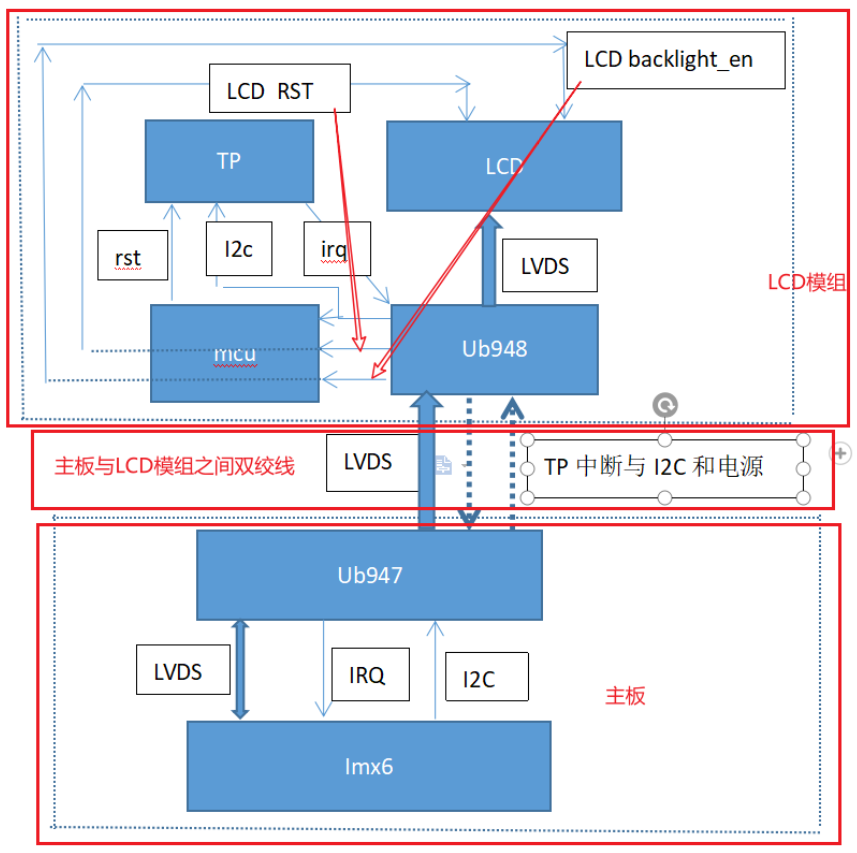
系统开机时概率性存在花屏。

除了花屏之外TP的中断，LCD复位，背光使能等的透传OK。

因项目中ub947和ub948针对lvds通路使用芯片默认，并没有配置针对LVDS通路的寄存器，即芯片上电LVDS通路直接导通。

想请教按照我们以下的设计方案，ub947和ub948的LVDS通路的寄存器部分需要配置吗？？要配置哪些寄存器，因为从ub947和ub948芯片手册上看并没有对应于LVDS通路的寄存器设置。怎么配置。

1. 系统硬件设计框图



* 1. 主板部分

Ub947链接到imx6的双路lvds接口上。

ub947需要实现TP的IRQ经过ub948透传到imx6输入。

Ub947需要实现imx6链接的I2C透传到TP以及MCU.

* 1. 双绞线电缆

电路上已经有ub947的GPIO0-3的GPIO模式管脚，但是实际并没使用ub947的GPIO0-3的透传，只使用了中断透传。其它部分为LVDS和I2C、中断以及电源的双绞电缆线链接主板和lcd模组

* 1. LCD模组
     1. 需要实现MCU与MPU之间的I2C通信驱动
     2. 需要实现TP驱动，TP的复位需要通过MCU与MPU之间的驱动接口给MCU发通信协议，由MCU复位TP。TP的中断需要ub948和ub947再到imx6透传。Ub947使用GPIO\_3作为中断输入。
     3. Ub948驱动，需要将ub948的GPIO0和GPIO1设置成DGPIO模式，由寄存器控制GPIO0和GPIO1拉高拉低来通知MCU，由MCU间接给LCD复位和使能背光。
     4. Imx6的LVDS输出需要根据屏的数据手册，设置imx6双路输出1920x720的LVDS信号
     5. Imx6到ub947，再到ub948使用芯片默认配置，即可默认上电，LVDS就通了。

1. 涉及到通路的电路原理图

主板的ub947部分：

请参考附件中的ub947原理图.pdf

其中ub947上的GPIO\_3用作中断管脚输入

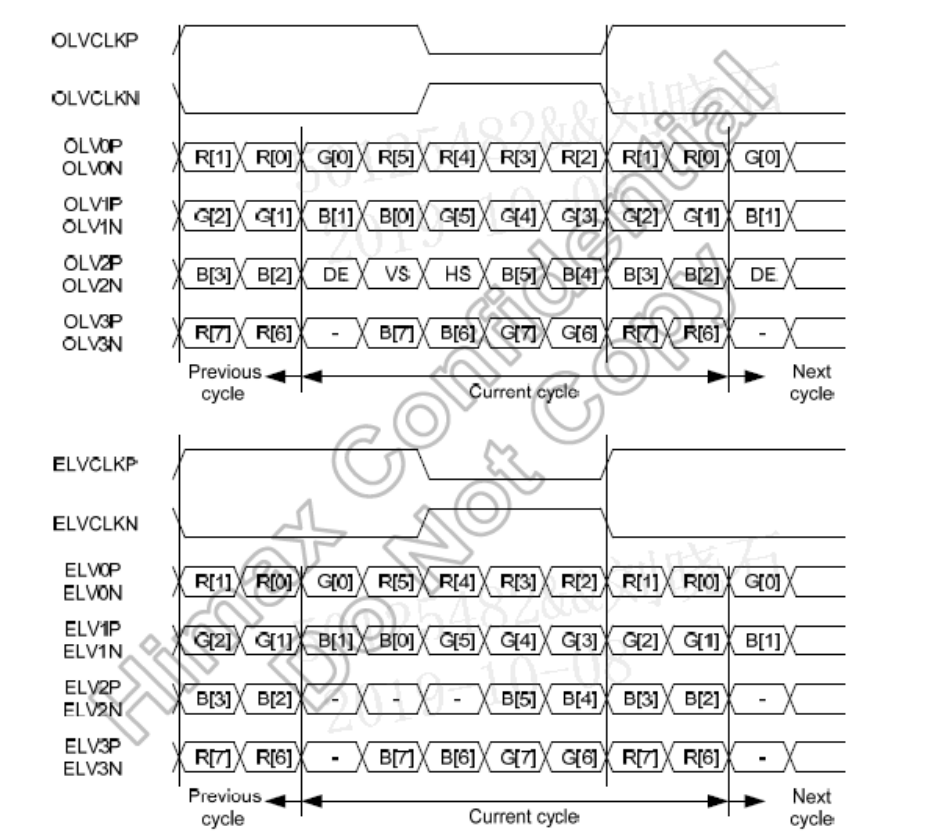
LCD模组的ub947部分：

Ub948原理图看附件ub948原理图.png

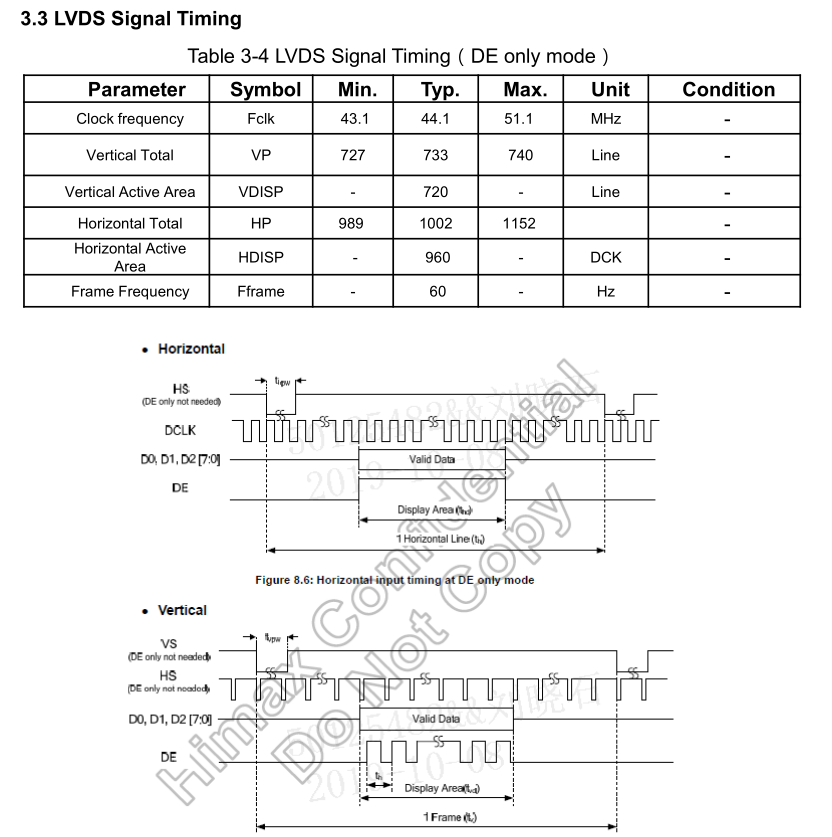
Ub948的GPIO0和GPIO作为lcd的RST和背光使能管脚用，LVDS直接到LCD屏

LCD屏的参数和时序：

时序图：



参数表：



1. **ub947和ub948驱动的实现的寄存器参数配置**

**3.1 ub947寄存器设置**

ret=disp947\_reg\_write(pdata->disp947\_client,0x03,0xDA);//设置I2C透传

ret=disp947\_reg\_write(pdata->disp947\_client,0x17,0xDE);//设置I2C透传

ret=disp947\_reg\_write(pdata->disp947\_client,0x0F,0x03);//设置947 GPIO3 为input 设置GPIO3中断管脚为输入

ret=disp947\_reg\_read(pdata->disp947\_client,0x0F);

printk("[disp947.37]:read disp947 0x0F= 0x%x\n",ret);

ret=disp947\_reg\_write(pdata->disp947\_client,0x0F,0x03);

msleep(2);

ret=disp947\_reg\_write(pdata->disp947\_client,0x0F,0x03);

msleep(3);

ret=disp947\_reg\_write(pdata->disp947\_client,0xc6,0x21); //设置中断透传

**3.2 ub947的寄存器设置**

**ub948驱动初始化不开背光和对LCD复位,由系统启动之后中间件层调用接口来开：**

result = disp948\_reg\_write(g\_client,I2C\_CONTROL\_1,0xDE);

reg\_val =i2c\_smbus\_read\_byte\_data(client, GENERAL\_CONFIGURATION\_1);

reg\_val = reg\_val|0x08;

result = disp948\_reg\_write(g\_client,GENERAL\_CONFIGURATION\_1,reg\_val);

printk("[disp948]:read chip ID ok equal to 0x%x \n",reg\_val);

//D\_GPIO3 LCD\_EN

reg\_val = i2c\_smbus\_read\_byte\_data(client, 0x34);

reg\_val |= 0x02;

result = disp948\_reg\_write(g\_client,0x34,reg\_val);//设置GPIO为控制DGPIO模式

printk("[disp948]:write reg\_val:0x%x\n",reg\_val);

result = disp948\_reg\_write(g\_client,0x1F,0x01);//ub947的GPIO3为开光使能管脚：09---开；01---关

//add D\_ Gpio0 default output 0

reg\_val=i2c\_smbus\_read\_byte\_data(client, 0x1D); //ub948的GPIO0为LCD为复位管脚 D\_GPIO0

reg\_val = ((reg\_val&0xF0) |0x01);

result=disp948\_reg\_write(g\_client,0x1D,reg\_val);//enable D\_GPIO0 output 0

//end add

以上两个接口由中间件层调用去设置，ub948本身驱动的寄存器初始化并不对LCD复位和开背光。

3.3 LVDS通路的寄存器设置

目前针对该项目ub947和ub948并没有专门针对UB947和ub948的LVDS通路的寄存器专门的设置，使用芯片默认。即可给ub947和ub948上电之后，LVDS通路就已经OK了。