

LT8911EX & LT8911EXB

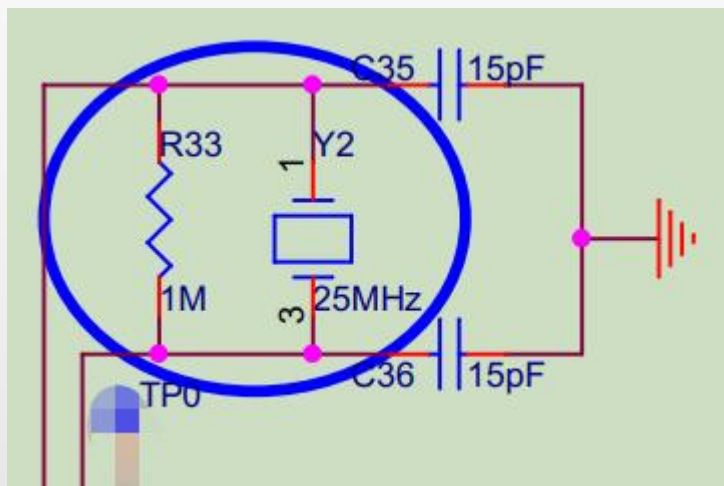
软/硬件方面调试

LT8911EX / LT8911EXB

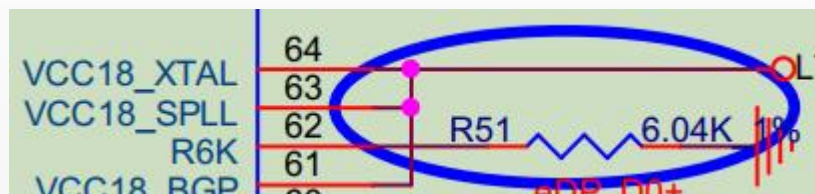
- 由于芯片内部PLL 带宽的限制，LT8911EX/LT8911EXB 不支持超过200M pixel CLK的eDP屏。如果要点pixel clk超过200M 的eDP屏，可以用LT9711.
- 像2160x1440、2240x1400分辨率eDP屏在配置输入信号的屏参的时候，Pixel CLK不要超过200M。
- LT8911EX : 单/双 port LVDS 转 eDP
- LT8911EXB: MIPI 转 eDP

1.1、硬件review:

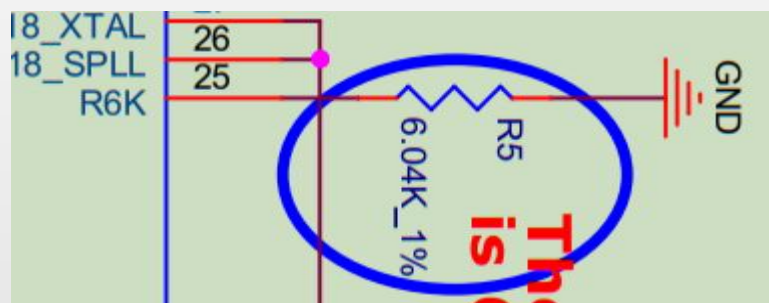
晶振选择25M



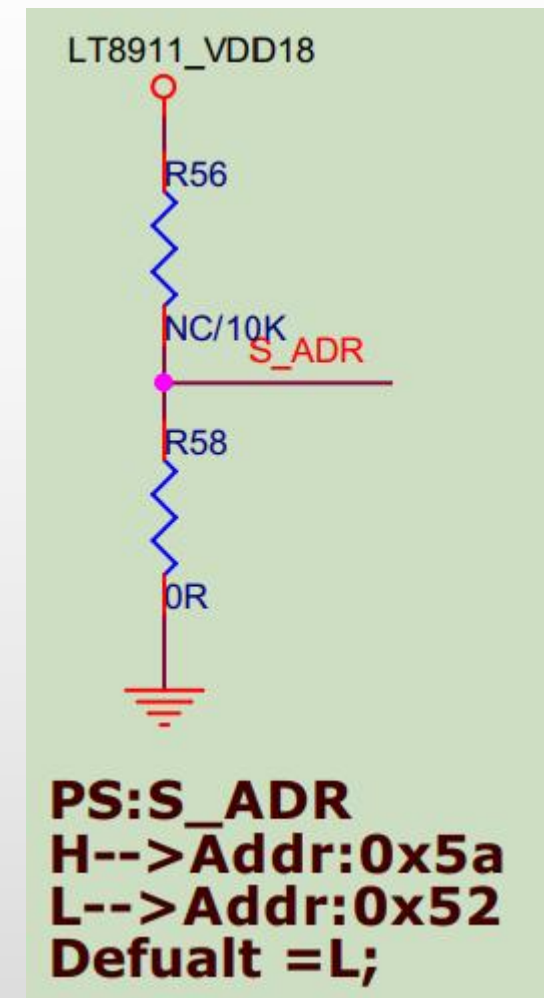
LT 8911EX 的62脚:



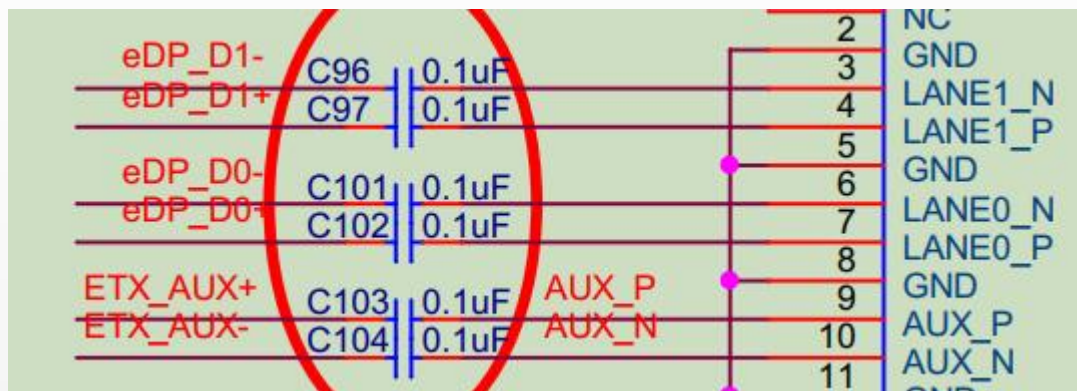
LT8911EXB的25脚:



IIC地址选择脚的电平，
寄存器配置设置的IIC地址要跟
这里对应，这管脚不能悬空。

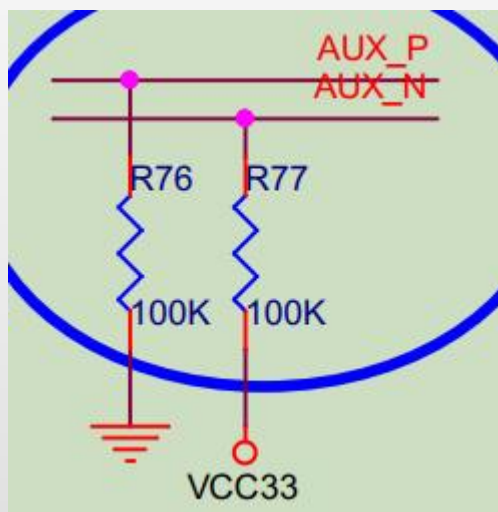
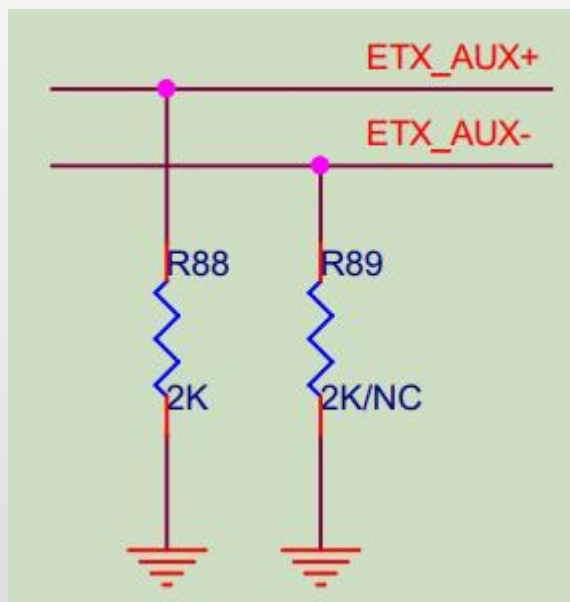


1.2、硬件review:



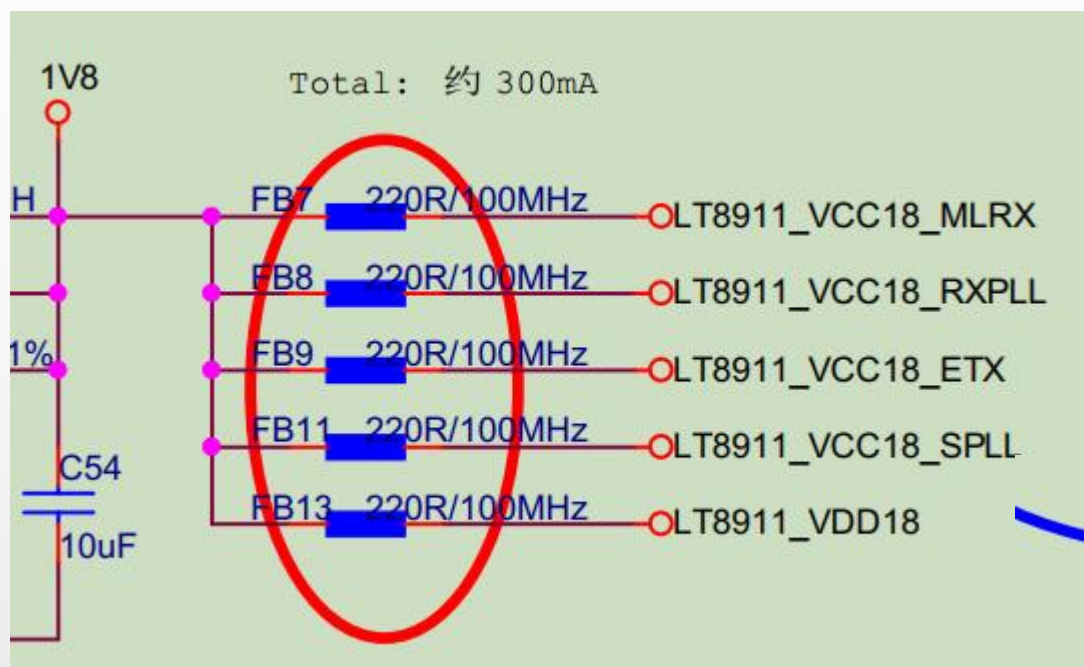
1、eDP输出要接0.1uF电容到屏端，靠近屏线座子摆放。

2、在 ETX_AUX+ 上加 2K 电阻下拉到 GND，ETX_AUX-上预留 2K 电阻下拉到 GND，是为了提高芯片内部 AUX RX 抗干扰能力。



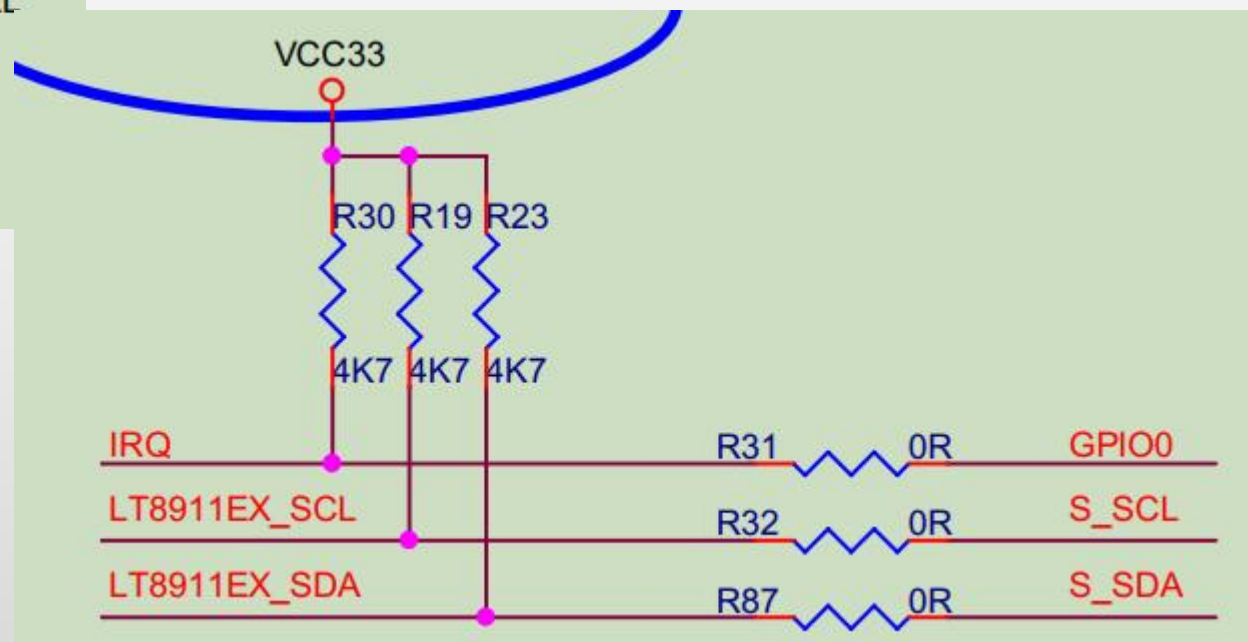
3、AUX_P 和AUX_N 分别接100K电阻到GND 和 3.3V (AUX电平翻转)，有些屏内部有接上下拉，这两个100K电阻接不接都可以；有些屏内部没有上下拉，就需要接这两个100K电阻。

1.3、硬件review:



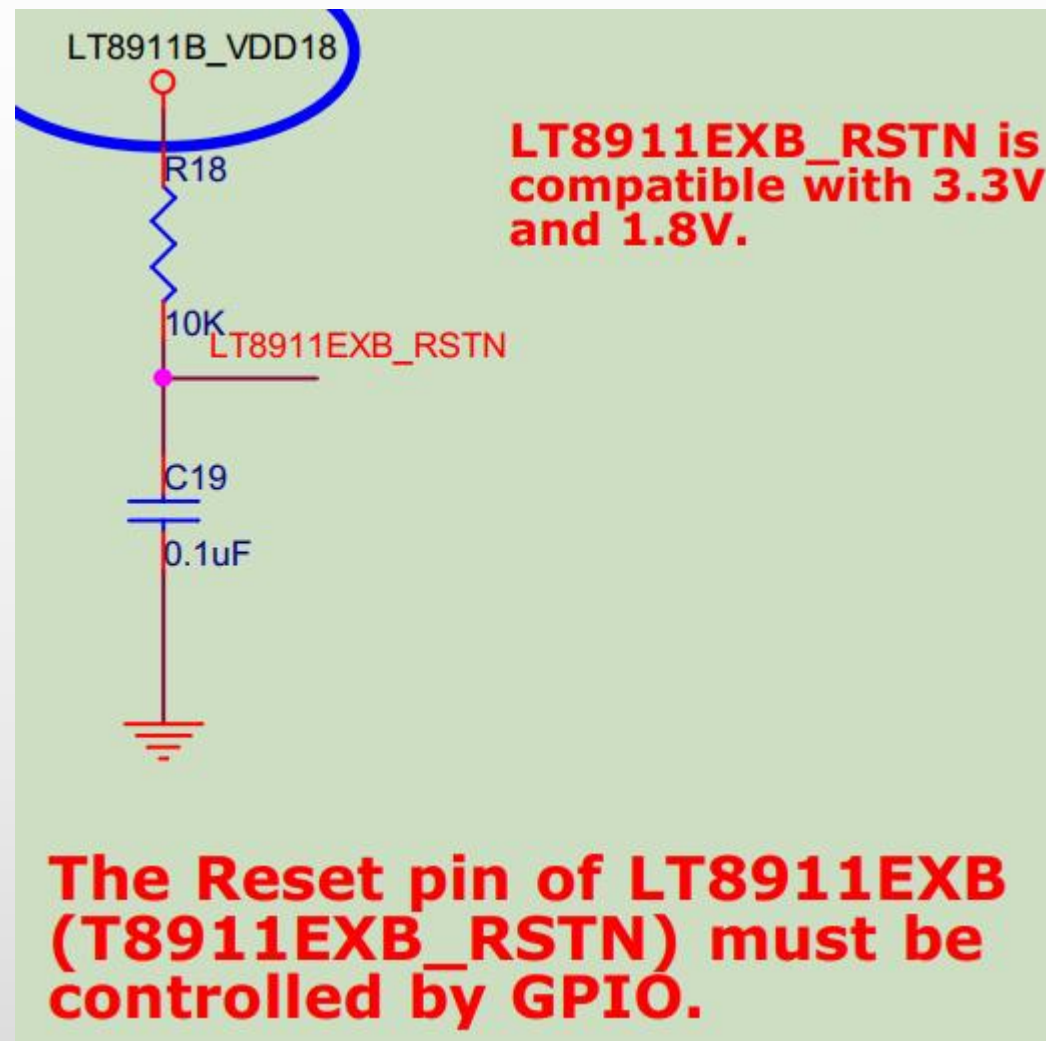
LT8911EX/LT 8911EXB 各电源输入需要用磁珠隔开，不要用0欧姆电阻。

LT8911EX/LT 8911EXB的IIC兼容3.3V 和 1.8V 电平，不需要做电平转换。



1.4、硬件review:

- LT8911EX/LT8911EXB 需要用主控SOC 的GPIO 去控制芯片的复位。
 - 从芯片角度来说，只要芯片复位脚保持1ms时间低电平就可以了，但考虑到复位脚上的RC 复位电路充放电时间，在配置寄存器前，这个复位拉低的时间一般设置50ms以上，是为了保证能稳定可靠复位。



2.1、LT8911EX输入信号 (LVDS)

- 要求先提供LVDS信号/给屏上电，再复位、初始化LT8911EX芯片，（可以在最后再打开屏背光）
- LT8911EX 支持单/双 port, 6/8bit 的LVDS 输入。
 - LVDS port A和port B 口可以swap (0xD812 ~0xD815寄存器)
 - LVDS data (D0 ~ D3) 线序可以swap (LVDS CLK的输入脚不能跟LVDS data输入脚swap) 。
 - LVDS data/CLK 的 P / N 可以swap (0xD837, 0x8237, 0x8241) 。
 - 如果按正常线序的PCB走线需要交叉、走较多的过孔，可以按方便PCB走线做线序、P/N的swap。

// LVDS CLK P / N swap:

```
HDMI_WriteI2C_Byte( 0xff, 0x82 );
```

```
HDMI_WriteI2C_Byte( 0x37, 0x0F );// default 0x1F; swap port A clk PN,setting 0x0f or 0x2f
```

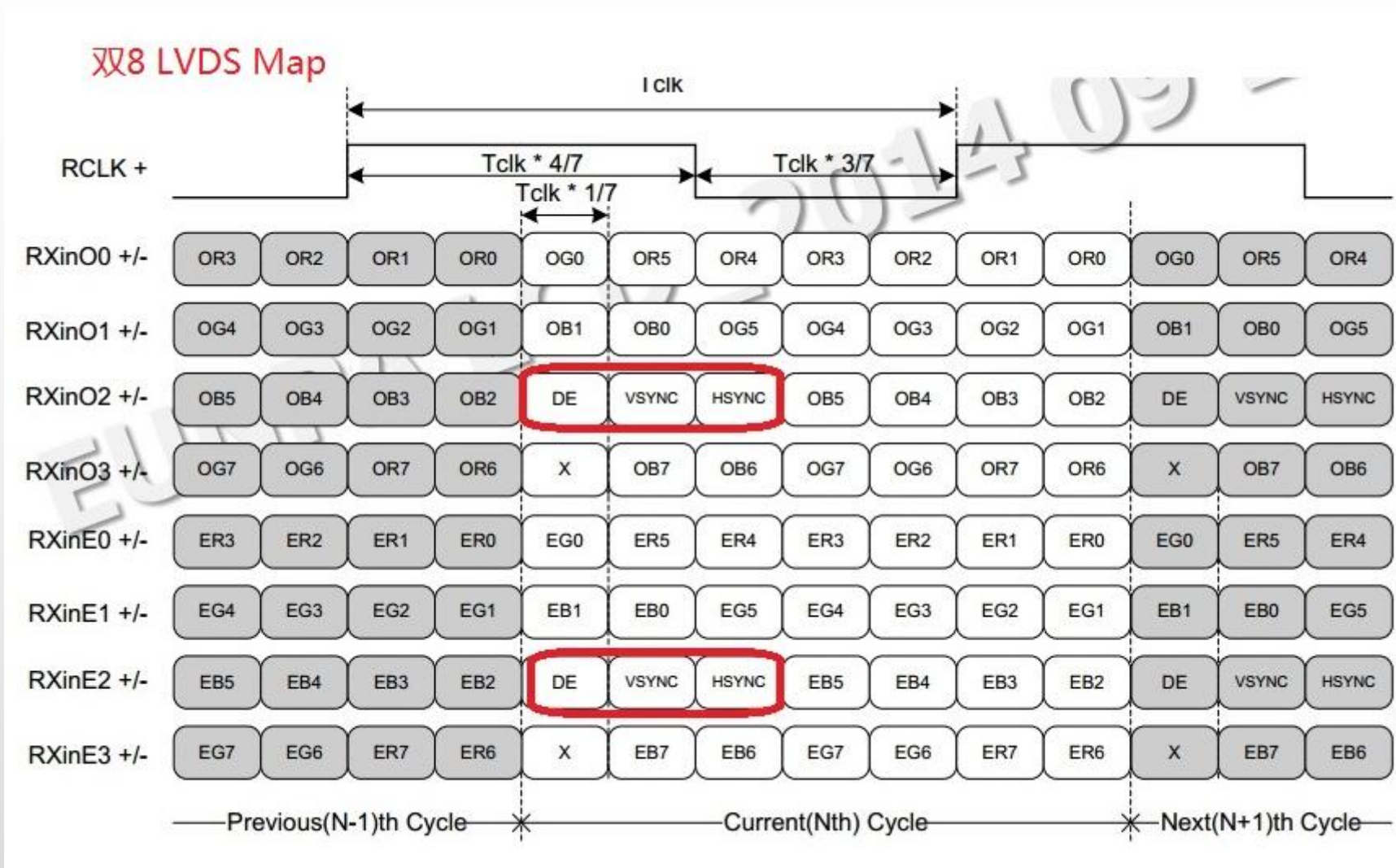
```
HDMI_WriteI2C_Byte( 0x41, 0x0F );// default 0x1F; swap port B clk PN,setting 0x0f or 0x2f
```


2.2、LT8911EX输入信号 (LVDS)

Reg addr	0xD812	0xD813	0xD814	0xD815	0xD837
bit7	RESERVED	RESERVED	RESERVED	RESERVED	Prot B channel 3 data PN swap enable.
bit[6:4]	The data select of channel 1 in port A : 3'd0 = Channel 0 in port A; 3'd1 = Channel 1 in Port A; 3'd2 = Channel 2 in Port A; 3'd3 = Channel 3 in Port A; 3'd4 = Channel 0 in Port B; 3'd5 = Channel 1 in Port B; 3'd6 = Channel 2 in Port B; 3'd7 = Channel 3 in Port B.	The data select of channel 3 in port A : 3'd0 = Channel 0 in port A; 3'd1 = Channel 1 in Port A; 3'd2 = Channel 2 in Port A; 3'd3 = Channel 3 in Port A; 3'd4 = Channel 0 in Port B; 3'd5 = Channel 1 in Port B; 3'd6 = Channel 2 in Port B; 3'd7 = Channel 3 in Port B.	The data select of channel 1 in port B : 3'd0 = Channel 0 in port A; 3'd1 = Channel 1 in Port A; 3'd2 = Channel 2 in Port A; 3'd3 = Channel 3 in Port A; 3'd4 = Channel 0 in Port B; 3'd5 = Channel 1 in Port B; 3'd6 = Channel 2 in Port B; 3'd7 = Channel 3 in Port B.	The data select of channel 3 in port B : 3'd0 = Channel 0 in port A; 3'd1 = Channel 1 in Port A; 3'd2 = Channel 2 in Port A; 3'd3 = Channel 3 in Port A; 3'd4 = Channel 0 in Port B; 3'd5 = Channel 1 in Port B; 3'd6 = Channel 2 in Port B; 3'd7 = Channel 3 in Port B.	Prot B channel 2 data PN swap enable.
					Prot B channel 1 data PN swap enable.
					Prot B channel 0 data PN swap enable.
bit3	RESERVED	RESERVED	RESERVED	RESERVED	Prot A channel 3 data PN swap enable
bit[2:0]	The data select of channel 0 in port A : 3'd0 = Channel 0 in port A; 3'd1 = Channel 1 in Port A; 3'd2 = Channel 2 in Port A; 3'd3 = Channel 3 in Port A; 3'd4 = Channel 0 in Port B; 3'd5 = Channel 1 in Port B; 3'd6 = Channel 2 in Port B; 3'd7 = Channel 3 in Port B.	The data select of channel 2 in port A : 3'd0 = Channel 0 in port A; 3'd1 = Channel 1 in Port A; 3'd2 = Channel 2 in Port A; 3'd3 = Channel 3 in Port A; 3'd4 = Channel 0 in Port B; 3'd5 = Channel 1 in Port B; 3'd6 = Channel 2 in Port B; 3'd7 = Channel 3 in Port B.	The data select of channel 0 in port B : 3'd0 = Channel 0 in port A; 3'd1 = Channel 1 in Port A; 3'd2 = Channel 2 in Port A; 3'd3 = Channel 3 in Port A; 3'd4 = Channel 0 in Port B; 3'd5 = Channel 1 in Port B; 3'd6 = Channel 2 in Port B; 3'd7 = Channel 3 in Port B.	The data select of channel 2 in port B : 3'd0 = Channel 0 in port A; 3'd1 = Channel 1 in Port A; 3'd2 = Channel 2 in Port A; 3'd3 = Channel 3 in Port A; 3'd4 = Channel 0 in Port B; 3'd5 = Channel 1 in Port B; 3'd6 = Channel 2 in Port B; 3'd7 = Channel 3 in Port B.	Prot A channel 2 data PN swap enable
					Prot A channel 1 data PN swap enable
					Prot A channel 0 data PN swap enable
default:	0x10	0x32	0x54	0x76	
portA/portB swap	0x54	0x76	0x10	0x32	

2.3、LT8911EX输入信号 (LVDS)

- LVDS CLK输入可以支持开展频 (SSC)。如果LVDS CLK 开了展频，那芯片的eDP输出就不要再同时打开展频，会影响显示效果。
- 有些主控SOC输出的双8 LVDS 信号，PortB的D2上没有DE/H/V 同步信号，导致输出图像显示的字符是一坨坨的。
 - 验证方法：设置PortB D2选择PortA D2源的输入，0x8015寄存器设置0x72，显示字符变正常，但颜色稍微有点差异。
 - 需要客户改前端LVDS 信号，让portB D2也输出DE/H/V 同步信息。



2.4、LT8911EX输入信号 (LVDS) --video check

```
// read 8911EX LVDS Timing
```

```
52 ff 81 00 // change Reg bank
```

```
52 08 7f 00
```

```
52 00 03 ff // read ID
```

```
//
```

```
52 ff d8 00 // change Reg bank
```

```
52 1f 00 00 // 00 Sync Mode;20 DE mode
```

```
//
```

```
52 ff 85 00 // change Reg bank
```

```
52 86 02 ff // H total
```

```
52 84 02 ff // V total
```

```
52 8a 02 ff // H act
```

```
52 88 02 ff // V act
```

```
52 82 02 ff // H fp
```

```
52 7c 02 ff // H sync
```

```
52 80 02 ff // H bp
```

```
52 7F 01 ff // V fp
```

```
52 7B 01 ff // V sync
```

```
52 7E 01 ff // V bp
```

```
//
```

```
52 1d 08 00 // sel pixel CLK
```

```
52 4d 03 ff // 4d[3:0]=[19:16],4e[7:0]=[15:8],4f[7:0]=[7:0]
```

0xD81F寄存器设置为sync mode (0x00) , 读到的 LVDS timing才是真正输入的 timing。

3.1、LT8911EXB输入信号 (MIPI)

- 要求先提供MIPI信号/给屏上电，再复位、初始化LT8911EXB芯片，（可以在最后再打开屏背光）
- LT8911EXB MIPI data线序可以swap (0xD003) 。
- LT8911EXB MIPI data P/N 可以swap (0xD000) 。

	0xD000
bit7	MIPI data Lane3 PN swap, high active.
bit6	MIPI data Lane2 PN swap, high active.
bit5	MIPI data Lane1 PN swap, high active.
bit4	MIPI data Lane0 PN swap, high active.

	0xD003
	MIPI lane swap: 5'd0 = Lane3210;// default
	5'd1 = Lane2310;
	5'd2 = Lane3120;
	5'd3 = Lane2130;
	5'd4 = Lane1230;
	5'd5 = Lane1320;
	5'd6 = Lane3201;
	5'd7 = Lane2301;
	5'd8 = Lane3021;
	5'd9 = Lane2031;
	5'd10 = Lane0231;
	5'd11 = Lane0321;
	5'd12 = Lane3012;
	5'd13 = Lane0312;
	5'd14 = Lane3102;
	5'd15 = Lane0132;
	5'd16 = Lane1032;
	5'd17 = Lane1302;
	5'd18 = Lane2013;
	5'd19 = Lane0213;
	5'd20 = Lane2103;
	5'd21 = Lane0123;
	5'd22 = Lane1023;
	5'd23 = Lane1203;
	Others = Lane3210.

3.2、LT8911EXB输入信号 (MIPI)

- LT8911EXB 对MIPI 输入信号的要求：
 - 需要**关闭**MIPI CLK的展频 (SSC) 。★★★★★
 - 需要**打开**MIPI data的EOTP (End Of Transmite Packet) 。★★★★★
- 关于EOTP：
 - 进入LP之前的那一段是叫trail的东西，是按照最后一个数据包的状态决定1还是0，trail的高电平或者低电平是由最后一个byte的bit7的反相的状态决定的。
 - Eotp是一个短包，内容是固定的 (0x08、0x0F、0x0F、0x01)，分配到每条lane (P lane) 后每次进入EOT波形就是固定的，最后一个byte的bit7都是0，反相之后，trail是固定高电平。
 - 如果没有打开eotp，那么最后传输的那个byte就是随机的，trail的高电平/低电平就是最后一个byte的bit7的反相，所以示波器看到的trail就是跳动的。



3.3、LT8911EXB输入信号 (MIPI) -videocheck

```
//8911EXB video check
52 ff 81 00 // change bank
52 08 7f 00
52 00 03 ff // read ID
//
52 ff 85 00
52 4d 03 ff // MIPI byteclk
//
52 76 02 ff // V total
//
52 ff d0 00
52 85 02 ff // V act
52 82 02 ff // H act *3
```

0x854D ~0x854F寄存器读到的是MIPI **byte** clk,
不是MIPI CLK, 读出来的clk值是KHz.

$\text{MIPI_byte_CLK} = \text{MIPI_CLK} / 4;$

4d[3:0]=[19:16],

4e[7:0]=[15:8],

4f[7:0]=[7:0]

0XD082、0xD083寄存器读到的值是Hactive的3倍,
比如1920x1080分辨率的MIPI信号输入, 0xD082、
0xD083读到的值分别是0x16, 0x80; $0x1680 = 5760 = 1920 * 3$

4.1、LT8911EX/LT8911EXB 的输出

- 芯片支持eDP data swap (0xAC15)
- 芯片支持eDP P/N swap
(0x0AC16[7:4], 分别对应
Lane3/2/1/0的P/N swap使能)
- 芯片支持AUX P/N swap (0xa63b[2])
- 如果不是为了eDP输出走线方便, 一般不建议做eDP输出swap。

Reg 0xAC15 bit:	bit7/bit6	bit5/bit4	bit3/bit2	bit1/bit0	
default :	lane 3	lane 2	lane 1	lane 0	
eDP lane :					Reg0xAC15 Value
3/2/1/0	11	10	01	00	0xE4(default)
3/2/0/1	11	10	00	01	0xE1
0/1/2/3	00	01	10	11	0x1B

4.2、LT8911EX / LT8911EXB eDP输出 SSC

0x8209寄存器设置0x00，配置SSC寄存器就打开eDP的SSC（0.5%展频幅度）。

LT8911EXB的eDP输出SSC配置，跟LT8911EX一样。

如果要关闭eDP的SSC，屏蔽0x8209寄存器设置（默认值是0x01）。

```
00424: // txpll Analog
00425: HDMI_WriteI2C_Byte( 0xff, 0x82 );
00426:
00427: HDMI_WriteI2C_Byte( 0x09, 0x00 ); // div hardware mode, for ssc.
00428:
00429: // HDMI_WriteI2C_Byte( 0x01, 0x18 );// default : 0x18
00430: HDMI_WriteI2C_Byte( 0x02, 0x42 );
00431: HDMI_WriteI2C_Byte( 0x03, 0x00 ); // txpll en = 0
00432: HDMI_WriteI2C_Byte( 0x03, 0x01 ); // txpll en = 1
00433: // HDMI_WriteI2C_Byte( 0x04, 0x3a );// default : 0x3A
00434:
00435: HDMI_WriteI2C_Byte( 0xff, 0x87 );
00436: HDMI_WriteI2C_Byte( 0x0c, 0x10 ); // cal en = 0
00437:
00438: HDMI_WriteI2C_Byte( 0xff, 0x81 );
00439: HDMI_WriteI2C_Byte( 0x09, 0xfc );
00440: HDMI_WriteI2C_Byte( 0x09, 0xfd );
00441:
00442: HDMI_WriteI2C_Byte( 0xff, 0x87 );
00443: HDMI_WriteI2C_Byte( 0x0c, 0x11 ); // cal en = 1
00444:
00445: //ssc
00446: HDMI_WriteI2C_Byte( 0xff, 0x87 );
00447: HDMI_WriteI2C_Byte( 0x13, 0x83 );
00448: HDMI_WriteI2C_Byte( 0x14, 0x41 );
00449: HDMI_WriteI2C_Byte( 0x16, 0x0a );
00450: HDMI_WriteI2C_Byte( 0x18, 0x0a );
00451: HDMI_WriteI2C_Byte( 0x19, 0x33 );
00452:
```

5.1、关于LT8911EX/EXB 点屏没显示的问题1

- 1、确保IIC读写是正常的。
 - IIC 数量不要超过100KHz .

1、IIC不通：	A、检查电源（1.8V）、晶振（25MHz）、复位（低电平复位，正常工作时为高）是否正常。IIC bus是否有上拉电阻？芯片底部的EPAD要接地。	如果IIC bus上挂的IIC从器件比较多，将上拉电阻改小一点，可以试试1K的上拉电阻。
	B、LT8911EX/EXB的IIC地址是0x52/0x5A，如果是Linux系统，bit7作为读写标志位，IIC 地址需要右移一位，既是0x29/0x2D。 如果LT8911EX/EXB的IIC地址选择脚（S_ADR）为低，则LT8911EXB的I2C 地址为 0x52 ； 如果LT8911EX/EXB的IIC地址选择脚（S_ADR）为高，则LT8911EXB的I2C 地址为 0x5A ； IIC 速率不要超过100KHz。	// 读LT8911EX/EXB ID值的操作 如下： IIC_Addr = 0x52; //bit0 是读写标志位； HDMI_WriteI2C_Byte(0xff, 0x81); //register bank HDMI_WriteI2C_Byte(0x08, 0x7f); Temp_ID0 = HDMI_ReadI2C_Byte(0x00); // 读 0x00 寄存器的值，正常值是0x17 Temp_ID1 = HDMI_ReadI2C_Byte(0x01); // 读 0x01 寄存器的值，正常值是0x05 Temp_ID2 = HDMI_ReadI2C_Byte(0x01); // 读 0x02 寄存器的值，正常值是0xe0 // HDMI_ReadI2C_Byte(Reg_Addr); 是标准 IIC 读操作函数，IIC 速率不要超过100KHz。
	C、可以用逻辑分析仪抓IIC 波形，可以看到IIC bus上写的是什么数据，有没有Ack。	

5.2、关于LT8911EX/EXB 点屏没显示的问题2

- 2、先配置芯片的test pattern输出，看看能不能点亮eDP。
 - 如果test pattern也点不亮eDP屏，
 - 读eDP的EDID，看看芯片配置的分辨率跟屏是不是一样，不能用1080P的timing配置去点非1080P分辨率的屏。
 - 检查芯片的外围硬件：
 - 电源（1.8V）、晶振（25M）、复位（低电平复位，正常工作时为高），是否正常。
 - 6K对地电阻阻值是否异常。
 - IIC 地址选择脚的电平跟寄存器配置的IIC地址是否对应。
 - 检测LT8911EX输出的eDP信号、AUX+/AUX- 信号跟eDP屏连接正常，线序、极性没有问题。
 - 确认eDP屏、屏线是OK的，eDP屏的panel VCC、背光电压、背光使能（BL_EN）、BL_PMW电压正常。
 - 读DPCD0202的值，看看是不是0x77（2lane eDP屏）/ 0x07（1lane eDP屏）
 - 如果不是0x77/0x07，重新training试试看。

5.3、关于LT8911EX/EXB 点屏没显示的问题3

- 3、如果test pattern可以点亮eDP屏，配置正常输出的寄存器配置
 - 读芯片的video check。看输入信号的timing是不是点屏所需的分辨率。
 - 能检测到输入信号timing
 - 看PCR是否稳定（LT8911EXB，0xD087寄存器bit4不是1，PCR就是unstable的）
 - 如果LT8911EXB PCR不稳定，需要测量MIPI CLK波形。
 - 看看输入的MIPI CLK是否有关闭展频。
 - 看看配置的MIPI CLK频率是否偏大、偏小。
 - 检测不到输入信号的timing
 - 确认输入信号data 线序、P/N 线序，跟寄存器的配置是对应。
 - 用示波器测量输入信号data/clock的波形。

6.1、常用的IIC 调试工具命令（1） --DPCD read

```
//DPCD read
52 ff 81 00 // change bank register
52 00 03 ff // read ID
//
52 ff ac 00
52 00 20 00
52 ff a6 00
52 2a 01 00
//
52 2b 90 00// H+90 read
52 2b 02 00// M
52 2b 02 00// L
52 2b 00 00//00--1;01--2
52 2c 00 00
//
52 ff a6 00
52 25 01 ff
52 39 01 ff//00--22;01--23
52 2b 01 ff// void read
52 2b 01 ff
//
52 ff ac 00
52 82 01 ff
```

读DPCD0202寄存器值，判断training是否成功，如果2 lane eDP屏，读DPCD0202d 值不是0x77；1 lane eDP屏，读DPCD0202的值不是0x07，说明training不成功。用IIC工具重新training一次看看（看6.2）

此调试命令是在主控SOC或者MCU已经对芯片有过初始化之后才有效果的。在这之前，需要对芯片进行初始化配置。

6.2、常用的IIC 调试工具命令 (2) --training

```
52 ff 81 00 // change bank register
52 00 03 ff // read Reg 0x8100
//
52 06 df 00// rst VID TX
52 06 ff 00
//
52 ff 85 00
52 a1 01 ff
//
/* Aux setup */
52 ff ac 00
52 00 60 00 //Soft Link train
52 ff a6 00
52 2a 00 00
//
// Aux reset
52 ff 81 00
52 07 fe 00
52 07 ff 00
52 0a fc 00
52 0a fe 00
```

```
//接左边
/* link train */
52 ff 85 00
52 1a 02 00 // 2 lane
52 ff ac 00
52 00 64 00
52 01 0a 00
52 0c 85 00
52 0c c5 00
```

此调试命令是在主控SOC或者MCU已经对芯片有过初始化之后才有效果的。在这之前，需要对芯片进行初始化配置。

7.1、关于视频信号屏参的常用公式：

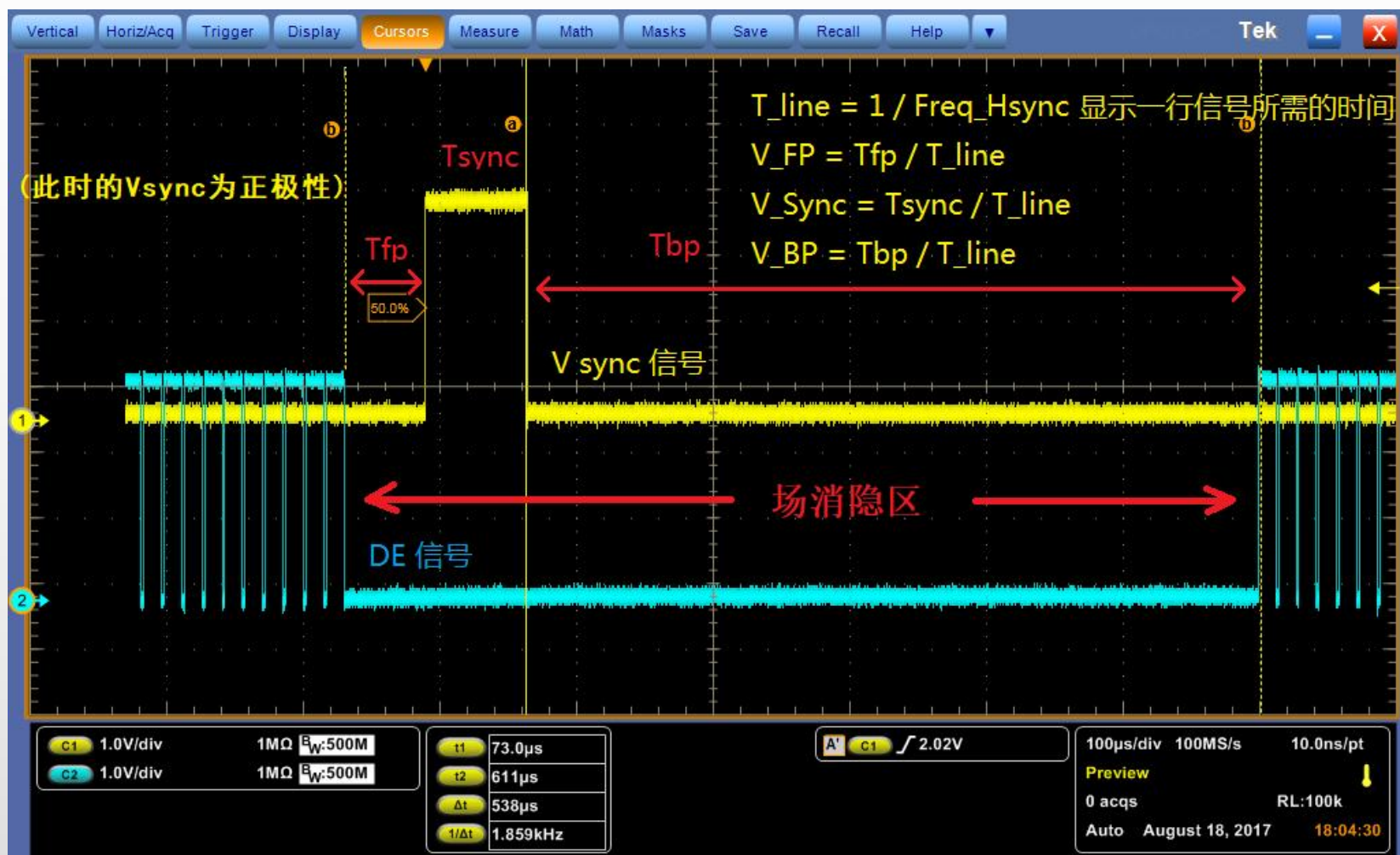
- $H_{total} = H_{active} + H_{BackPorch} + H_{FrontPorch} + H_{syncWidth} = H_{active} + H_{Blanking}$
- $V_{total} = V_{active} + V_{BackPorch} + V_{FrontPorch} + V_{syncWidth} = V_{active} + V_{Blanking}$
- $H_{Blanking} = H_{BackPorch} + H_{FrontPorch} + H_{syncWidth}$
- $V_{Blanking} = V_{BackPorch} + V_{FrontPorch} + V_{syncWidth}$

- $Pixel_CLK = H_{total} * V_{total} * \text{帧率(Hz)} = H_{total} * \text{行频(KHz)}$
- $\text{行频(KHz)} = V_{total} * \text{帧率(Hz)}$
- $H_{total} = Pixel_CLK / \text{行频(KHz)}$
- $V_{total} = \text{行频(KHz)} / \text{帧率(Hz)}$

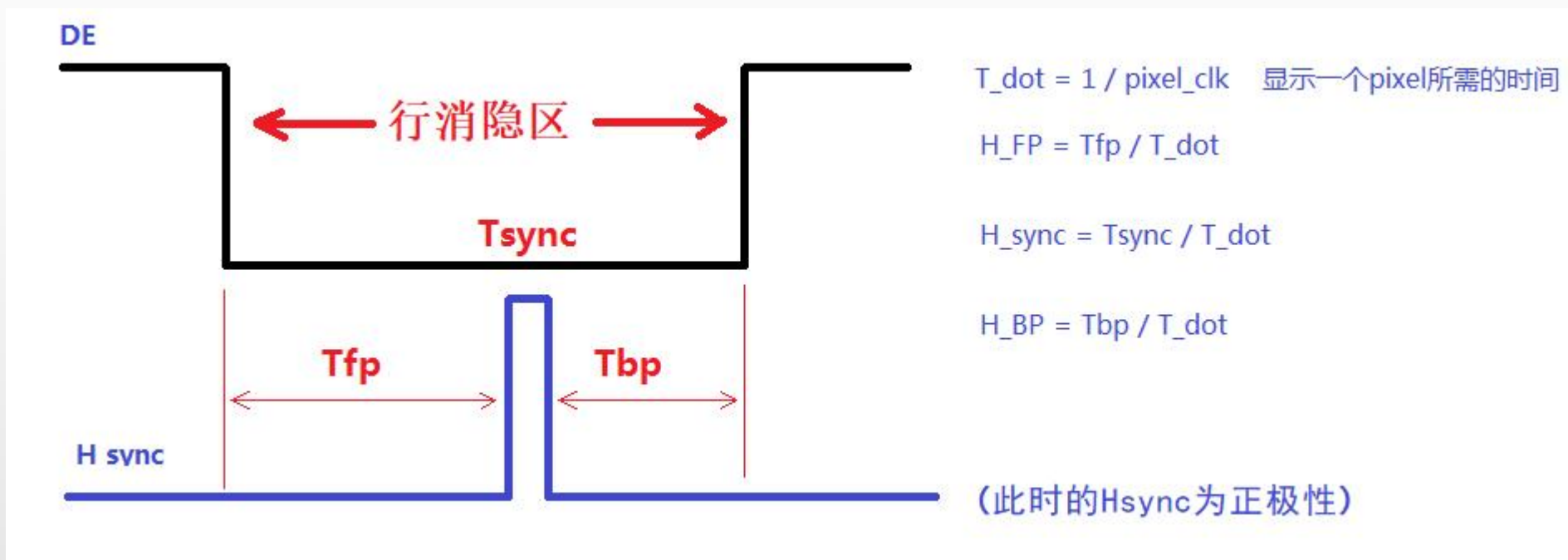
7.2、关于视频信号屏参的常用公式（MIPI）：

- $\text{MIPI_CLK} = \text{Pixel_CLK} * 3 * 8\text{bit} / (2 * 4 \text{ Lane}) // \text{RGB888}$
- $\text{MIPI_bit_rate} = \text{MIPI_CLK} * 2;$
- $\text{MIPI_byte_CLK} = \text{MIPI_bit_rate} / 8 = \text{MIPI_CLK} / 4;$

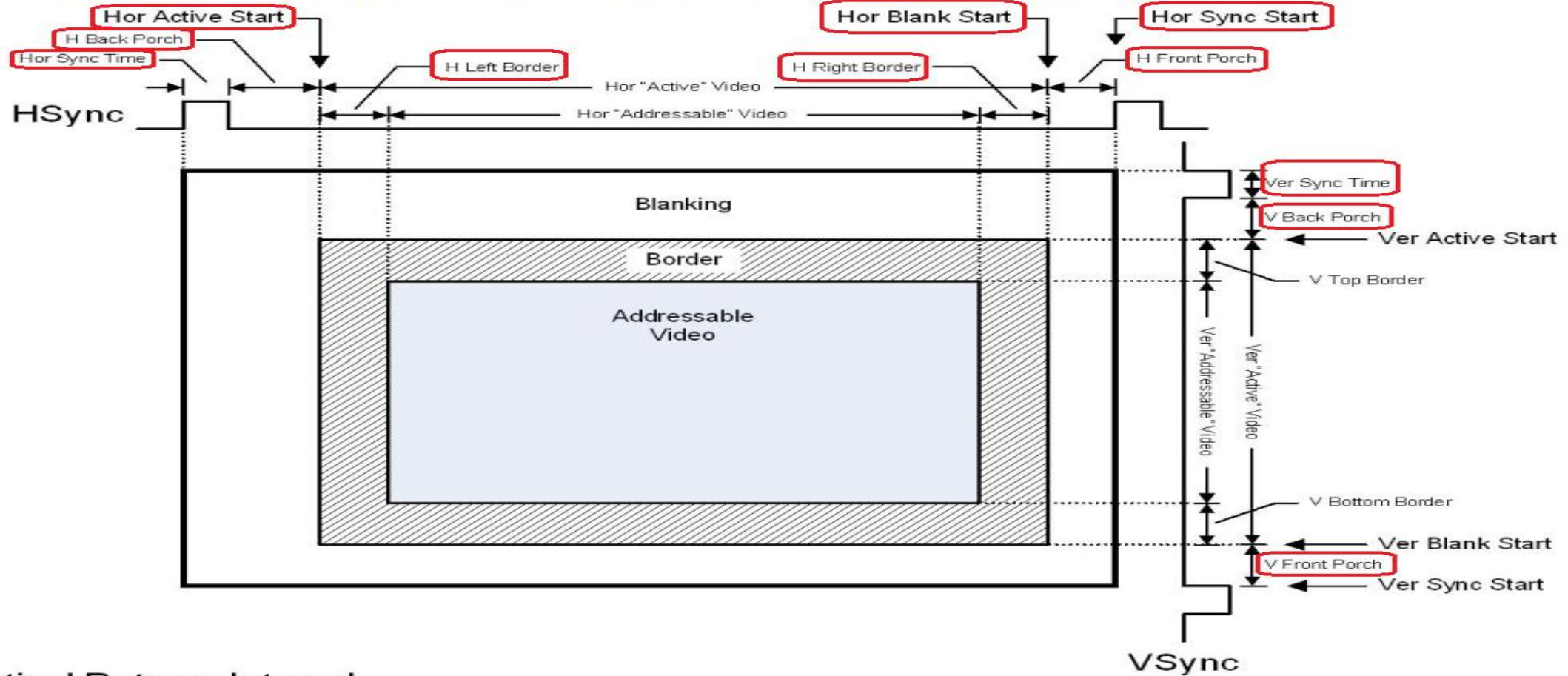
7.3、V sync & DE



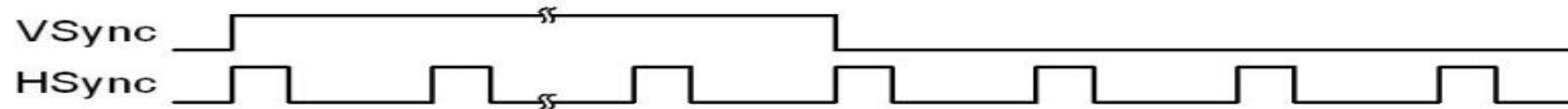
7.4、Hsync & DE



3.5 DMT Video Timing Parameter Definitions - Total Frame Timing:



Vertical Retrace Interval



Note:

All syncs shown as active high. For active low invert the waveform as shown below

