LT8912B MIPI 转 HDMI & LVDS 寄存器配置说明文档

LT8912B 是 IIC 从器件，需要前端主控 IIC bus 去控制、初始化 LT8912B。

1、LT8912B 内部有三个 IIC 模块，IIC 地址分别是 0x90/0x92/0x94.

I2C_Adr = 0x90 / 0x92 / 0x94;// bit0 是读写标志位；

如果是 Linux 系统，IIC address 的 bit7 作为读写标志位，IIC 地址需要右移一位，则 I2C_Adr 应该是 0x48 / 0x49 / 0x4a.

LT8912 IIC Address		Linux IIC address
0x90	==》	0x48
0x92	==》	0x49
0x94	==》	0x4a

LT8912B slave IIC 速率不要超过 100KHz。

2、LT8912B 对 MIPI 输入信号的要求：

a) MIPI DSI

b) Video mode

c) non-burst mode（continue mode）--（MIPI 的 CLK 是要连续的）（RK 平台要选择 MIPI_DSI_MODE_VIDEO_BURST ）

d) sync event or pulse。

MIPI 信号要关闭展频；MIPI 的 CLK 要连续，不能有 LP。

MIPI 信号要关闭 EOTP：

a、MTK 平台 的 dis_eotp_en 的值，改成 TRUE; LK 和 kernel 都需要修改成 TRUE

b、展讯平台的 tx_eotp 的值要置 0 。

c、高通 cfg->append_tx_eot = false.

1、LT8912B 支持 HDMI（DVI）和 LVDS 输出，在寄存器配置文档里，如果需要 LVDS 输出，那么需要打开 **_LVDS_Output_** 的宏定义（如下图）。

如果需要输出 DVI(准确说是 Single link 的 DVI-D，只有视频信号，没有音频)，那么就屏蔽_HDMI_Output_宏定义，同时打开_DVI_Output_ 宏定义。

同样，要输出 HDMI（带音频），那么就是屏蔽_DVI_Output_宏定义，打开 _HDMI_Output_宏定义。

MIPI 转 HDMI(DVI)输出是 Bypass 的，前端 MIPI 信号是什么分辨率，输出的 HDMI（DVI）就是什么分辨率。

MIPI 转 LVDS 输出，可以是 Bypass 输出，也可以 scale 输出（就是 LVDS 输出分辨率可以缩放成 1080P60 以下分辨率输出）。

```
00055: //-------------------------------------//
00056:
00057: #define _HDMI_Output_  // With video and audio output.
00058: //#define _DVI_Output_ // Only video output, no audio.
00059:
00060: //#define _LVDS_Output_ // LVDS output to drive panel.
00061:
00062: //-------------------------------------//
00063:
```

2、根据前端 MIPI 信号 Timing 配置 LT8912B MIPI 输入参数（如下图）：
下图设置的 Timing 值要跟 MIPI 信号 Timing 保持一致。

```
00262: //-----------------------------------------//
00263:
00264: // 设置输入的MIPI信号的Lane数
00265: #define MIPI_Lane 4                          // 4 Lane MIPI input
00266:
00267: // 根据前端MIPI输入信号的Timing修改以下宏定义的值:
00268:
00269: //* 1024x600  MIPI , pixel_clock  51.25MHz
00270:
00271: #define MIPI_H_Active   1024
00272: #define MIPI_V_Active   600
00273:
00274: #define MIPI_H_Total    1344
00275: #define MIPI_V_Total    635
00276:
00277: #define MIPI_H_FrontPorch   160
00278: #define MIPI_H_SyncWidth    20
00279: #define MIPI_H_BackPorch    140
00280:
00281: #define MIPI_V_FrontPorch   12
00282: #define MIPI_V_SyncWidth    3
00283: #define MIPI_V_BackPorch    20
00284:
00285: //*/
00286: //-----------------------------------------//
00287:
```

3、如果需要 LVDS 输出去点屏，需要设置 LVDS 的一些配置（如下图）：

如果 MIPI 信号分辨率跟 LVDS 屏分辨率一致，打开_Bypass_Mode_ 宏定义，屏蔽 _Scaler_Mode_ 宏定义。

如果 MIPI 信号分辨率跟 LVDS 屏分辨率不一样，打开_Scaler_Mode_ 宏定义，屏蔽 _Bypass_Mode_ 宏定义。

根据 LVDS 屏规格书，设置色深（6bit / 8bit）、输出模式（VESA/JEIDA）、DE 模式 / Sync 模式 ，打开关闭相应的宏定义。

```
00111: #ifdef _LVDS_Output_
00112:
00113: #define _Scaler_Mode_   // Mipi signal resolution and LVDS panel resolution are different.
00114: //#define _Bypass_Mode_ // Mipi signal resolution is the same as LVDS panel resolution.
00115:
00116: //-------------------------------------------//
00117: // 设置LVDS屏色深
00118: #define _8_Bit_Color_   // 24 bit
00119: //  #define _6_Bit_Color_ // 18 bit
00120:
00121: // 定义LVDS输出模式
00122: #define _VESA_
00123: //#define _JEIDA_
00124:
00125: //#define _De_mode_
00126: #define _Sync_Mode_
00127:
```

如果 LVDS 是设置 scaler mode 输出，还需要配置 LVDS 屏的屏参，如下图：

```
00128: #ifdef _Scaler_Mode_
00129: //-------------------------------------------//
00130: // 根据屏的规格书设置下面 LVDS 屏的Timing:
00131:
00132: //* 1024x600 panel
00133: #define Panel_Pixel_CLK 5125 // 51.25MHZ
00134:
00135: #define Panel_H_Active  1024
00136: #define Panel_V_Active  600
00137:
00138: #define Panel_H_Total   1344
00139: #define Panel_V_Total   635
00140:
00141: #define Panel_H_FrontPorch  160
00142: #define Panel_H_SyncWidth   20
00143: #define Panel_H_BackPorch   140
00144:
00145: #define Panel_V_FrontPorch  12
00146: #define Panel_V_SyncWidth   3
00147: #define Panel_V_BackPorch   20
00148:
```

4、LT8912B 支持 IIS 或者 SPDIF 数字音频输入，跟 MIPI 信号一起转成 HDMI 输出（如下图）。

根据 IIS 或者 SPDIF 输入，配置不同的寄存器。

根据输入音频的不同的采样率、数据长度，设置不同的寄存器。

```
00448: void Audio_Config( void )
00449: {
00450:     // Audio config
00451:     I2CADR = 0x90;
00452:     HDMI_WriteI2C_Byte( 0xB2, 0x01 );    // 0x01:HDMI; 0x00: DVI
00453:
00454: #if 1 // IIS input
00455: //  AudioIIsEn(); // IIS Input
00456:     I2CADR = 0x94;
00457:     HDMI_WriteI2C_Byte( 0x06, 0x08 );    // 0x09
00458:     HDMI_WriteI2C_Byte( 0x07, 0xF0 );    // enable Audio: 0xF0;  Audio Mute: 0x00
00459:
00460:     HDMI_WriteI2C_Byte( 0x09, 0x00 );    // 0x00:Left justified; default
00461: // 0x02:Right justified;
00462:
00463: #else // SPDIF input
00464:     //  AudioSPDIFEn(); // SPDIF Input
00465:     I2CADR = 0x94;
00466:     HDMI_WriteI2C_Byte( 0x06, 0x0e );
00467:     HDMI_WriteI2C_Byte( 0x07, 0x00 );
00468: //*/
00469: #endif
00470:
00471:     HDMI_WriteI2C_Byte( 0x0f, 0x0b + Sample_Freq[_48KHz] );
00472:
00473:     HDMI_WriteI2C_Byte( 0x37, (u8)( IIS_N[_48KHz] / 0x10000 ) );
00474:     HDMI_WriteI2C_Byte( 0x36, (u8)( ( IIS_N[_48KHz] & 0x00FFFF ) / 0x100 ) );
00475:     HDMI_WriteI2C_Byte( 0x35, (u8)( IIS_N[_48KHz] & 0x0000FF ) );
00476:
00477:     HDMI_WriteI2C_Byte( 0x34, 0xD2 );    // 32 bit的数据长度
00478: //  HDMI_WriteI2C_Byte( 0x34, 0xE2 ); // 16 bit的数据长度
00479:
00480:     HDMI_WriteI2C_Byte( 0x3c, 0x41 );    // Null packet enable
00481: } ? end Audio_Config ?
```

IIS input

SPDIF input

48KHz 采样率

5、AVI（Auxiliary Video Information）设置（具体可以参考 HDMI 规范）：

HDMI_VIC 的值，根据 MIPI 信号分辨率来决定，如果是标准的 HDMI 分辨率，在下面第二张图的列表里找相应的对应值，如果是其它分辨率，设置为 0。

AVI_PB1：如果色彩空间是 RGB，设置 0x10；如果是 YUV422 色彩空间，设置为 0x30，LT8912B 是 DSI 的 MIPI 信号输入，DSI 的 MIPI 信号是 RGB 色彩空间，所以这里设置为 0x10。

AVI_PB2 ：设置图像的宽高比，4:3 显示比例设置 0x19；16:9 显示比例设置 0x2A。720P、1080P、4K30 都是 16:9 的显示比例，800x600/1024x768 这些分辨率是 4:3 显示比例；默认设置 0x19.

```
00483: void AVI_Config(void)
00484: {
00485:     // AVI Packet config()
00486:
00487:     I2CADR = 0x94;
00488:     HDMI_WriteI2C_Byte( 0x3e, 0x0A );
00489:
00490:     // 0x43寄存器是checksums，改变了0x45或者0x47 寄存器的值，0x43寄存器的值也要跟着变，
00491:     // 0x43，0x44，0x45，0x47四个寄存器值的总和是0x6F
00492:
00493:     //  HDMI_VIC = 0x04; // 720P 60; Corresponding to the resolution to be output
00494:     HDMI_VIC = 0x10;                    // 1080P 60
00495:     //  HDMI_VIC = 0x1F; // 1080P 50
00496:     //  HDMI_VIC = 0x00; // If the resolution is non-standard, set to 0x00
00497:
00498:     AVI_PB1 = 0x10;   // PB1,color space: YUV444 0x70;YUV422 0x30; RGB 0x10
00499:
00500:     AVI_PB2 = 0x2A;   // PB2; picture aspect rate: 0x19:4:3 ;     0x2A : 16:9
00501:
00502:
00503:     /*******************************************************************************
00504:        The 0x43 register is checksums,
00505:        changing the value of the 0x45 or 0x47 register,
00506:        and the value of the 0x43 register is also changed.
00507:        0x43, 0x44, 0x45, and 0x47 are the sum of the four register values is 0x6F.
00508:      *******************************************************************************/
00509:     AVI_PB0 = ( ( AVI_PB1 + AVI_PB2 + HDMI_VIC ) <= 0x6f ) ? ( 0x6f - AVI_PB1 - AVI_PB2 - HDMI_VIC ) : ( 0x16f -
00510:
00511:     HDMI_WriteI2C_Byte( 0x43, AVI_PB0 );    //avi packet checksum ,avi_pb0
00512:     HDMI_WriteI2C_Byte( 0x44, AVI_PB1 );    //avi packet output RGB 0x10
00513:     HDMI_WriteI2C_Byte( 0x45, AVI_PB2 );    //0x19:4:3 ; 0x2A : 16:9
00514:     HDMI_WriteI2C_Byte( 0x47, HDMI_VIC );   //VIC(as below);1080P60 : 0x10
00515: } ? end AVI_Config ?
```

```
00228: u8  HDMI_VIC = 0x00; // vic ,0x10: 1080P ;  0x04 : 720P ; Refer to the following list
00229:
00230: /**************************************
00231:    Resolution           HDMI_VIC
00232:    ---------------------------------------
00233:    640x480              1
00234:    720x480P 60Hz        2
00235:    720x480i 60Hz            6
00236:
00237:    720x576P 50Hz        17
00238:    720x576i 50Hz            21
00239:
00240:    1280x720P 24Hz       60
00241:    1280x720P 25Hz       61
00242:    1280x720P 30Hz       62
00243:    1280x720P 50Hz       19
00244:    1280x720P 60Hz       4
00245:
00246:    1920x1080P 24Hz      32
00247:    1920x1080P 25Hz      33
00248:    1920x1080P 30Hz      34
00249:
00250:    1920x1080i 50Hz      20
00251:    1920x1080i 60Hz      5
00252:
00253:    1920x1080P 50Hz      31
00254:    1920x1080P 60Hz      16
00255:
00256:    3840x2160 30Hz       95 // 4K30
00257:
00258:    Other resolution 0(default)
00259:
00260:  **************************************/
```