**字幕帧结构定义**

|  |  |  |
| --- | --- | --- |
| 版本号 | 修改日期 | 修改内容 |
| 1.01 |  | 最初版本 |
| 1.02 |  | 添加了最小时间单位的默认值 |
| 1.03 | 2014-09-25 | 添加了帧头 |
| 1.04 | 2014-12-25 | 增加字幕整帧拆分、压缩等 |
|  |  |  |
|  |  |  |

**字幕帧应当包含的信息**：

1. SpotNumber；
2. 坐标，字幕图片左上角像素坐标和字幕图片的长度和宽度；
3. 时间，包含TimeIn和TimeOut，单位为帧；
4. 数据长度；
5. 数据实体（32bits Per Pixel）；

**字幕帧结构的具体定义：**

|  |  |  |
| --- | --- | --- |
| 字段 | 字幕信息 | 说明 |
| 帧头 | 帧头 | 128bit，其值为CCCC\_CCCC\_5555\_5555\_FFFF\_FFFF\_0000\_0000 |
| SpotNumber | SpotNumber | 2Bytes，从“1”开始，每一个字幕图片对应一个SpotNumber |
| 字幕图片坐标（均以像素点为单位） | 左上角像素横坐标 | 2Bytes，小于背景图片的长度 |
| 左上角像素纵坐标 | 2Bytes，小于背景图片的高度 |
| 字幕图片长度 | 2Bytes |
| 字幕图片高度 | 2Bytes |
| 时间（以帧为单位） | TimeIn | 3Bytes，本字幕开始时间 |
| TimeOut | 3Bytes，本字幕结束时间 |
| 字幕总帧数 | 0x0~0xFF | 1Byte,当前SpotNumber中字幕包含的总帧数 |
| 当前分帧数 | 0x0~0xFF | 1Byte,当前帧中的分帧号 |
| 渐入Alpha步长 |  | 1Byte,渐入Alpha值增量  （注意最后一帧的Alpha值的满量程跳变） |
| 渐出Alpha步长 |  | 1Byte,渐出Alpha值减量  （注意最后一帧的Alpha值的满量程跳变） |
| 淡入结束点 |  | 4Bytes，字幕从TimeIn到该帧之间实现淡入渐变( 绝对值) |
| 淡出开始点 |  | 4Bytes，字幕从该帧到TimeOut之间实现淡出渐变( 绝对值) |
| 数据长度 | 数据长度 | 4Bytes，以字节为单位 |
| 数据体 | 数据体 | 32bits，压缩过的数据 |

1. **字幕图片位置的确定**

字幕图片的位置信息由XML中的Halign、Hposition、Valign和Vposition，以及字幕图片的长度和高度确定。

Halign有三种对其方式，分别是left、center和right。默认为center。Hposition的默认值为0。

|  |  |  |
| --- | --- | --- |
| 对齐方式 | Hposition定义 | 左上角像素横坐标计算公式 |
| Left | 背景图片左边沿和字幕图片左边沿的距离 | Hposition |
| Center | 背景图片中线和字幕图片中线的距离 | 1/2背景图片长度 + Hposition- 1/2字幕图片长度  **(更好理解：分辨率宽-字幕宽)/2+Hposition.** |
| Right | 背景图片右边沿和字幕图片右边沿的距离 | 背景图片长度 – Hposition - 字幕图片长度 |

Valign也是三种对齐方式，top、center和bottom。默认为center。Vposition的默认值为0。

|  |  |  |
| --- | --- | --- |
| 对齐方式 | Hposition定义 | 左上角像素纵坐标计算公式 |
| Top | 背景图片上边沿和字幕图片上边沿的距离 | Vposition |
| Center | 背景图片中线和字幕图片中线的距离 | 1/2背景图片高度 + Vposition- 1/2字幕图片高度  **（更好理解：分辨率高-字幕高 )/2+VPosition** |
| Bottom | 背景图片下边沿和字幕图片下边沿的距离 | 背景图片高度 – Vposition - 字幕图片高度 |

1. **时间的换算**

XML文件中的时间信息是以“HH-MM-SS-EEE”的格式表示的，需要将其转换成单位“帧” ：

* 其中的EEE是通过TimeCodeRate来定义，默认的值为250；如果其值为24，就表示1S内含有24个E；
* 而EditRate通过比例来定义步进值。比如定义<EditRate>24 1</EditRate>，那么当TimeCodeRate为24时，此时每次EEE增加1；

根据以上分析，假设TimeCodeRate为1000，并定义<EditRate>24 1</EditRate>，

那么当前图片的帧数公式为：(“HH-MM-SS”- StartTime)\*24 + 24\*EE/TimeCodeRate。

TimeIn和TimeOut的计算：假设StartTime为00-00-00-00，当前时间为00-00-15-217，则当前帧数为15\*24+24\*217/1000 = 360 + 5.208 =366帧。

FadeUpTime和FadeDownTime的计算：如果FadeUpTime值为00-00-00-840，则840\*24/1000 = 20.16 = 21，表示图片的透明度从全透明（0X00）到全不透明（0XFF）经过21个图片，该值每个图片递增255/21 = 12。

1. **字幕本身的特效**

Direction，方向。定义了字符读取的顺序，默认为ltr(left to right)。此外还有rtl(right to left)、ttb(top to bottom)和btt(bottom to top)三种方式。

字库本身定义的效果：

Script（super、sub、normal），Effect（border、shadow、none），Italic，Underline，Weight（bold、normal），Color（AARRGGBB），EffectColor，Size（default42），Space（0.5em），HGroup，Rotate（right、left、none）。

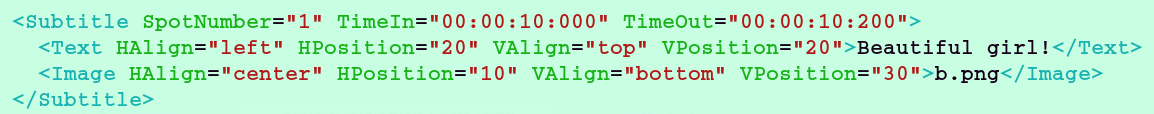
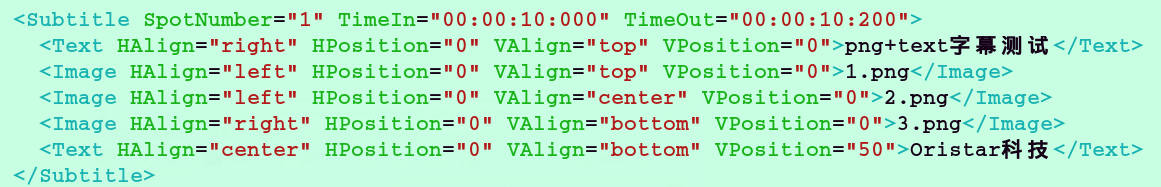
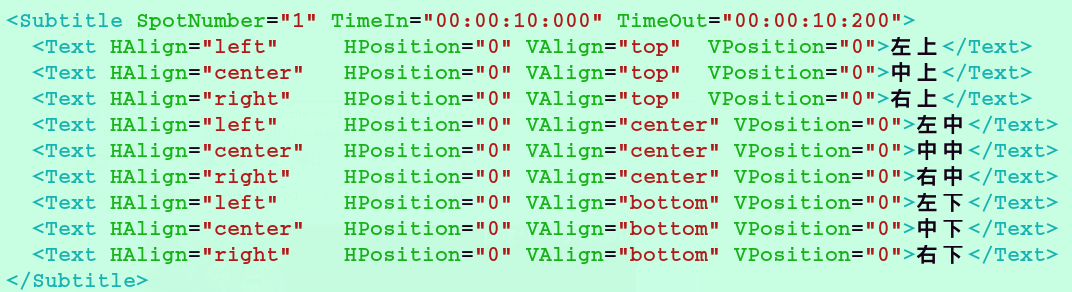
Size，假定背景图片的高度为11inches，而72pt（point）等于1inch。默认值为42pt；

Space，字体之间的空间，以字体的大小em为单位。默认为0.5em；

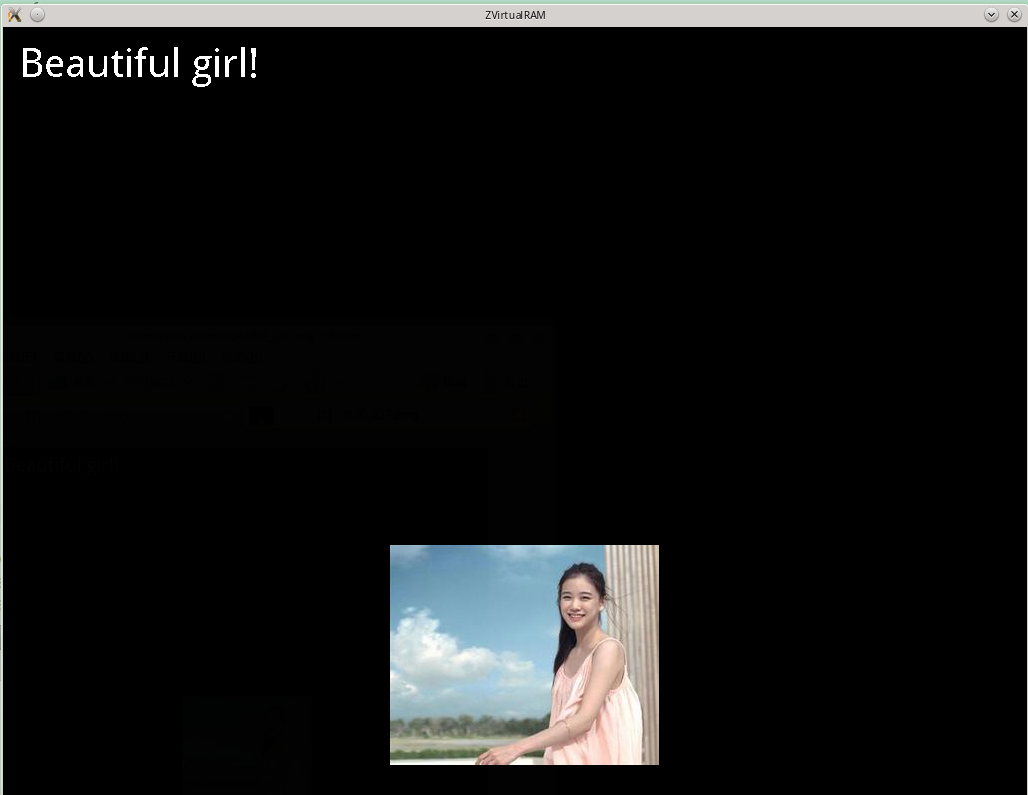
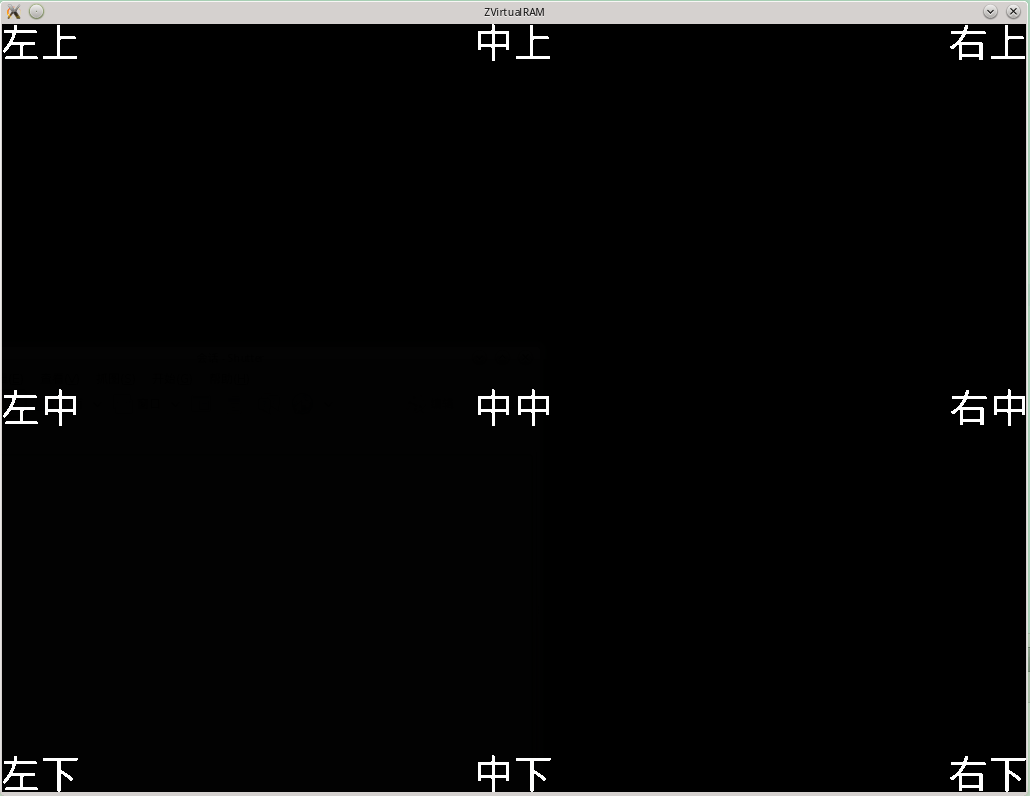
HGroup，Horizontal Group，当text direction为Vertical时才有效。

４、字幕拆分详细说明

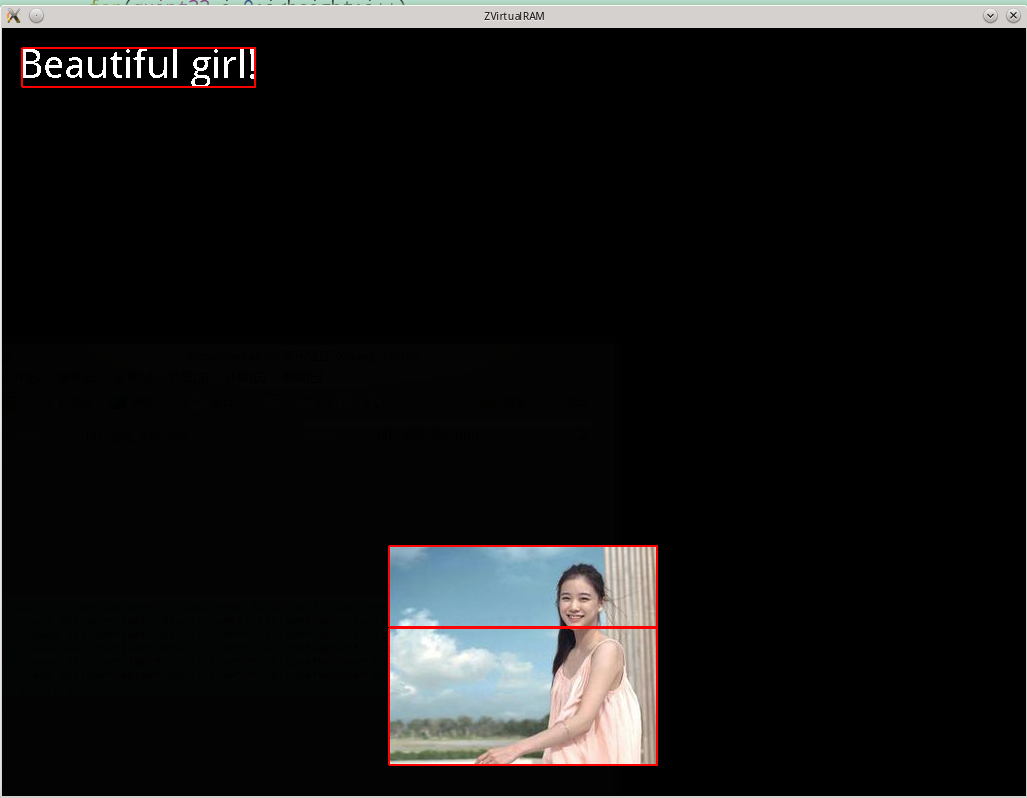
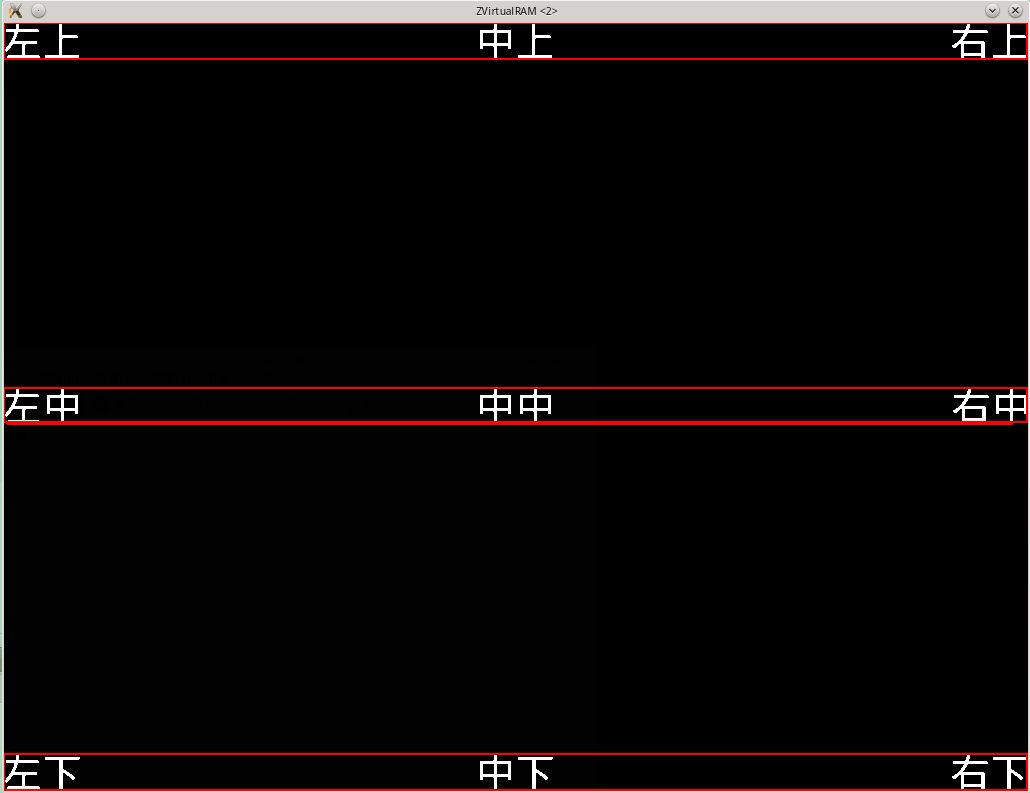
对于复杂的字幕描述xml文件，可能会在整屏的可见分辨率范围内任何位置处显示字幕相关数据，例如，下面所示的坐标情况比较复杂，如下：



这其中不局限于单独<Text>和<Image>以及混合体。



为了方便FPGA的处理，根据逐行扫描的原理，现将字幕数据按照保持x轴为最小单位，y轴逐步递增，进行分帧处理，如下图所示：



即，同属于一个SpotNumber中的字幕可能会被拆开N个分帧，分N次下发给FPGA。下发的字幕分帧中包含有总的分帧数和当前分帧数，每一分帧中均包含字幕像素矩阵的坐标和尺寸，用于FPGA叠加时的定位。

例如：

上面的第３张图，会被从上到下拆开３个分帧，每个分帧包含的信息，

第1个分帧包含：Beautiful girl! 字符的矩形框；

第２个分帧包含：图片的上小半部分（美女的头！）；

第３个分帧包含：图片的下大半部分（美女的身体！）。

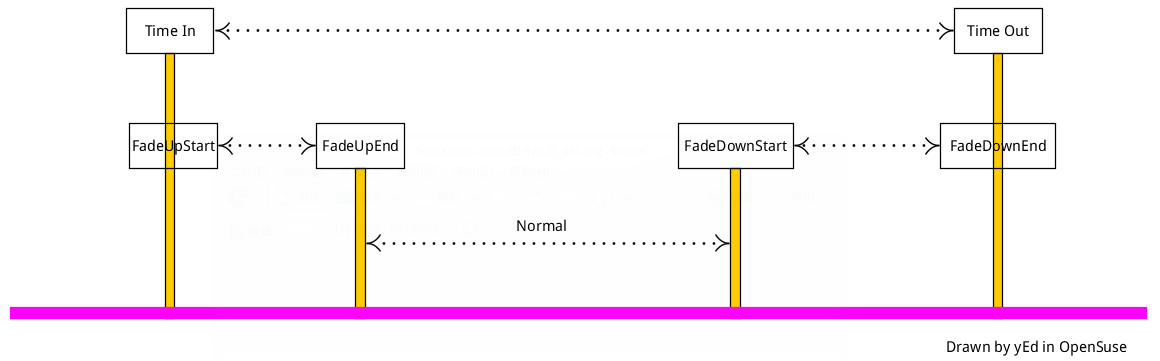
FPGA收到字幕数据后，应该判断该SpotNumber的总的分帧数目和当前分帧数目，只有两者相等，(分帧编号从1开始)，才证明属于该SpotNumber的字幕数据接收完成。

(后续陆续补充…)

５、淡入淡出效果

FadeUpTime，默认为20个ticks，即80ms，若当前帧频为24fps，则刷新一帧图像需要41.6ms，即要在2帧内完成淡入效果。第１帧的Alpha为0x0，第２帧的Alpha为0xFF。

FadeDownTime，默认也为20个ticks，即80ms，若当前帧频为24fps，则刷新一帧图像需要41.6ms，即要在2帧内完成淡入效果。第１帧的Alpha为0xFF，第２帧的Alpha为0x0。



ARM只下发一次字幕数据，FPGA根据协议中的“淡入结束点”和“TimeIn”两者差值计算淡入时每次Alpha的递增步长值，同时根据“淡出开始点”和“TimeOut”两者差值计算淡出时每次Alpha的递减步长值。

例：ARM向FPGA提供以下参数（当前帧频为24fps）：

TimeIn=240，FadeUpEnd=480，FadeDownStart=960，TimeOut=1100。

分析：

1. 在240帧处开始显示字幕，从240帧到480帧之间完成淡入效果，

这之间的差值为(480-240+1=241)，即需要完成241次Alpha值的变化，从0x0~0xFF的递增，

故有255/241=1.0580912863，为方便计算，这里取整为1。

即在240帧到479帧内，Alpha从0一直递增为240(0xF0)，

由于之前不能整除，所以最后的满值Alpha在最后一帧给出，即在480帧处Alpha变为满值0xFF。

1. 在480帧到906帧之间，字幕的Alpha保持为满值0xFF。
2. 在906帧处开始淡出字幕，在1100帧处清除字幕。

差值帧数为(1100-960+1=141)，即需要完成141次Alpha值的变化，从0xFF~0x0的递减，

故有255/141=1.8085106382，为方便计算，这里取整为2。

其实Alpha值从0xFF变为0x0，只需要255/2=127.5000000000帧(取整为128)即可完成，

所以余下的帧数(141-128=13)帧的Alpha已经变为0x0了。

(不知道这样处理是否可行？).