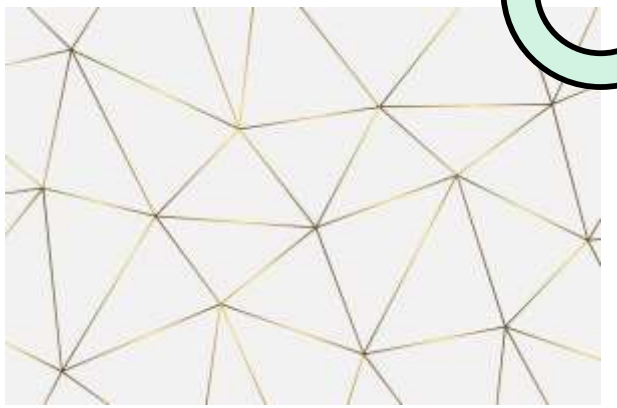
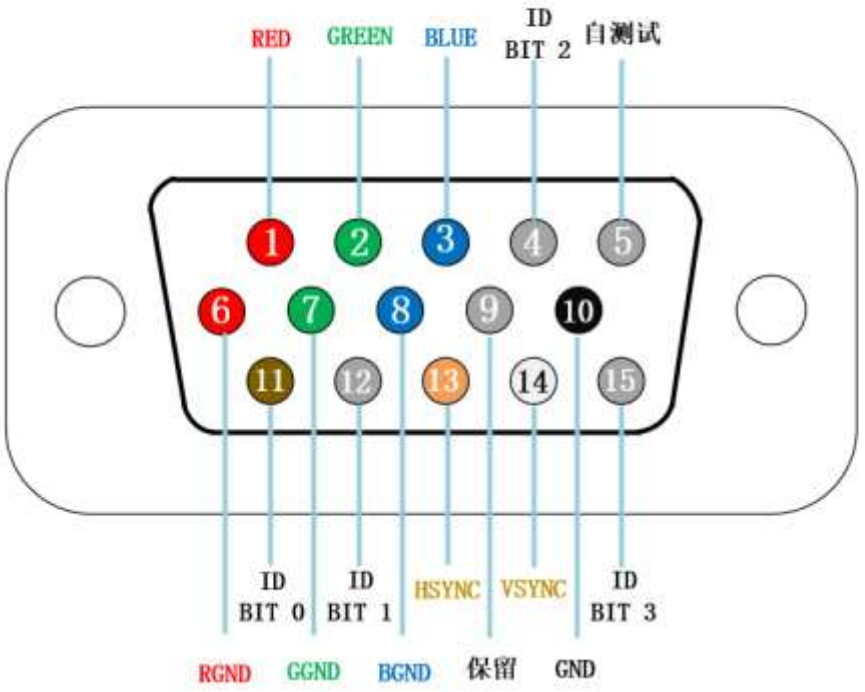
VGA LAB NOTE



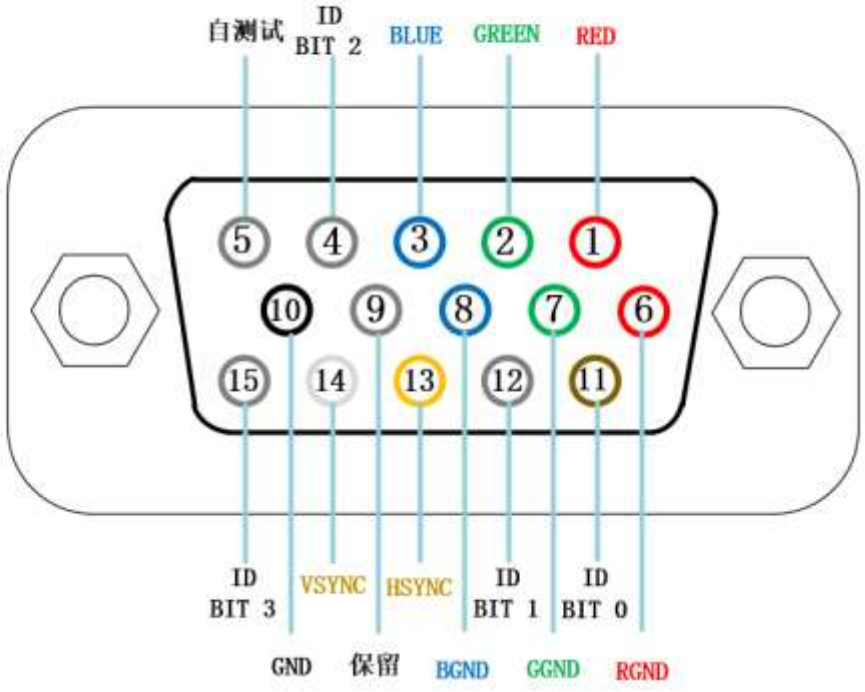
VGA 設計與驗證

Three type port : VGA > DVI > HDMI

公頭


RED
GREEN
BLUE
ID BIT 2
自測试
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
ID BIT 0
ID BIT 1
HSYNC
VSYNC
ID BIT 3
RGND
GGND
BGND
保留
GND

母頭

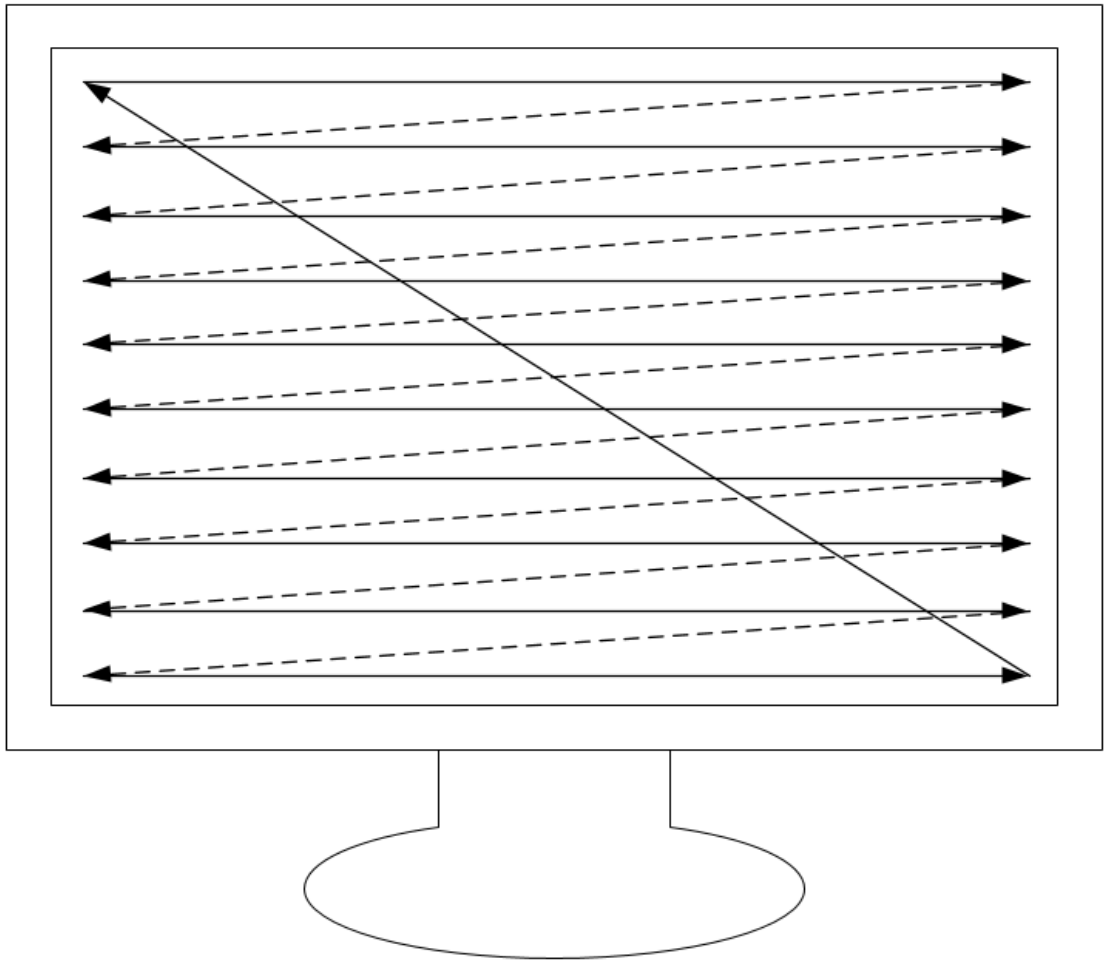

自測试
ID BIT 2
BLUE
GREEN
RED
5
4
3
2
1
10
9
8
7
6
15
14
13
12
11
ID BIT 3
VSYNC
HSYNC
ID BIT 1
ID BIT 0
GND
保留
BGND
GGND
RGND

引脚	定义	引脚	定义
1	红基色(RED)	9	保留(各厂家定义不同)
2	绿基色(GREEN)	10	数字地(GND)
3	蓝基色(BLUE)	11	地址码0(ID BIT0)
4	地址码2(ID BIT2)	12	地址码1(ID BIT1)
5	自测试(各厂家定义不同)	13	行同步(HSYNC)
6	红色地(RGND)	14	场同步(VSYNC)
7	绿色地(GGND)	15	地址码3(ID BIT3)
8	蓝色地(BGND)		

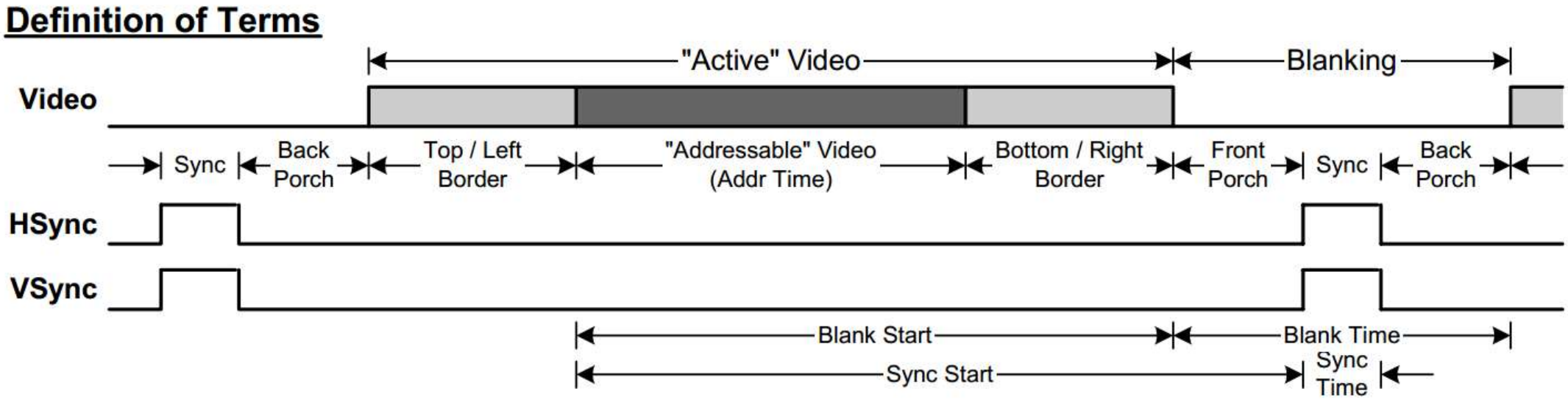
VGA介面共有15個接腳，分為3排，每排各5個，依照自上而下、由左向右的順序排列。其中第一排的腳位1、2、3和第三排的腳位**13、14最為重要**。

- VGA使用工業界通用的RGB色彩模式作為色彩顯示標準，這種色彩顯示標準是根據三原色中紅色、綠色、藍色所佔比例多少及三原色之間的相互疊加得到各式各樣的顏色。 腳位1紅基色(RED)、接腳2綠基色(GREEN)、接腳3藍基色(BLUE)就是VGA介面中負責傳送三原色的傳輸通道。 要注意的是，這3個接腳傳輸的 是類比訊號
- 腳位13行同步訊號(HSYNC)、接腳14場同步訊號(VSYNC)，這兩個訊號，是在VGA顯示影像時，負責同步影像色彩資訊的同步訊號
- 腳位5、9：這兩個接腳分別是VGA接口的自測試和預留接口，不過不同生產廠家對這兩個接口定義不同，在接線時，兩腳可懸空不接
- 腳位4、11、12、15：這四個是VGA介面的位址碼，可以懸空不接。
- 腳位6、7、8、10：這四個腳接地

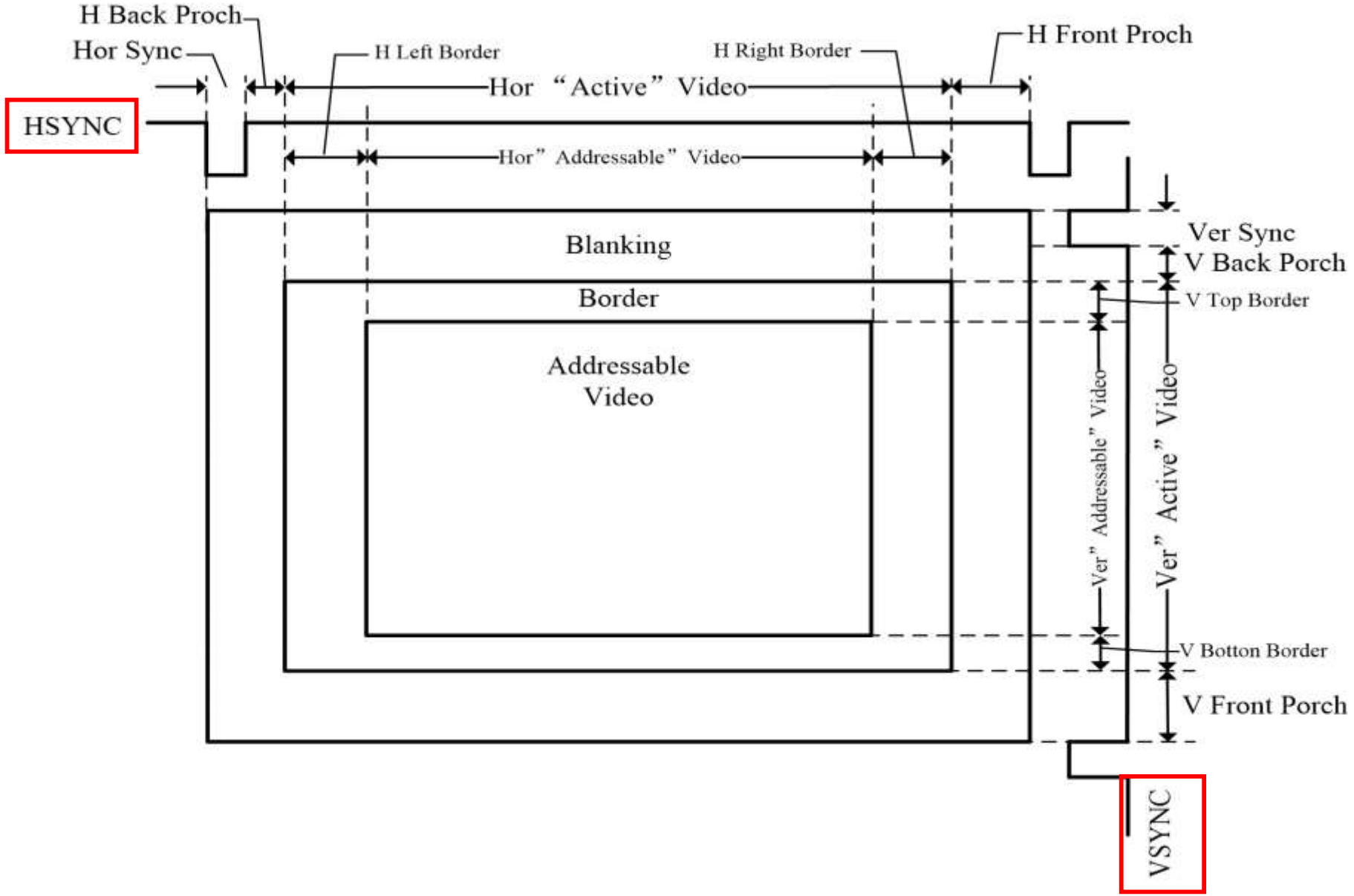
顯示原理



1.
左上角第一個像素點
2.
自左上角(第一行)第一個像素點座標，逐個像素點向右掃描(圖中第一個水平方向箭頭)
3.
掃描到第一行最後一個數據，一行影像掃描完成，進行影像消隱，掃描座標自第一行行尾轉移到第二行行首(圖中第一條虛線)；
4.
重複若干次掃描至最後一行行尾，一幀影像掃描完成，進行影像消隱，掃描座標跳轉回到左上角第一行行首(圖中對角線箭頭)，開始下一幀影像的掃描 。



1.
一個完整的行掃描週期，包含6部分：Sync（同步）、Back Porch（後沿）、Left Border（左邊框）、“Addressable” Video（有效圖像）、Right Border（右邊框）、Front Porch（前沿），這6部分的基本單位是pixel（像素），即一個像素時脈週期。
2.
在一個完整的行掃描週期中，Video影像資訊在HSync行同步訊號的同步下完成一行影像的掃描顯示，Video影像資訊只有在「**Addressable**」 Video（有效影像）階段，影像資訊有效，其他階段影像訊息 無效。
3.
HSync、VSync行同步訊號在Sync（同步）階段，維持高電平，其他階段均保持低電平，在下一個行掃描週期的Sync（同步）階段，HSync行掃描訊號會再次拉高


H Back Proch
Hor Sync
H Left Border
H Right Border
H Front Proch
Hor "Active" Video
HSYNC
Hor" Addressable" Video
Blanking
Ver Sync
V Back Porch
Border
V Top Border
Addressable
Video
Ver" Addressable" Video
Ver" Active" Video
V Botton Border
V Front Porch
VSYNC

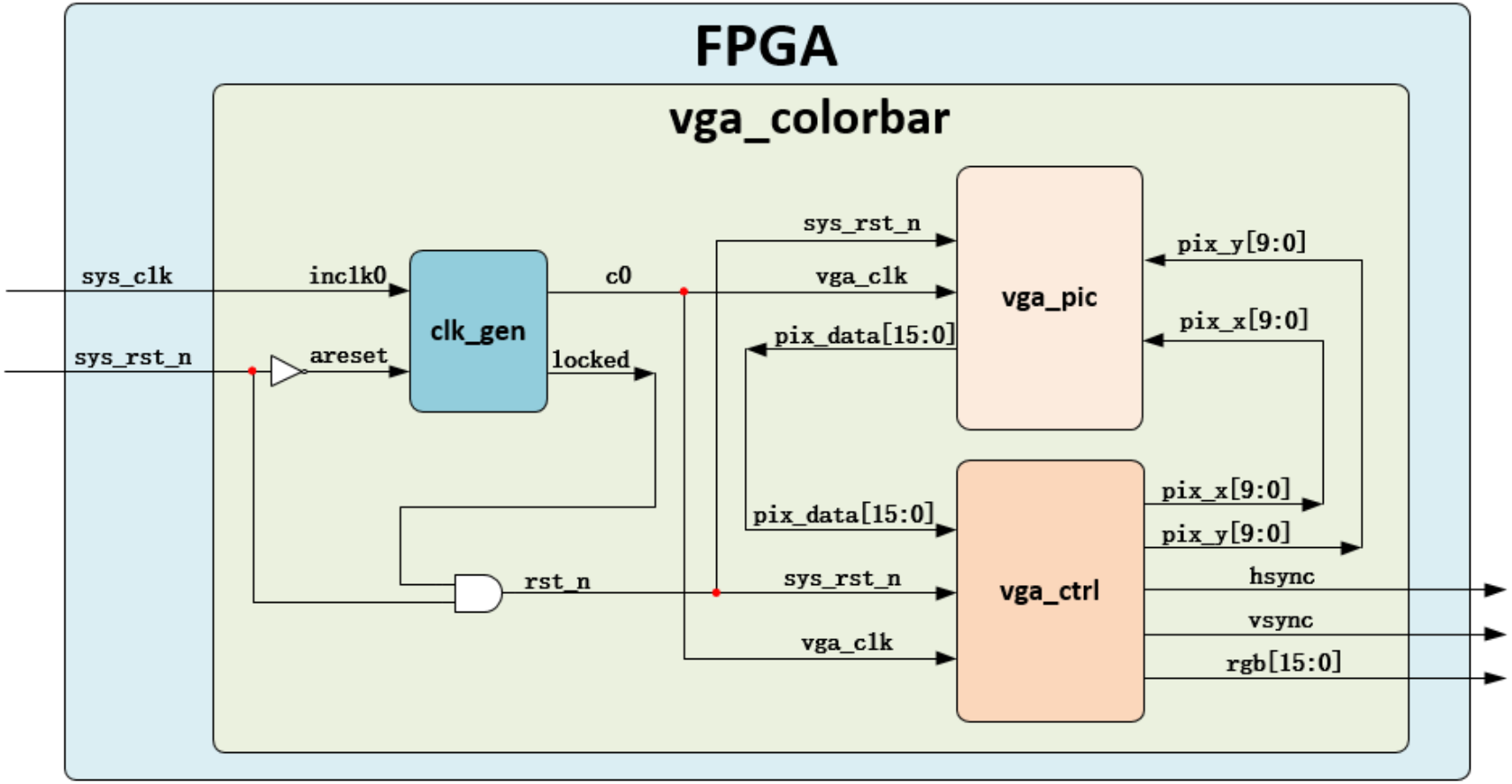
显示模式	时钟 (MHz)	行同步信号时序(像素)							场同步信号时序(行数)						
		同步	后沿	左边框	有效图像	右边框	前沿	行扫描周期	同步	后沿	上边框	有效图像	底边框	前沿	场扫描周期
640x480@60	25.175	96	40	8	640	8	8	800	2	25	8	480	8	2	525
640x480@75	31.5	64	120	0	640	0	16	840	3	16	0	480	0	1	500
800x600@60	40.0	128	88	0	800	0	40	1056	4	23	0	600	0	1	628
800x600@75	49.5	80	160	0	800	0	16	1056	3	21	0	600	0	1	625
1024x768@60	65	136	160	0	1024	0	24	1344	6	29	0	768	0	3	806
1024x768@75	78.8	176	176	0	1024	0	16	1312	3	28	0	768	0	1	800
1280x1024@60	108.0	112	248	0	1280	0	48	1688	3	38	0	1024	0	1	1066

VGA 常用分辨率时序参数

显示模式	时钟 /MHz	行时序参数(单位：像素)					列时序参数(单位：行)				
		a	b	c	d	e	f	g	h	i	k
640x480@60Hz	25.175	96	48	640	16	800	2	33	480	10	525
800x600@60Hz	40	128	88	800	40	1056	4	23	600	1	623
1024x768@60Hz	65	136	160	1024	24	1344	6	29	768	3	806
1280x720@60Hz	74.25	40	220	1280	110	1650	5	20	720	5	750
1280x1024@60Hz	108	112	248	1280	48	1688	3	38	1024	1	1066
1920x1080@60Hz	148.5	44	148	1920	88	2200	5	36	1080	4	1125

Pixel Clock = (Screen Refresh Frequency)*(Hor Active Video + Hor Front Porch + Hor Synv Pulse + Hor Back Porch)* (Ver Active Video + Ver Front Porch + Ver Synv Pulse + Ver Back Porch)

640x480,60Hz這種解析度格式來說，25.175MHz = 25175000Hz = 60*(640 + 16 + 96 + 48)*(480 + 11 + 2 + 31) = 60 * 800



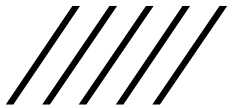
模块名称	功能描述
vga_colorbar	顶层模块
clk_gen	时钟生成模块，生成VGA驱动时钟
vga_ctrl	VGA时序控制模块，驱动VGA图像显示
vga_pic	图像数据生成模块，生成VGA待显示图像

Hor 時序: 開始計數，像素時鐘CNT到清零，利用assign在計數值為Hor Sync，信號拉低，產生一個低脈衝。

Ver 時序:開始計數，像素時鐘CNT到清零，利用assign在計數值為Ver Sync，信號拉低，產生一個低脈衝。

行時序與場時序以後，接下來就是在Hor Active Video和Ver Active Video均有效的期間往Red,Green,Blue三個分量送數據，數據就會在在螢幕上顯示出來了。而Hor Active Video有效的期間正是行計數器的計數值在大於(Hor Sync + Hor Back Porch)，小於(Hor Sync + Hor Back Porch + Hor Active Video)的時候，而Ver Active Video有效的期間正是 場計數器的計數值在大於(Ver Sync + Ver Back Porch)，小於(Ver Sync + Ver Back Porch + Ver Active Video)的時候，所以在程式碼裡面可以利用assign語句產生一個激活標誌，當激活標誌為高 的時候給Red,Green,Blue三個分量送數據

640x480為例來寫vga_driver的程式碼，由前面的解析度時序參數表可知，640x480解析度的像素時脈為25.175Hz，但實際上並不需要這麼精確的時脈頻率，我們取25MHz就可以了，我的開發 板的時脈頻率為50MHz，所以只需要簡單的寫一個二分頻邏輯就可以得到這個像素時脈了。如果你想顯示其他解析度的圖片，例如800x600解析度的時脈頻率是40MHz，這時候就需要用FPGA內部的Clocking Wizard IP核來得到這個40MHz的時鐘，Clocking Wizard IP核內部回呼叫FPGA的PLL資源 對輸入頻率進行處理以得到想要的輸出頻率



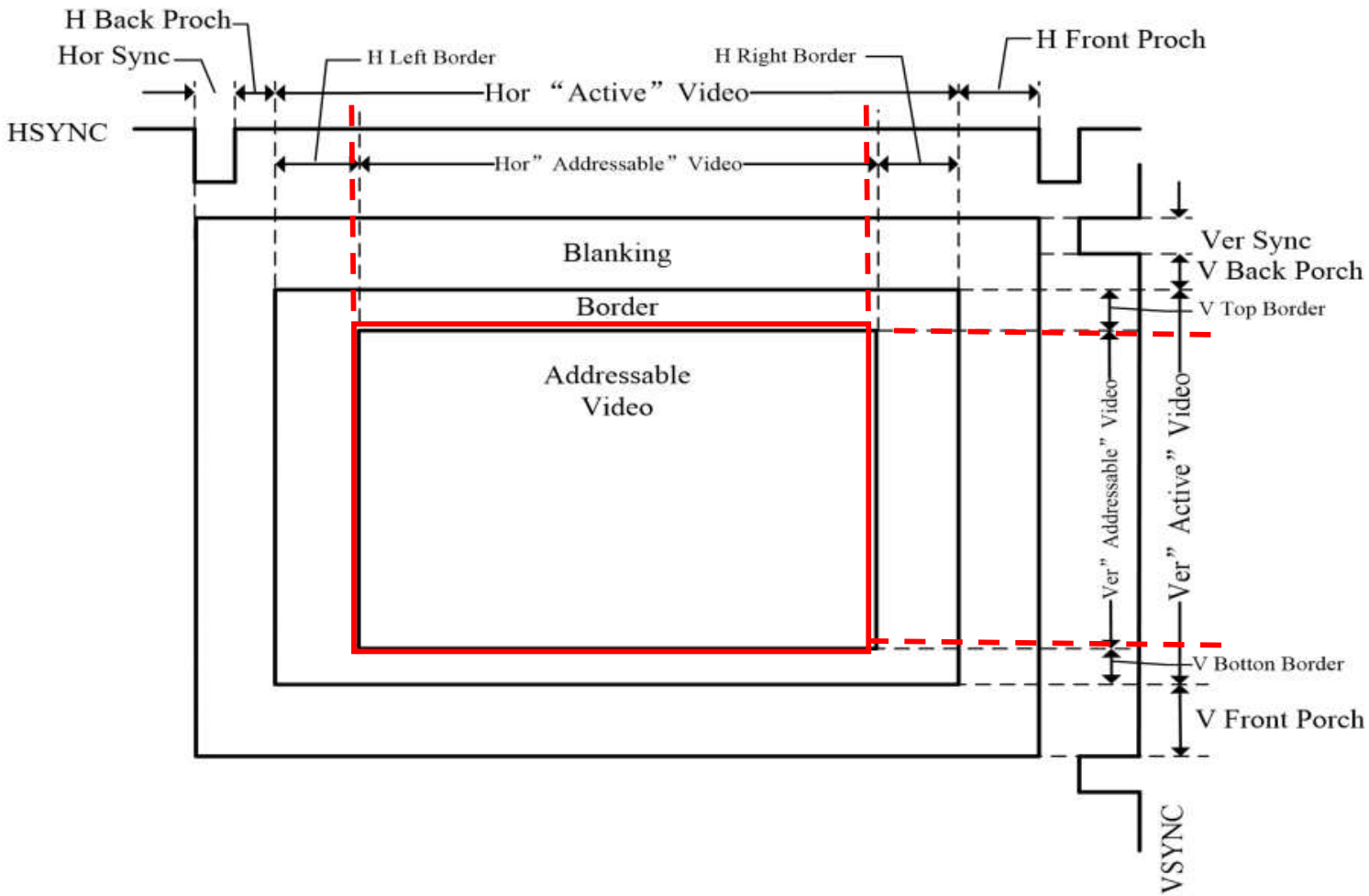
```
//640*480  Hor
parameter       C_H_SYNC_PULSE      =       'd96  ,
                C_H_BACK_PORCH      =       'd48  ,
                C_H_ACTIVE_TIME     =       'd640 ,
                C_H_FRONT_PORCH     =       'd16  ,
                C_H_LINE_PERIOD     =       'd800 ;

//640*480  Ver
parameter       C_V_SYNC_PULSE      =       'd2   ,
                C_V_BACK_PORCH      =       'd33  ,
                C_V_ACTIVE_TIME     =       'd480 ,
                C_V_FRONT_PORCH     =       'd10  ,
                C_V_FRAME_PERIOD    =       'd525 ;
```

VGA 常用分辨率时序参数

显示模式	时钟	行时序参数(单位：像素)					列时序参数(单位：行)				
	/MHz	a	b	c	d	e	f	g	h	i	k
640x480@60Hz	25.175	96	48	640	16	800	2	33	480	10	525
800x600@60Hz	40	128	88	800	40	1056	4	23	600	1	623
1024x768@60Hz	65	136	160	1024	24	1344	6	29	768	3	806
1280x720@60Hz	74.25	40	220	1280	110	1650	5	20	720	5	750
1280x1024@60Hz	108	112	248	1280	48	1688	3	38	1024	1	1066
1920x1080@60Hz	148.5	44	148	1920	88	2200	5	36	1080	4	1125

```
// actice signle
assign   W_act_flag =   (R_h_cnt >= (C_H_SYNC_PULSE + C_H_BACK_PORCH                 )) &&
                        (R_h_cnt <= (C_H_SYNC_PULSE + C_H_BACK_PORCH + C_H_ACTIVE_TIME)) &&
                        (R_v_cnt >= (C_V_SYNC_PULSE + C_V_BACK_PORCH                 )) &&
                        (R_v_cnt <= (C_V_SYNC_PULSE + C_V_BACK_PORCH + C_V_ACTIVE_TIME)) ;
```



分別給rgb進行配置。

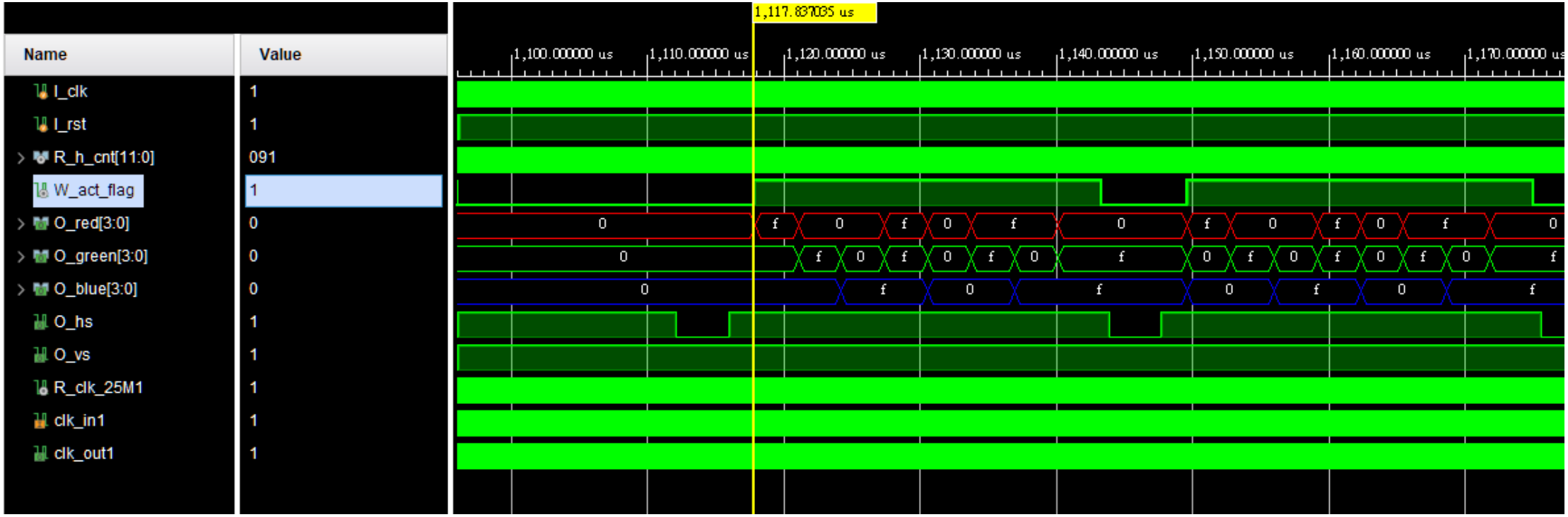
```
always@(*)
begin
    if(!I_rst)
        begin
            O_red   <=  4'b0000;
            O_green <=  4'b0000;
            O_blue  <=  4'b0000;
        end
    else if(W_act_flag)
        begin
            if(R_h_cnt < (C_H_SYNC_PULSE + C_H_BACK_PORCH + C_COLOR_BAR_WIDTH)) // red
                begin
                    O_red   <=  4'b1111    ;
                    O_green <=  4'b0000    ;
                    O_blue  <=  4'b0000    ;
                end
            else if(R_h_cnt < (C_H_SYNC_PULSE + C_H_BACK_PORCH + C_COLOR_BAR_WIDTH*2)) // green
                begin
                    O_red   <=  4'b0000    ;
                    O_green <=  4'b1111    ;
                    O_blue  <=  4'b0000    ;
                end
            else if(R_h_cnt < (C_H_SYNC_PULSE + C_H_BACK_PORCH + C_COLOR_BAR_WIDTH*3)) // blue
                begin
                    O_red   <=  4'b0000    ;
                    O_green <=  4'b0000    ;
                    O_blue  <=  4'b1111    ;
                end
            else if(R_h_cnt < (C_H_SYNC_PULSE + C_H_BACK_PORCH + C_COLOR_BAR_WIDTH*4)) // white
                begin
                    O_red   <=  4'b1111    ;
                    O_green <=  4'b1111    ;
                    O_blue  <=  4'b1111    ;
                end
            else if(R_h_cnt < (C_H_SYNC_PULSE + C_H_BACK_PORCH + C_COLOR_BAR_WIDTH*5)) // black
                begin
                    O_red   <= 4'b0000  ;
                    O_green <= 4'b0000  ;
                    O_blue  <= 4'b0000  ;
                end
            else if(R_h_cnt < (C_H_SYNC_PULSE + C_H_BACK_PORCH + C_COLOR_BAR_WIDTH*6)) // yellow
                begin
                    O_red   <= 4'b1111    ;
                    O_green <= 4'b1111    ;
                    O_blue  <= 4'b0000    ;
```

ISSUE

(*)

Posedge 25Mclk

Posedge 25Mclk



○ (*)


TB_VGA_Driver.v
VGA_Driver.v
TB_VGA_Driver_behav.wcfg
Name
Value
I_clk
I_rst
R_h_cnt[11:0]
W_act_flag
O_red[3:0]
O_green[3:0]
O_blue[3:0]
O_hs
O_vs
R_clk_25M1
clk_in1
clk_out1
1
1
091
1
0
0
0
1
1
1
1
1,117.837035 us
1,100.000000 us
1,120.000000 us
1,140.000000 us
1,160.000000 us
1,180.000000 us
1,200.000000 us

ISSUE

- 需要再去計算為何，正緣觸發以及敏感觸發的相差。

目前Simulation 看起來正緣觸發在早10us觸發，並且一個週期為30us，敏感觸發20us以內。