

数字逻辑设计进阶实验

实验六 存储器

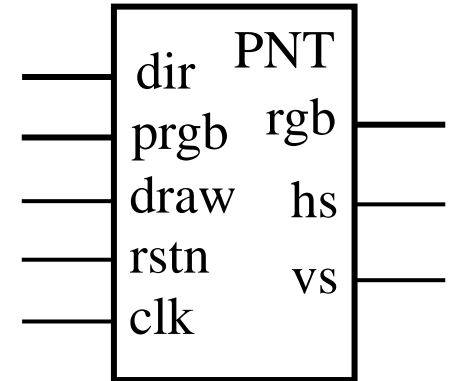
实验目的

- 掌握存储器**IP**核的使用，分布式与块式存储器的区别，以及存储器的读写时序
- 理解视频显示基本原理，**VGA**显示接口和定时时序
- 掌握数字系统模块化的设计方法

实验内容

- 控制画笔在画布(显示屏)上绘画

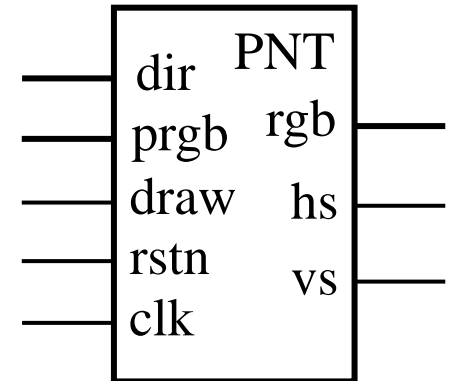
- 画布分辨率: 200x150, 画笔位置(x, y), $x = 0 \sim 199$, $y = 0 \sim 149$, 复位时 (50, 50)
- 画笔移动方向(dir): 上/下/左/右, 按钮控制
- 画笔颜色(prgb): 12位(红r绿g蓝b各4位), 开关设置
- 绘画开关(draw): 1--移动画笔同时绘制颜色, 0--仅移动画笔
- rgb, hs, vs: 显示器接口信号



	0	1	...	199
0	(0, 0)	(1, 0)	...	(199, 0)
1	(0, 1)	(1, 1)	...	(199, 1)
...
149	(0,149)	(1,149)	...	(199,149)

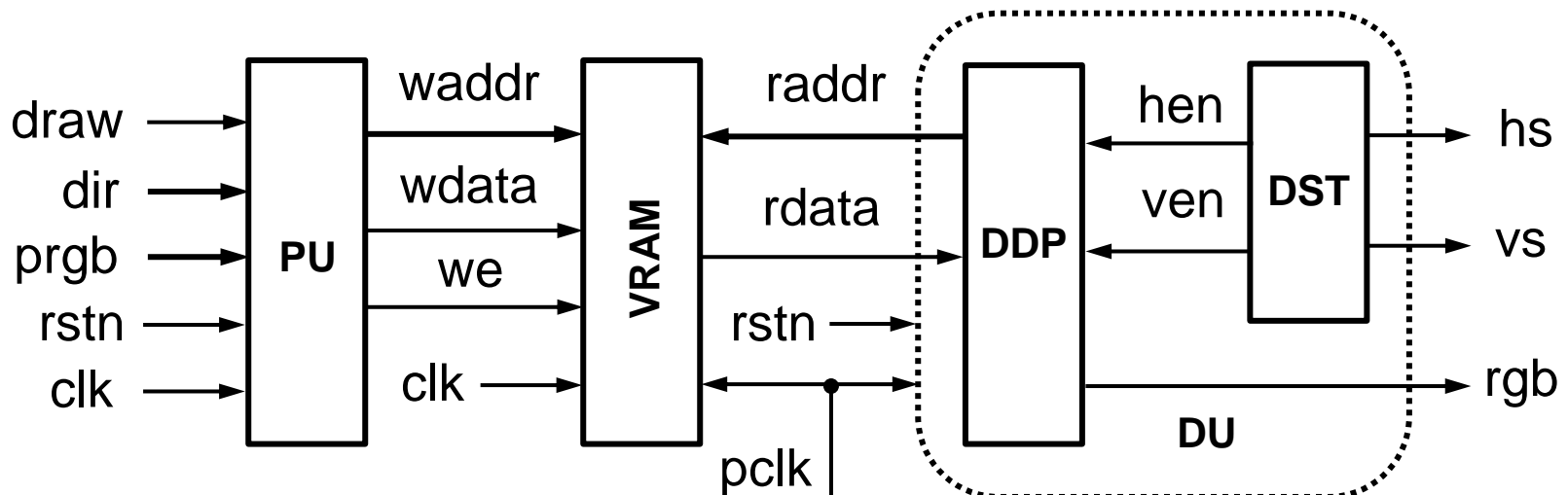
绘画模块端口

```
module painter (  
    input clk, rstn,           //时钟, 复位 (低电平有效)  
    input up,                  //移动方向 (dir): 上  
    input down,                //移动方向 (dir): 下  
    input left,                //移动方向 (dir): 左  
    input right,               //移动方向 (dir): 右  
    input [3:0] pred,          //画笔颜色 (prgb): 红  
    input [3:0] pgreen,        //画笔颜色 (prgb): 绿  
    input [3:0] pblue,         //画笔颜色 (prgb): 蓝  
    input draw,                //绘画开关: 1-绘画  
    output [3:0] red,          //像素颜色 (rgb): 红  
    output [3:0] green,        //像素颜色 (rgb): 绿  
    output [3:0] blue,         //像素颜色 (rgb): 蓝  
    output hs,                 //行同步  
    output vs                   //场同步  
);
```



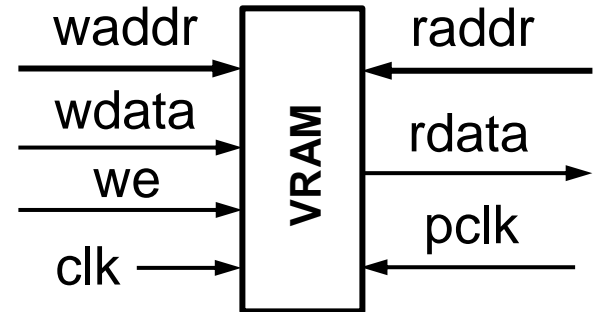
绘画模块逻辑结构

- **PU: Painting Unit**, 绘画单元
- **VRAM: Video RAM**, 视频存储器
- **DU: Display Unit**, 显示单元
 - **DDP: Display Data Processing**, 显示数据处理
 - **DST: Display Scan Timing**, 显示扫描定时



视频存储器

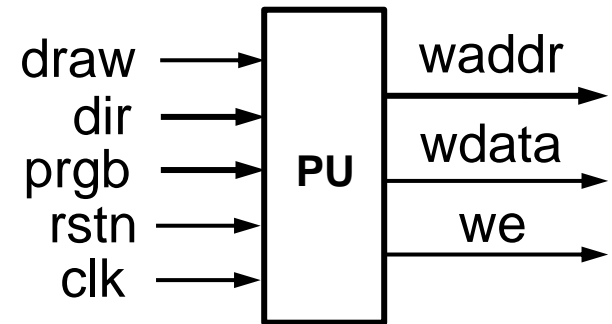
- 存储绘画信息，用块式简单双端口存储器实现
 - 存储容量：32Kx12位(每单元存储画布的1个像素，共需要200x150x12位)
 - 绘画写端口：waddr, wdata, we，写地址、写数据、写使能，同步写端口
 - 显示读端口：raddr, rdata，读地址、读数据，同步读端口



	0	1	2	...	199
0	0	1	2	...	199
1	200	201	...		
2					
...					
149					

绘画单元

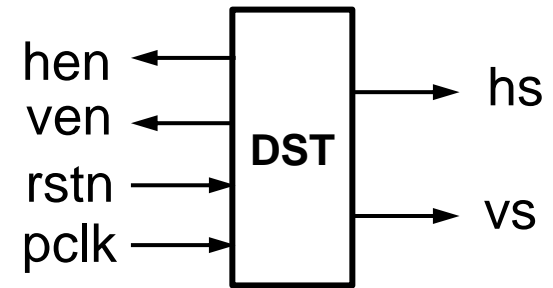
- 处理绘画操作，修改VRAM中图形信息
 - 通过12个拨动开关设置画笔颜色 (prgb)
 - 通过上/下/左/右(dir)按钮开关，移动画笔位置(x, y)
 - 直角移动：单一按钮按下一次，x或y增加或减小1
 - 绘画 (draw=1) 时，依据rgb 和 (x, y)，通过VRAM写端口(waddr, wdata, we)存储绘图信息



	0	1	2	...	199
0	0	1	2	...	199
1	200	201	...		
2					
...					
149					

显示扫描定时

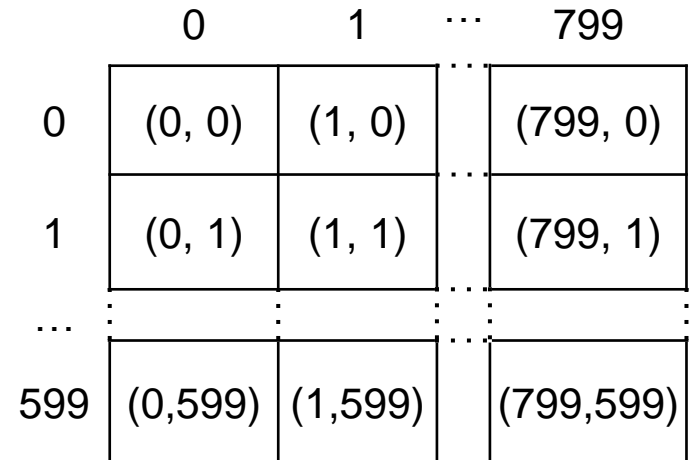
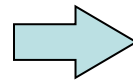
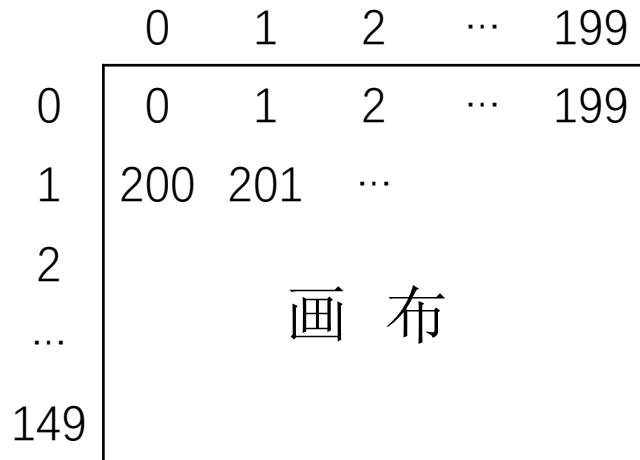
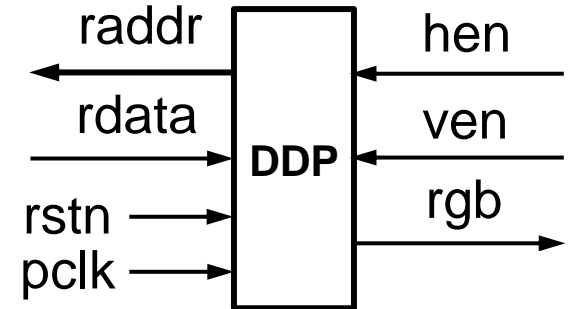
- 按标准的显示规格，产生刷新显示器的定时信号
 - 显示器分辨率：800x600
 - hs, vs: 行同步，场同步
 - hen, ven: 水平显示有效，垂直显示有效
 - pclk: 像素(pixel)时钟



	0	1	2	...	799
0	0	1	2	...	799
1	800	801	...		
2					
...					
599					

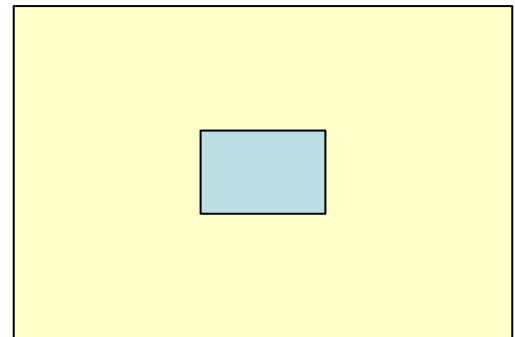
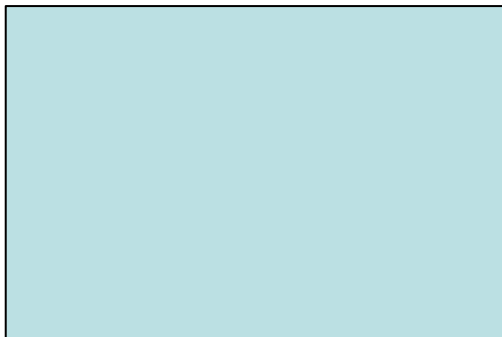
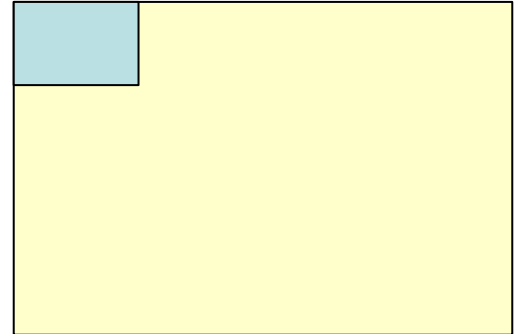
显示数据处理

- 将画布与显示屏适配
 - 画布分辨率：200x150
 - 根据hen、ven产生raddr
 - 根据hen、ven、rdata生成rgb



画布与显示屏适配

- 画布原始大小显示
 - 左上角（无偏移）
 - 正中央（有偏移）
- 画布整屏缩放显示



存储器IP核

- **Vivado**中有存储器IP核可以直接使用
- IP核类型：分布式（**Distributed**）、块式（**Block**）存储器
- 存储器类型：**ROM/RAM**、单端口/简单双端口/真正双端口等

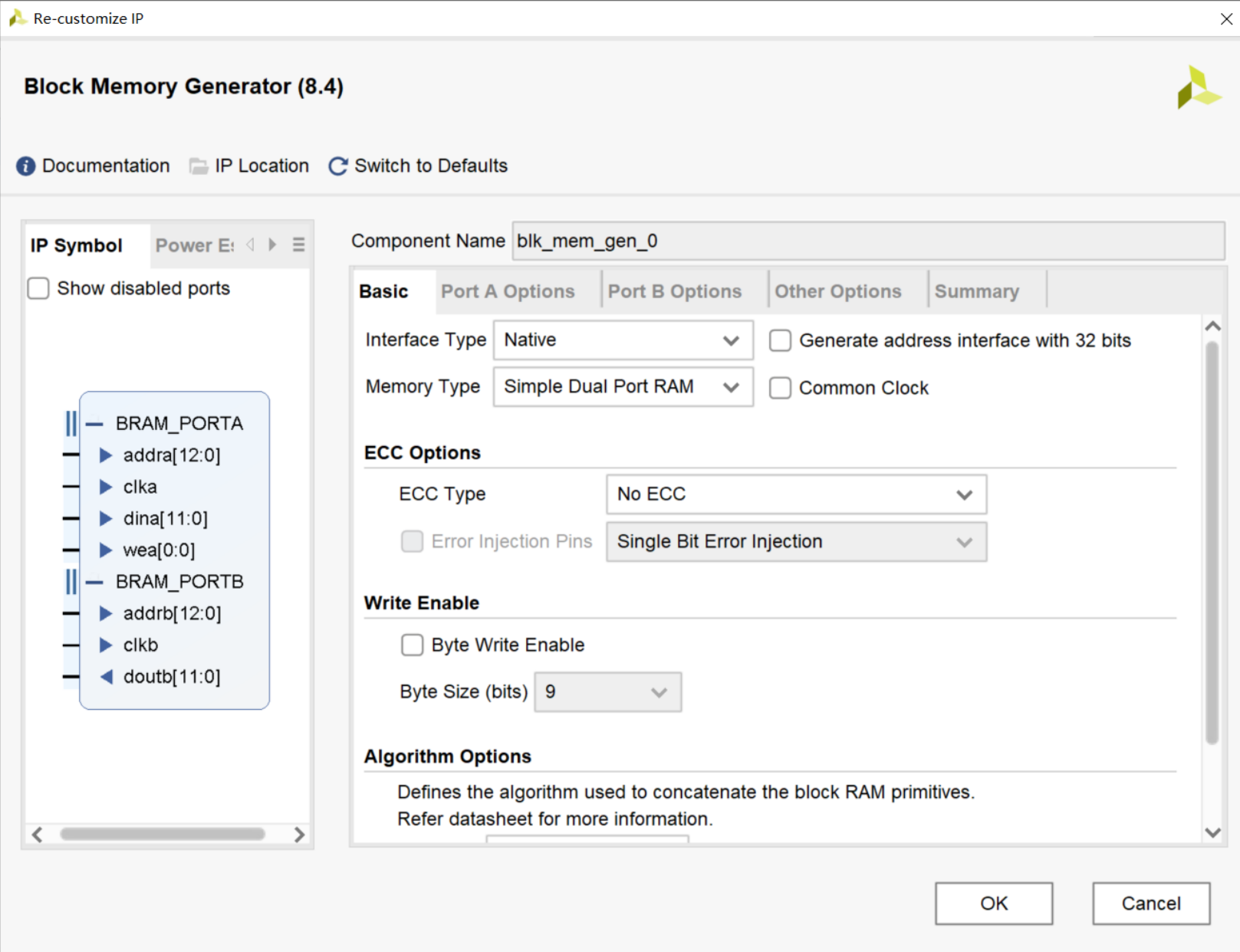
定制块式存储器IP核

- **Flow Navigator >> Project Manager >> IP Catalog**

- Memories & Storage Elements >> RAMs & ROMs >> Block Memory Generator
- 或者 Basic Elements >> Memory Elements >> Block Memory Generator
 - Basic >> Memory Type: [Simple Dual Port RAM](#)
 - Other Option >> Memory Initialization >> [Load Init File](#)

同步写端口: addra (地址), dina (数据), wea (写使能), clka

同步读端口: addrb (地址), doutb (数据), clkb



Re-customize IP

Block Memory Generator (8.4)

[Documentation](#) [IP Location](#) [Switch to Defaults](#)

IP Symbol

◀ ▶ ≡

☐ Show disabled ports

|| — BRAM_PORTA

▶ addra[12:0]

▶ clka

▶ dina[11:0]

▶ wea[0:0]

|| — BRAM_PORTB

▶ addrb[12:0]

▶ clkb

◀ doutb[11:0]

◀ ▶

Component Nameblk_mem_gen_0

Basic

Port A Options

Port B Options

Other Options

Summary

Memory Size

Port A Width12Range: 1 to 4608 (bits)

Port A Depth8192Range: 2 to 1048576

The Width and Depth values are used for Write Operations in Port A

Operating ModeWrite FirstEnable Port TypeAlways Enabled

Port A Optional Output Registers

☐ Primitives Output Register☐ Core Output Register

☐ SoftECC Input Register☐ REGCEA Pin

READ Address Change A

☐ Read Address Change A

OK

Cancel

Re-customize IP

Block Memory Generator (8.4)

[Documentation](#) [IP Location](#) [Switch to Defaults](#)

IP Symbol

☐ Show disabled ports

BRAM_PORTA

▶ addra[12:0]

▶ clka

▶ dina[11:0]

▶ wea[0:0]

BRAM_PORTB

▶ addrb[12:0]

▶ clkb

◀ doutb[11:0]

Component Name

blk_mem_gen_0

Basic

Port A Options

Port B Options

Other Options

Summary

Memory Size

Port B Width12

Port B Depth : 8192

The Width and Depth values are used for Read Operation in Port B

Operating ModeWrite First

Enable Port TypeAlways Enabled

Port B Optional Output Registers

☐ Primitives Output Register☐ Core Output Register

☐ SoftECC Output Register☐ REGCEB Pin

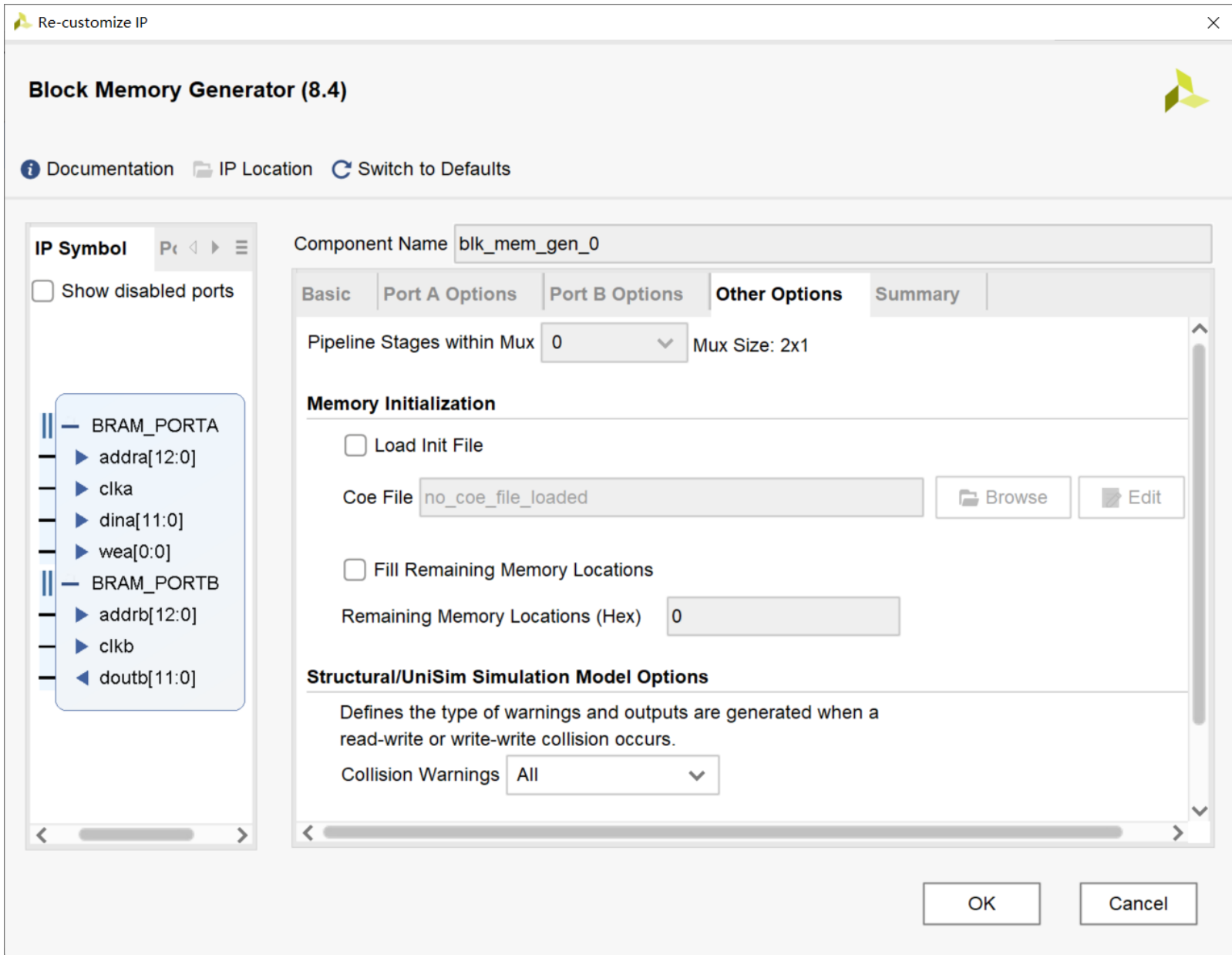
Port B Output Reset Options

☐ RSTB Pin (set/reset pin)Output Reset Value (Hex)0

☐ Reset Memory LatchReset PriorityCE (Latch or Register Enable)

OK

Cancel



COE文件格式

- **An example COE file:**

; Sample Initialization file for a 32x16 distributed ROM

memory_initialization_radix = 16;

memory_initialization_vector =

23f4 0721 11ff ABe1 0001 1 0A 0 逗号或空格分隔每项

23f4 0721 11ff ABe1 0001 1 0A 0 数据（不允许为负数）

23f4 721 11ff ABe1 0001 1 A 0

23f4 721 11ff ABe1 0001 1 A 0;

例化存储器IP核

- **Project Manager – display >> Sources >> IP Sources**
 - IP >> blk_mem_gen_0 >> Instantiation Template >> blk_mem_gen_0.vco

```
blk_mem_gen_0 your_instance_name (  
    .clka(clka),      // input wire clka  
    .wea(wea),        // input wire [0 : 0] wea  
    .addra(addra),    // input wire [14 : 0] addra  
    .dina(dina),      // input wire [11 : 0] dina  
    .clkb(clkb),      // input wire clkb  
    .addrb(addrb),    // input wire [14 : 0] addrb  
    .doutb(doutb)     // output wire [11 : 0] doutb  
);
```

时钟IP核

- 需要定制化频率的时钟，最好由时钟IP核（如 **Clocking Wizard**）生成
 - IP Catalog >> Vivado Repository >> FPGA Features and Design >> Clocking >> Clocking Wizard
 - Clocking Options: 选择MMCM 或者 PLL，在Input Clock Information设置输入时钟
 - Output Clocks: 设置输出时钟(MMCM: 4.68~800MHz, PLL: 6.250~800MHz)，以及复位和锁定(locked)选项

Customize IP

Clocking Wizard (6.0)

[Documentation](#) [IP Location](#) [Switch to Defaults](#)

IP Symbol

☐ Show disabled ports

resetclk_out1
clk_in1locked

Component Nameclk_wiz_1

Clocking Options

Output Clocks

Port Renaming

MMCM Settings

Summary

Clock Monitor

☐ Enable Clock Monitoring

Primitive

☒ MMCM ☐ PLL

Clocking Features

Jitter Optimization

OK

Cancel

2022/11/10

数字逻辑设计进阶实验

20

Customize IP

Clocking Wizard (6.0)

[Documentation](#) [IP Location](#) [Switch to Defaults](#)

IP Symbol

☐ Show disabled ports

resetclk_out1
clk_in1locked

Component Nameclk_wiz_1

Clocking OptionsOutput ClocksPort RenamingMMCM SettingsSummary

Dynamic Reconfig Interface
Options
☒ AXI4Lite ☐ DRP
☐ Phase Duty Cycle Config ☐ Write DRP registers

Input Clock Information

	Input Clock	Port Name	Input Frequency(MHz)		Jitter Op
<input checked="" type="checkbox"/>	Primary	clk_in1	100.000	10.000 - 800.000	UI
<input type="checkbox"/>	Secondary	clk_in2	100.000	76.800 - 153.600	

OKCancel

Customize IP

Clocking Wizard (6.0)

Documentation

IP Location

Switch to Defaults

IP Symbol

Show disabled ports

reset

clk_in1

clk_out1

locked

Component Name

clk_wiz_1

Clocking Options

Output Clocks

Port Renaming

MMCM Settings

Summary

The phase is calculated relative to the active input clock.

Output Clock	Port Name	Output Freq (MHz)		Phase (degrees)
		Requested	Actual	Requested
<input checked="" type="checkbox"/> clk_out1	clk_out1	1	10.000	0.000
<input type="checkbox"/> clk_out2	clk_out2	1		
<input type="checkbox"/> clk_out3	clk_out3	1		
<input type="checkbox"/> clk_out4	clk_out4	1		
<input type="checkbox"/> clk_out5	clk_out5	100.000	N/A	0.000

Validation failed on parameter 'CLKOUT1 REQUESTED OUT FREQ(CLKOUT1_REQUESTED_OUT_FREQ)' for Please enter valid freq in range (4.687 - 800.000) . IP 'clk_wiz_1'

OK

Cancel

Customize IP

Clocking Wizard (6.0)

[Documentation](#) [IP Location](#) [Switch to Defaults](#)

IP Symbol

☐ Show disabled ports

clk_in1

clk_out1

Component Name

Clocking Options

Output Clocks

Port Renaming

MMCM Settings

Summary

clk_out5	1
clk_out6	1
clk_out7	1

Enable Optional Inputs / Outputs for MMCM/PLL

☐ reset ☐ power_down ☐ input_clk_stopped ☐ locked ☐ clkfbstopped

Reset Type

☒ Active High ☐ Active

OK

Cancel

2022/11/10

数字逻辑设计进阶实验

23

时钟IP核例化模板

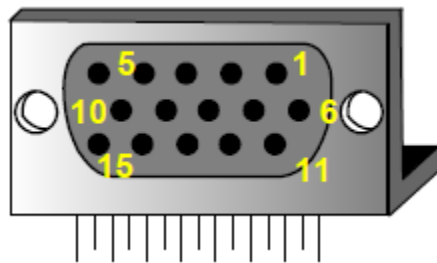
- **Project Manager – xxx >> Sources >> IP Sources**

- IP >> clk_wiz_0 >> Instantiation Template >> clk_wiz_0.veo

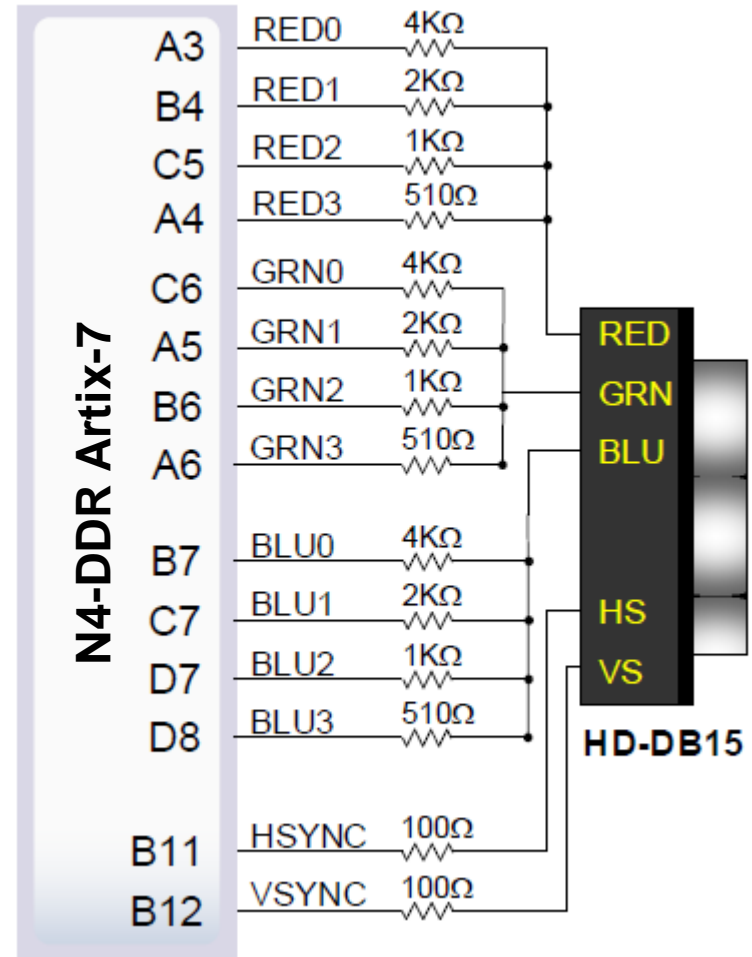
```
clk_wiz_0 instance_name (  
    .clk_out1(clk_out1),    // output clk_out1  
    .clk_in1(clk_in1)      // input clk_in1  
);
```


显示接口信号

- **RGB:** 像素颜色，红/绿/蓝各4位
- **HS:** 水平同步，也称行同步
- **VS:** 垂直同步，也称列/场/帧同步



Pin 1: Red	Pin 5: GND
Pin 2: Grn	Pin 6: Red GND
Pin 3: Blue	Pin 7: Grn GND
Pin 13: HS	Pin 8: Blu GND
Pin 14: VS	Pin 10: Sync GND

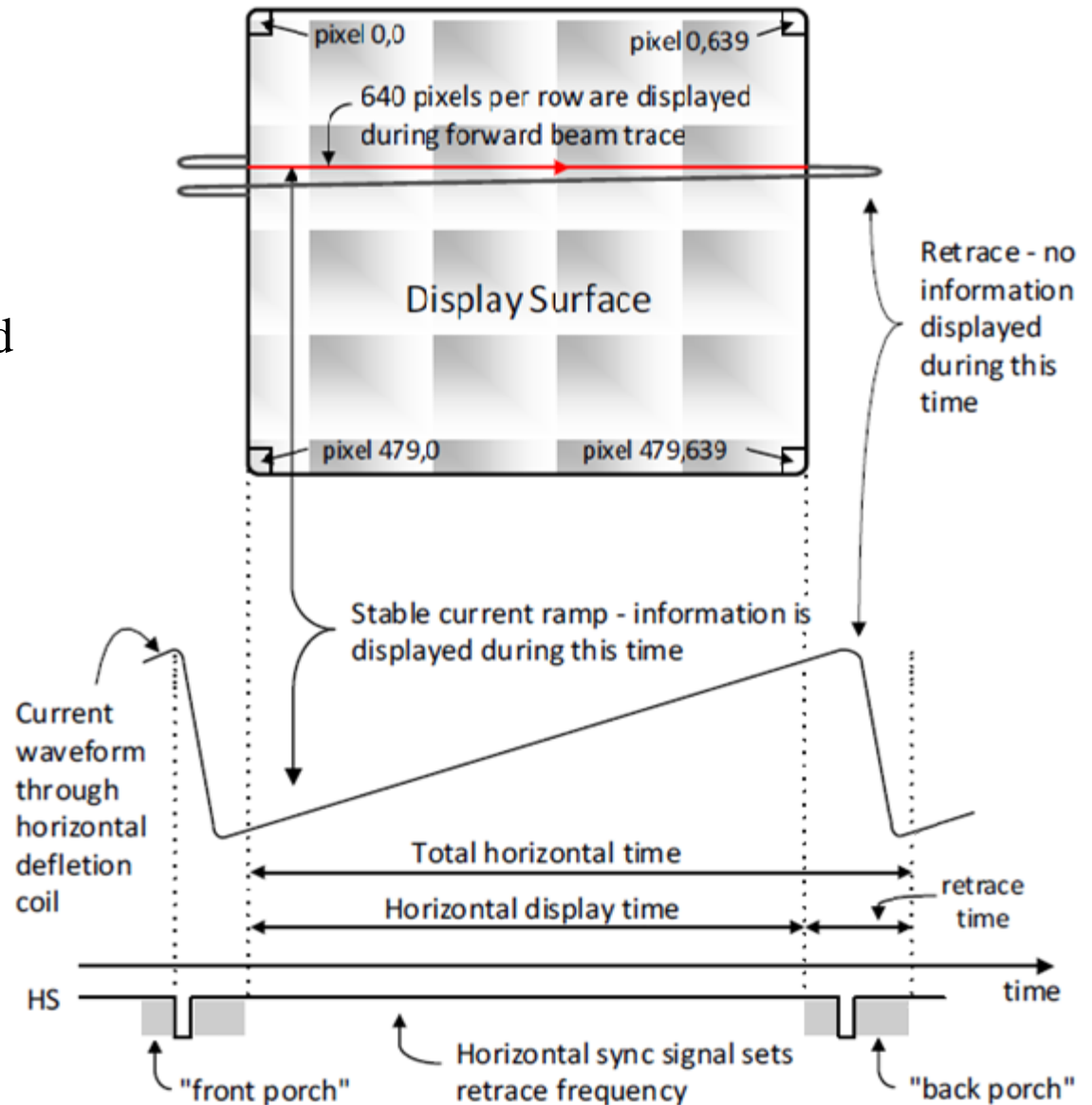


显示原理

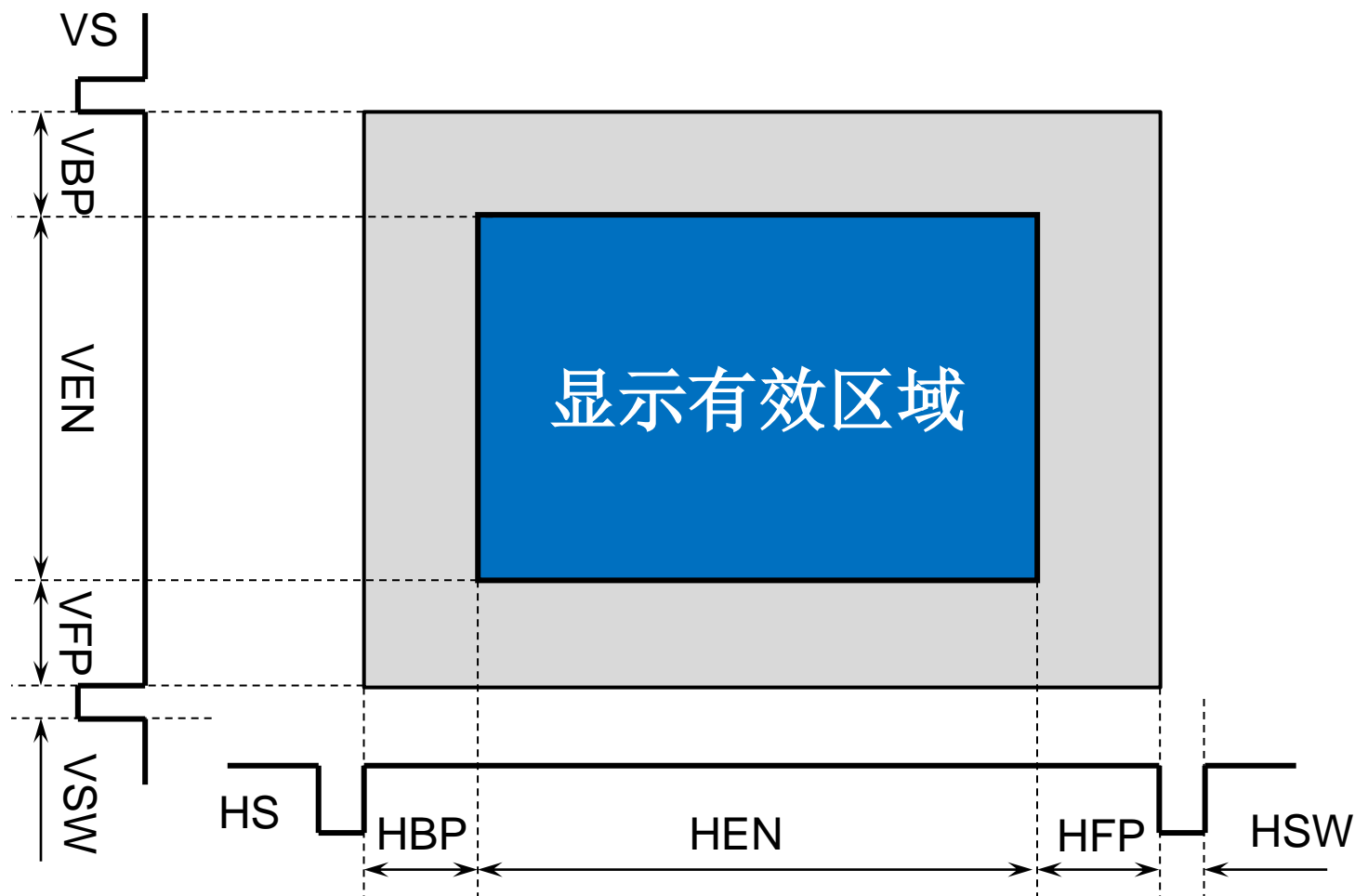
- 参考文档:

Nexys4 DDR FPGA Board
Reference Manual

(Nexys4-DDR_rm.pdf)



显示区域和定时参数



显示定时参数

- **HBP (horizontal back porch)**: 表示从水平同步信号开始到一行的有效数据开始之间的PCLK的个数，对应驱动中的left_margin;
- **HEN**: 水平显示有效区域，对应水平像素分辨率
- **HFP (horizontal front porch)**: 表示一行的有效数据结束到下一个水平同步信号开始之间的PCLK的个数，对应驱动中的right_margin;
- **HSW (horizontal sync width)**: 表示水平同步信号的宽度，以PCLK为单位计算，对应驱动中的hsync_len;
- **VBP (vertical back porch)**: 表示在一帧图像开始时，垂直同步信号以后的无效的行数，对应驱动中的upper_margin;
- **VEN**: 垂直显示有效区域，对应垂直像素分辨率
- **VFB (vertical front porch)**: 表示在一帧图像结束后，垂直同步信号以前的无效的行数，对应驱动中的lower_margin;
- **VSX (vertical sync width)**: 表示垂直同步脉冲的宽度，以行数为单位计算，对应驱动中的vsync_len

显示分辨率标准

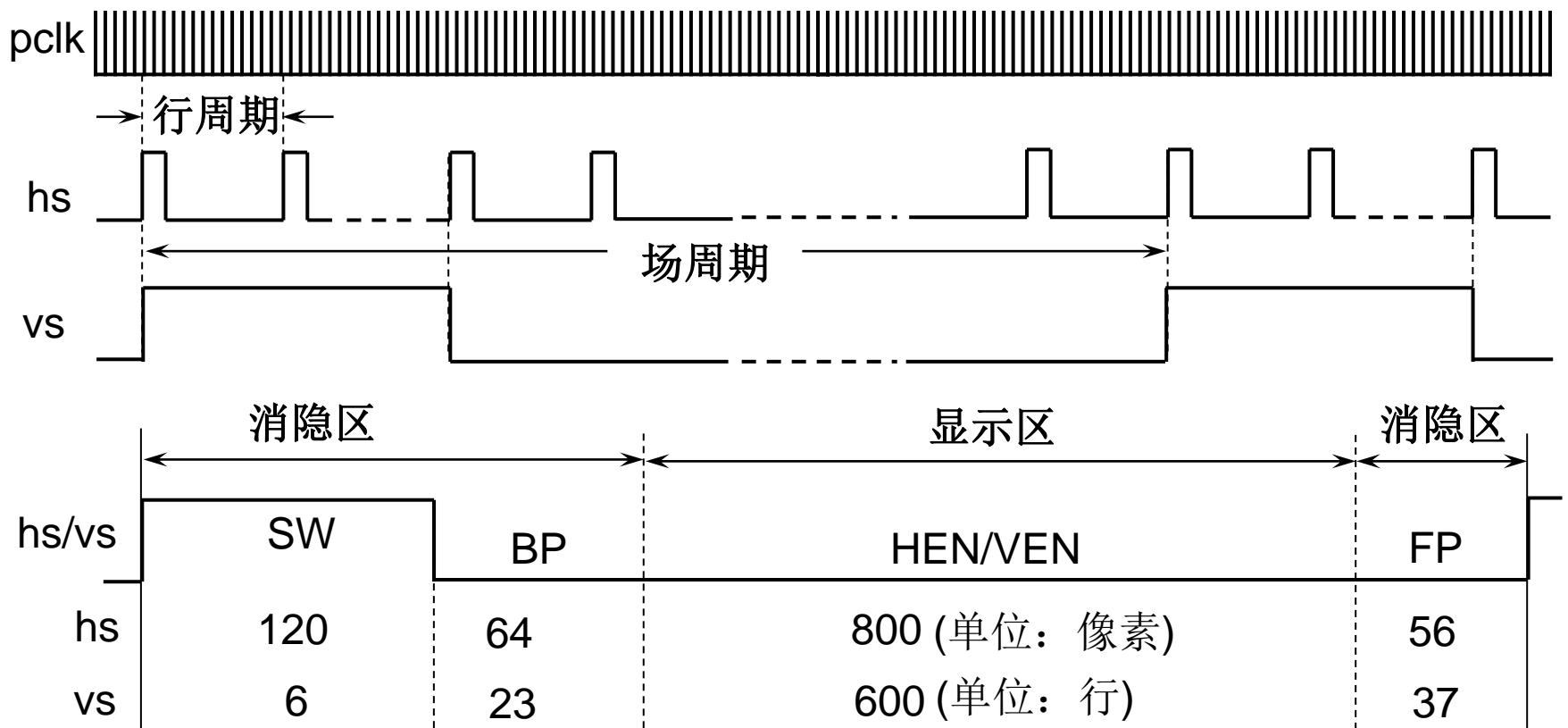
标屏	分辨率	宽屏	分辨率
QVGA	320×240	WQVGA	400×240
VGA	640×480	WVGA	800×480
SVGA	800×600	WSVGA	1024×600
XGA	1024×768	WXGA	1280×720, 1280×768, 1280×800
XGA+	1152×864	WXGA+	1366×768
SXGA	1280×1024, 1280×960	WSXGA	1440×900
SXGA+	1400×1050	WSXGA+	1680×1050
UXGA	1600×1200	WUXGA	1920×1200
QXGA	2048×1536	WQXGA	2560×1600

示例：显示定时参数

分辨率	像素频率 (MHz)	行/场同步极性	行总像素	行同步脉冲宽度	行同步后沿	行显示像素	行同步前沿	场总行数	场同步脉冲宽度	场同步后沿	场显示行数	场同步前沿
640x480@60	25.175	-/-	800	96	48	640	16	525	2	33	480	10
640x480@72	31.5	-/-	832	40	120	640	16	520	3	20	480	1
640x480@75	31.5	-/-	840	64	120	640	16	500	3	16	480	1
720x400@70	28.322	-/+	900	108	54	720	18	449	2	35	400	12
800x600@56	36	+/+	1024	72	128	800	24	625	2	22	600	1
800x600@72	50	+/+	1040	120	64	800	56	666	6	23	600	37
1024x768@60	65	-/-	1344	136	160	1024	24	806	6	29	768	3
1024x768@70	75	-/-	1328	136	144	1024	24	806	6	29	768	3

示例：SVGA显示时序

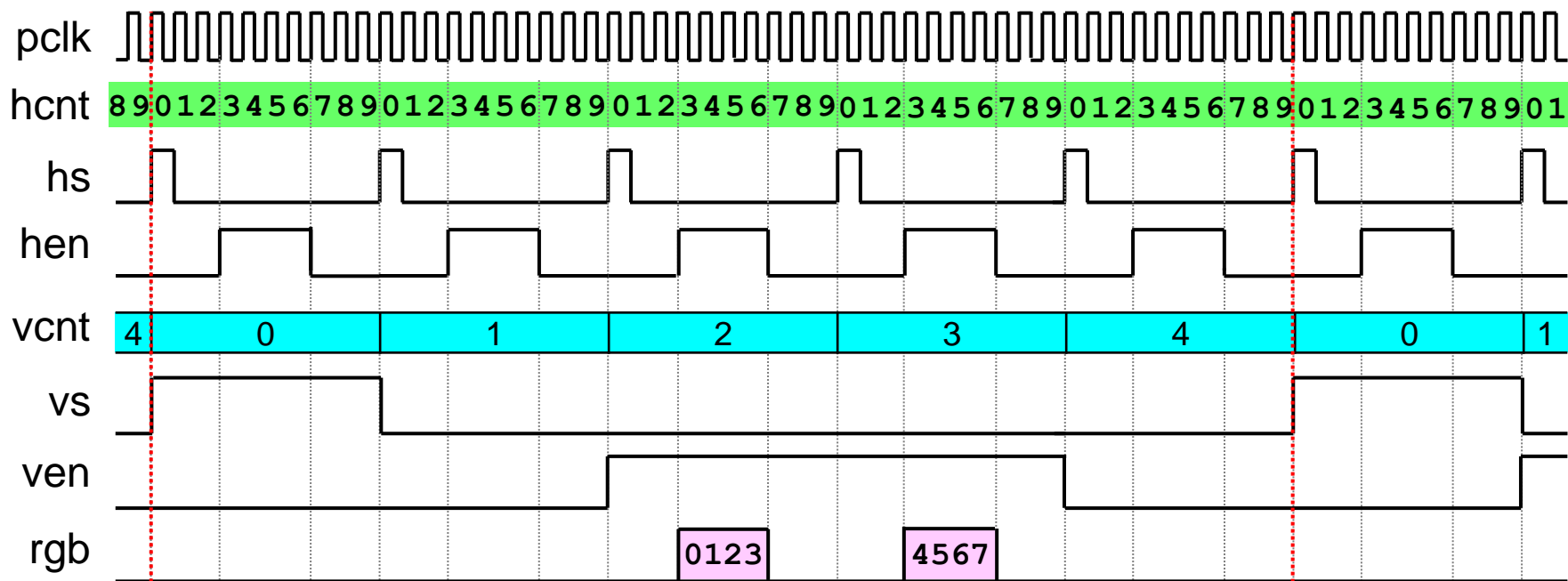
- 800x600 @72Hz, 50MHz像素时钟(pclk)



示例：显示仿真时序

- 假定显示定时参数如右表所示

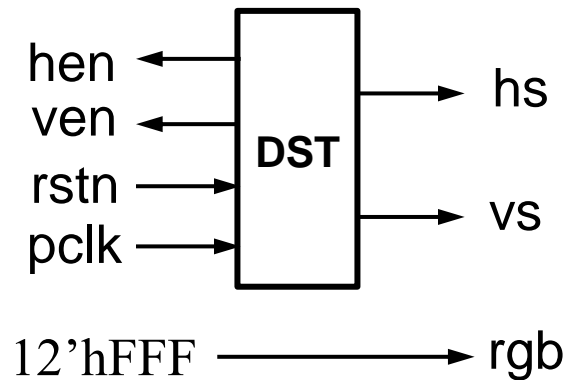
	SW	BP	EN	FP	单位
HS	1	2	4	3	像素
VS	1	1	2	1	行



实验步骤

1. 实现DST，显示白色背景

- 显示模式：800x600
@72Hz，50MHz像素
时钟(pclk)
- $rgb = 12'hFFF$

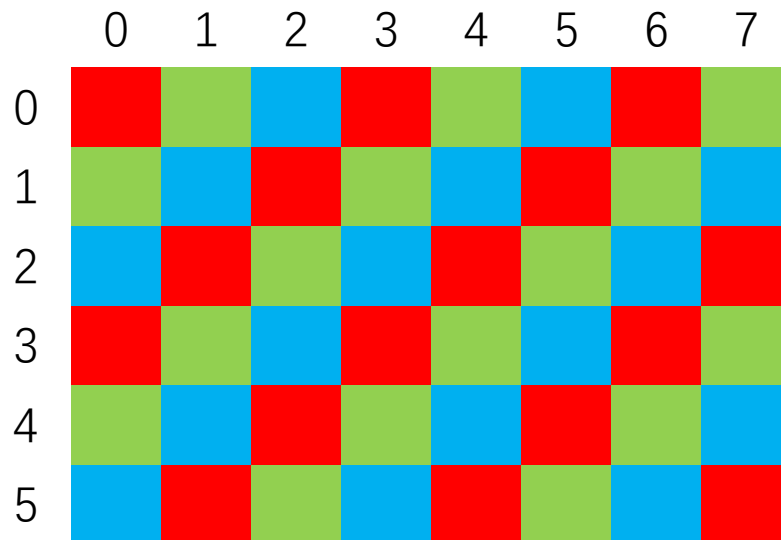
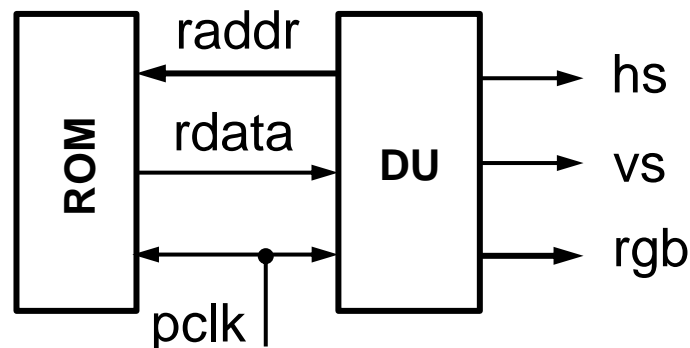


	0	1	2	...	799
0	0	1	2	...	799
1	800	801	...		
2					
...					
599					

实验步骤 (续1)

2. 实现DU，显示彩色方格图案

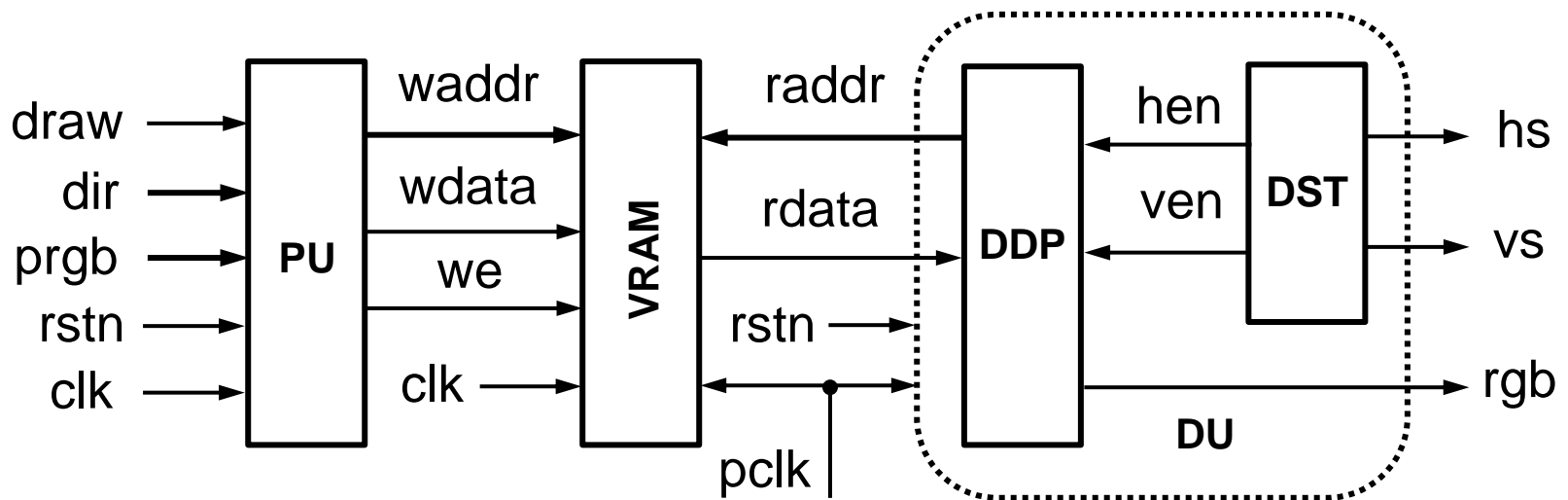
- 显示模式：800x600
@72Hz，50MHz像素时钟(pclk)
- ROM：存储图案，采用块式存储器，容量为64x2位
 - 颜色编码：0-红色，1-绿色，2-蓝色，3-白色



实验步骤 (续2)

3. 实现PU绘画

- 显示模式：800x600 @72Hz，50MHz像素时钟(pclk)
- VRAM：块式简单双端口存储器，容量为32Kx12位，初始化默认值为0xFF，即显示白色绘画背景



实验步骤 (续3)

4. 选做内容

- 复位清除绘画
- 画笔连续移动、 45° 移动、画任意直线
- 显示十字光标，指示画笔当前位置
- 实现其他分辨率的DU，如1024x768
- 设置画布背景为图片
- 显示输入文本数字
-

The End