

# 实验四 存储器与显示控制器

2019.4.18

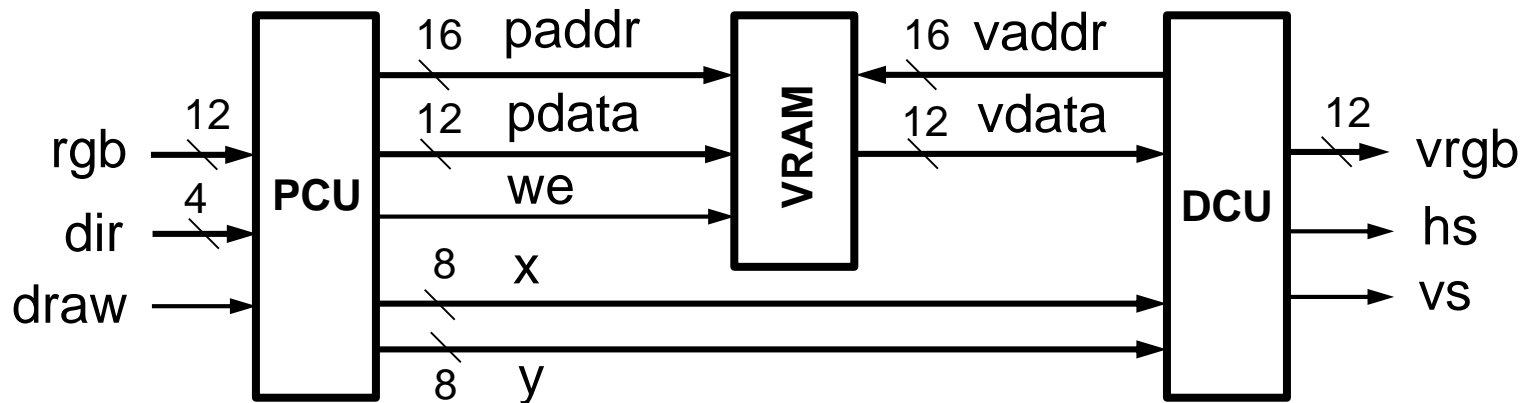
# 实验内容

1. 控制画笔在**800x600**分辨率的显示器上随意涂画，画笔的颜色**12位**(红**r**绿**g**蓝**b**各**4位**)，绘画区域位于屏幕正中部，大小为**256x256**
  - 画笔位置(**x, y**): **x = y = 0 ~ 255**, 复位时 (**128, 128**)
  - 移动画笔(**dir**): 上/下/左/右按钮
  - 画笔颜色(**rgb**): **12位**开关设置
  - 绘画状态(**draw**): **1-是, 0-否**;  
处于绘画状态时，移动画笔同时绘制颜色，否则仅移动画笔

	0	1	...	254	255
0	(0,0)	(0,1)	.....	(0,254)	(0,255)
1	(1,0)	(1,1)	.....	(1,254)	(1,255)
2			.....		
⋮	⋮	⋮	⋮	⋮	⋮
254			.....		
255			.....		(255, 255)

# 实验内容 (续1)

- **VRAM:** 视频存储器，存储**256x256**个像素的颜色信息，采用简单双端口存储器实现
  - **paddr, pdata, we:** 地址、数据、写使能，用于绘画的同步写端口
  - **vaddr, vdata:** 地址、数据，用于显示的异步读端口



(图中省略clk和rst信号)

# 实验内容 (续2)

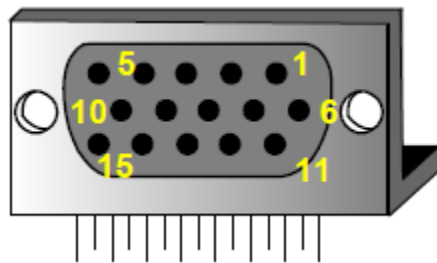
- **PCU: Paint Control Unit**, 绘画控制单元, 修改**VRAM**中像素信息
  - 通过**12**个拨动开关设置像素颜色 (**rgb**)
  - 通过上/下/左/右(**dir**)按钮开关, 移动画笔位置(**x, y**)
    - 直角移动: 单一按钮按下一次, **x**或**y**增加或减小**1**
    - 对角移动: 两按钮同时按下一次, **x**和**y**同时加或减**1**
    - 连续移动: 按钮按下超过**t**秒后, 等效为**s**速率的连续点击, 直至松开 (调试时确定合适的**t**和**s**取值)
  - 绘画 (**draw=1**) 时, 依据 **rgb** 和 (**x, y**), 通过写端口 (**paddr, pdata, we**) 修改**VRAM**像素信息

# 实验内容 (续3)

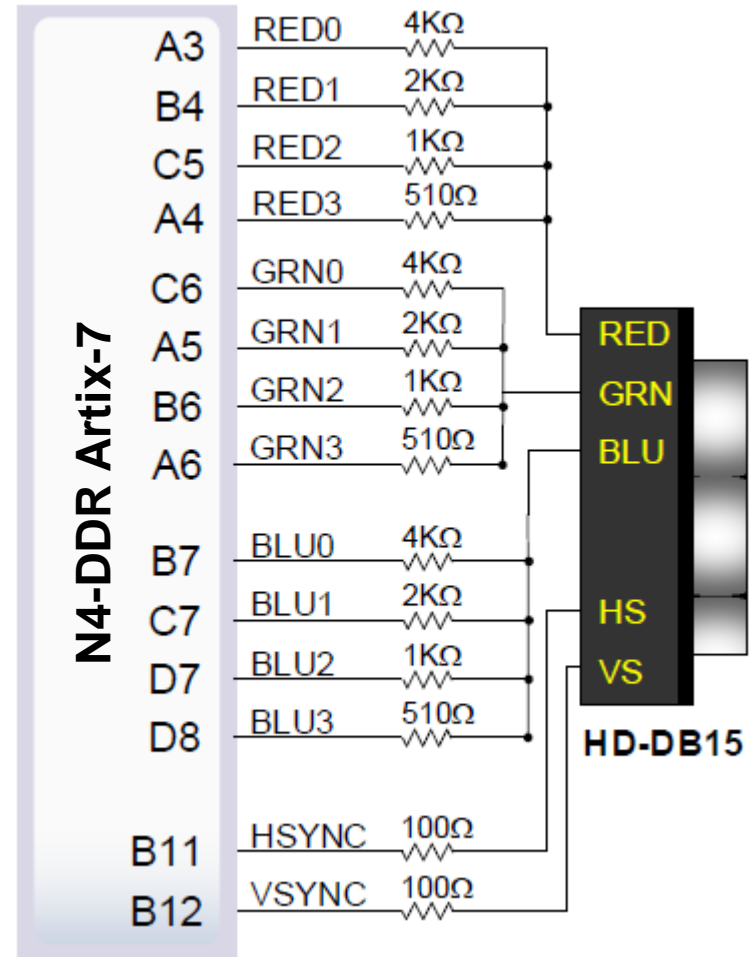
- **DCU: Display Control Unit**, 显示控制单元, 显示**VRAM**中像素信息
    - 通过读端口(**vaddr, vdata**)取出**VRAM**信息并显示
    - **vrgb, hs, vs**: 显示器接口信号
- 显示模式: 分辨率**800x600**, 刷新频率**72Hz**, 像素时钟频率**50MHz**
- VRAM**中的1个像素对应显示屏上1个像素
- 在屏幕上显示十字光标, 指示画笔当前位置 (**x, y**)

# 显示接口信号

- **RGB**: 像素点颜色信号, 红/绿/蓝各4位
- **HSYNC**: 水平同步信号, 也称行同步
- **VSYNC**: 垂直同步信号, 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

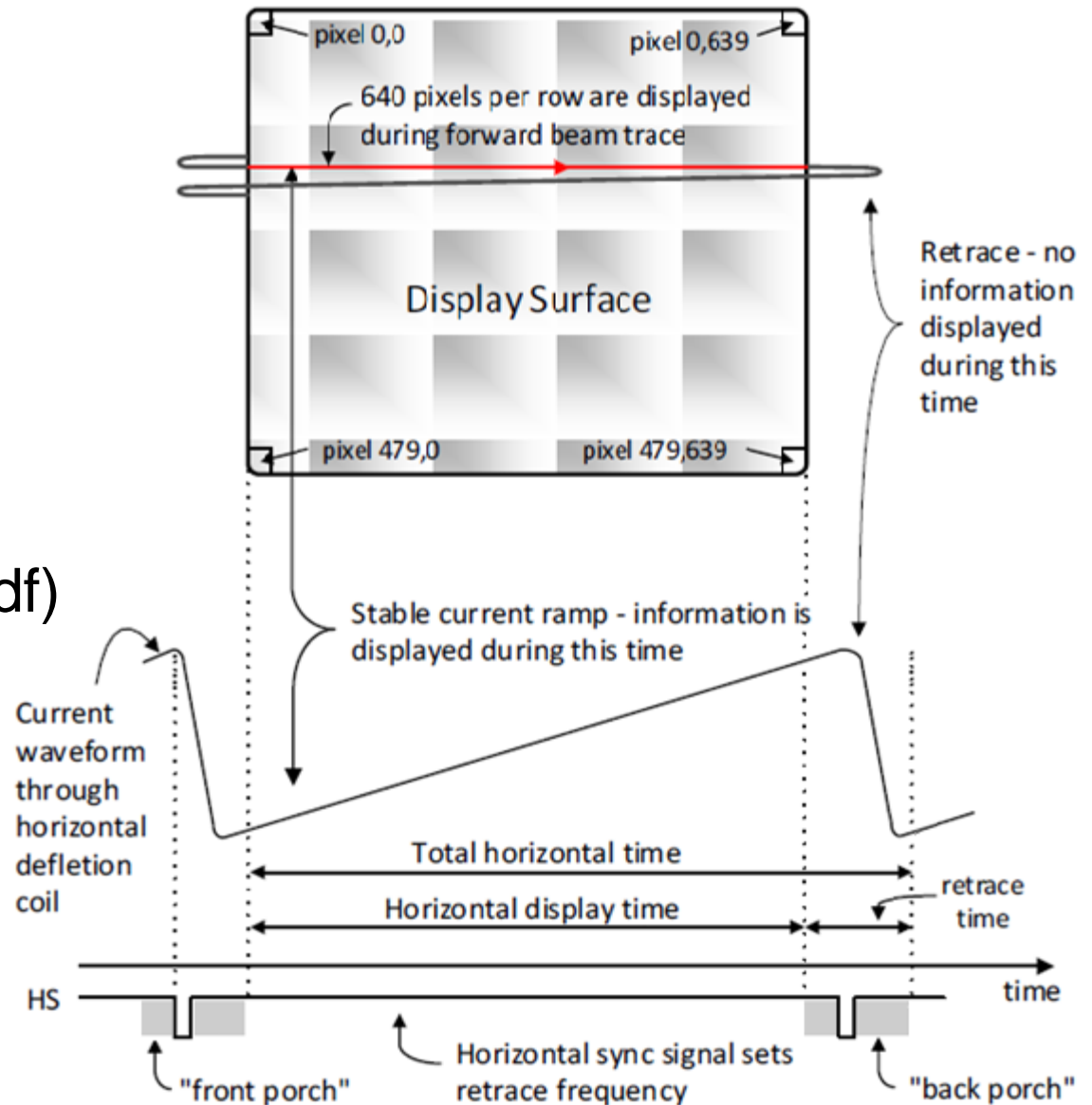


# 显示原理

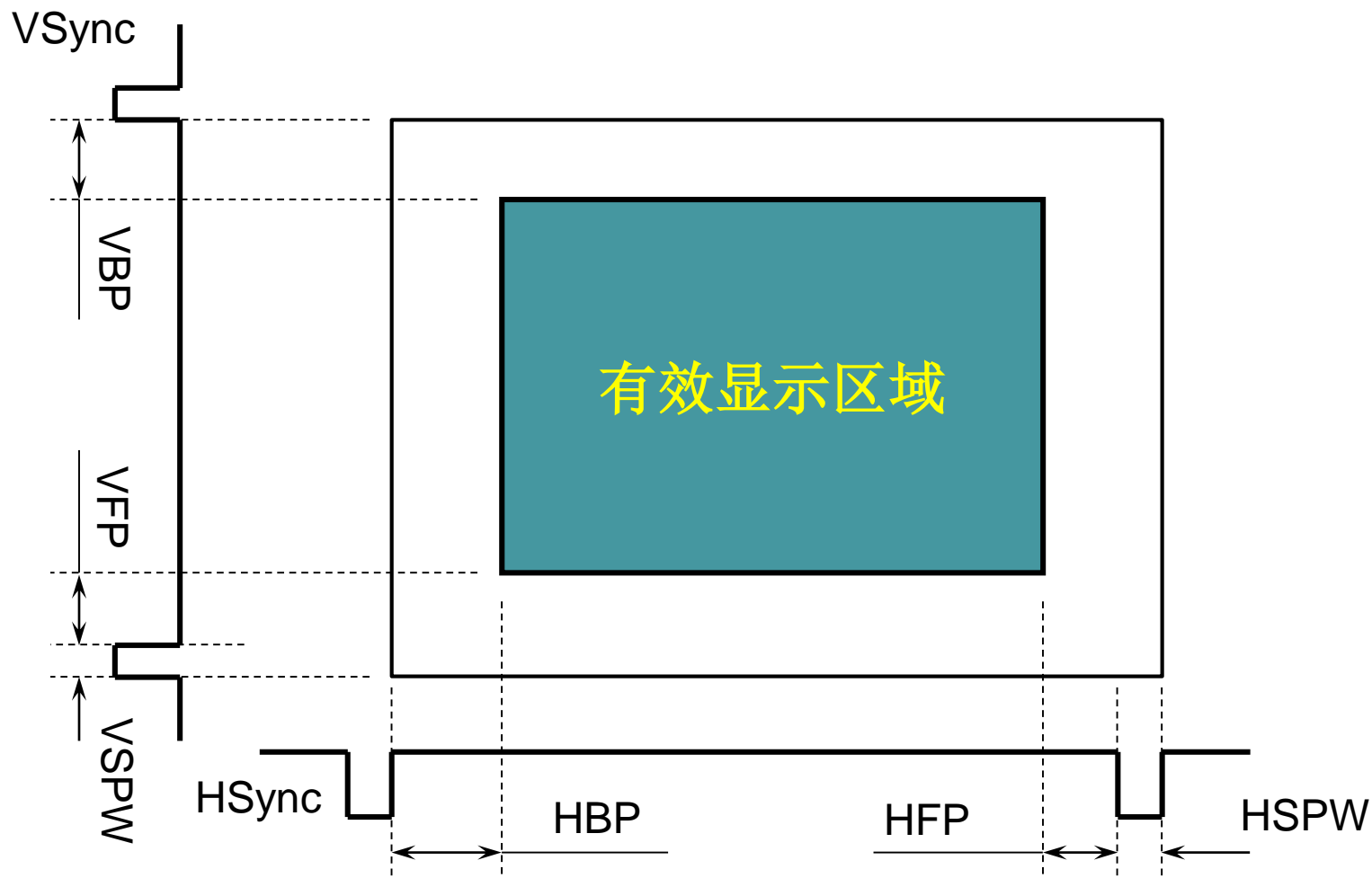
- 参考文档:

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**;
- **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**;
- **HSPW(horizontal sync pulse 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**;
- **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**;
- **VSPW(vertical sync pulse 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**;

# 显示器分辨率

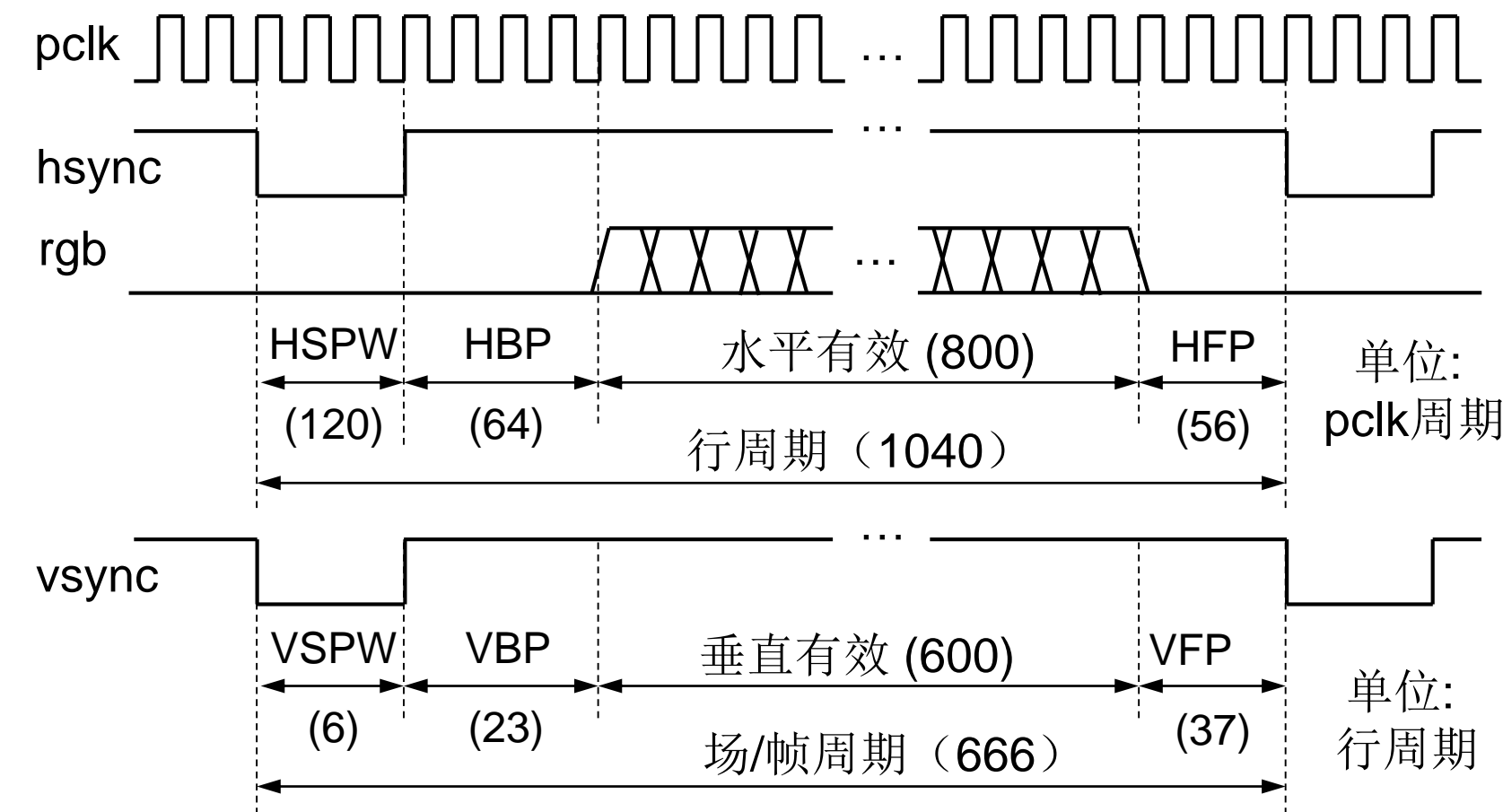
标屏	分辨率	宽屏	分辨率
QVGA	320×240	WQVGA	400×240
VGA	640×480	WVGA	800×480
SVGA	800×600	WSVGA	1024×600
XGA	1024×768	WXGA	1280×720/1280×768/1280×800
XGA+	1152×864	WXGA+	1366×768
SXGA	1280×1024/1280×960	WSXGA	1440×900
SXGA+	1400×1050	WSXGA+	1680×1050
UXGA	1600×1200	WUXGA	1920×1200
QXGA	2048×1536	WQXGA	2560×1600

# 显示标准定时参数

分辨率	像元频率 (兆赫兹)	每行像元数	行同步 (消隐) 脉冲开始时钟周期	行同步 (消隐) 脉冲结束时钟周期	每行时钟周期数	每帧图像行数	帧同步脉冲开始扫描行数	帧同步脉冲结束扫描行数	每帧扫描行数
640x480@60	25.2	640	656	752	800	480	490	492	525
640x480@72	31.5	640	664	704	832	480	489	491	520
640x480@75	31.5	640	656	720	840	480	481	484	500
640x480@85	36.0	640	696	752	832	480	481	484	509
800x600@56	36.0	800	824	896	1024	600	601	603	625
800x600@72	50.0	800	856	976	1040	600	637	643	666
800x600@75	49.5	800	816	896	1056	600	601	604	625
800x600@85	56.3	800	832	896	1048	600	601	604	631
800x600@60	40.0	800	840	968	1056	600	601	605	628

# SVGA显示定时参数

- 800x600 @72Hz, 50MHz像素时钟(pclk)



# 存储器IP实例化

- Flow Navigator >> Project Manager >> IP Catalog
  - Memories & Storage Elements >> RAMs & ROMs >> Distributed Memory Generator
  - 或者 Basic Elements >> Memory Elements >> Distributed Memory Generator
    - Memory config >> Memory Type: [Simple Dual Port RAM](#)
    - RST & Initialization >> [Load COE File](#)

同步写端口：a (地址), d (数据), we (写使能), clk

异步读端口：dpra (地址), dpo (数据)

# 存储器IP实例化 (续)

- Project Manager – display >> Sources >> IP Sources
  - IP >> dist\_mem\_gen\_0 >> Instantiation Template >> dist\_mem\_gen\_0.vco

```
dist_mem_gen_0  your_instance_name (  
  .a(a),          // input wire [14 : 0] a  
  .d(d),          // input wire [11 : 0] d  
  .dpra(dpra),    // input wire [14 : 0] dpra  
  .clk(clk),      // input wire clk  
  .we(we),        // input wire we  
  .dpo(dpo)       // output wire [11 : 0] dpo  
);
```

实例化模板

# 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
23f4 0721 11ff ABe1 0001 1 0A 0
23f4 721 11ff ABe1 0001 1 A 0
23f4 721 11ff ABe1 0001 1 A 0;
```

逗号或空格分隔  
每项数据 (不允许为负数)

# 实验要求和检查

- 完成实验内容1的逻辑设计和下载测试
  - 加分选项：实现PCU的画笔对角和连续移动处理，DCU的十字光标显示功能
- 查看电路性能和资源使用情况
- 检查仿真结果是否正确
- 检查下载测试是否正确
- 检查代码设计，代码是否独立完成



# 实验报告

- 内容包括但不限于：逻辑设计、核心代码、仿真/下载结果、结果分析、实验总结、意见/建议等，附设计和仿真代码
- 实验检查后一周内提交实验报告
  - <ftp://202.38.79.134/> 相应文件夹
  - 文件名格式：Labn\_学号\_姓名\_vi.pdf （其中，n表示第n次实验，vi表示第i版本，不满足该格式的视为未提交实验报告）
- 严禁抄袭，否则作零分处理

# The End