

## 存储器实验日志

### 一、实验代码

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_unsigned.all;
4  entity sw_pc_ar is
5  port (clk_cdu,pcclr,pcld,pcen:in std_logic;
6        sw_bus,pc_bus,ldar:in std_logic;
7        inputd: in std_logic_vector(7 downto 0);
8        arout:out std_logic_vector(7 downto 0);
9        d :inout std_logic_vector(7 downto 0));
10 end sw_pc_ar;
11 architecture rtl of sw_pc_ar is
12 signal pc,ar,bus_Reg:std_logic_vector(7 downto 0);
13 begin
14 seq1:process(clk_cdu,ldar,bus_Reg)
15 begin
16     if clk_cdu'event and clk_cdu='1' then
17         if ldar='1' then
18             ar<=bus_Reg;
19         end if;
20     end if;
21 end process;
22 seq2:process(clk_cdu,pcclr,pcld,pcen,bus_Reg)
```



End Time

×

Time:  us

Default extension options:

Extension value: Last clock pattern

End time extension per signal:

Signal Name	Direction	Radix	Extension value
clk_cdu	Input	Hexadecimal	Default extension value
+ swlpc_bus	Group	Binary	Default extension value
+ m wlr	Group	Binary	Default extension value
ldar	Input	Hexadecimal	Default extension value
+ pc_clr dlen	Group	Binary	Default extension value
+ inputd	Input	Hexadecimal	Default extension value
+ d	Bidir	Hexadecimal	Default extension value
+ d~result	Group	Hexadecimal	Default extension value
+ sw_pc_ar:inst1 pc	Buried	Hexadecimal	Default extension value
+ sw_pc_ar:inst1 ar	Buried	Hexadecimal	Default extension value

OK

Cancel

Grid Size

×

Base grid on

☐ Clock settings:

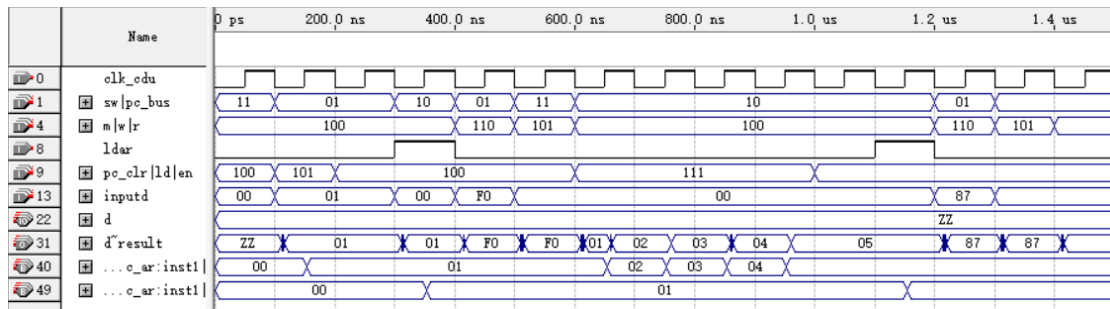
☒ Time period:

Period:  ns

OK

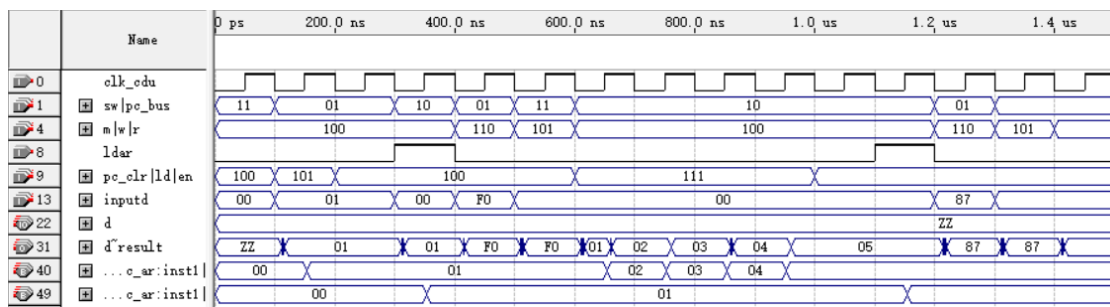
Cancel

- (2) 记录芯片设置及引脚设置。
- (3) 记录电路初始状态时 input 输入信号设置。



inputd 置为 00, sw\_bus,pc\_bus 低电平有效, 置为 11, m 为片选信号, 置为 1, w|r 分别为写|读信号, 置为 00, ldar 高电平有效, 置为 0, pc\_clr|ld|en 置为 100, 使 PC 处于保持状态。

(4) 记录全部实验内容中出现的数。



0~100ns,设置初始状态;

## 置数法

100~200ns,使 PC 处于置数状态, 将外部输入 01 传入总线, 再写入 PC;  
 200~300ns,使 PC 处于保持状态;  
 300~400ns,使 PC 处于保持状态, 关闭 sw\_bus,打开 pc\_bus,使 pc 数据出现在总线上, 将 ldar 置 1, 将总线数据写入 ar;  
 400~500ns,打开 sw\_bus, 将外部输入 F0 传入总线再写入 RAM 中 01 单元;  
 500~600ns,关闭 sw\_bus, 将 RAM 中 01 单元数据读出送入总线;

## 计数法

600~1.0us,使 PC 处于计数状态, 打开 pc\_bus,观察 pc 中数据;  
 1.0~1.1us,使 PC 处于保持状态;  
 1.1~1.2us,使 PC 处于保持状态,将 ldar 置 1, 将总线数据写入 ar;  
 1.2~1.3us,打开 sw\_bus, 将外部输入 87 传入总线再写入 RAM 中 05 单元;  
 1.3~1.4us,关闭 sw\_bus, 将 RAM 中 05 单元数据读出送入总线;  
 1.4us~4us,停机。

## 四、思考题

- 1) LPM\_RAM\_IO 参数设置中的 LPM\_NUMWORDS、LPM\_WIDTH、LPM\_WIDTHAD 分别代表什么含义?如何设置?  
 LPM\_NUMWORDS 表示字节数量, LPM\_WIDTH 表示输出数据宽度, LPM\_WIDTHAD 表示输入地址宽度。

- 2) LPM\_RAM\_IO 参数设置中的 lpm\_File 含义是什么?如何编写此类文件?  
lpm\_File 为 mif 文件或其他类型的记忆文件, 用来以数组的方式存储指令, 通过新建 mif 文件在其中设置即可。
- 3) 1024\*8b 的 RAM 应有几根地址线?存储单元为 4b 的 RAM 其存储容量为多少?  
由于  $2^{10}=1024$ , 所以至少需要 10 根, 256x4b。
- 4) 地址寄存器的数据源一般是哪些器件?  
pc 自身, 总线, ir 寄存器, alu 运算器。
- 5) 地址计数器 PC 如何用置数法产生地址?操作过程中为何需要经过一个保持状态?  
通过置数状态保持运行置数指令, 由此来实现在置数法过程中不会因 pc 改变而影响到结果。
- 6) 如何修改电路使其能连续读出存入连续地址单元中存放的数据?  
将 pc 的控制端设置为 111, 即计数状态。
- 7) 如何将原理图输入的逻辑电路转换成 HDL 语言描述的元件?  
通过 creat 指令实现。
- 8) 分析仿真结果图 4-3-14(P241)中 200ns 后执行读 M【01】中数据时为何会先出现 E1?  
因为 RAM 的地址输入端开始时为 00H, 时钟信号上升沿后更新为 01H, 00H 中存储的数据为 E1。