

实验四 模型机时序部件的实现

班级 计科 1808 姓名 张继伟 学号 201808010829

一、实验目的

1. 熟悉计数器、寄存器和 RAM 的工作原理。
2. 了解模型机中 SM 的作用。
3. 学会使用 VHDL 语言设计时序电路。

二、实验内容

1. 用 VHDL 语言设计 SM;
2. 用 VHDL 语言设计一个 8 位的指令计数器 PC;
3. 用 VHDL 语言设计 3 个 8 位寄存器组成的寄存器组, 实现读写操作。
4. 用 LPM_RAM_IO 定制一个 256*8 的 RAM, 实现对 RAM 的读写操作。

三、实验方法

1、实验方法

采用基于 FPGA 进行数字逻辑电路设计的方法。
采用的软件工具是 Quartus II。

2、实验步骤

1、新建, 编写源代码。

(1).选择保存项和芯片类型: **【File】 - 【new project wizard】 - 【next】** (设置文件路径+设置 project name 为 xor2) - **【next】** (设置文件名 xor2.vhd—在**【add】**)-**【properties】**(type=AHDL) - **【next】**(family=FLEX10K; name=EPF10K10TI144-4) - **【next】** - **【finish】**

(2).新建: **【file】 - 【new】** (第二个 AHDL File) - **【OK】**

2、写好源代码, 保存文件 (xor2.vhd)。

3、编译与调试。确定源代码文件为当前工程文件, 点击 **【processing】 - 【start compilation】** 进行文件编译, 编译成功。

4、波形仿真及验证。新建一个 vector waveform file。按照程序所述插入 a,b,c 三个节点 (a、b 为输入节点, c 为输出节点)。(操作为: 右击 - **【insert】 - 【insert node or bus】 - 【node finder】** (pins=all; **【list】**) - **【>>】 - 【ok】 - 【ok】**)。任意设置 a,b 的输入波形...点击保存按钮保存。(操作为: 点击 name (如: A)) - 右击 - **【value】 - 【clock】** (如设置 period=200; offset=0), 同理设置 name B (如 120, ,60), 保存)。然后 **【start simulation】**, 出 name C 的输出图。

5、时序仿真或功能仿真。

6、查看 RTL Viewer: **【Tools】 - 【netlist viewer】 - 【RTL viewer】**。

四、实验过程

SM

1、编译过程

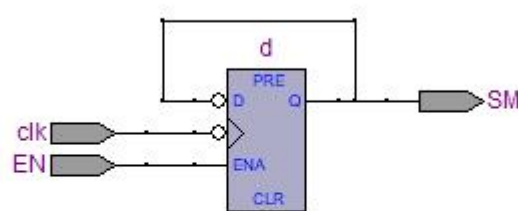
a) 源代码如图（VHDL 设计）

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_unsigned.all;
4
5  entity zjw_SM is
6  port(EN: in std_logic;
7        clk: in std_logic;
8        SM: out std_logic);
9  end zjw_SM;
10
11 architecture behavioral of zjw_SM is
12     signal d: std_logic:='0';
13 begin
14     process(clk)
15     begin
16         if (clk='0' and clk'event and EN='1') then
17             d <= not d;
18         else d <= d;
19         end if;
20     end process;
21     SM <= d;
22 end behavioral;
```

b) 编译、调试过程

Type	Message
Warning	(10492): VHDL Process Statement warning at zjw_SM.vhd(16): signal "EN" is read inside the Process Statement but isn't in the Process Statement's sensitivity list
Warning	(10492): VHDL Process Statement warning at zjw_SM.vhd(18): signal "d" is read inside the Process Statement but isn't in the Process Statement's sensitivity list
Warning	Feature LogicLock is not available with your current license
Warning	No exact pin location assignment(s) for 3 pins of 3 total pins
Warning	The Reserve All Unused Pins setting has not been specified, and will default to 'As output driving ground'.
Warning	Found pins functioning as undefined clocks and/or memory enables

c) RTL 视图



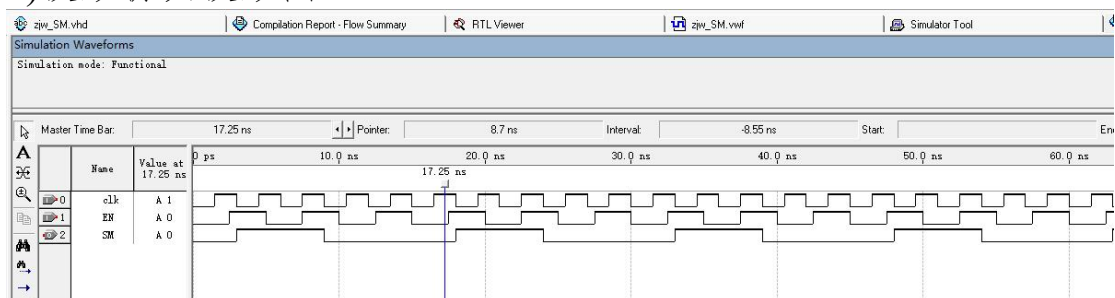
d) 结果分析及结论

当 $SM=0$ 时，该周期完成取指令，当 $SM=1$ 时，该周期执行指令。下降沿触发，根据使能 EN 决定 SM 是取反还是保持， $EN=1$ ，取反， $EN=0$ 保持不变。

2、波形仿真

a) 波形仿真过程（详见实验步骤）

b) 波形仿真波形图



c) 结果分析及结论

以 10ns 为一周期， SM 初始为 0，得到结果均正确：

在 10ns 时，下降沿触发， $EN=0$ ， SM 保持不变

在 20ns 时，下降沿触发， $EN=1$ ， SM 取反

在 30ns 时，时钟和使能同时变化，不触发， SM 保持不变

在 40ns 时，下降沿触发， $EN=0$ ， SM 保持不变

在 50ns 时，下降沿触发， $EN=1$ ， SM 取反

3、时序仿真

a) 时序仿真过程

做好上述步骤后，编译【classic timing analysis】-在 compilation report 中选择【timing analysis】-【tpd】（引脚到引脚的延时）

计数器 PC

1, 编译过程

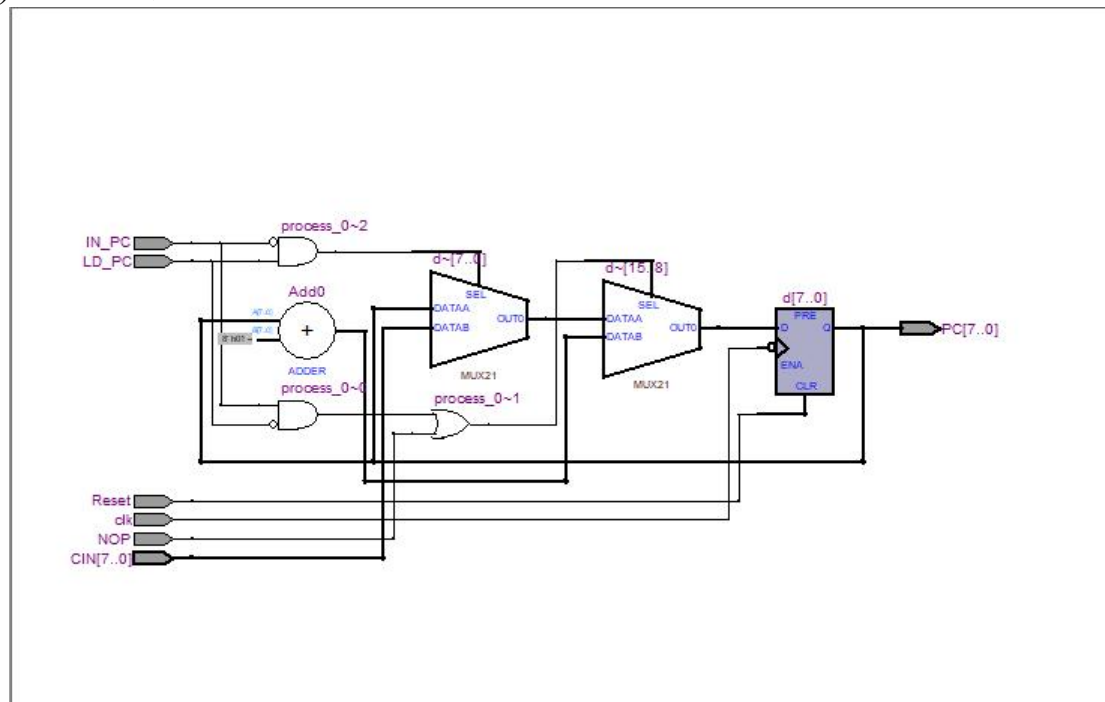
a) 源代码如图 (VHDL 设计)

```
zjw_PCjishuqi.vhd
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_unsigned.all;
4
5  entity zjw_PCjishuqi is
6  port (CIN: in std_logic_vector(7 downto 0);
7        IN_PC, LD_PC: in std_logic;
8        Reset,clk,NOP: in std_logic;
9        PC: out std_logic_vector(7 downto 0));
10 end zjw_PCjishuqi;
11
12 architecture behavioral of zjw_PCjishuqi is
13 signal d: std_logic_vector(7 downto 0):="00000000";
14 begin
15 process(LD_PC,Reset,clk,NOP)
16 begin
17 if Reset='1' then
18 d <= "00000000";
19 elsif clk='0' and clk'event then
20 if NOP='1' or (LD_PC='0' and IN_PC='1') then
21 d<=d+1;
22 elsif LD_PC='1' and IN_PC='0' then
23 d<=CIN;
24 end if;
25 end if;
26 end process;
27 PC <= d;
28 end behavioral;
```

b) 编译、调试过程

Type	Message
Warning	Warning: Feature LogicLock is not available with your current license
Warning	Warning: No exact pin location assignment(s) for 21 pins of 21 total pins
Warning	Warning: The Reserve All Unused Pins setting has not been specified, and will default to 'As output driving ground'.
Warning	Warning: Found pins functioning as undefined clocks and/or memory enables

c) RTL 视图



结果分析及结论

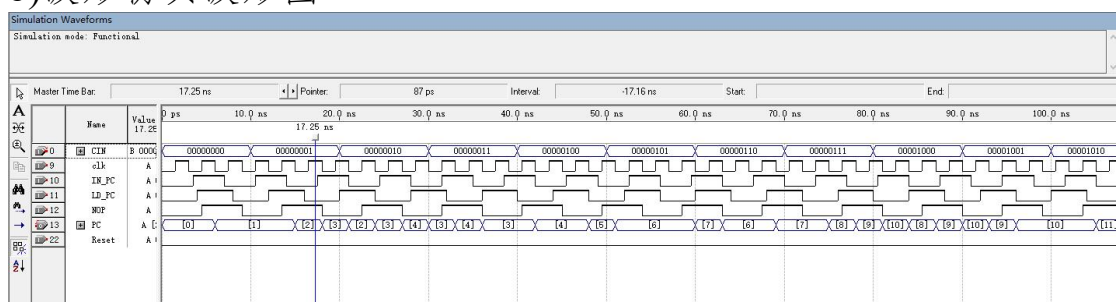
当执行一条指令时，首先需要根据 PC 中存放的指令地址，将指令由 RAM 读取至指令寄存器 IR 中，此过程称为“取指令”，与此同时，PC 中的地址自动加 1。跳转指令如 JMP、JZ、JC 让程序跳转至指定地址去执行，这时 PC 需要装载跳转地

址。

2.波形仿真

a)波形仿真过程（详见实验步骤）

b)波形仿真波形图



b) 结果分析及结论

Reset 首先置为 1，之后为 0，保证时钟信号有效

一开始 Reset 设置为 1，对整个加法器进行重置。

在 IN PC 和 LD PC，取值相反时，计数器有操作，在时钟下降沿触发。

IN PC=1 和 LD PC=0 时，下降沿触发计数器 $PC \leq PC+1$ 。

IN PC=1 和 LD PC=0 时，下降沿触发计数器 $PC \leq CIN$ 。

以上结果符合预期，正确。

3.时序仿真

c) 时序仿真过程

做好上述步骤后，编译【classic timing analysis】-在 compilation report 中选择【timing analysis】-【tpd】（引脚到引脚的延时）

b)时序仿真图



d) 结果分析及结论

时序仿真不仅反应出输出和输入的逻辑关系，同时还计算了时间的延时信息，是与实际系统更接近的一种仿真结果。不过，要注意的是，这个时间延时是仿真软件“估算”出来的。

寄存器组

1.编译过程

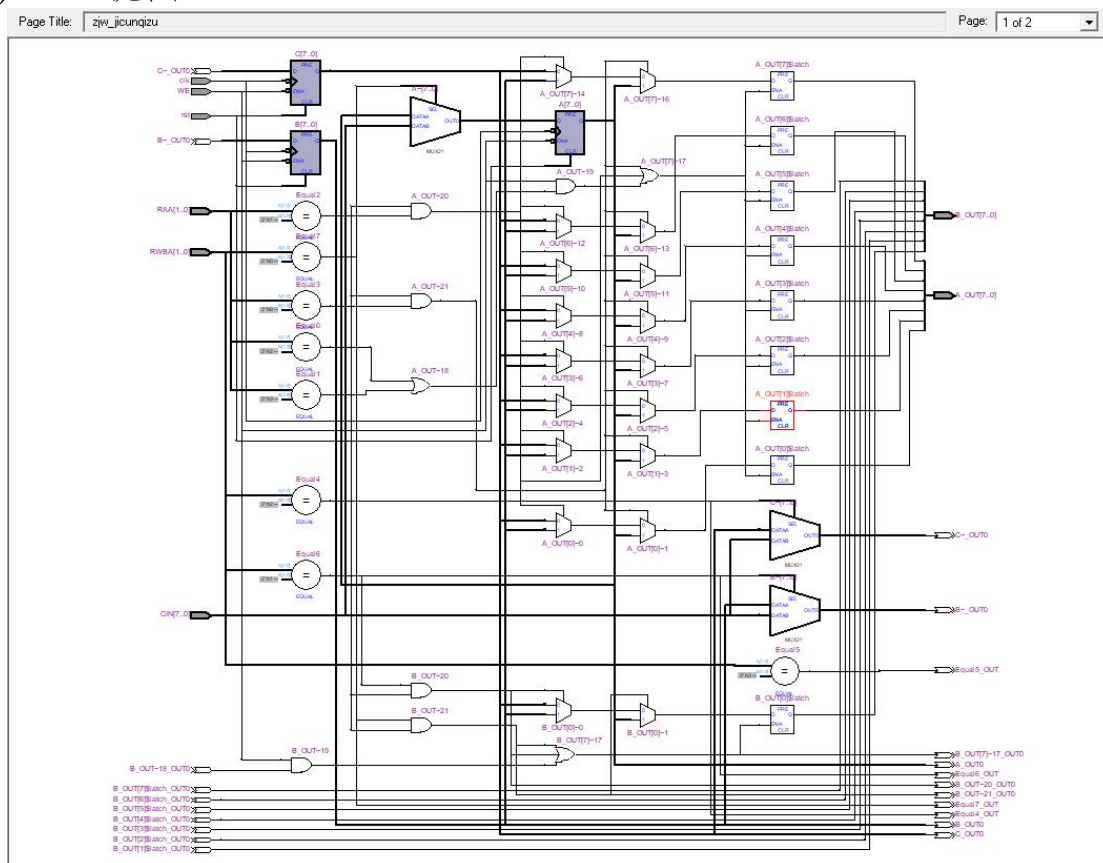
a) 源代码如图 (VHDL 设计)

```
zjw_jicunqizu.vhd
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity zjw_jicunqizu is
5  port(CIN: in std_logic_vector(7 downto 0);
6       RAA, RWBA: in std_logic_vector(1 downto 0);
7       WE,rst,clk: in std_logic;
8       A_OUT, B_OUT: out std_logic_vector(7 downto 0));
9  end zjw_jicunqizu;
10
11 architecture behavioral of zjw_jicunqizu is
12 signal A,B,C: std_logic_vector(7 downto 0):="00000000";
13 begin
14     A_OUT<=A when (WE='1' and RAA="00") else
15         B when (WE='1' and RAA="01") else
16         C when (WE='1' and (RAA="10" or RAA="11"));
17     B_OUT<=A when (WE='1' and RWBA="00") else
18         B when (WE='1' and RWBA="01") else
19         C when (WE='1' and (RWBA="10" or RWBA="11"));
20
21 process(rst,WE,RWBA,RAA,clk)
22 begin
23     if rst = '1' then
24         A <= "00000000";
25         B <= "00000000";
26         C <= "00000000";
27     elsif (WE='0' and (clk'event and clk = '0')) then
28         if(RWBA="00") then
29             A<=CIN;
30         elsif(RWBA="01") then
31             B<=CIN;
32         elsif(RWBA="10") then
33             C<=CIN;
34         end if;
35     else
36         end if;
37     end process;
```

b)编译、调试过程

Type	Message
Warning	Warning: Feature LogicLock is not available with your current license
Warning	Warning: No exact pin location assignment(s) for 31 pins of 31 total pins
Warning	Warning: The Reserve All Unused Pins setting has not been specified, and will default to 'As output driving ground'.
Warning	Warning: Timing Analysis is analyzing one or more combinational loops as latches
Warning	Warning: Node "A_OUT[0]\$latch" is a latch
Warning	Warning: Node "A_OUT[1]\$latch" is a latch
Warning	Warning: Node "A_OUT[2]\$latch" is a latch
Warning	Warning: Node "A_OUT[3]\$latch" is a latch
Warning	Warning: Node "A_OUT[4]\$latch" is a latch
Warning	Warning: Node "A_OUT[5]\$latch" is a latch
Warning	Warning: Node "A_OUT[6]\$latch" is a latch
Warning	Warning: Node "A_OUT[7]\$latch" is a latch
Warning	Warning: Node "B_OUT[0]\$latch" is a latch
Warning	Warning: Node "B_OUT[1]\$latch" is a latch
Warning	Warning: Node "B_OUT[2]\$latch" is a latch
Warning	Warning: Node "B_OUT[3]\$latch" is a latch
Warning	Warning: Node "B_OUT[4]\$latch" is a latch
Warning	Warning: Node "B_OUT[5]\$latch" is a latch
Warning	Warning: Node "B_OUT[6]\$latch" is a latch
Warning	Warning: Node "B_OUT[7]\$latch" is a latch
Warning	Warning: Found pins functioning as undefined clocks and/or memory enables

c) RTL 视图



d) 结果分析及结论

WE=1 时，输出（写的功能）

RAA[1..0]的值从 A,B,C 中选择一个寄存器的值由 A0 口输出

如 RAA[1..0]=00, $A0 \leftarrow A$ 寄存器的值

RAA[1..0]=01, $A0 \leftarrow B$ 寄存器的值

RAA[1..0]=10, $A0 \leftarrow C$ 寄存器的值

根据 RWBA[1..0]的值从选择 A,B,C 中选择一个寄存器的值由 B0 口输出，

如 RWBA[1..0]=00, $B0 \leftarrow A$ 寄存器的值

RWBA[1..0]=01, $B0 \leftarrow B$ 寄存器的值

RWBA[1..0]=10, $B0 \leftarrow C$ 寄存器的值

WE=0 时（读的功能）

根据 RWBA[1..0]的值,将外部输入写入 A,B,C 三个寄存器中的一个寄存器内。

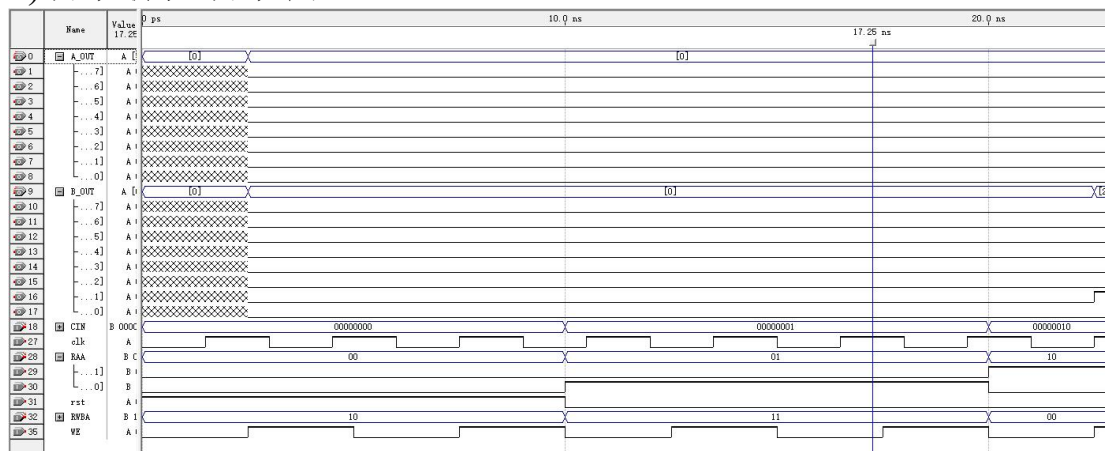
为了不出现未定义的情况，先进行一次重置，Reset 在一开始置为 0，将所有的触发器触发

成复位。然后连续 12 个时钟周期，保证时钟信号有效。

2.波形仿真

a) 波形仿真过程（详见实验步骤）

b) 波形仿真波形图



c) 结果分析及结论

WE=1 时，输出（写的功能）

RAA[1..0]的值从 A,B,C 中选择一个寄存器的值由 AO 口输出

如 RAA[1..0]=00, $AO \leftarrow A$ 寄存器的值

RAA[1..0]=01, $AO \leftarrow B$ 寄存器的值

RAA[1..0]=10, $AO \leftarrow C$ 寄存器的值

根据 RWBA[1..0]的值从选择 A,B,C 中选择一个寄存器的值由 BO 口输出，

如 RWBA[1..0]=00, $BO \leftarrow A$ 寄存器的值

RWBA[1..0]=01, $BO \leftarrow B$ 寄存器的值

RWBA[1..0]=10, $BO \leftarrow C$ 寄存器的值

WE=0 时（读的功能）

根据 RWBA[1..0]的值,将外部输入写入 A,B,C 三个寄存器中的一个寄存器内。

为了不出现未定义的情况，先进行一次重置，Reset 在一开始置为 0，将所有的触发器触发

成复位。然后连续 12 个时钟周期，保证时钟信号有效。

3.时序仿真

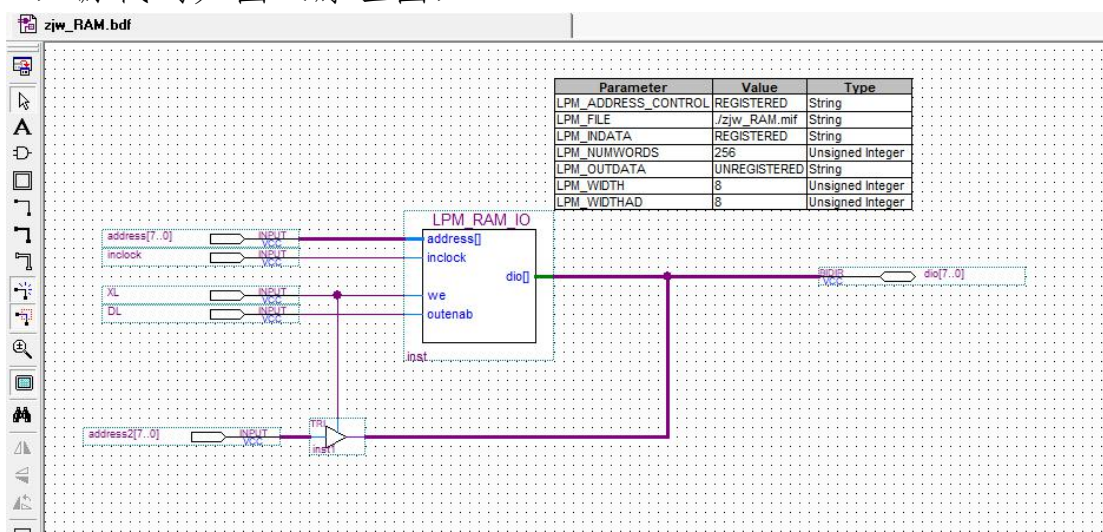
e) 时序仿真过程

做好上述步骤后，编译【classic timing analysis】-在 compilation report 中选择【timing analysis】-【tpd】（引脚到引脚的延时）

RAM

1.编译过程

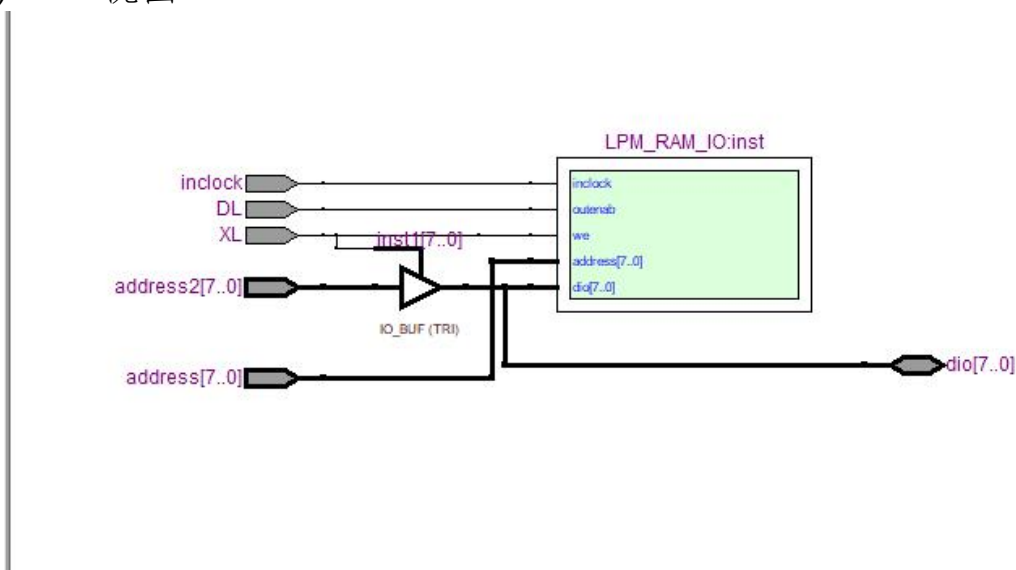
a) 源代码如图（原理图）



b)编译、调试过程

Type	Message
Warning	Warning: Assertion warning: altfasm does not support Cyclone device family -- attempting best-case memory conversions, but power-up states and read during write behavior will be different for...
Warning	Warning: Feature LogicLock is not available with your current license
Warning	Warning: No exact pin location assignment(s) for 27 pins of 27 total pins
Warning	Warning: The Reserve All Unused Pins setting has not been specified, and will default to 'As output driving ground'.
Warning	Warning: Found pins functioning as undefined clocks and/or memory enables

c) RTL 视图



d)结果分析及结论

数据通路确定后,就可以设计指令系统中每条指令所需要的机器周期数。对于微程序控制的

计算机,根据总线结构,需考虑哪些微操作可以安排在同一条微指令中,哪些微操作不能安排

在同一条微指令中。

2.波形仿真

a)波形仿真过程（详见实验步骤）

b)波形仿真波形图

c)结果分析及结论

3.时序仿真

f) 时序仿真过程

做好上述步骤后，编译【**classic timing analysis**】-在 **compilation report** 中选择【**timing analysis**】-【**tpd**】（引脚到引脚的延时）

b)时序仿真图

无需仿真。

在加入寄存器之后可以看见输出。

五、实验结论（实验总结与实验心得）

1. 时钟周期的上升沿实现对 RAM 的读写操作，为何 PC、SM 以及寄存器

的操作是下降沿完成？

答：一个时钟周期包含上升沿和下降沿，所以一个周期内可以同时读写。

根据 CPU 设计图可以知道，RAM 要处理传来的 PC，SM 和寄存器组的信息，考虑到延

时，并且要紧随其后处理信息，所以 RAM 采用上升沿，而 PC，SM 和寄存器采用下降

沿。

2. 总结 VHDL 语言描述时序部件的方法和常用语句。

数据流控制语句：

1.case...when (VHDL 顺序语句)

2.if...else (VHDL 顺序语句)

3.with...select when (VHDL 并行语句)

4.when...else (VHDL 并行语句)

模块化设计语句：

Process, component, generate.....

时钟沿的描述

上升沿：clk' event and clk=' 1' ;

下降沿：clk' event and clk=' 0