湖南大學

数字电路与逻辑设计

课程实验报告

题　　目：寄存器、计数器及 RAM

学生姓名：魏子铖

学生学号：201726010308

专业班级：软件 1703

完成时间：2018.12.13

实验四 寄存器、计数器及 RAM

班级 软件 1703 姓名 魏子铖 学号 201706010308

一、实验目的

1. 用 VHDL 语言设计一个 8 位的指令计数器 PC;
2. 用 VHDL 语言设计一个 8 位寄存器;
3. 用 LPM_RAM_IO 定制一个 256*8 的 RAM，再与 8 位寄存器配合实现对 RAM 的读取和写入操作;
4. 设计一个包含 3 个 8 位寄存器的寄存器组，并对其读写操作。

二、实验内容

1. 用 VHDL 语言编写一个 8 位的程序计数器 PC;
2. 用 VHDL 语言编写一个 8 位寄存器;
3. 用 VHDL 语言编写包含 3 个 8 位寄存器的通用寄存器组。
4. 采用 LPM_RAM_IO 定制一个 256*8 的 RAM，再与 8 位寄存器配合实现对 RAM 中读取和写入操作。

三、实验方法

用 VHDL 语言设计一个 8 位的指令计数器 PC

1. 新建，编写源代码。

(1).选择保存项:【File】-【new project wizard】-【next】(设置文件路径+设置 project name 为 FullAdder_VHDL +设置 top-level design entity name 为 PC_VHDL _VHDL）-【finish】

(2).新建:【file】-【new】-【VHDL File】-【OK】

2.写好源代码，保存文件（PC_VHDL.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。编译结果有 4 个警告，文件编译成功。

4.波形仿真及验证。

(1).新建一个 vector waveform file，保存文件（PC_VHDL.vwf)。。

(2).设置仿真模式【Assignments】-【Setting】-【Simulator Settings】-【Simulation mode】选择 Functional

(3).设置 end time（操作为:【edit】-【End Time】(设置为 8ns）-【ok】)。

(4).按照程序所述插入 X,Y,Z,C,S 三个节点（X,Y,Z 为输入节点，C,S 为输出节点)。

(操作为:右击 -【insert】-【insert node or bus】-【node finder】(pins=all;【list】) -【>>】-【ok】-【ok】)。

(5).设置 X,Y,Z 的输入波形…点击保存按钮保存。

（操作为:点击 name（如:Z）-右击-【value】-【clock】(如设置 period=2ns; offset=0ns)，同理设置 name X,Y（如 period=8/4ns; offset=0ns)，保存)。

(6).然后【Processing】-【Generate Functional Simulation Netlist】;

(7).然后【Processing】-【start simulation】，形成 name C，S 的输出图。

5.查看 RTL Viewer:【Tools】-【netlist viewer】-【RTL viewer】。

用 VHDL 语言设计一个 8 位寄存器

1.新建，编写源代码。

(1).选择保存项:【File】-【new project wizard】-【next】(设置文件路径+设置 project name 为 register8+设置 top-level design entity name 为 register8）-【finish】

(2).新建：【file】-【new】-【Block Diagrame/Schematic File】-【OK】
2.画好原理图，保存文件（register8.bdf）
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。编译结果有 5 个警告，文件编译成功。
4.波形仿真及验证。
(1).新建一个 vector waveform file，保存文件（register8_VHDL.vwf）。。
(2).设置仿真模式【Assignments】-【Setting】-【Simulator Settings】-【Simulation mode】选择 Functional
(3).设置 end time（操作为：【edit】-【End Time】（设置为 8ns）-【ok】）。
(4).按照程序所述插入 X,Y,Z,C,S 三个节点（X,Y,Z 为输入节点，C,S 为输出节点）。
(操作为：右击 -【insert】-【insert node or bus】-【node finder】（pins=all;【list】）-【>>】-【ok】-【ok】）。
(5).设置 X,Y,Z 的输入波形…点击保存按钮保存。
（操作为：点击 name（如：Z）-右击-【value】-【clock】（如设置 period=2ns; offset=0ns），同理设置 name X,Y（如 period=8/4ns; offset=0ns），保存）。
(6).然后【Processing】-【Generate Functional Simulation Netlist】；
(7).然后【Processing】-【start simulation】，形成 name C，S 的输出图。
5.查看 RTL Viewer:【Tools】-【netlist viewer】-【RTL viewer】。

设计一个包含 3 个 8 位寄存器的寄存器组，并对其读写操作

1.新建，编写源代码。
(1).选择保存项:【File】-【new project wizard】-【next】(设置文件路径+设置 project name 为 gen_register_VHDL+ 设置 top-level design entity name 为 gen_register_VHDL）-【finish】
(2).新建：【file】-【new】-【VHDL File】-【OK】
2.写好源代码，保存文件（gen_register_VHDL.vhd 和 register8.vhd 和 mux4_1.vhd 和 decoder2_4.vhd）
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。编译结果有 5 个警告，文件编译成功。
4、波形仿真及验证。
(1).新建一个 vector waveform file,保存文件(gen_register_VHDL.vwf)
(2).设置仿真模式【Assignments】-【Setting】-【Simulator Settings】-【Simulation mode】选择 Functional
(3).设置 end time（操作为：【edit】-【End Time】（设置为 160us）-【ok】）。
(4).按照程序所述插入 Shift,Clock,Reset,SI 为输入节点，Q,O 为输出节点）。
(操作为：右击 -【insert】-【insert node or bus】-【node finder】（pins=all;【list】）-【>>】-【ok】-【ok】）。
(5).设置 Shift,Clock,Reset,SI 的输入波形…点击保存按钮保存。
（操作为：点击 name（如：Reset））-在波形图中选取合适的一段设定,值为 0，之后设置为 1，同理设置 name Shift(全部为 1)Clock(在 Reset 后连续 12 个周期)的各个节点，SI（可以任意设置），保存）。
(6).然后【Processing】-【Generate Functional Simulation Netlist】；
(7).然后【Processing】-【start simulation】，形成 name Q，O 的输出图。

5.查看 RTL Viewer:【Tools】-【netlist viewer】-【RTL viewer】。

四、实验过程

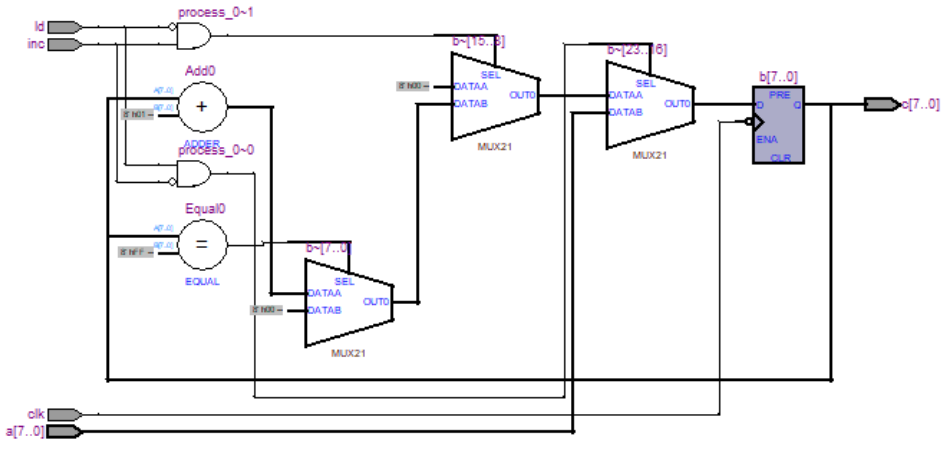
1、 a．用 VHDL 语言编写一个 8 位的程序计数器 PC

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 PC is
6      port( ld, inc, clk: in std_logic;
7                  a: in std_logic_vector(7 downto 0);
8                  c: out std_logic_vector(7 downto 0)
9                  );
10 end PC;
11
12 architecture main of PC is
13 signal b: std_logic_vector(7 downto 0);
14 begin
15     process(clk)
16     begin
17         if (clk'event and clk = '0') then
18             if (ld = '1' and inc = '0') then
19                 b <= a;
20             elsif (ld = '0' and inc = '1') then
21                 if ( b = "11111111") then
22                     b <= "00000000";
23                 else
24                     b <= b + "00000001";
25                 end if;
26             else
27                 b <= "00000000";
28             end if;
29         end if;
30     end process;
31     c <= b;
32 end main;
```

RTL 如图



b)编译、调试过程

编译通过，有 4 个警告;

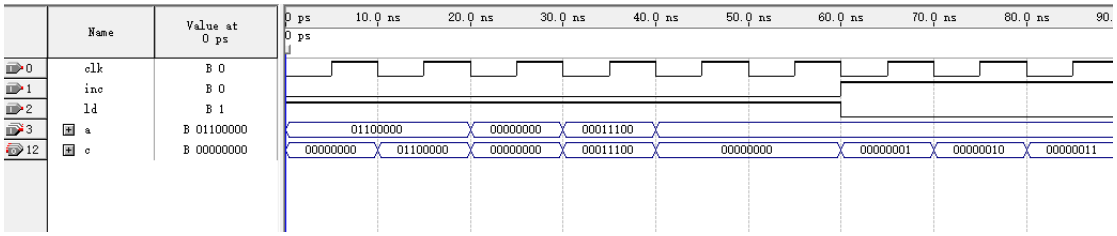
c)结果分析及结论

输出正确，符合预期结果

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

输出正确，符合预期结果

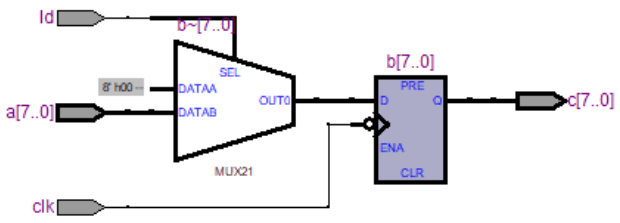
b．用 VHDL 语言设计一个 8 位寄存器

编译过程

a) 源代码如图（VHDL 设计）

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity register8 is
5       port(ld, clk : in std_logic;
6                a: in std_logic_vector(7 downto 0);
7                c: out std_logic_vector(7 downto 0)
8                );
9   end register8;
10
11  architecture main of register8 is
12  signal b: std_logic_vector(7 downto 0);
13  begin
14      process(clk)
15      begin
16          if (clk'event and clk = '0') then
17              if (ld = '1') then
18                  b <= a;
19              else
20                  b <= "00000000";
21              end if;
22          end if;
23      end process;
24      c <= b;
25  end main;
```

RTL 如图



b)编译、调试过程

编译通过，有 5 个警告;

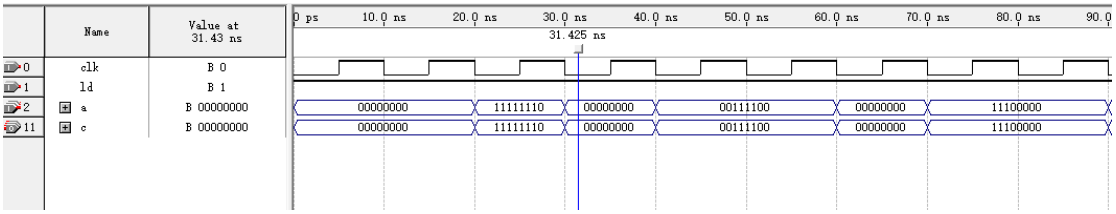
c)结果分析及结论

输出符合预期，结果正确

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



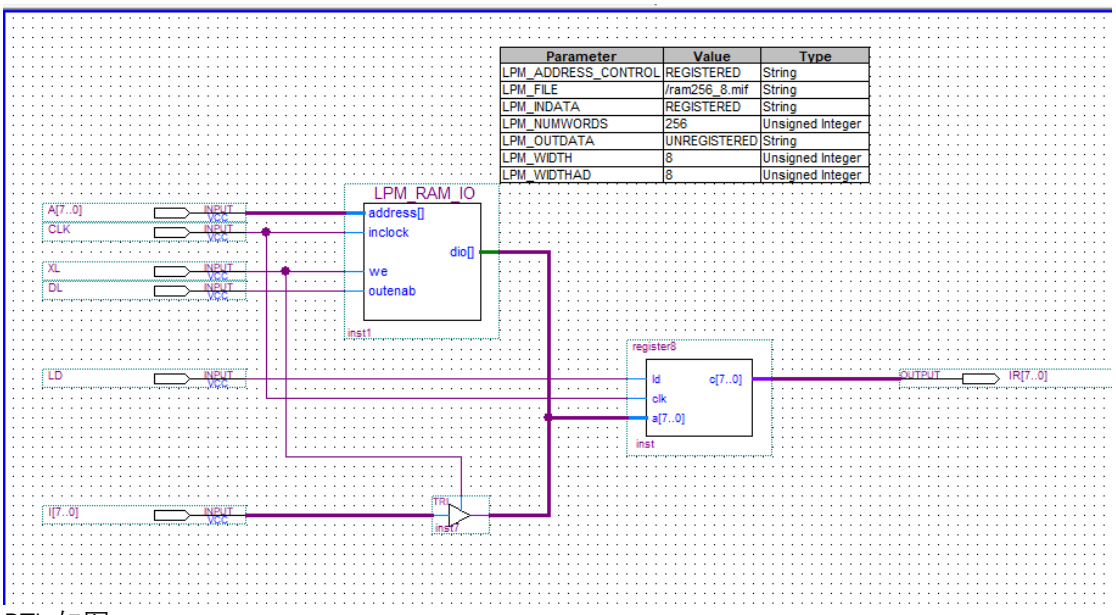
c)结果分析及结论

输出正确，符合预期结果

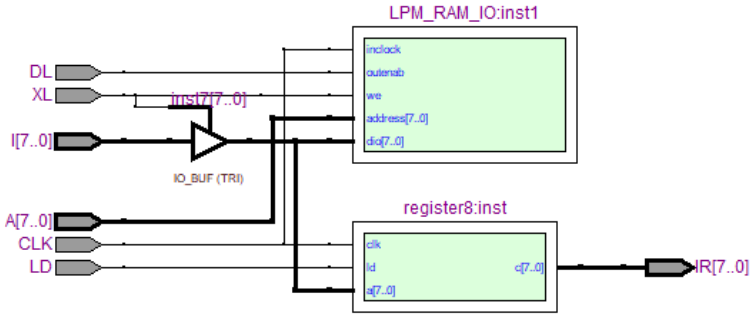
c. 用 LPM_RAM_IO 定制一个 256*8 的 RAM，再与 8 位寄存器配合实现对 RAM 的读取和写入操作

编译过程

a） 原理图如图:



RTL 如图

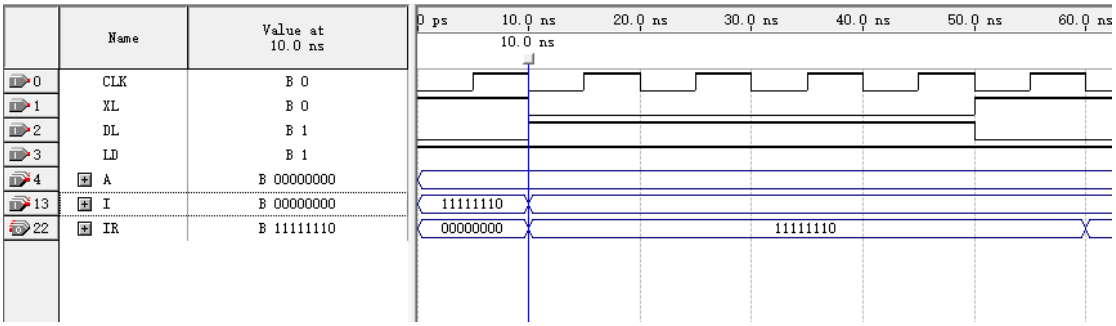


b)编译、调试过程

编译通过，有 5 个警告;
c)结果分析及结论
输出正确，符合预期结果

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
输出正确，符合预期结果

以上结果符合预期，正确

d．设计一个包含 3 个 8 位寄存器的寄存器组，并对其读写操作

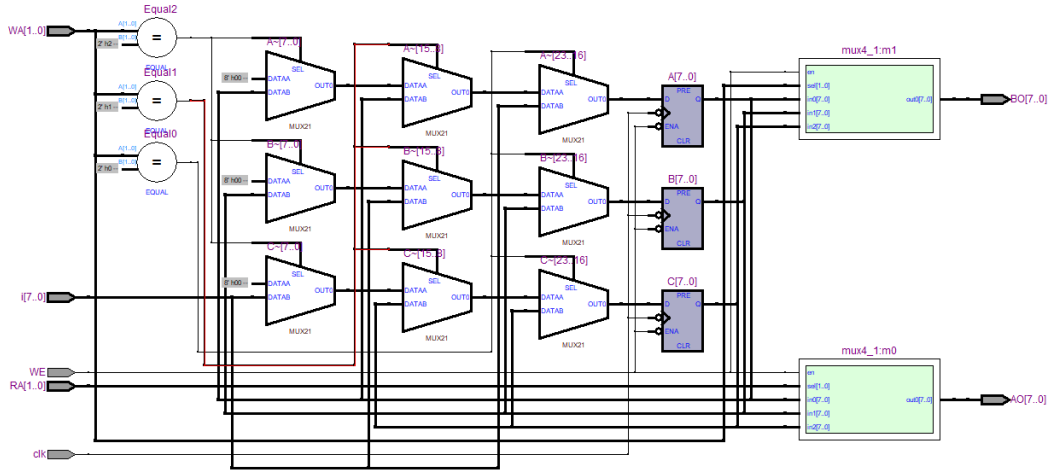
编译过程

a） 源代码如图（VHDL 设计）

```
1  --mux4_1
2  library ieee;
3  use ieee.std_logic_1164.all;
4
5  entity mux4_1 is
6      port(    en: in std_logic;
7               sel: in std_logic_vector(1 downto 0);
8               in0, in1, in2: in std_logic_vector(7 downto 0);
9               out0: out std_logic_vector(7 downto 0)
10          );
11 end mux4_1;
12
13 architecture main of mux4_1 is
14 begin
15     process(sel)
16     begin
17         if en = '1' then
18             if sel = "00" then
19                 out0 <= in0;
20             elsif sel = "01" then
21                 out0 <= in1;
22             elsif sel = "10" then
23                 out0 <= in2;
24             else
25                 out0 <= "00000000";
26             end if;
27         end if;
28     end process;
29 end main;
```

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  entity gen_register is
5      port(  clk, WE: in std_logic;
6             RA, WA: in std_logic_vector(1 downto 0);
7                  i: in std_logic_vector(7 downto 0);
8             AO, BO: out std_logic_vector(7 downto 0)
9             );
10 end gen_register;
11
12 architecture main of gen_register is
13
14 component mux4_1
15     port(    en: in std_logic;
16              sel: in std_logic_vector(1 downto 0);
17              in0, in1, in2: in std_logic_vector(7 downto 0);
18              out0: out std_logic_vector(7 downto 0)
19           );
20 end component;
21
22 signal A, B, C: std_logic_vector(7 downto 0) := "00000000";
23 begin
24     process(clk)
25     begin
26         if (clk'event and clk = '0' and WE = '0') then
27             if WA = "00" then
28                 A <= i;
29             elsif WA = "01" then
30                 B <= i;
31             elsif WA = "10" then
32                 C <= i;
33             end if;
34         end if;
35     end process;
36
37     m0: mux4_1 port map(WE, RA, A, B, C, AO);
38     m1: mux4_1 port map(WE, WA, A, B, C, BO);
39
40 end main;
```

RTL 如图



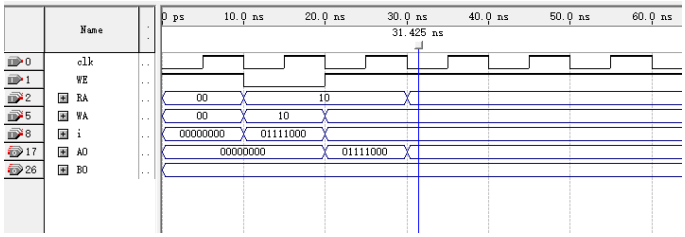
b)编译、调试过程

编译通过，有 4 个警告;

c）结果分析及结论
输出正确，符合预期结果

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
输出正确，符合预期结果

五、实验结论

本次实验内容多，难度较大，完成这次实验是一次很大的挑战。在做预习报告之前，对实验内容的很多概念都不是很了解，比如锁存器，触发器，寄存器。在做实验之前先认真的看了书本，也用 quarters 自己实现了一遍锁存器和触发器，让我对于书本知识和实验内容有了一个深刻的认识。本次实验内容很多要用到 VHDL 语音，经常查资料和问同学，再加上自己的尝试，最后也掌握了不少 VHDL 的语法。本次实验难度虽大，但也收获很多。