

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

1. **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。

1. **实验内容**
2. 用VHDL语言编写一个8位的程序计数器PC；
3. 用VHDL语言编写一个8位寄存器；
4. 用VHDL语言编写包含3个8位寄存器的通用寄存器组。
5. 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。
6. **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）。

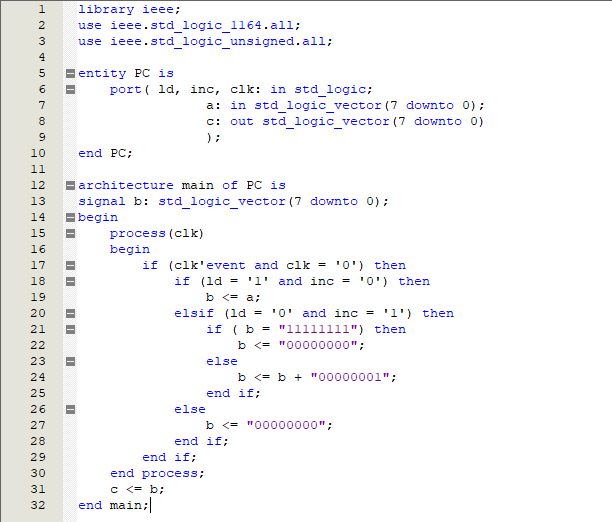
(6).然后【Processing】-【Generate Functional Simulation Netlist】;

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

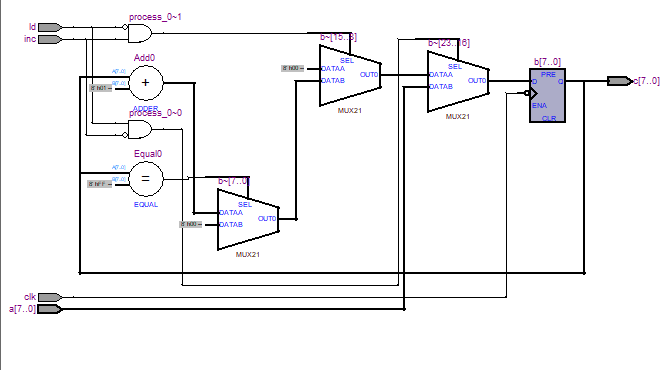
5.查看RTL Viewer:【Tools】-【netlist viewer】-【RTL viewer】。

1. **实验过程**
2. **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）

RTL如图



b)编译、调试过程

编译通过，有4个警告；

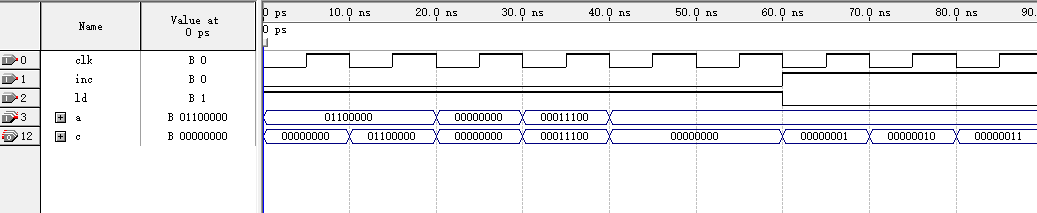
c)结果分析及结论

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



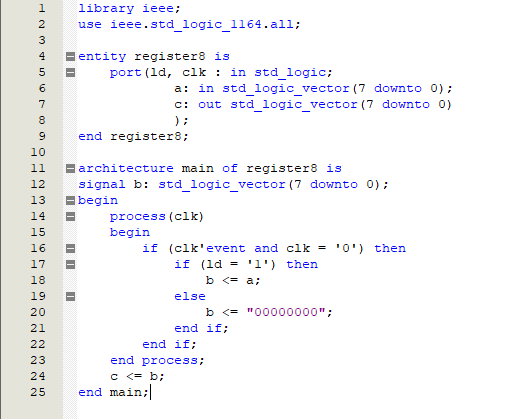
c)结果分析及结论

输出正确，符合预期结果

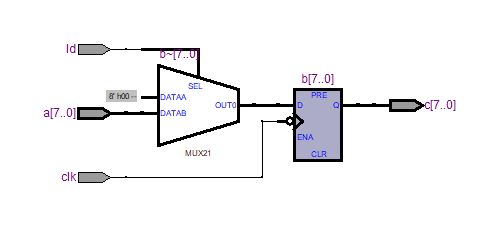
**b．用VHDL语言设计一个8位寄存器**

**编译过程**

a）源代码如图（VHDL设计）



RTL如图



b)编译、调试过程

编译通过，有5个警告；

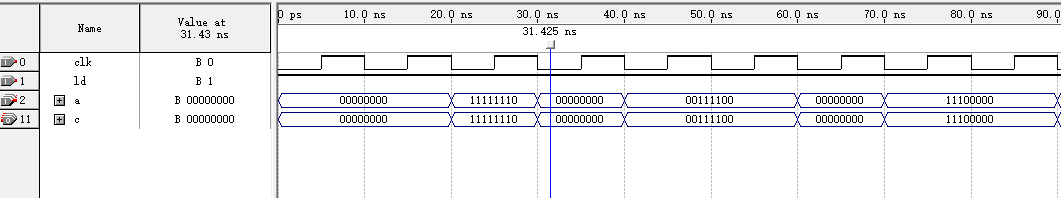
c)结果分析及结论

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



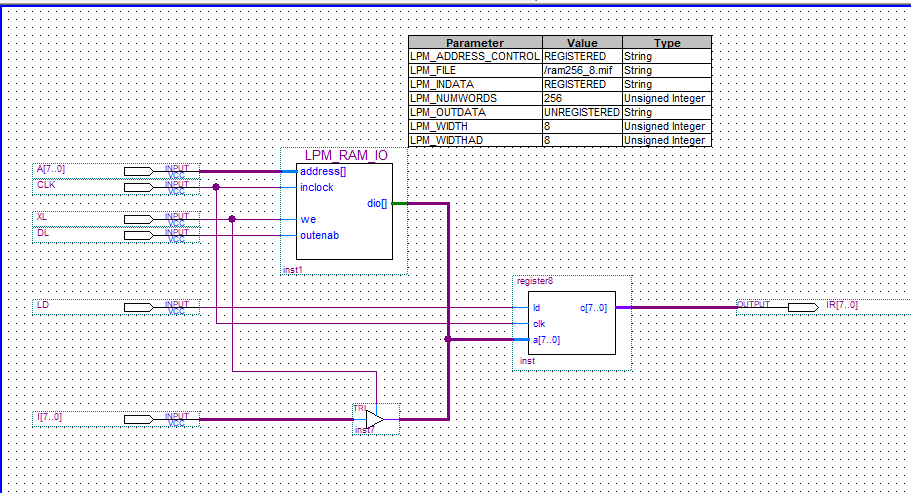
c)结果分析及结论

输出正确，符合预期结果

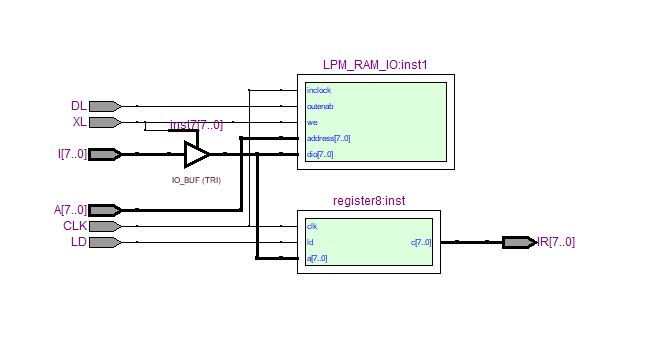
**c.** **用LPM\_RAM\_IO定制一个256\*8的RAM，再与8位寄存器配合实现对RAM的读取和写入操作**

**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；

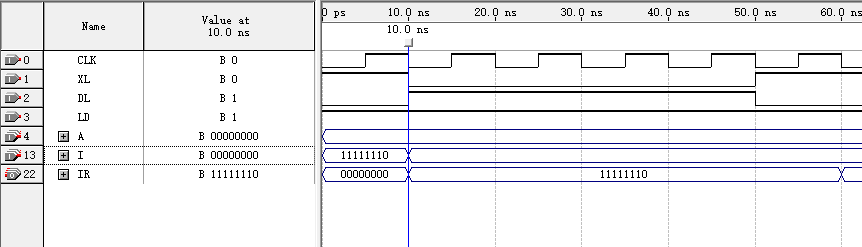
c)结果分析及结论

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

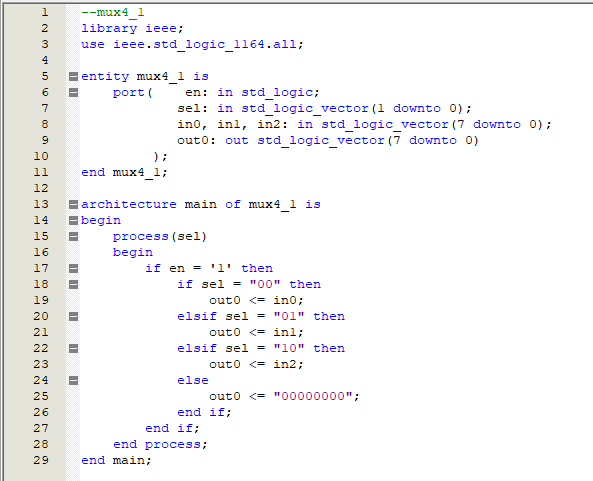
输出正确，符合预期结果

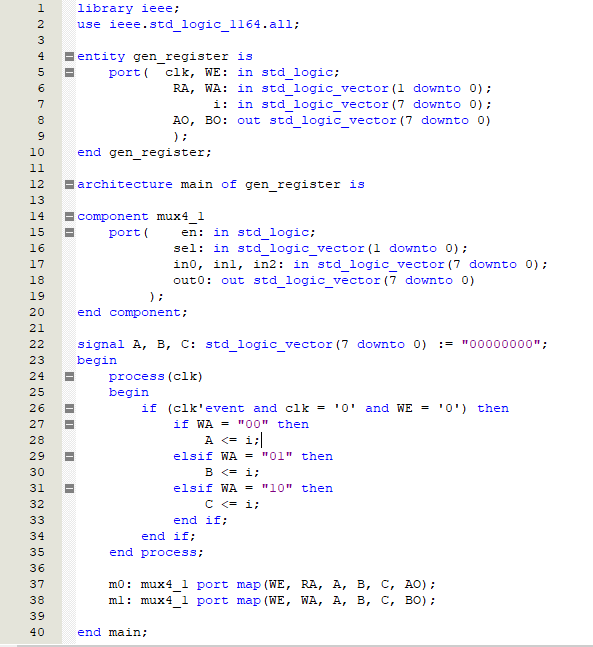
以上结果符合预期，正确

**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**

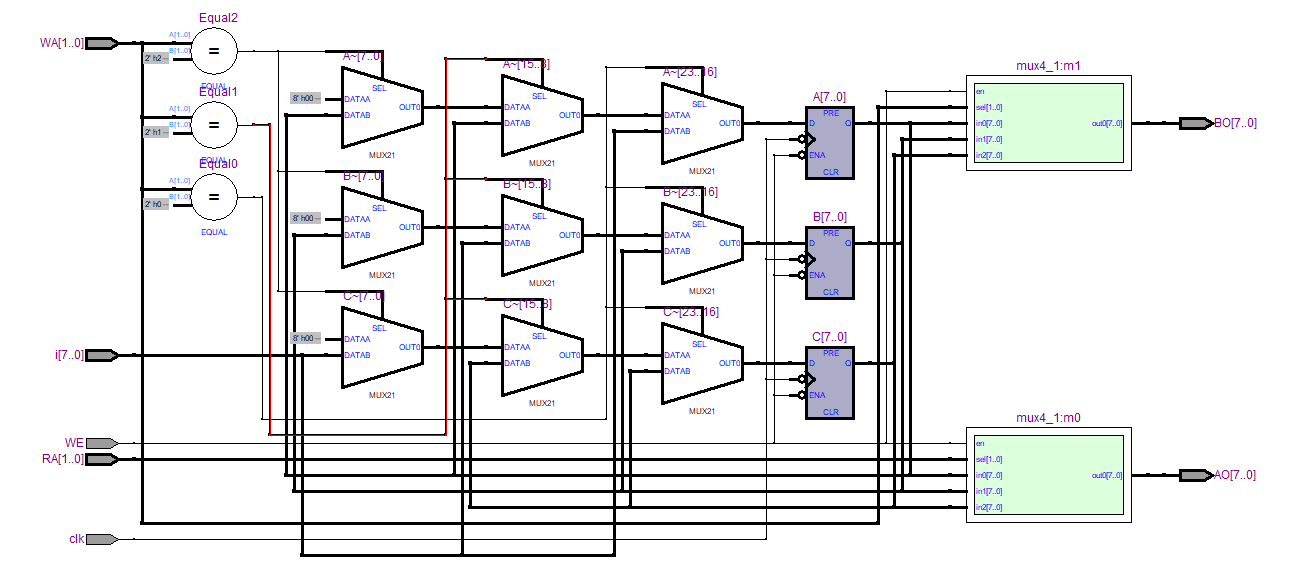
**编译过程**

1. 源代码如图（VHDL设计）





RTL如图



b)编译、调试过程

编译通过，有4个警告；

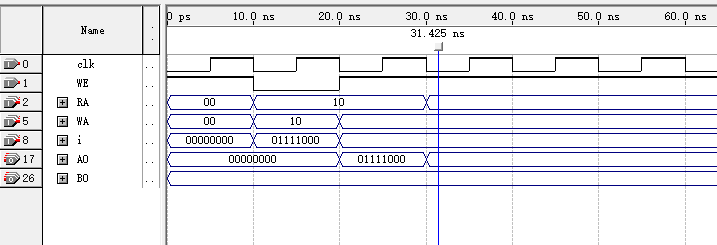
c)结果分析及结论

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

1. **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。