《计算机组成原理》实验报告

年级、专业、班级	2019 级计算机科学与技术 01 班、2019 级计算机科学与技术 01 班、2019 级计算机科学与技术
实验题目	实验二处理器译码实验
实验时间	2019 年 11 月 15 日
实验成绩	优秀/良好/中等

教师评价：

□算法/实验过程正确； □源程序/实验内容提交； □程序结构/实验步骤合

□实验结果正确； □语法、语义正确； □报告规范；

其他:

实验目的

(1)掌握单周期 CPU 控制器的工作原理及其设计方法。

(2)掌握单周期 CPU 各个控制信号的作用和生成过程。

(3)掌握单周期 CPU 执行指令的过程。

(4)掌握取指、译码阶段数据通路执行过程。

报告完成时间: 2021 年 5 月 22 日

1 实验内容

1. PC。D 触发器结构，用于储存 PC(一个周期)。需实现 2 个输入，分别为 *clk, rst*, 分别连接时钟和复位信号；需实现 2 个输出，分别为 *pc, inst_ce*, 分别连接指令存储器的 *addra, ena* 端口。其中 *addra* 位数依据 coe 文件中指令数定义；
2. 加法器。用于计算下一条指令地址，需实现 2 个输入，1 个输出，输入值分别为当前指令地址 *PC*、*32'h4*；
3. Controller。其中包含两部分：
   (a). main_decoder。负责判断指令类型，并生成相应的控制信号。需实现 1 个输入，为指令 inst 的高 6 位 *op*，输出分为 2 部分，控制信号有多个，可作为多个输出，也作为一个多位输出，具体参照参考指导书进行设计；*aluop*，传输至 alu_decoder，使 alu_decoder 配合 *inst* 低 6 位 *funct*，进行 ALU 模块控制信号的译码。
   (b). alu_decoder。负责 ALU 模块控制信号的译码。需实现 2 个输入，1 个输出，输入分别为 *funct, aluop*；输出位 *alucontrol* 信号。
   (c). 除上述两个组件，需设计 controller 文件调用两个 decoder，对应实现 *op*,*funct* 输入信号，并传入调用模块；对应实现控制信号及 *alucontrol*，并连接至调用模块相应端口。
4. 指令存储器。使用 Block Memory Generator IP 构造。(参考指导书)
   注意：Basic 中 Generate address interface with 32 bits 选项不选中；PortA Options 中 Enable Port Type 选择为 Use ENA Pin
5. 时钟分频器。将板载 100Mhz 频率降低为 1hz，连接 PC、指令存储器时钟信号 clk。(参考数字逻辑实验)
   注意：Xilinx Clocking Wizard IP 可分的最低频率为 4.687Mhz，因而只能使用自实现分频模块进行分频

2 实验设计

2.1 控制器 (Controller)

2.1.1 功能描述

在 CPU 运行过程中为不同的输入输出提供选择的信号。

2.1.2 接口定义

表 1: 控制器接口定义图

信号名	方向	位宽	功能描述
inst	input	32-bit	输入指令,依次决定控制信号的输出
regdst	output	1-bit	写入寄存器堆的地址是 rt 还是 rd,0 为 rt,1 为 rd
regwrite	output	1-bit	是否需要写寄存器堆
alusrc	output	1-bit	送入 ALU B 端口的值是立即数的 32 位扩展/寄存器堆读取的值
memwrite	output	1-bit	是否需要写数据存储器
memtoreg	output	1-bit	回写的数据来自于 ALU 计算的结果/存储器读取的数据
branch	output	1-bit	是否为 branch 指令，且满足 branch 的条件
jump	output	1-bit	是否为 jump 指令
alucontrol	output	3-bit	ALU 控制信号，代表不同的运算类型

2.1.3 逻辑控制

控制器输出信号为 regdst,regwrite,alusrc,memwrite,memtoreg,branch,jump,alucontrol。

regdst：写入寄存器堆的地址是 rt 还是 rd,0 为 rt,1 为 rd；

regwrite：是否需要写寄存器堆,0 为不需要,1 为需要；

alusrc：送入 ALU B 端口的值是立即数的 32 位扩展/寄存器堆读取的值，0 为寄存器堆读取的值,1 为立即数的 32 位扩展;

memwrite：是否需要写数据存储器,0 表示不需要,1 表示需要;

memtoreg：回写的数据来自于 ALU 计算的结果/存储器读取的数据，0 表示回写的数据来自于 ALU 计算的结果,1 表示回写的数据来自存储器读取的数据;

branch：是否为 branch 指令，且满足 branch 的条件，0 表示不是 branch 指令，1 表示是 branch 指令;

jump：是否为 jump 指令,0 表示不是 jump 指令,1 表示是 jump 指令;

alucontrol：ALU 控制信号，代表不同的运算类型；

2.2 存储器 (Block Memory)

存储器使用 vivado 内置的 Block Memory IP Core，并根据实验需求更改类型和参数。

2.2.1 类型选择

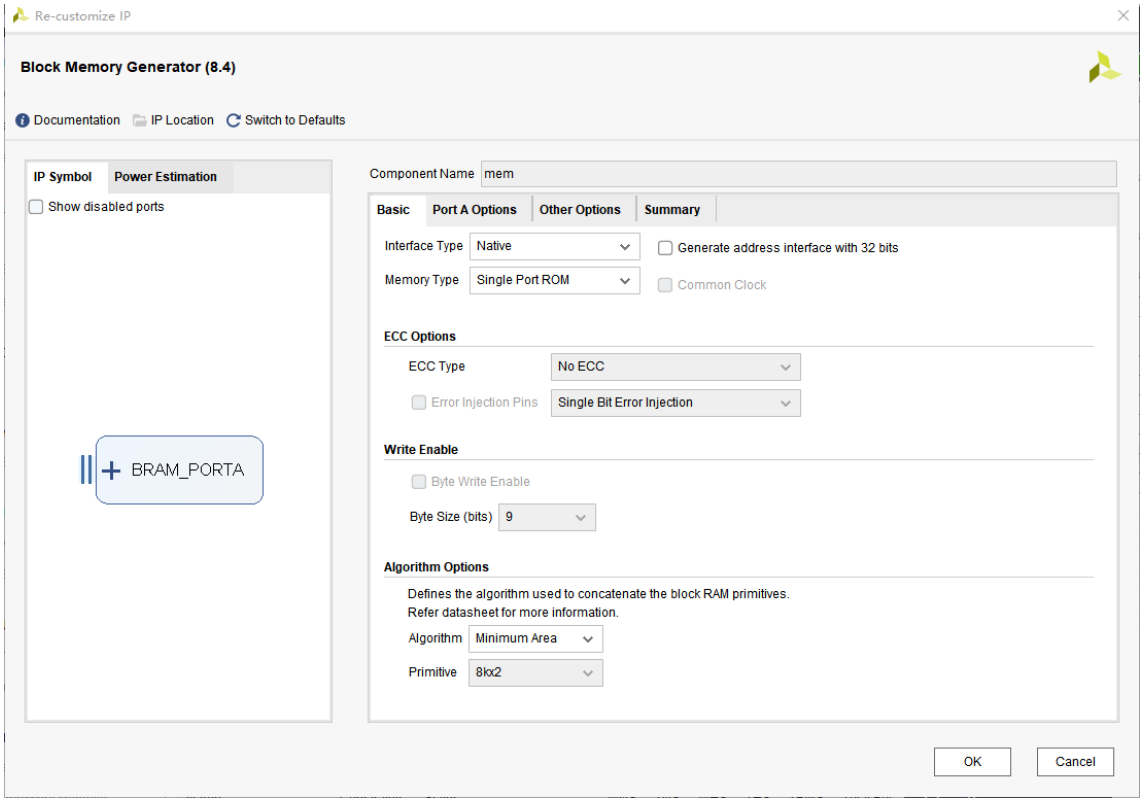

图 1: 类型选择

2.2.2 参数设置

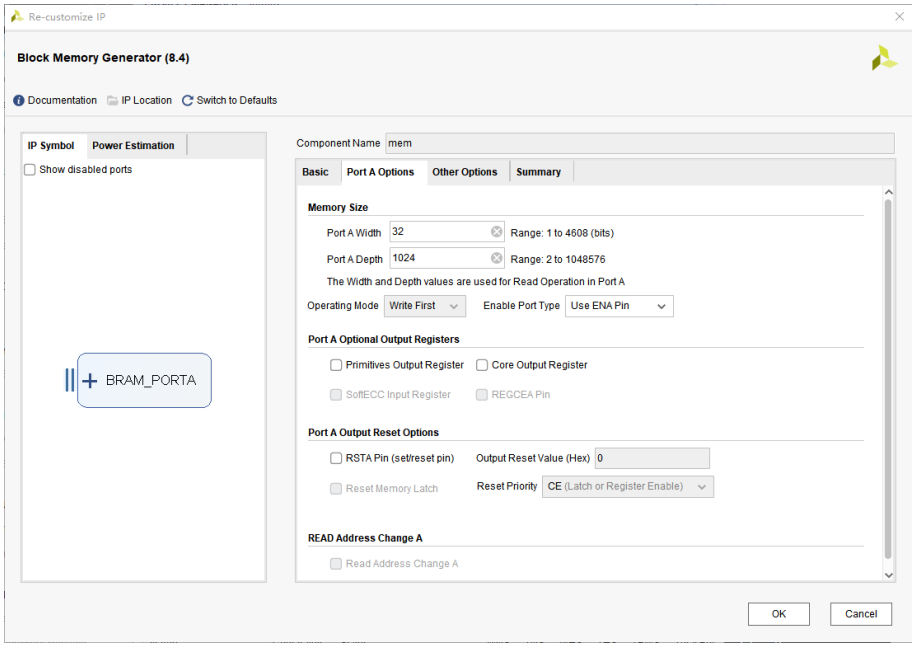

图 2: 参数设置 1

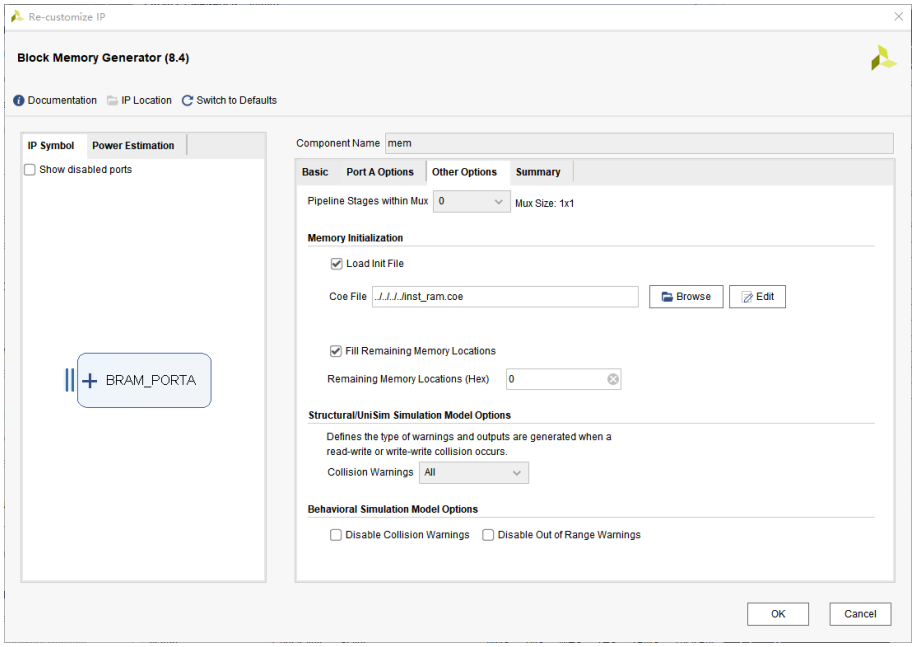

图 3: 参数设置 2

3 实验过程记录

3.1 完成工作

首先创建 datapath.v 的 rtl 文件根据实验指导书,一共需要以下部件来构建数据通路:

|——pc.v PC 模块,使用实验二代码

|——alu.v ALU 模块,使用实验一代码

|——sl2.v 移位模块,参考《其他组件实现.pdf》
|——signext.v 有符号扩展模块,参考《其他组件实现.pdf》
|——mux2.v 二选一选择器,自行实现
|——regfile.v 寄存器堆,已提供
|——adder.v 加法器,已提供

3.1.1 LW 指令

LW 指令除了需要寄存器堆,之外,还需要 pc 触发器,pc+4 加法器,符号拓展器,ALU 这几个部件,依次在 datapath.v 引用相应的模块,并用中间变量连接,连接后图示如下:

图 4: LW 指令连接图示

3.1.2 SW 指令

在完成 LW 指令的数据通路后,SW 指令仅需进行些许改动即可。只需用中间变量将 RT 连接至 A2,RD2 连接至 WD 即可,不需要增加器件,连接后图示如下:

图 5: SW 指令连接图示

3.1.3 R-type 指令

R-Type 指令。除了指令进行的运算类型不同外,其操作均为将 rs,rt 所在寄存器的值进行相应运算后,存入 rd 中。因而,对已经满足 lw、sw 的数据通路进行一定改造即可。只需要加入多路选择器和相应的控制信号即可。lw 指令写入 regfile 的地址为 rt,而 R-Type 为 rd,在此处加入一个多路选择器,输入分别为指令的 [20:16] 和 [15:11],使用 RegDst(register destination) 信号控制。ALU 输入为 rs,rt, 分别对应 srcA,srcB, 因此需要在 srcB 处加入多路选择器, 选择来源为 RD2 或立即数 SignImm, 控制信号为 ALUSrc(ALU source)。最后写回到 regfile 的值应该为 ALU 计算得到的值, 为 ALUResult, 加入多路选择器控制 9 + ALU Result 来源为 ALU 或数据存储器, 控制信号为 MemtoReg(Memory to regfile)。连接后图示如下:

图 6: R-type 指令连接图示

3.1.4 Branch 指令

此处以 beq 指令为例。只需添加一个与门(无需单独写, assign 赋值即可)得到 pcsrc, 再用一个两路选择器选择 branch 语句 pc 和 pc+4, 控制信号为 pcsrc。连接后图示如下:

图 7: Branch 指令连接图示

3.1.5 Jump 指令

跳转指令实际为无条件跳转，不需要做任何判断，因此更加简单。只需要计算地址，更改 PC 即可。首先用一个移位器将指令 [25:0] 位左移两位后和 pc 高四位拼接（Verilog 语言实现，无需设计拼接），再用一个两路选择器选择 jump 指令的 pc 和之前已经选择过的 pc，控制信号为 jump。连接后图示如下：在完成上述步骤之后，datapath 基本上就连接完成了，配合 maindec 和 aludec 和

图 8: Jump 指令连接图示

controller 和 mips 文件构成了 MIPS 软核，然后在 top 文件中和 IP 调用创建的 instruction ram 和 data ram 连接，就完成了一个单周期简单 CPU 的设计

4 实验过程记录

4.1 完成工作

4.1.1 控制器

控制器分为两个部分，一个是 maindec，另一个是 aludeec，这两个合并之后就构成了 controller。subsubsectionaludec

4.2 问题

4.3 解决方案

4.3.1 PC 取址问题

5 实验结果及分析

需要仿真图一张，要求仿真图中包含 pc、指令存储器输出、8 种控制信号，PC 自增无误。

5.1 仿真图

5.2 控制台输出

A Controller 代码

仅需要根据需要，在一个模块完成控制器的，直接填写，多个模块（maindec、aludec）分别加入新的 lstlisting 并填写 (红字为内容说明，请删除)