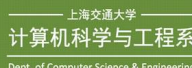


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42



计算机系统结构实验指导书-LAB3

 上海交通大学 计算机科学与工程系 Dept. of Computer Science & Engineering	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB3	CSE-COA-LAB-003	0.5	1 of 15
	作者	修改日期	公开	
	CETC	03/21/2019		

1. 概述

1.1 实验名称

简单的类 MIPS 单周期处理器部件实现 – 控制器，ALU

1.2 实验目的

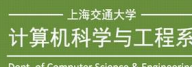
1. 理解 CPU 控制器，ALU 的原理
2. 主控制器 Ctr 的实现
3. 运算单元控制器 ALUCtr 的实现
4. ALU 的实现
5. 使用功能仿真

1.3 实验预计时间

120~150 分钟

1.4 实验报告与验收办法

- 1) 实验报告和工程文件在第十二周星期五晚上 23 点前提交
- 2) 无法验收，报告中检查三个仿真结果（Ctr、AluCtr 和 Alu）

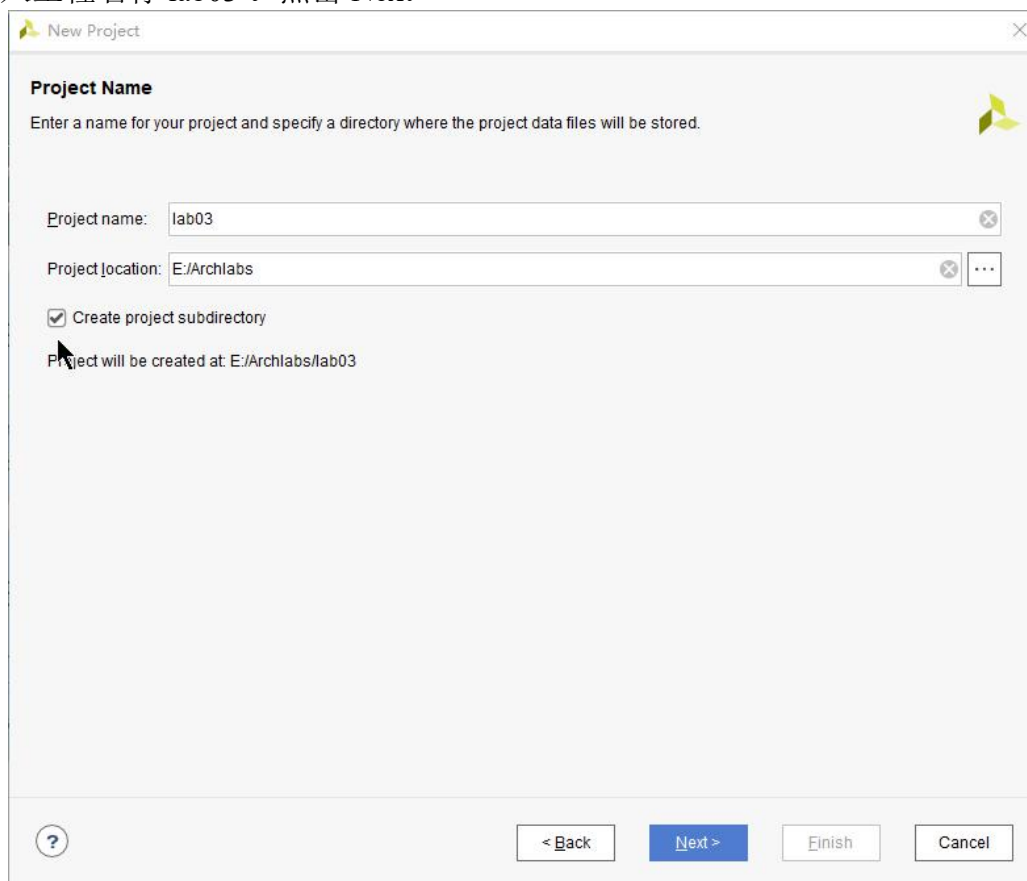
 上海交通大学 计算机科学与工程系 Dept. of Computer Science & Engineering	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB3	CSE-COA-LAB-003	0.5	2 of 15
	作者	修改日期	公开	
	CETC	03/21/2019		

2. 新建工程

2.1 实验描述


2.1.1 新建工程

1. 启动 Vivado 2018.3
2. 输入工程名称 lab03 。点击 Next



3. 选择实验板对应的 FPGA 参数:

Product Category: ALL
Family: Kintex-7
Package: ffg676
Speed: -2
Temperature: ALL Remaining

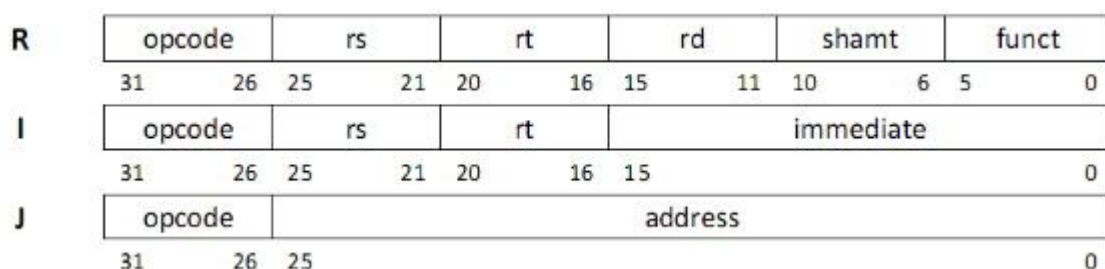
 上海交通大学 计算机科学与工程系 Dept. of Computer Science & Engineering	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB3	CSE-COA-LAB-003	0.5	3 of 15
	作者	修改日期	公开	
	CETC	03/21/2019		

3. 主控制器单元模块

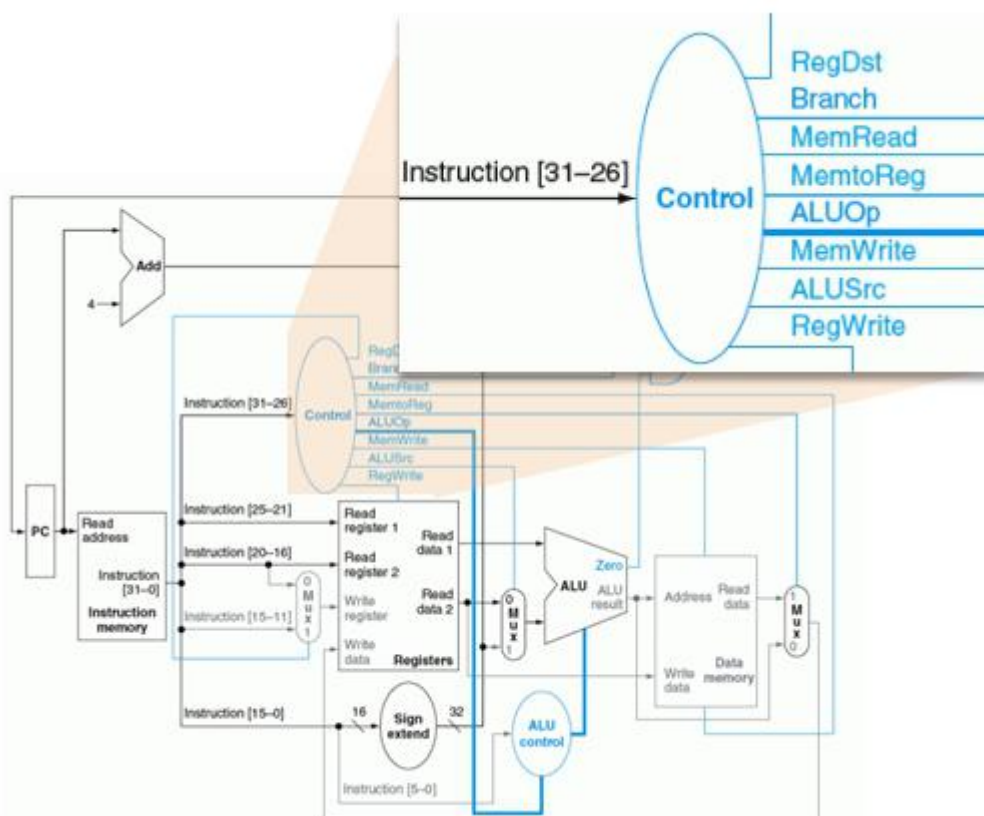
3.1 实验描述

3.1.1 模块描述

主控制器单元（Ctr）的输入为指令的 opCode 字段，操作码经过 Ctr 的译码，给 ALUCtr, Data Memory, Registers, Muxs 等部件输出正确的控制信号。



Mips 基本指令格式

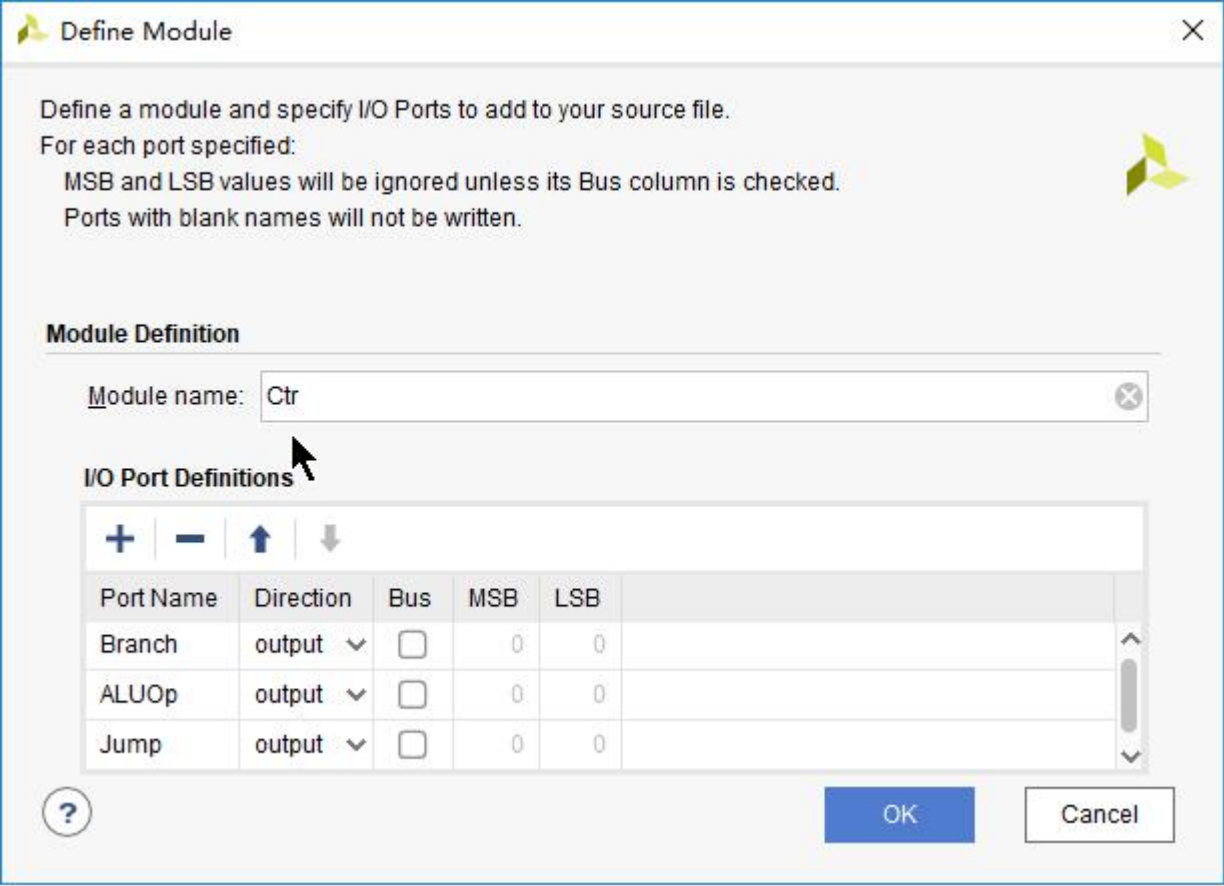


主控制模块的 IO 定义


<div> <div>上海交通大学</div> <div>计算机科学与工程系</div> <div>Dept. of Computer Science & Engineering</div> </div>	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB3	CSE-COA-LAB-003	0.5	4 of 15
	作者	修改日期		
	CETC	03/21/2019	公开	

3.1.2 新建模块 Verilog 文件

定义控制器 I/O 端口



```
20 //////////////////////////////////////////////////
21 module Ctr(
22     input [5:0] opCode,
23     output regDst,
24     output aluSrc,
25     output memToReg,
26     output regWrite,
27     output memRead,
28     output memWrite,
29     output branch,
30     output [1:0] aluOp,
31     output jump
32 );
33
34
35 endmodule
```

 上海交通大学 计算机科学与工程系 Dept. of Computer Science & Engineering	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB3 作者 CETC	CSE-COA-LAB-003 修改日期 03/21/2019	0.5	5 of 15 公开

3.1.3 编写译码功能


Input or output	Signal name	R-format	lw	sw	beq
Inputs	Op5	0	1	1	0
	Op4	0	0	0	0
	Op3	0	0	1	0
	Op2	0	0	0	1
	Op1	0	1	1	0
	Op0	0	1	1	0
Outputs	RegDst	1	0	X	X
	ALUSrc	0	1	1	0
	MemtoReg	0	1	X	X
	RegWrite	1	1	0	0
	MemRead	0	1	0	0
	MemWrite	0	0	1	0
	Branch	0	0	0	1
	ALUOp1	1	0	0	0
	ALUOp0	0	0	0	1

主控制模块真值表

注：Jump 指令编码是 000010，Jump 信号输出 1，其余输出 0

指令	opCode
R 型：add, sub, and, or, slt	000000
I 型：lw	100011
I 型：sw	101011
I 型：beq	000100
J 型：J	000010

指令操作码

 上海交通大学 计算机科学与工程系 Dept. of Computer Science & Engineering	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB3	CSE-COA-LAB-003	0.5	6 of 15
	作者	修改日期	公开	
	CETC	03/21/2019		

用 verilog 硬件描述语言写出上述真值表，实现方式多种多样，这里仅给出一段使用 case 语句的参考方案，如下图：

```

14      reg RegDst;
15      reg ALUSrc;
16      reg MemToReg;
17      reg RegWrite;
18      reg MemRead;
19      reg MemWrite;
20      reg Branch;
21      reg [1:0] ALUOp;
22      reg Jump;

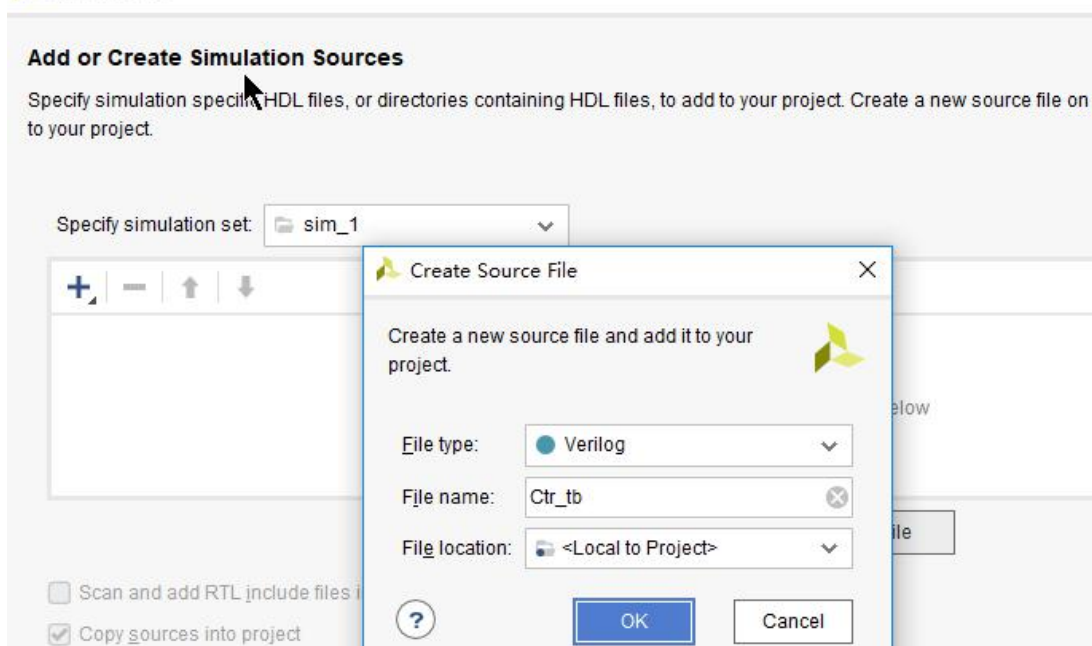
23
24      always @(OpCode)
25      begin
26          case(OpCode)
27              6'b000000: //R type
28              begin
29                  RegDst = 1;
30                  ALUSrc = 0;
31                  MemToReg = 0;
32                  RegWrite = 1;
33                  MemRead = 0;
34                  MemWrite = 0;
35                  Branch = 0;
36                  ALUOp = 2'b10;
37                  Jump = 0;
38              end
          //add beq
          //6'bxxxxxx;
          begin
          ...
          end
          //add lw
          //add sw
          //add Jump
          default:
          begin
              RegDst = 0;
              ALUSrc = 0;
              MemToReg = 0;
              RegWrite = 0;
              MemRead = 0;
              MemWrite = 0;
              Branch = 0;
              ALUOp = 2'b00;
              Jump = 0;
          end
          endcase
    
```


PS:代码中要把真值表里的所有情况按样例都要覆盖补齐

3.1.4 功能仿真

1. 新建激励文件 Ctr_tb

 Add Sources



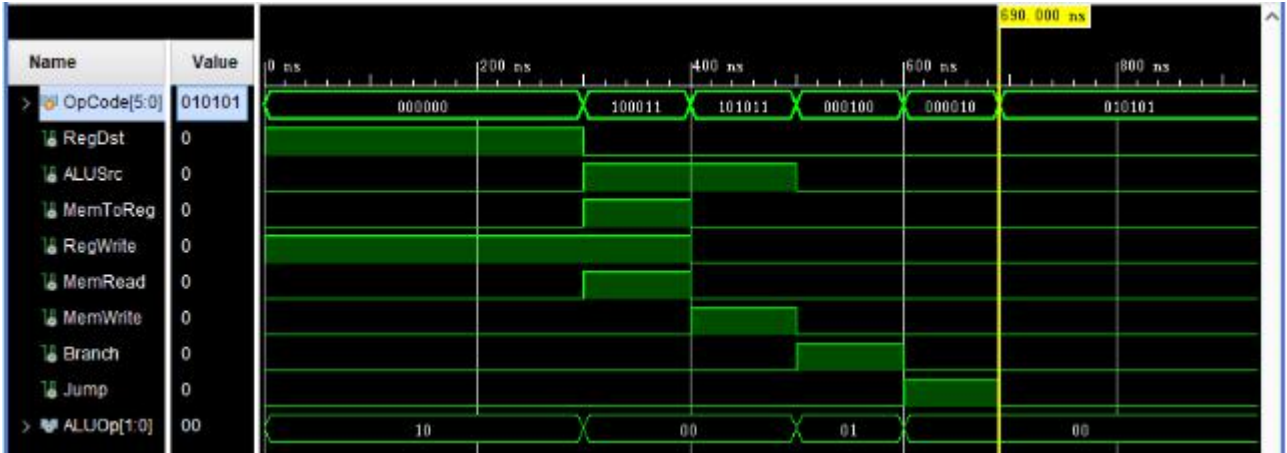
 上海交通大学 计算机科学与工程系 Dept. of Computer Science & Engineering	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB3 作者 CETC	CSE-COA-LAB-003 修改日期 03/21/2019	0.5	7 of 15 公开

2. 添写如下激励。要设定不同的输入，覆盖所有的情况，以保证逻辑的正确。观察波形是否满足逻辑。若由误，修改代码，重新仿真

```
51  
52 initial begin  
53     // Initialize Inputs  
54     OpCode = 0;  
55  
56     // Wait 100 ns for global reset to finish  
57     #100;  
58  
59     #100 OpCode = 6'b0000000; //R-type  
60     //Add orther stimulus here  
61
```

PS:编写激励代码时要像 Lab1 或 Lab2 一样对要仿真的模块进行实例化

3. 下面给出仿真波形样例（这是原先线下要检查的也是你现在仿真要得到的 请截图）：



Ctr 的仿真波形

4. 观察波形，查看仿真结果，是否满足预期设计。如果有错，检查代码逻辑，重新仿真

3.2 实验报告

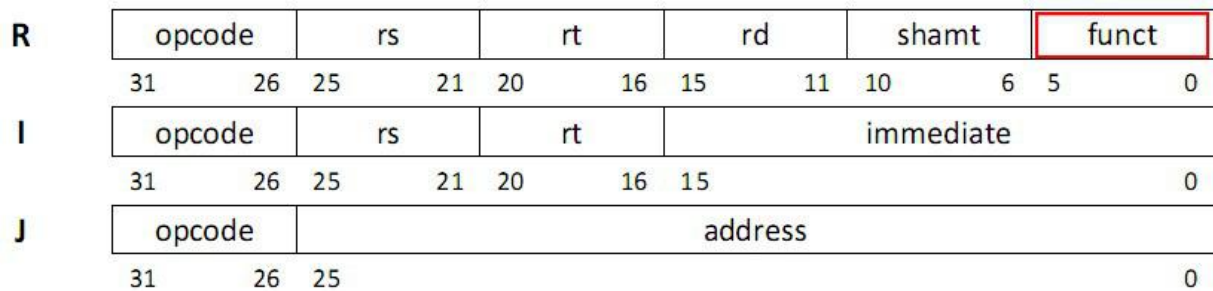
<div>上海交通大学</div> <div>计算机科学与工程系</div> <div>Dept. of Computer Science & Engineering</div>	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB3	CSE-COA-LAB-003	0.5	8 of 15
	作者	修改日期	公开	
	CETC	03/21/2019		

4. 算术逻辑单元（ALU）控制器模块

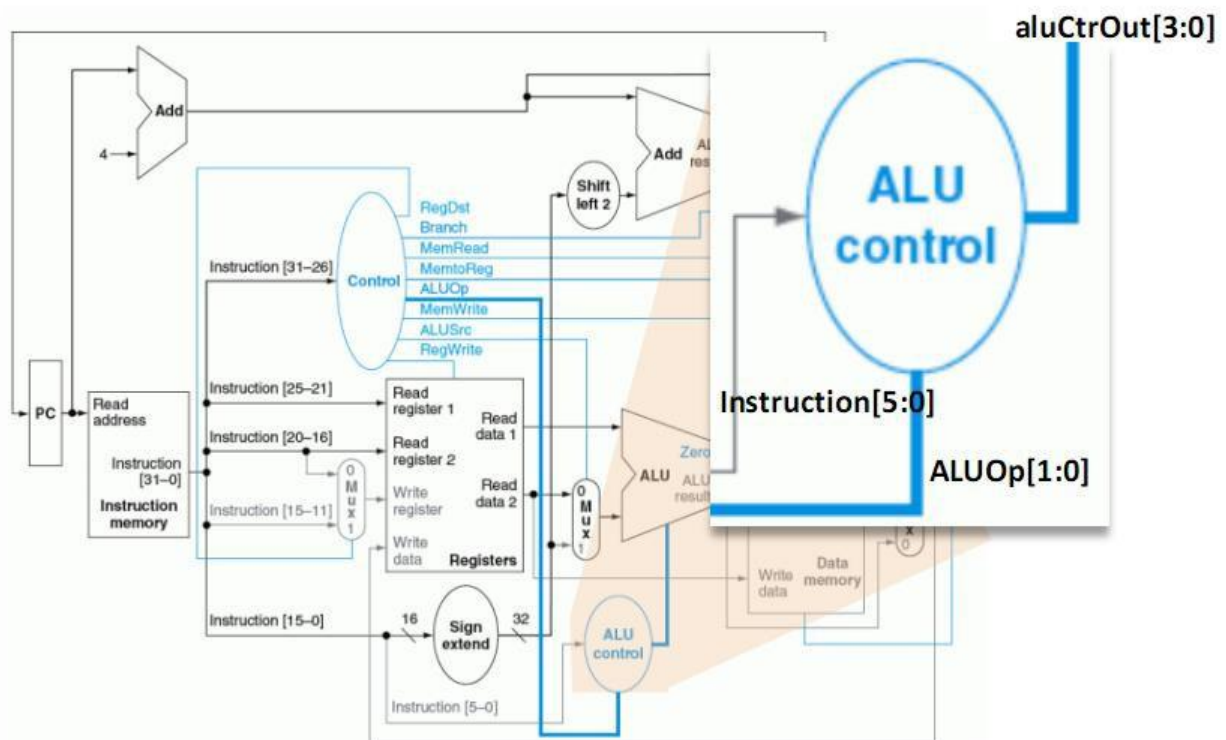
4.1 实验描述

4.1.1 模块描述

ALU 的控制器模块（ALUCtr）是根据主控制器的 ALUOp 来判断指令类型。根据指令的后 6 位区分 R 型指令。综合这两种输入，控制 ALU 做正确的操作。



Mips 基本指令格式



ALU 控制器模块

<div>上海交通大学</div> <div>计算机科学与工程系</div> <div>Dept. of Computer Science & Engineering</div>	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB3	CSE-COA-LAB-003	0.5	9 of 15
	作者	修改日期		
	CETC	03/21/2019	公开	

4.1.2 新建模块源文件

略（可参照之前的方法，比如 3.1.2 的步骤）

PS: 现在起所有的实验主要基于所学的理论知识和前期的实验积累依靠发挥同学们的聪明才智和实干探索精神独立完成（实际遇到不解之处可搜索问题资源的解决经验、代码编写技巧、设计实现思路等）

4.1.3 编写代码实现功能

ALU Control（即 ALU 的输出 `aluCtrOut[3:0]`）的值与 ALU 操作的对应关系如下：

ALU control lines	Function
0000	AND
0001	OR
0010	add
0110	subtract
0111	set on less than
1100	NOR


aluCtrOut 和 alu 操作的对应关系

Instruction opcode	ALUOp	Instruction operation	Func field	Desired ALU action	ALU control input
LW	00	load word	XXXXXX	add	0010
SW	00	store word	XXXXXX	add	0010
Branch equal	01	branch equal	XXXXXX	subtract	0110
R-type	10	add	100000	add	0010
R-type	10	subtract	100010	subtract	0110
R-type	10	AND	100100	and	0000
R-type	10	OR	100101	or	0001
R-type	10	set on less than	101010	set on less than	0111

Func field, ALUOp 与 ALU Control 编码关系

ALUOp		Func field						Operation
ALUOp1	ALUOp0	F5	F4	F3	F2	F1	F0	
0	0	X	X	X	X	X	X	0010
X	1	X	X	X	X	X	X	0110
1	X	X	X	0	0	0	0	0010
1	X	X	X	0	0	1	0	0110
1	X	X	X	0	1	0	0	0000
1	X	X	X	0	1	0	1	0001
1	X	X	X	1	0	1	0	0111

输入输出真值表

 上海交通大学 计算机科学与工程系 Dept. of Computer Science & Engineering	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB3	CSE-COA-LAB-003	0.5	10 of 15
	作者	修改日期	公开	
	CETC	03/21/2019		

请用 verilog 代码写出上述真值表内容

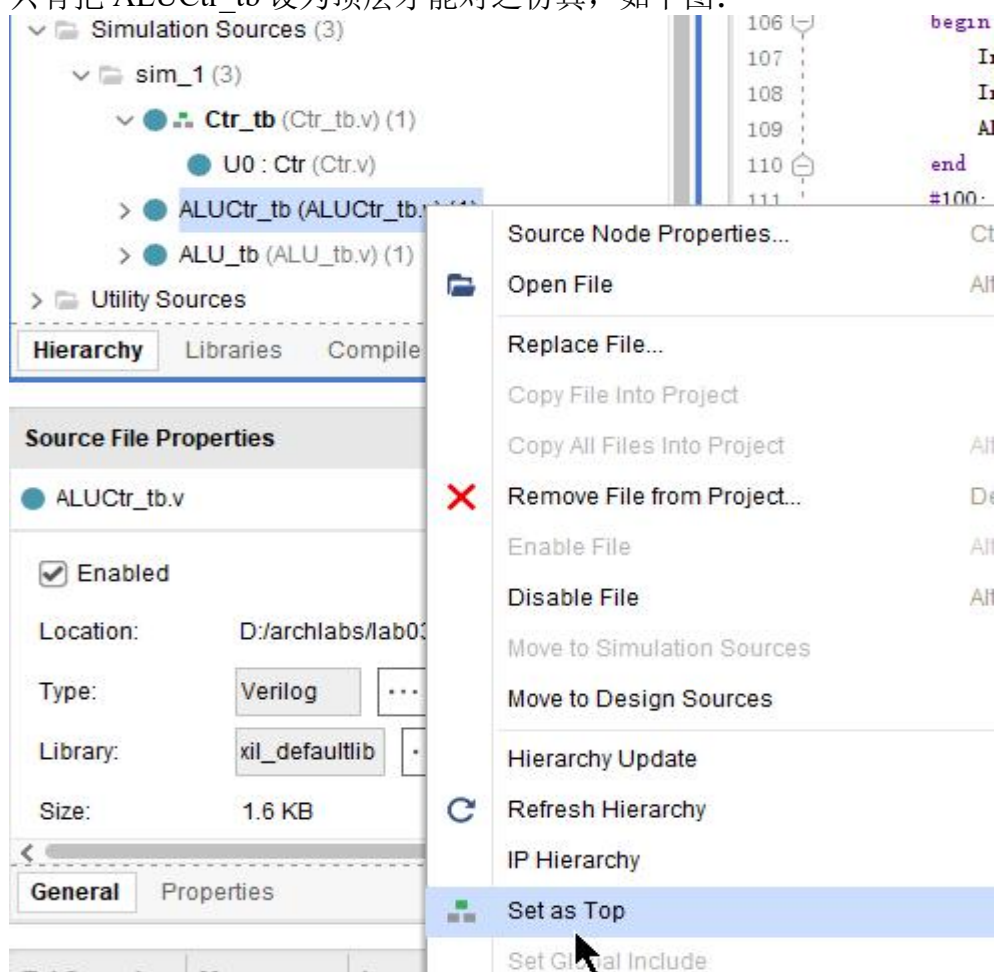
实现方式多种多样，这里给出一种使用 case 语句参考方案（只有一种情况哟，需要补全的），如下图：

```
29     always @ (ALUOp or Funct )
30     begin
31         casex ({ALUOp, Funct})
32
33             8'b00xxxxxxx : ALUCtrOut = 4'b0010;
34
35             //add orther few situations here
36             ...
37
38     endcase
```

注：{a,b}是 verilog 位拼接运算符

4.1.4 仿真测试

1. 新建 ALUCtr_tb
2. 在测试文件中设定不同的输入，需覆盖全部情形
3. 只有把 ALUCtr_tb 设为顶层才能对之仿真，如下图：



上海交通大学 计算机科学与工程系 Dept. of Computer Science & Engineering	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB3	CSE-COA-LAB-003	0.5	11 of 15
	作者	修改日期	公开	
	CETC	03/21/2019		

4. 下面给出仿真波形样例（这是原先线下要检查的也是你做仿真要得到的 请截图）：

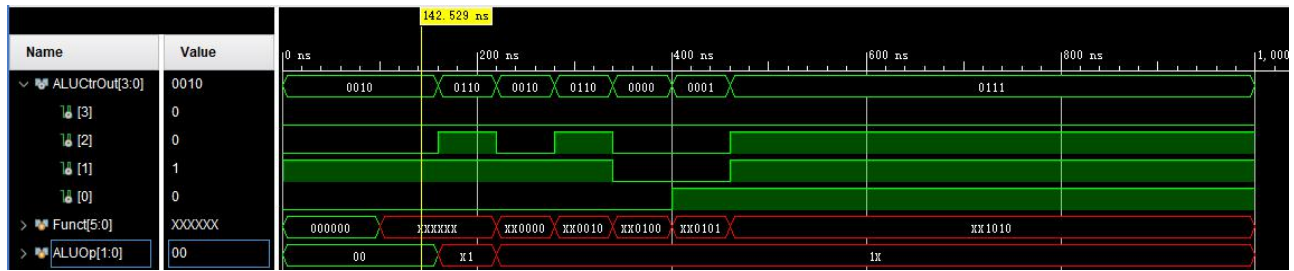


图 A 仿真波形

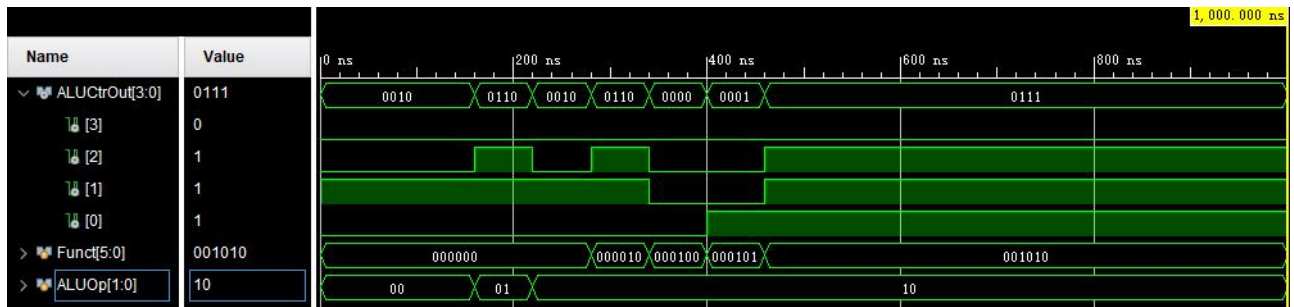


图 B 仿真波形

以上两种仿真波形都正确。注意图 A 和图 B 的代码区别

4.2 实验报告

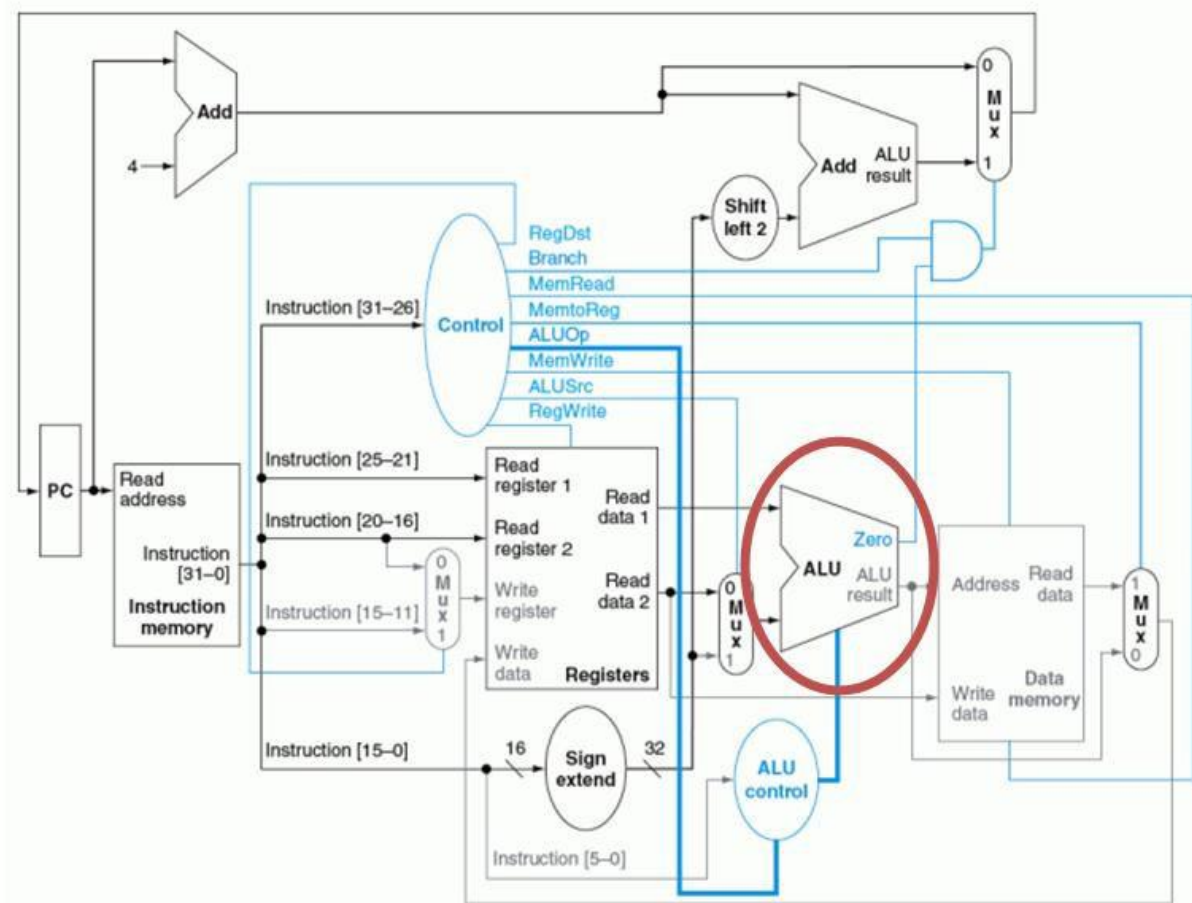
<div>上海交通大学</div> <div>计算机科学与工程系</div> <div>Dept. of Computer Science & Engineering</div>	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB3	CSE-COA-LAB-003	0.5	12 of 15
	作者	修改日期	公开	
	CETC	03/21/2019		

5. ALU 模块

5.1 实验描述

5.1.1 模块描述


算术逻辑单元 ALU 根据 ALUCtr 信号将两个输入执行对应的操作，ALURes 为输出结果。若做减法操作，当 ALURes 结果为 0 时，则 Zero 输出置为 1。



ALU 模块

5.1.2 新建模块源文件

略

 上海交通大学 计算机科学与工程系 Dept. of Computer Science & Engineering	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB3	CSE-COA-LAB-003	0.5	13 of 15
	作者 CETC	修改日期 03/21/2019	公开	

5.1.3 实现功能

我们约定 aluCtrOut[3:0]的值与 ALU 操作的对应关系如下：

ALU control lines	Function
0000	AND
0001	OR
0010	add
0110	subtract
0111	set on less than
1100	NOR

注：beq 实际是个减法操作

用 verilog 代码实现 ALU 功能

实现方式多种多样，比如 case 语句；但这里给出另一种参考方案（仅部分代码哟），如下图：

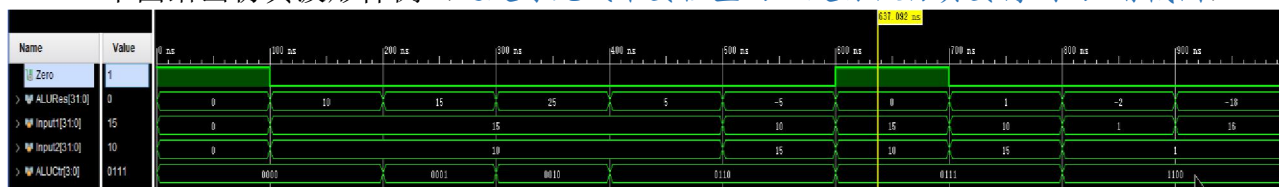
```
21 module Alu(input1, input2, aluCtr, zero, aluRes);
22     input [31:0] input1;
23     input [31:0] input2;
24     input [3:0] aluCtr;
25     output zero;
26     output [31:0] aluRes;
27     reg zero;
28     reg [31:0] aluRes;
29
30     always @ (input1 or input2 or aluCtr)
31     begin
32         if (aluCtr == 4'b0010) // add
33             aluRes = input1 + input2;
34         else if (aluCtr == 4'b0110) // sub
35         begin
36             aluRes = input1 - input2;
37             if (aluRes == 0)
38                 zero = 1;
39             else
40                 zero = 0;
41         end
42         // add and, or, slt here
43     end
44 endmodule
```

上海交通大学 计算机科学与工程系 Dept. of Computer Science & Engineering	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB3 作者 CETC	CSE-COA-LAB-003 修改日期 03/21/2019	0.5	14 of 15 公开

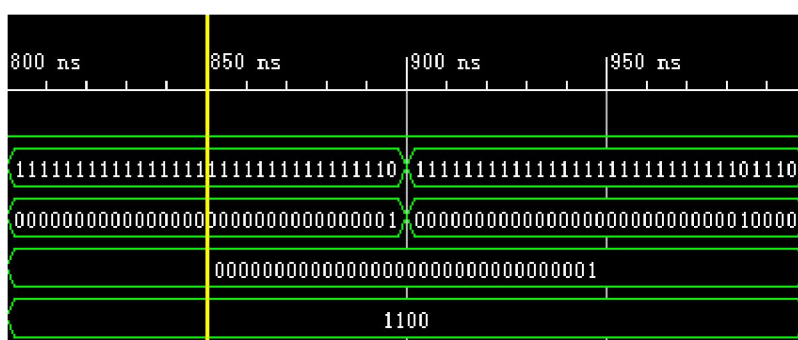
5.1.4 行为仿真

1. 新建测试文件 ALU_tb

2. 下面给出仿真波形样例（这是原先线下要检查的也是你做仿真要得到的 请截图）



Alu 仿真波形



其中 NOR 运算展示

5.2 实验报告

PS: Ctr、ALUctr 和 ALU 这三个模块的实验报告也可以合并起来最后总结写

上海交通大学 计算机科学与工程系 Dept. of Computer Science & Engineering	标题	文档编号	版本	页
	计算机系统结构实验指导书 LAB3	CSE-COA-LAB-003	0.5	15 of 15
	作者 CETC	修改日期 03/21/2019	公开	