数字逻辑与处理器基础大作业

单周期处理器

本次作业中，你需要完成一个单周期处理器的控制器部分。待完成的处理器应能实现 MIPS 指令集的一个子集，包括：

lw, sw, lui

add, addu, sub, subu, addi, addiu

and, or, xor, nor, andi, sll, srl, sra, slt, sltu, sltiu

beq, j, jal, jr, jalr

指令格式可以参考教材附录 A，[1]也可以参考下表。

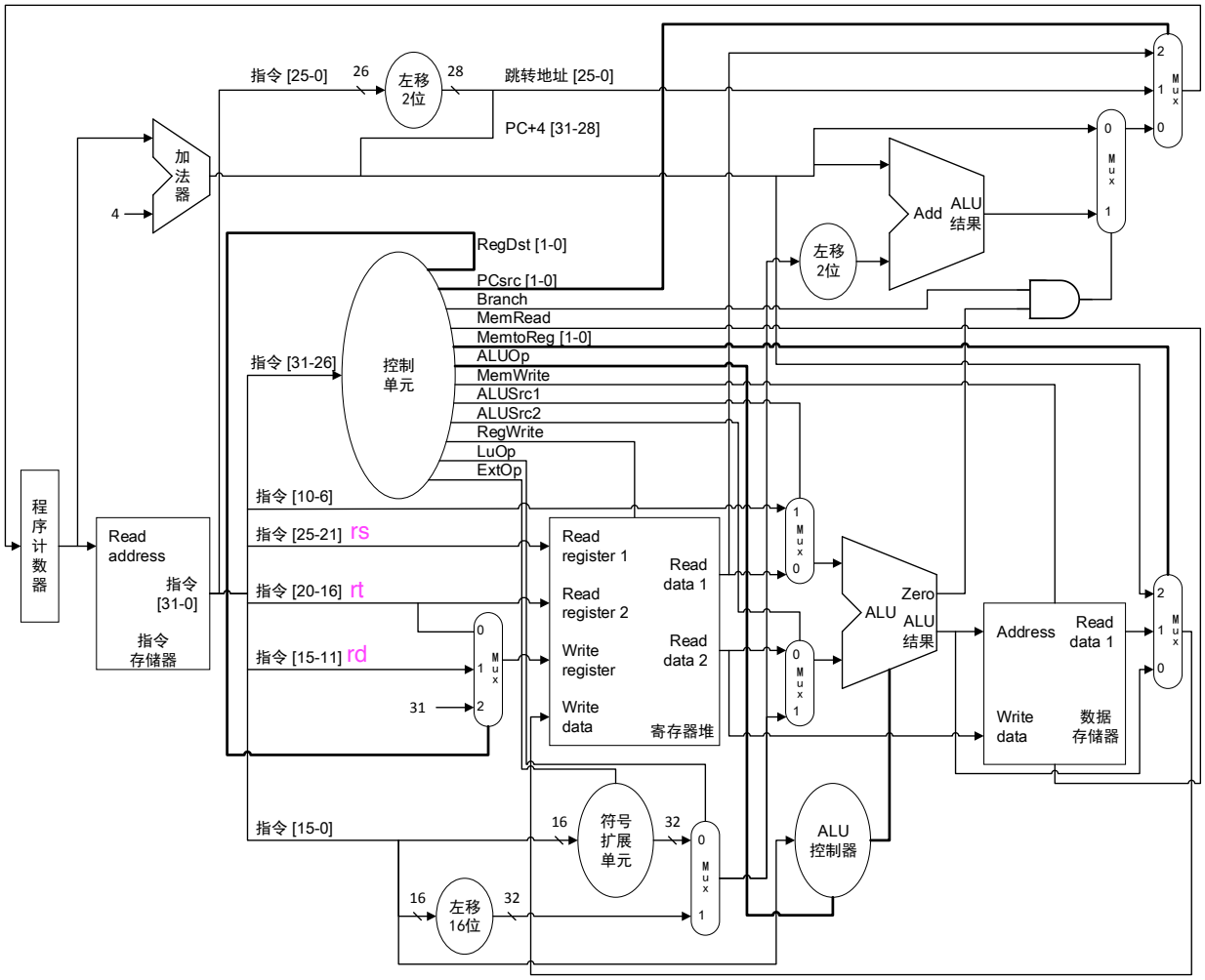
Instruction	OpCode[5:0]	rs[4:0]	rt[4:0]	rd[4:0]	shamt[4:0]	funct[5:0]
lw rt, offset(rs)	0x23	rs	rt	offset		
sw rt, offset(rs)	0x2b	rs	rt	offset		
lui rt, imm	0x0f	0	rt	imm		
add rd, rs, rt	0	rs	rt	rd	0	0x20
addu rd, rs, rt	0	rs	rt	rd	0	0x21
sub rd, rs, rt	0	rs	rt	rd	0	0x22
subu rd, rs, rt	0	rs	rt	rd	0	0x23
addi rt, rs, imm	0x08	rs	rt	imm		
addiu rt, rs, imm	0x09	rs	rt	imm		
and rd, rs, rt	0	rs	rt	rd	0	0x24
or rd, rs, rt	0	rs	rt	rd	0	0x25
xor rd, rs, rt	0	rs	rt	rd	0	0x26
nor rd, rs, rt	0	rs	rt	rd	0	0x27
andi rt, rs, imm	0x0c	rs	rt	imm		
sll rd, rt, shamt	0	0	rt	rd	shamt	0
srl rd, rt, shamt	0	0	rt	rd	shamt	0x02
sra rd, rt, shamt	0	0	rt	rd	shamt	0x03

[1] （第三版）教材附录有两处错误。475 页 nor 指令应为“或非”，而非“异或”；480 页“jalr rs, rd”应改为“jalr rd, rs”。

slt rd, rs, rt	0	rs	rt	rd	0	0x2a
sltu rd, rs, rt	0	rs	rt	rd	0	0x2b
slti rt, rs, imm	0x0a	rs	rt	imm		
sltiu rt, rs, imm	0x0b	rs	rt	imm		
beq rs, rt, label	0x04	rs	rt	offset		
j target	0x02	target				
jal target	0x03	target				
jr rs	0	rs	0			0x08
jalr rd, rs	0	rs	0	rd	0	0x09

1 处理器结构

本次作业采用的处理器结构如下。



阅读处理器的结构图，理解其中各控制信号的含义。注意与教材（第三版）204 页图 5-24 的异同。

1．试回答以下问题：

a) 由 RegDst 信号控制的多路选择器，输入 2 对应常数 31。这里的 31 代表什么？在执行哪些指令时需要 RegDst 信号为 2？为什么？

b) 由 ALUSrc1 信号控制的多路选择器，输入 1 对应的指令[10-6]是什么？在执行哪些指令时需要 ALUSrc1 信号为 1？为什么？

c) 由 MemtoReg 信号控制的多路选择器，输入 2 对应的是什么？在执行哪些指令时需要 MemtoReg 信号为 2？为什么？

d) 图中的处理器结构并没有 Jump 控制信号，取而代之的是 PCSrc 信号。PCSrc 信号控制的多路选择器，输入 2 对应的是什么？在执行哪些指令时需要 PCSrc 信号为 2？为什么？

e) 为什么需要 ExtOp 控制信号？什么情况下 ExtOp 信号为 1？什么情况下 ExtOp 信号为 0？

f) 若想再多实现一条指令 nop（空指令），指令格式为全 0，需要如何修改处理器结构？

2．根据对各控制信号功能的理解，填写如下真值表（填 0,1,2,x 等）。

	PCSrc[1:0]	Branch	RegWrite	RegDst[1:0]	MemRead	MemWrite	MemtoReg[1:0]	ALUSrc1	ALUSrc2	ExtOp	LuOp
lw											
sw											
lui											
add											
addu											
sub											
subu											
addi											
addiu											
and											
or											
xor											
nor											

andi											
sll											
srl											
sra											
slt											
sltu											
slti											
sltiu											
beq											
j											
jal											
jr											
jalr											

2 完成控制器

code 文件夹中是处理器的 Verilog 文件。

1. CPU.v 实现了处理器的整体结构。阅读 CPU.v，理解其实现方式。

2. Control.v 是控制器模块的代码。完成 Control.v。

3. 阅读 InstructionMemory.v，根据注释理解指令存储器中的程序。

MIPS Assembly

```
 0        addi $a0, $zero, 12345
 1        addiu $a1, $zero, -11215
 2        sll $a2, $a1, 16
 3        sra $a3, $a2, 16
 4        beq $a3, $a1, L1
 5        lui $a0, -11111
    L1:
 6        add $t0, $a2, $a0
 7        sra $t1, $t0, 8
 8        addi $t2, $zero, -12345
 9        slt $v0, $a0, $t2
10        sltu $v1, $a0, $t2
    Loop:
11        j Loop
```

这段程序执行足够长时间后会发生什么？此时寄存器$a0~$a3,$t0~$t2,$v0~$v1 中的值应是多少？写出计算过程。注意理解有符号数、无符号数以及各种进制表示的数之间的关系。如果已知某一时刻在某寄存器中存放着数 0xffffcfc7，能否判断出它是有符号数还是无符号数？为什么？

4. 使用 ModelSim 等仿真软件进行仿真。仿真顶层模块为 test_cpu，这是一个 testbench，用于向 CPU 提供复位和时钟信号。观察仿真结果中各寄存器和控制信号的变化。回答以下问题：

   a) PC 如何变化？

   b) Branch 信号在何时为 1？它引起了 PC 怎样的变化？

   c) 100~200ns 期间，PC 是多少？对应的指令是哪条？此时$a1 的值是多少？200~300ns 期间$a1 的值是多少？为什么会这样？下一条指令立即使用到了$a1 的值，会出现错误吗？为什么？

   d) 运行时间足够长之后（如 1100ns 时）寄存器$a0~$a3,$t0~$t2,$v0~$v1 中的值是多少？与你的预期是否一致？

3 执行汇编程序

阅读并理解下面这段汇编程序。

MIPS Assembly

```
 0        addi $a0, $zero, 3
 1        jal sum
    Loop:
 2        beq $zero, $zero, Loop
    sum:
 3        addi $sp, $sp, -8
 4        sw $ra, 4($sp)
 5        sw $a0, 0($sp)
 6        slti $t0, $a0, 1
 7        beq $t0, $zero, L1
 8        xor $v0, $zero, $zero
 9        addi $sp, $sp, 8
10        jr $ra
    L1:
11        addi $a0, $a0, -1
12        jal sum
13        lw $a0, 0($sp)
14        lw $ra, 4($sp)
15        addi $sp, $sp, 8
16        add $v0, $a0, $v0
17        jr $ra
```

1. 如果第一行的 3 是任意正整数 n，这段程序能实现什么功能？Loop，sum，L1 各有什么作用？为每一句代码添加注释。

2. 将这段汇编程序翻译成机器码。
   对于 beq 和 jal 语句中的 Loop，sum，L1，你是怎么翻译的？立即数-1、-8 被翻译成了什么（用 16 进制或 2 进制表示）？

3. 修改 InstructionMemory.v，使 CPU 运行上面这段程序。注意 case 语句的输入是地址的[9-2]比特。仿真观察各控制信号和寄存器的变化。

   a) 运行时间足够长之后（如 5000ns 时），寄存器$a0,$v0 的值是多少？和你预期的程序功能是否一致？

   b) 观察、描述并解释 PC,$a0,$v0,$sp,$ra 如何变化。