计算机组成原理

课程设计报告

学　　号 17020310

姓　　名 李泊岩

指导教师 魏坚华

提交日期 2019 年 5 月 11 日

成绩评价表

报告内容	报告结构	报告最终成绩
□丰富正确 □基本正确 □有一些问题 □问题很大	□完全符合要求 □基本符合要求 □有比较多的缺陷 □完全不符合要求	
报告与 Project 功能一致性	报告图表	总体评价
□完全一致 □基本一致 □基本不一致	□符合规范 □基本符合规范 □有一些错误 □完全不正确	

教师签字:_____

目录

一、 设计说明

1. 处理器应实现 MIPS-Lite1 指令集。
   a) MIPS-Lite1＝{MIPS-Lite，addi，addiu， slt，jal，jr}。
   b) MIPS-Lite 指令集：addu，subu，ori，lw，sw，beq，lui，j。
   c) addi 应支持溢出。
2. 处理器为单周期设计。

二、 模块定义

1. GPR（寄存器组）

a) 基本描述

GPR 主要部件是 32 个 32 位寄存器，作为 MIPS 32 微处理器的 32 个寄存器单元。

b) 模块接口

信号名	方向	描述
WD	I	需要写入寄存器组的数据。
RegWr	I	寄存器写使能信号。 1：写使能 0：写无效
Clk	I	时钟信号
Rst	I	复位信号。 1：复位 0：无效
Rs	I	寄存器地址 rs。
Rt	I	寄存器地址 rt。
Rd	I	寄存器地址 rd。
RD1	O	寄存器输出数据 1。
RD2	O	寄存器输出数据 2。

c) 功能定义

序号	功能名称	功能描述
1	复位	当复位信号有效时，32 个寄存器全部置为 0x0000_0000。
2	读写数据	当写使能信号有效时，按照指令要求向寄存器中读取/写入内容。 当写使能信号无效时，可读取寄存器中的内容。

2. ALU（运算器）

a) 基本描述

ALU 是微处理器的运算单元。主要处理输入两数据之间的运算关系。本运

算器无数据溢出功能。

b) 模块接口

信号名	方向	描述
RD1[31:0]	I	输入操作数 1。
RD2[31:0]	I	输入操作数 2。
Imm32[31:0]	I	32 位立即数 imm32。
ALUSrc	I	运算器第二个输入 B 数据来源。
ALUOp	I	运算器功能选择信号。 00：加法运算（除 ADDI） 01：减法运算 10：或运算 11：SLT 运算 100：ADDI 运算
Result	O	输出数据。
Overflow	O	数据溢出标志。 0：否 1：是
Zero	O	判断输出值是否为 0。 0：否 1：是

c) 功能定义

序号	功能名称	功能描述
1	加法	当功能选择信号是 00 时，两操作数相加。
2	减法	当功能选择信号是 01 时，两操作数相减。
3	或	当功能选择信号是 10 时，两操作数相或。
4	SLT 比较	当功能选择信号是 11 时，两操作数执行 slt 比较
5	Lui	当功能选择信号是 100 时，操作数执行 lui 功能。

3. EXT（扩展单元）

a) 基本描述

EXT 为微处理器的立即数扩展单元。

b) 模块接口

信号名	方向	描述
Imm16	I	输入 16 位立即数。
ExtOp	I	数据扩展单元功能选择信号。 00：零扩展 01：符号扩展 10：LUI 扩展
Imm32	O	输出 32 位数据。

c) 功能定义

序号	功能名称	功能描述

1	零扩展	当功能选择信号是 00 时，立即数进行零扩展。
2	符号扩展	当功能选择信号是 01 时，立即数进行符号扩展。
3	LUI 扩展	当功能选择信号是 10 时，立即数进行 LUI 扩展。

4. DM（数据存储器）

a) 基本描述

DM 为微处理器的数据存储模块，其大小为 32 字 32 位数据存储。

b) 模块接口

信号名	方向	描述
clk	I	时钟信号。
Addr[11:2]	I	目的数据存储器地址
MemWr	I	存储器写使能信号。 1：写使能 0：写无效
Din[31:0]	I	数据输入。
Dout[31:0]	O	数据输出。

c) 功能定义

序号	功能名称	功能描述
1	数据写入	当写使能信号有效时，按照指令要求向寄存器中写入内容。
2	数据读取	任何时候可按照指令要求向寄存器中读取内容。

5. Controller（控制器）

a) 基本描述

Controller 是微处理器的控制器，通过读取 Instr 指令，并对指令进行译码，最终转换为对其他元件的控制信号。

b) 模块接口

信号名	方向	描述
Opcode[5:0]	I	输入指令高 6 位，判断指令种类。
Funct[5:0]	I	输入指令低 6 位，判断 R 型指令的种类。
Overflow	I	输入是否存在溢出信号。
RegDst	O	寄存器目的位置。 0：Instr[21:25] 1：Instr[16:20]
ALUSrc	O	运算器数据来源选择信号。 0：寄存器 1：立即数扩展
Mem2Reg	O	存入寄存器数据来源选择信号。 0：来源于运算器 1：来源于存储器
RegWr	O	寄存器写使能信号。

		1：写使能 0：写无效
MemWr	O	存储器写使能信号。 1：写使能 0：写无效
nPC_sel	O	下一条指令地址选择信号。 0：PC+4 1：PC+4+imm
ExtOp[1:0]	O	数据扩展单元功能选择信号。 00：零扩展 01：符号扩展 10：LUI 扩展
ALUOp[2:0]	O	运算器功能选择信号。 00：加法 01：减法 10：或运算
J/JAL	O	跳转 j/jal 指令信号。 1：有效 0：无效
JR	O	跳转 jr 指令信号。 1：有效 0：无效

c) 控制信号真值表

1）R 型指令

Funct	100000	100001	100010	100011	101010	001000	100000	100001
Opcode	000000	000000	000000	000000	000000	000000	000000	000000
	add	addu	sub	subu	slt	jr	add	addu
RegDst	1	1	1	1	X	X	1	1
ALUSrc	0	0	0	0	0	X	0	0
Mem2Reg	0	0	0	0	0	X	0	0
RegWr	1	1	1	1	1	0	1	1
MemWr	0	0	0	0	0	0	0	0
nPC_sel	0	0	0	0	0	0	0	0
ExtOp[1:0]	X	X	X	X	X	X	X	X
ALUOp[2:0]	0	0	1	1	11	X	0	0
J/JAL	0	0	0	0	0	0	0	0
JR	0	0	0	0	0	1	0	0

2）I 型指令

Opcode	001000	001001	001101	100011	101011	001111	000100	001000
	addi	addiu	ori	lw	sw	lui	beq	addi
RegDst	0	0	0	0	X	X	X	0
ALUSrc	1	1	1	1	1	1	X	1

Mem2Reg	0	0	0	1	X	X	X	0
RegWr	1	1	1	1	0	1	0	1
MemWr	0	0	0	0	1	0	0	0
nPC_sel	0	0	0	0	0	0	1	0
ExtOp[1:0]	0	0	0	1	1	10	X	0
ALUOp[2:0]	100	0	10	0	0	10	1	100
J/JAL	0	0	0	0	0	0	0	0
JR	0	0	0	0	0	0	0	0

3）J 型指令

Opcode	000010	000011
	j	jal
RegDst	X	10
ALUSrc	X	X
Mem2Reg	X	10
RegWr	0	1
MemWr	0	0
nPC_sel	0	0
ExtOp[1:0]	X	X
ALUOp[2:0]	X	X
J/JAL	1	1
JR	0	0

6. PC（指令计算）

(1) 基本描述

PC 主要功能是完成输出当前指令地址并保存下一条指令地址。复位后，PC 指向 0x0000_3000，此处为第一条指令的地址。

(2) 模块接口

信号名	方向	描述
NPC[31:2]	I	下条指令的地址
clk	I	时钟信号
Reset	I	复位信号。 1：复位 0：无效
PC[31:2]	O	30 位指令存储器地址(最低 2 位省略)

(3) 功能定义

序号	功能名称	功能描述
1	复位	当复位信号有效时，PC 被设置为 0x0000_3000。
2	保存 NPC 并输出	在每个 clock 的上升沿保存 NPC，并输出。

7. nPC（下一条指令读取）

(1) 基本描述

nPC 主要功能是根据指令内容。

(2) 模块接口

信号名	方向	描述
curPC[29:0]	I	30 位当前指令存储器地址(最低 2 位省略)
nPC_sel	I	下地址选择信号功能。 1：BEQ 0：PC+4
Zero	I	复位信号。 1：复位 0：无效
J	I	跳转 j/jal 指令信号。 1：有效 0：无效
JR	I	跳转 jr 指令信号。 1：有效 0：无效
j_imm26[25:0]	I	跳转指令 26 位立即数地址。
imm16[15:0]	I	普通 I 指令 16 位立即数。
Data[31:0]	I	读入的 32 位数据。
nPC[29:0]	O	下一个指令存储器地址。
PC32[31:0]	O	32 位指令存储器地址。

(3) 功能定义

序号	功能名称	功能描述
1	复位	当复位信号有效时，PC 被设置为 0x0000_3000。
2	保存 NPC 并输出	在每个 clock 的上升沿保存 NPC，并输出。

三、 测试程序及原理

```
1  addu $8, $0, 0x8                #$s8:   0x8
2  ori $16, $0, 1                  #$16:   0x1
3  ori $17, $0, 3                  #$8:    0x3
4  ori $8, $0, 0x2                 #$8:    0x2
5  ori $12, $0,0xabab              #$12:   0xabab
6  lui $13, 10                     #$13:   0xa
7  start:addu $4, $0, $16          #$4:    0x1
8  addu $5, $0,$8                  #$5:    0x2
9  jal newadd
10 addu $16, $0, $2                #$16:   3
11 subu $17,$17,$8                 #$17:   0xfffffffe
12 beq $16, $17, start             #false
13 ori $8, $0, 1                   #$8:    0x1
14 addiu $24,$0,0x7fffffff         #$24:   0x7fffffff
15 addiu $9,$24,3                  #$9:    0x80000002
16 addiu $10,$24,5                 #$10:   0x80000004
17 addu $0,$0,$0
18 start2:sw $9, 0($8)
19 lw $14, 0($8)                   #$14:   0x80000002
20 sw $10,4($8)
21 lw $15,4($8)                    #$15:   0x80000004
22 sw $4, -4($8)
23 lw $18, -4($8)                  #$18:   0x00000001
24 addu $4,$0,$8                   #$4:    0x1
25 addu $5,$0,$9                   #$5:    0x80000002
26 jal newadd                      #$2:    0x80000003
27 slt $25,$10,$8                  #$25:   0
28 beq $25, $0,end2                #true
29 slt $20,$12,$4
30 beq $20, $0, end1
31 lui $12, 65535
32 end1:ori $0, $0,1
33 lui $19, 0xefef
34 addiu $3,$0,0xababcdcd
35 start3:addiu $4, $3, 2
36 addi $23, $3, 5
37 jal newadd
38 addu $8, $0, $2
39 addu $4, $0, $8
40 addu $5, $0, $9
41 jal newadd
```

```
addu $9, $0, $2
addu $9, $8, $0
lui $10, 0x69
beq $8, $9, start4
beq $0, $0, start3
start4:j end
newadd:addu $2, $4, $5          #$2:    3
addi $0,$12,0x7823              #$0:    0
jr $31
end2:addi $25,$0,0x1234         #25:    0x1234
end:
```

四、 问答

1. 请说明为什么在忽略溢出的前提下，addi 与 addiu 是等价的，add 与 addu 是等价的。

在忽略溢出的前提下，由于二进制的特殊性，有循环功能，即加到最大值后，再加 1 自动归零。因此，在忽略溢出的前提下，addi 和 addiu 在数值取值范围和计算规律上都没有任何区别，add 和 addu 亦然。