

计算机组成原理实验报告

一、CPU 设计方案综述

（一）总体设计概述

本 CPU 为 Logisim 实现的单周期 MIPS - CPU, 支持的指令集包含 {addu、subu、ori、lw、sw、beq、lui、sll、srl、sra、nop}。为了实现这些功能, CPU 主要包含了 PC、NPC、IM、GRF、ALU、DM、EXT……, 这些模块按照自顶向下的顶层设计逐级展开。

（二）关键模块定义

1. GRF

GRF 中包含 32 个 32 位寄存器, 分别对应 0~31 号寄存器, 其中 0 号寄存器读取的结果恒为 0。具体模块端口定义如下:

信号名	方向	描述
CLK	I	时钟信号
RST	I	异步复位信号
WE3	I	写使能信号
A1	I	读地址
A2	I	读地址
A3	I	写地址
WD	I	32 位写入数据
RD1	O	32 位读出数据
RD2	O	32 位读出数据

2. DM

信号名	方向	描述
CLK	I	时钟信号

A	I	读地址
WD	I	待写入 DM 的 32 位数据信号
WE	I	写数据使能信号
RST	I	异步复位信号
RD	O	DM 中 A 地址读出的数据

3. ALU

ALUControl[3:0]	功能	ALUControl[3:0]	功能
0000	A AND B	0100	A AND ~B
0001	A OR B	0101	A OR ~B
0010	A + B	0110	A - B
0011	\	0111	SLT
1000	A 逻辑左移 B	1001	A 逻辑右移 B
1010	A 算数右移 B	1011	\
1100	\	1101	\
1110	\	1111	\

4. Controller

控制器 Controller 分为主译码器 MainDecoder 和 ALU 译码器 ALUDecoder。

主译码器端口声明如下：

主译码器			
端口声明	方向	位宽	描述
OP	I	6	Opcode
REGorMEM	O	1	回写 GRF 信号 WD3 选择
MemWrite	O	1	DM 写使能信号
Branch	O	1	是否为 Beq 指令
ALUSrc	O	1	ALU 中 srcB 信号来源选择
RegDst	O	1	GRF 中 A3 信号选择

RegWrite	O	1	GRF 写使能信号
ZeroEXT	O	1	EXT 扩展方式选择
Lui	O	1	是否为 lui 指令
ALUOp	O	3	作为 ALU 译码器的输入信号

主译码器真值表如下：（本电路实现中 X 全置 0）

指令	OP	REGor MEM	Mem Write	Branch	ALUSrc	RegDst	RegWrite	ZeroEXT	Lui	ALUOp
R 型	000000	0	0	0	0	0	1	X	0	010
lw	100011	1	0	0	1	1	1	0	0	000
sw	101011	X	1	0	1	X	0	0	0	000
beq	000100	X	0	1	0	X	0	X	0	001
ori	001101	0	0	0	1	1	1	1	0	100
lui	001111	X	0	0	X	1	1	0	1	XXX

ALU 译码器端口声明如下：

ALU 译码器			
端口声明	方向	位宽	描述
Funct	I	6	指令最低六位
ALUOp	I	3	从主译码器传入
ALUControl	O	4	控制 ALU 的功能（具体见上面 ALU 部分）

主译码器产生 3 位信号 ALUOp[2:0]传给 ALU 译码器，ALUOp 编码及其对应功能如下：

ALUOp[2:0]	功能	ALUOp[2:0]	功能
000	加法	010	表示指令为 R 指令，依赖于 funct 字段
001	减法	011	与

100	或	101	/
110	/	111	/

ALU 译码器真值表如下：

ALUOp[2:0]	Funct[5:0]	Mid(中间变量)	ALUcontrol[3:0]
000	X	ADD	0010(加)
001	X	SUB	0110(减)
011	X	AND	0000(与)
100	X	OR	0001(或)
010	100001(addu)	ADD	0010(加)
010	100011(subu)	SUB	0110(减)
010	100100(and)	AND	0000(与)
010	100101(or)	OR	0001(或)
010	101010(slt)	SLT	0111(SLT)
010	000000(sll)	SLL	1000(SLL)
010	000011(srl)	SRL	1001(SRL)
010	000010(sra)	SRA	1010(SRA)

(三) 重要机制实现方法

1. 跳转

NPC 模块和 ALU 模块协同工作支持指令 beq 的跳转机制。

当指令为 beq 指令时，ALU 进行减法运算并输出判 0 结果 ALUZero，同时控制器将 Branch 信号置 1。只有当 ALUZero 和 Branch 信号都为 1 时，NPC 才执行 $PC' = PC + 4 + OFFSET \parallel 00$ 跳转操作。

二、测试方案

（一）典型测试样例

1. ALU 功能测试

首先分为两大类功能进行测试，分别测试 $F[3] = 0$ 时的加减与或运算和 $F[3] = 1$ 时的移位运算，通过 MIPS 代码导出至 Logism 中 IM（ROM）内进行测试。

2. DM 功能测试

同样利用 MARS 的代码导出功能对 sw 和 lw 两个指令进行测试。

三、思考题

（一）题目描述：现在我们的模块中 IM 使用 ROM，DM 使用 RAM，GRF 使用 Register，这种做法合理吗？请给出分析，若有改进意见也请一并给出。

合理。首先，IM、DM、GRF 都具有储存数据的功能。其中，IM 为指令存储器，一般情况下不需要写入，只需要读出，故选用 ROM；DM 为数据存储器，既要支持读功能，也要支持写功能，故选用随机访问存储器 RAM；而 GRF 为通用寄存器堆，显然要采用 Register。

（二）题目描述：事实上，实现 nop 空指令，我们并不需要将它加入控制信号真值表，为什么？请给出你的理由。

因为空指令不执行任何操作，只进行 PC 自增。其对应 16 进制机器码为 0x00000000，在不将其添加至控制信号真值表的情况下，其不满足真值表中的任何其他指令输入，故寄存器写入信号为 0，存储器写入信号也为 0，这保证了在指令为 nop 时电路存储不变。而同时，ALU 等算数运算模块依旧进行运算，但不会影响存储，故不需要添加 nop。