

流水线 cpu 设计文档

王郁含 16182672

目录

- 一、 设计与测试说明
- 二、 模块规格
- 三、 数据通路设计
- 四、 控制器设计
- 五、 测试程序
- 六、 思考题

一、设计与测试说明

- 处理器应支持指令集为：{addu, subu, ori, lw, sw, beq, lui, j, jal, jr, nop}。
- addu, subu 可以不支持溢出。
- 处理器为流水线设计。
- 需要考虑延迟槽。
- 顶层文件为 mips.v，接口定义如下：

表 1 模块接口定义

文件	模块接口定义
mips.v	<pre>module mips(clk,reset); input clk; //clock input reset; //reset</pre>

二、模块规格

1. IFU（取指令单元）

- 包括 PC（程序计数器）、IM（指令存储器），及相关逻辑。
- IM 容量为 4KB（32bit×1024 字）。
- 端口定义：

信号名	方向	描述
clk	I	时钟信号
reset	I	复位信号
enable	I	使能端
Npc	I	是否取 npc
NPC	I	Npc 的值
[31:0]PC	O	当前 PC 值
[31:0]PC4	O	下一个 PC 值
[31:0]PC8	O	PC+8
Instr	O	32 位指令信号

- 功能描述：

表 3 IFU 功能描述

功能	功能描述
复位清零	当时钟上升沿到来时，如果复位信号有效，则清零。
取下一条地址上的指令	当时钟上升沿到来且跳转控制失效时，PC + 4 以获取下一地址，使 IM 读取下一条指令
转移至对应地址获取指令	当时钟上升沿到来且转移控制有效，转移至目标地址获取该地址内指令

2. GRF（通用寄存器组）

- 用 32 个具有写使能的寄存器实现。
- 0 号寄存器保持为 0。
- 端口定义：

表 4 GRF 端口定义

信号名	方向	描述
[4:0]RA1	I	读取寄存器 1
[4:0]RA2	I	读取寄存器 2
[4:0]WA	I	写入寄存器地址
[31:0]WD	I	写入数据
Reset	I	复位信号
clk	I	时钟信号
RWE	I	写使能信号
[31:0]PC	I	当前指令信号
[31:0]D1	O	读取数据 1
[31:0]D2	O	读取数据 2

- 功能描述：

表 5 GRF 功能描述

功能	功能描述
复位清零	当时钟上升沿到来时，如果复位信号有效，则清零。
读取寄存器中的数据	当时钟上升沿到来且写使能 WE 失效时，读取 RA1 和 RA2 两个对应号的寄存器中的数据并分别输出至 RD1 和 RD2
读取寄存器中的数据并将数据写入寄存器	当时钟上升沿到来且写使能 WE 有效时，读取 RA1 和 RA2 两个对应号的寄存器中的数据并分别输出至 RD1 和 RD2 同时将 WD 的数据写入 WA 对应号的寄存器

3. ALU（算术逻辑单元）

- 提供 32 位加、减、或运算及大小比较功能。
- 可以不支持溢出。
- 端口定义：

表 6 ALU 端口定义

信号名	方向	描述
[31:0] A	I	32 位被运算数据
[31:0] B	I	32 位被运算数据

[1:0] ALUOp	I	功能选择信号 00:加运算 01:减运算 10:按位或运算
[31:0] ALUOut	0	计算结果

- 功能描述:

表 7 ALU 功能描述

功能	功能描述
加运算	$ALUOut = A + B$
减运算	$ALUOut = A - B$
按位或运算	$ALUOut = A B$

4. DM（数据存储器）

- DM 容量为 4KB（32bit×1024 字）。
- 端口定义:

表 8 DM 端口定义

信号名	方向	描述
[4:0] Add	I	访问的地址
[31:0] WD	I	写入地址的数据
DWE	I	写入使能端
DRE	I	读取使能端
Reset	I	复位信号
Clk	I	时钟信号
[31:0] RD	O	读取地址的数据

- 功能描述

表 9 DM 功能描述

功能	功能描述
复位清零	当时钟上升沿时，如果信号有效，则置零
写入地址	当时钟上升沿且 WE 为 1 时在 Addr 所指地址中写入 WD 所代表数据
读取地址	当时钟上升沿且 RE 为 1 时将 Addr 所指地址中的数据读出至 RD

5. EXT(位数扩展器)

- 端口定义:

表 10 EXT 端口定义

信号名	方向	描述
[15:0] Imm	I	16 位立即数/偏移量输入
[1:0]EXTOp	I	EXT 控制端： 00: 无符号扩展 01: 有符号扩展 10: 左移 16 位
[31:0] EXT	O	扩展后的 32 位立即数/偏移量输出

- 功能描述:

表 11 EXT 功能描述

功能	功能描述
无符号扩展	将 16 位输入进行无符号扩展至 32 位
有符号扩展	当指令为 sw/lw 时，对 16 位输入进行有符号扩展至 32 位
左移 16 位	当指令为 lui 时，将 16 位输入扩展至 32 位后移至高 16 位

6. CMP(比较器)

- 用于 beq 指令前置比较的时候进行数据转发。
- 端口定义:

表 12 CMP 端口定义

信号名	方向	描述
[31:0] A	I	第一个取出来的数
[31:0] B	I	第二个取出来的数
ZERO	O	相等时为 1，不等时为 0

- 功能描述:

功能	功能描述
比较两数大小	相等时为 1，不等时为 0，有助于构造数据转发通路

7. NPC(地址计算器)

- 用于计算下一条指令的地址。
- 端口定义:

表 13 NPC 端口定义

信号名	方向	描述
[31:0] PC4	I	下一条指令的物理地址
[25:0] IMM26	I	可能会被用到的扩展数
[31:0] JRPC	I	Jr 返回的地址
zero	I	Beq 相等的判断
[31:0]NPC	O	输出下一条地址的 pc 值

- 功能描述

表 14 NPC 功能描述

功能	功能描述
----	------

跳转至 beq 指令地址	根据 imm26 和 pc4 计算出 beq 地址，根据 zero 判断是否跳转
跳转至 j/jal 指令地址	根据 imm26 算出的跳转地址
跳转至 jr 指令地址	根据 jrpc 可直接得到

三、 数据通路设计

部件	输入	ADDU	SUBU	ORI	LW	SW	BEQ	LUI	J	JAL	JR
PC											
ADD4		PC	PC	PC	PC	PC	PC	PC	PC	PC	PC
ADD8											
IM		PC	PC	PC	PC	PC	PC	PC	PC	PC	PC
PC		ADD4	ADD4	ADD4	ADD4	ADD4	ADD4/NPC	ADD4	NPC	NPC	RF.RD1
IR@D		IM	IM	IM	IM	IM	IM	IM	IM	IM	IM
PC@D		PC	PC	PC	PC	PC	PC	PC	PC	PC	PC
PC4@D							ADD4				
PC8@D										ADD8	
RF	A1	IR@D[RS]	IR@D[RS]	IR@D[RS]	IR@D[RS]	IR@D[RS]	IR@D[RS]				IR@D[RS]
	A2	IR@D[RT]	IR@D[RT]			IR@D[RT]	IR@D[RT]				
EXT				IR@D[16]	IR@D[16]	IR@D[16]		IR@D[16]			
CMP	D1						RF.RD1				
	D2						RF.RD2				
NPC	PC4						PC4@D		PC4@D	PC4@D	
	I26						IR@D[16]		IR@D[16]	IR@D[16]	
PC							NPC		NPC	NPC	

IR@E		IR@D	IR@D	IR@D	IR@D	IR@D		IR@D		IR@D	
PC@E		PC@D	PC@D	PC@D	PC@D	PC@D		PC@D		PC@D	
PC4@E											
PC8@E										PC8@D	
RD1@E		RF.RD1	RF.RD1	RF.RD1	RF.RD1	RF.RD1		RF.RD1			
RD2@E		RD.RD2	RD.RD2			RF.RD2					
EXT@E				EXT	EXT	EXT		EXT			
ALU	A	RD1@E	RD1@E	RD1@E	RD1@E	RD1@E		RD1@E			
	B	RD2@E	RD2@E	EXT@E	EXT@E	EXT@E		EXT@E			
IR@M		IR@E	IR@E	IR@E	IR@E	IR@E		IR@E			
PC@M		PC@E	PC@E	PC@E	PC@E	PC@E		PC@E		PC@E	
PC4@M											
PC8@M										PC8@E	
AO@M		ALU	ALU	ALU	ALU	ALU		ALU			ALU
RD2@M						RD2@E					
DM	PC					PC@M					
	A				AO@M	AO@M					
	WD					RD2@M					
IR@W		IR@M	IR@M	IR@M	IR@M	IR@M	IR@M	IR@M	IR@M	IR@M	IR@M
PC@W		PC@M	PC@M	PC@M	PC@M	PC@M	PC@M	PC@M	PC@M	PC@M	PC@M
PC4@W											
PC8@W										PC8@M	
AO@W		AO@M	AO@M	AO@M		AO@M		AO@M			
RD@W					DM						
RF	PC	PC@W	PC@W	PC@W	PC@W			PC@W		PC@W	
	WA	IR@W[RD]	IR@W[RD]	IR@W[RT]	IR@W[RT]			IR@W[RT]		31	
	WD	AO@W	AO@W	AO@W	RD@W			AO@W		PC8@W	

整合来源

输入来源		mux		控制
PC				
PC				
ADD4	NPC	RF.RD1	MUX_PC	PCOp
IM				
PC				
ADD4				
ADD8				
IR@D[RS]				
IR@D[RT]				
IR@D[I16]				
RF.RD1				
RF.RD2				
PC4@D				
IR@D[I26]				
NPC				
IR@D				
PC@D				
PC8@D				
RF.RD1				
RF.RD2				
EXT				
RD1@E				
RD2@E	EXT@E		MUX_ALUB	ALUBOp
IR@E				
PC@E				
PC8@E				
ALU				
RD2@E				
PC@M				
AO@M				
RD2@M				
IR@M				
PC@M				
PC8@M				
AO@M				
DM				
PC@W				
IR@W[RD]	IR@W[RT]	31	MUX_WA	WAOp
AO@W	PC8@W	RD@W	MUX_WD	WDOp

四、控制器设计

1. 功能部件控制信号

	funcode	100001	100011							001000	
	opcode	000000	000000	0011111	0011101	100011	101011	000010	000011	000000	000010
		addu	subu	lui	ori	lw	sw	j	jal	jr	beq
ID级	PCOP	0	0	0	0	0	0	1	1	2	3
	EXTOP	0	0	2	0	1	1	0	0	0	0
	NPC	0	0	0	0	0	0	1	1	1	1
EX级	ALUBOP	1	1	0	0	0	0	0	0	0	0
	ALUOP	0	1	0	2	0	0	0	0	0	0
MEM级	WAOP	1	1	0	0	0	0	0	2	0	0
	WDOP	0	0	0	0	1	0	0	2	0	0
	DWE	0	0	0	0	0	1	0	0	0	0
	DRE	0	0	0	0	1	0	0	0	0	0
WB级	RWE	1	1	1	1	1	0	0	1	0	0

2. 暂停控制

IF/ID 当前指令			ID/EX(T _{new})			EX/MEM (T _{new})	
指令类型	源寄存器	T _{use}	cal-r 1/rd	cal-l 1/rt	load 2/rt	load 1/rt	
beq	rs/rt	0	暂停	暂停	暂停	暂停	
jr	rs	0	暂停	暂停	暂停	暂停	
cal-r	rs/rt	1			暂停		
cal-i	rs	1			暂停		
load	rs	1			暂停		
store	rs	1			暂停		

3. 转发控制

流水级	源寄存器	涉及指令	转发MUX	控制信号	输入0	ID/EX(T_{new})	EX/MEM(T_{new})				MEM/WB(T_{new})			
IF/ID	rs	beq/jr	MFRSD	RSDOP	RF.RD1	PC8_E	cal-r 0/rd	cal-l 0/rt	jal 0/rs	cal-r 0/rd	cal-l 0/rt	load 0/rt	jal 0/rs	
	rt	beq	MFRSD	RTDOP	RF.RD2	PC8_E	AO_M	AO_M	PC8_M	WD_W	WD_W	WD_W	WD_W	
ID/EX	rs	cal-r,cal-l,ld,st	MFRSE	RSEOP	RS_E	无	AO_M	AO_M	PC8_M	WD_W	WD_W	WD_W	WD_W	
	rt	cal-r	MFRTE	RTEOP	RT_E	无	AO_M	AO_M	PC8_M	WD_W	WD_W	WD_W	WD_W	
EX/MEM	rt	store	MFRTM	RTMOP	RT_M	无	无	无	无	WD_W	WD_W	WD_W	WD_W	

4. 数据冒险控制

通过延迟槽解决。

五、测试程序

1. 不包含暂停/转发的测试程序

Address	Code	Basic	Source
0x00003000	0x34210064	ori \$1,\$1,0x00000064	1 ori \$1, \$1, 100
0x00003004	0x340200c8	ori \$2,\$0,0x000000c8	2 ori \$2, \$0, 200
0x00003008	0x3480012c	ori \$0,\$4,0x0000012c	3 ori \$0, \$4, 300
0x0000300c	0x3c050064	lui \$5,0x00000064	4 lui \$5, 100
0x00003010	0x3c0600c8	lui \$6,0x000000c8	5 lui \$6, 200
0x00003014	0x10000007	beq \$0,\$0,0x00000007	6 beq \$0, \$0, label1
0x00003018	0x00000000	nop	7 nop
0x0000301c	0xac270004	sw \$7,0x00000004(\$1)	9 sw \$7, 4(\$1)
0x00003020	0x00430823	subu \$1,\$2,\$3	10 subu \$1, \$2, \$3
0x00003024	0x00680023	subu \$0,\$3,\$8	11 subu \$0, \$3, \$8
0x00003028	0x000a4823	subu \$9,\$0,\$10	12 subu \$9, \$0, \$10
0x0000302c	0x03e00008	jr \$31	13 jr \$31
0x00003030	0x00000000	nop	14 nop
0x00003034	0x00210821	addu \$1,\$1,\$1	16 addu \$1, \$1, \$1
0x00003038	0x00430021	addu \$0,\$2,\$3	17 addu \$0, \$2, \$3
0x0000303c	0x00052021	addu \$4,\$0,\$5	18 addu \$4, \$0, \$5
0x00003040	0x0c000c07	jal 0x0000301c	19 jal label2
0x00003044	0x00000000	nop	20 nop
0x00003048	0x10010005	beq \$0,\$1,0x00000005	21 beq \$0, \$1, exit
0x0000304c	0x00000000	nop	22 nop
0x00003050	0x00000000	nop	23 nop
0x00003054	0x08000c17	j 0x0000305c	24 j label3
0x00003058	0x00000000	nop	25 nop
0x0000305c	0x8c610000	lw \$1,0x00000000(\$3)	27 lw \$1, 0(\$3)
0x00003060	0x1000ffff	beq \$0,\$0,0xffffffff	29 beq \$0, \$0, exit

2. 暂停/转发测试程序

Address	Code	Basic	Source
0x00003000	0x34210064	ori \$1,\$1,0x00000064	1 ori \$1, \$1, 100
0x00003004	0x1001001b	beq \$0,\$1,0x0000001b	2 beq \$0, \$1, label
0x00003008	0x00000000	nop	3 nop
0x0000300c	0x00411021	addu \$2,\$2,\$1	4 addu \$2, \$2, \$1


```

0x00003010 0x10020018 beq $0,$2,0x00000018 5 beq $0, $2, label
0x00003014 0x00000000 nop 6 nop
0x00003018 0xac010000 sw $1,0x00000000($0) 7 sw $1, 0($0)
0x0000301c 0x00000000 nop 8 nop
0x00003020 0x00000000 nop 9 nop
0x00003024 0x8c030000 lw $3,0x00000000($0) 10 lw $3, 0($0)
0x00003028 0x10030012 beq $0,$3,0x00000012 11 beq $0, $3, label
0x0000302c 0x00000000 nop 12 nop
0x00003030 0x8c040000 lw $4,0x00000000($0) 13 lw $4, 0($0)
0x00003034 0x00802823 subu $5,$4,$0 14 subu $5, $4, $0
0x00003038 0x1080000e beq $4,$0,0x0000000e 15 beq $4, $0, label
0x0000303c 0x0c000c11 jal 0x00003044 16 jal ja
0x00003040 0x00000000 nop 17 nop
0x00003044 0x3c010000 lui $1,0x00000000 19 addu $31, $31, 20
0x00003048 0x34210014 ori $1,$1,0x00000014
0x0000304c 0x03e1f821 addu $31,$31,$1
0x00003050 0x03e00008 jr $31 20 jr $31
0x00003054 0xac1f0000 sw $31,0x00000000($0) 21 sw $31, 0($0)
0x00003058 0x8e810000 lw $1,0x00000000($20) 22 lw $1, 0($20)
0x0000305c 0x3c010000 lui $1,0x00000000 23 addu $2, $1, 12
0x00003060 0x3421000c ori $1,$1,0x0000000c
0x00003064 0x00211021 addu $2,$1,$1
0x00003068 0x00400008 jr $2 24 jr $2
0x0000306c 0x8e820000 lw $2,0x00000000($20) 26 lw $2, 0($20)
0x00003070 0xae820004 sw $2,0x00000004($20) 27 sw $2, 4($20)
0x00003074 0x1000ffff beq $0,$0,0xffffffff 29 beq $0, $0, label

```

六、思考题

1. 在本实验中你遇到了哪些不同指令组合产生的冲突？你又是如何解决的？相应的测试样例是什么样的？请有条理的罗列出来。

1) 暂停

下图为需要暂停的指令类型

IF/ID 当前指令			ID/EX(T_{new})			EX/MEM (T_{new})	
指令类型	源寄存器	T_{use}	cal-r 1/rd	cal-l 1/rt	load 2/rt	load 1/rt	
beq	rs/rt	0	暂停	暂停	暂停	暂停	
jr	rs	0	暂停	暂停	暂停	暂停	
cal-r	rs/rt	1			暂停		
cal-l	rs	1			暂停		
load	rs	1			暂停		
store	rs	1			暂停		

冻结 IF/ID：sub 继续被保存。

清除 ID/EX：指令全为 0，等价于插入 NOP。

禁止 PC：防止 PC 继续计数，PC 应保持为 PC+4。

2) 转发

如下图所示，所需要的转发的数据通路。

	D	E	M	W	output
RSD	beq/jr	jal			PC8_E
	beq/jr		R		AO_M
	beq/jr		I		AO_M
	beq/jr		jal		PC8_M
	beq/R/I/ld/st/jr			R	WD_W
	beq/R/I/ld/st/jr			I	WD_W
	beq/R/I/ld/st/jr			ld	WD_W
	beq/R/I/ld/st/jr			jal	WD_W
RTD	beq	jal			PC8_E
	beq		R		AO_M
	beq		I		AO_M
	beq		jal		PC8_M
	beq/R/I/ld/st			R	WD_W
	beq/R/I/ld/st			I	WD_W
	beq/R/I/ld/st			ld	WD_W
	beq/R/I/ld/st			jal	WD_W
RSE		R/I/ld/st	R		AO_M
		R/I/ld/st	I		AO_M
		R/I/ld/st	jal		PC8_M
		R/I/ld/st		R	WD_W
		R/I/ld/st		I	WD_W
		R/I/ld/st		ld	WD_W
		R/I/ld/st		jal	WD_W
RTE		R/st	R		AO_M
		R/st	I		AO_M
		R/st	jal		PC8_M
		R/st		R	WD_W
		R/st		I	WD_W
		R/st		ld	WD_W
		R/st		jal	WD_W
RTM			st	R	WD_W
			st	I	WD_W
			st	ld	WD_W
			st	jal	WD_W

根据这些数据通路添加多路选择器，根据表中不同的选择产生控制信号。
完成数据转发。