

MIPS 流水线处理器设计文档

MIPS 单周期处理器设计文档由顶层模块、数据通路、控制器、冲突模块、测试程序组成。

一、顶层模块 mips.v

序号	信号	位数	方向	功能描述
1	clk	1	I	pc、grf、dm 的时钟信号
2	reset	1	I	当 reset=1 且时钟为上跳沿时，pc 被设置为 0x00003000，gpr 所有寄存器置零，dm 所有数据置零

顶层模块将数据通路模块 datapath.v 与控制器模块 control.v,hazard.v 相连。

二、数据通路 datapath.v

信号端口

序号	信号	位数	方向	描述
1	reset	1	I	复位信号
2	clk	1	I	时钟信号
3	PCOP	2	I	muxPCOP 模块控制信号
4	RegDst	2	I	muxRegDst 模块控制信号
5	ExtOP	1	I	Ext 模块控制信号
6	RegWrite	1	I	寄存器文件写使能
7	RegWData	1	I	muxRegWData 模块控制信号
8	ALUSrc	1	I	muxALUSrc 模块控制信号
9	ALUOP	4	I	选择 ALU 运算类型的信号
10	MemWrite	1	I	dm 写入控制信号
11	Instr_original	32	O	IM 输出指令
12	Instr_D	32	O	D 级指令
13	Instr_E	32	O	E 级指令
14	Instr_W	32	O	W 级指令

datapath.v 将 pc.v, dm.v , im.v , add4.v,npv.alu.v , grf.v , ext.v , muxPCOP.v , muxRegWData.v , muxRegDst.v , muxALUSrc.v 等模块连接在一起。下面将介绍

这些模块。

1、grf.v

信号端口

序号	信号	位数	方向	描述
1	RA1	5	I	读寄存器文件时第一个寄存器的下标
2	RA2	5	I	读寄存器文件时第二个寄存器的下标
3	Waddr	5	I	写入文件的寄存器下标
4	WData	32	I	寄存器文件写入数据
5	clk	1	I	时钟信号
6	reset	1	I	复位信号
7	RegWrite	1	I	寄存器文件写使能
8	RData1	32	O	读寄存器文件时第一个寄存器的输出
9	RData2	32	O	读寄存器文件时第二个寄存器的输出

模块功能

序号	功能名称	功能描述
1	读寄存器	RData1 输出 RA1 所寻址的寄存器的文件数据，RData2 输出 RA2 所寻址的寄存器的文件数据
2	写寄存器	当时钟上升沿到来时，并且 reset=0 且 RegWrite 有效时，WD 被写入 WA 所寻址的寄存器
3	复位	reset=1 且 clk 为上升沿时 gpr 所有寄存器置零

2、alu.v

信号端口

序号	信号	位数	方向	描述
1	ALU_A	32	I	ALU 的第一个操作数
2	ALU_B	32	I	ALU 的第二个操作数
3	ALUOP	4	I	选择 ALU 运算类型的信号
4	ALU_C	32	O	ALU 的运算结果
5	equal	1	O	判断 ALU_A 与 ALU_B 是否相等，若相等输出 1，否则输出 0

模块功能

序号	功能名称	功能描述
1	加法	输出端 $ALU_C=ALU_A+ALU_B$
2	减法	输出端 $ALU_C=ALU_A-ALU_B$
3	或	输出端 $ALU_C=ALU_A ALU_B$
4	判断相等	若 $ALU_A=ALU_B$, equal 信号输出 1, 否则输出 0

3、dm.v

信号端口

序号	信号	位数	方向	描述
1	addr	32	I	写入数据储存器的数据的地址/读取数据储存器的数据的地址
2	din	32	I	写入数据储存器的数据
3	MemWrite	1	I	写入控制信号
4	clk	1	I	时钟信号
5	reset	1	I	复位信号
6	dout	32	O	从数据存储器读取的数据

模块功能

序号	功能名称	功能描述
1	读数据存储器	dout 输出数据存储器在 addr[11:2]地址的数据
2	写数据存储器	当 MemWrite 有效,reset=0,并且时钟为上升沿时, din 被写入数据存储器地址为 addr[11:2]的区域
3	复位	reset=1 并且时钟为上升沿时,DM 所有数据置零

4、ext.v

信号端口

序号	信号	位数	方向	描述
1	ext_in	16	I	位扩展输入信号
2	EXTOP	1	I	信号为 1 时, 进行符号扩展, 信号为 0 时, 进行 0 扩展
3	ext_out	32	O	位扩展输出信号

模块功能

序号	功能名称	功能描述
1	符号扩展	当 EXTOP 信号为 1 时, ext_in 符号扩展后由 ext_out 输出
2	0 扩展	当 EXTOP 信号为 0 时, 在 ext_in 前置 16 个 0 后由 ext_out 输出

5、im.v

信号端口

序号	信号	位数	方向	描述
1	address	32	I	指令地址
2	instr	32	O	指令

模块功能

序号	功能名称	功能描述
1	取指令	取出位于(address-0x00003000)[11:2]处指令

6、pc.v

信号端口

序号	信号	位数	方向	描述
1	PC_in	32	I	下一条指令 PC 值
2	clk	1	I	时钟信号
3	reset	1	I	复位信号
4	PC_out	32	O	当前 PC 值

模块功能

序号	功能名称	功能描述
1	输出指令地址	时钟上升沿且 reset=0 是 PC 输出指令地址
2	复位	时钟上升沿且 reset=1, 将 PC 置为 0x00003000

7、muxPCOP.v

信号端口

序号	信号	位数	方向	描述
1	PCOP	2	I	muxPCOP 控制信号
2	PC4	32	I	PC+4
3	NPC	32	I	NPC 结果
4	newPC	32	O	下条指令地址

模块功能

序号	功能名称	功能描述
1	输出下条指令地址	PCOP==2'b00:2'b00:newPC<=PC4; PCOP==2'b01:2'b01:newPC<=NPC;

8、muxRegDst.v

信号端口

序	信号	位数	方向	描述
---	----	----	----	----

号				
1	RegDst	2	I	muxRegDst 控制信号
2	IR_W	32	I	W 级指令
3	WAddr	5	O	写入文件的寄存器下标

模块功能

序号	功能名称	功能描述
1	输出写入文件的寄存器下标	RegDst=2'b00, WAddr=rd RegDst=2'b01, WAddr=rt

9、muxRegWData.v

信号端口

序号	信号	位数	方向	描述
1	ALUC_W	32	I	W 级 ALU 结果
2	DM_W	32	I	W 级从数据存储器读取的数据
3	RegWData	2	I	muxRegWData 的控制信号
4	WData	32	O	寄存器文件写入数据
5	IR_W	32	I	W 级指令

模块功能

序号	功能名称	功能描述
1	输出寄存器文件写入数据	RegWData=2'b00:WData<=ALUC_W; RegWData=2'b01:WData<=DM_W;

10、muxALUSrc.v

信号端口

序号	信号	位数	方向	描述
1	ALUSrc	1	I	muxALUSrc 的控制信号
2	RD2	32	I	读寄存器文件时第二个寄存器的下标
3	EXT_out	32	I	位扩展输出信号

4	ALU_B	32	O	ALU 的第二个操作数
---	-------	----	---	-------------

模块功能

序号	功能名称	功能描述
1	选择 ALU 的第二个操作数	ALUSrc=0 时: ALU_B=RD2 ALUSrc=1 时: ALU_B=EXT_out

11、add4. v

信号端口

序号	信号	位数	方向	描述
1	PC	32	I	PC
3	PC4	32	O	PC+4

模块功能

序号	功能名称	功能描述
1	PC+4	PC4=PC+4

12、IF_ID. v

信号端口

序号	信号	位数	方向	描述
1	clk	1	I	时钟信号
2	reset	1	I	复位信号
3	IR_IM	32	I	IM 输出指令
4	PC4_ADD 4	32	I	PC+4
5	PC4_D	32	O	PC+4
6	IR_D	32	O	D 级指令

模块功能

序号	功能名称	功能描述
1	D 级流水线寄存器	传递 IR 与数据

13、ID_EX. v

信号端口

序号	信号	位数	方向	描述
1	clk	1	I	时钟信号
2	reset	1	I	复位信号

3	IR_D	32	I	D 级指令
4	RD1	32	I	读寄存器文件时第一个寄存器的输出
5	RD2	32	O	读寄存器文件时二个寄存器的输出
6	EXT_out	32	O	D 级指令
7	PC4_D	32	O	PC+4
8	IR_E	32	O	E 级指令
9	RD1_E	32	O	读寄存器文件时第一个寄存器的输出
10	RD2_E	32	O	读寄存器文件时第二个寄存器的输出
11	EXT_E	32	O	EXT 结果
12	PC4_E	32	O	PC+4

模块功能

序号	功能名称	功能描述
1	E 级流水线寄存器	传递 IR 与数据

14、EX_MEM. v

信号端口

序号	信号	位数	方向	描述
1	clk	1	I	时钟信号
2	reset	1	I	复位信号
3	IR_E	32	I	E 级指令
4	RD2_E	32	I	读寄存器文件时二个寄存器的输出
5	ALU_out	32	I	ALU 结果
6	PC4_E	32	I	PC+4
7	IR_M	32	O	M 级指令
8	ALUC_M	32	O	ALU 结果
9	PC4_M	32	O	PC+4
10	RD2_M	32	O	读寄存器文件时二个寄存器的输出

模块功能

序号	功能名称	功能描述
1	M 级流水线寄存器	传递 IR 与数据

15、MEM_WB. v

信号端口

序号	信号	位数	方向	描述
1	clk	1	I	时钟信号
2	reset	1	I	复位信号

3	IR_M	32	I	M 级指令
4	DM_Data	32	I	从数据存储器读取的数据
5	ALU_M	32	I	ALU 结果
6	PC4_M	32	I	PC+4
7	IR_W	32	O	W 级指令
8	ALUC_W	32	O	ALU 结果
9	PC4_W	32	O	PC+4
10	DM_W	32	O	从数据存储器读取的数据

模块功能

序号	功能名称	功能描述
1	W 级流水线寄存器	传递 IR 与数据

16、MF_RS_E.v

信号端口

序号	信号	位数	方向	描述
1	MF_RS_E_OP	2	I	MF_RS_E 控制信号
2	RD1_E	32	I	读寄存器文件时第二个寄存器的值
3	ALUC_M	32	I	ALU 结果
4	RegWData	32	I	muxRegWData 结果
5	RD1	32	O	转发结果

模块功能

序号	功能名称	功能描述
1	转发	控制信号为 01,10 时转发

17、MF_RT_E.v

信号端口

序号	信号	位数	方向	描述
1	MF_RT_E_OP	2	I	MF_RT_E 控制信号
2	RD2_E	32	I	读寄存器文件时第二个寄存器的值
3	ALUC_M	32	I	ALU 结果
4	RegWData	32	I	muxRegWData 结果
5	RD2	32	O	转发结果

序号	功能名称	功能描述
1	转发	控制信号为 01,10 时转发

三、控制器 control.v

以下为控制信号产生的真值表：

func	100001	100011	无			
opcode	000000	000000	001101	000010	001111	
控制信号	addu	subu	ori	j	lui	nop
PCOP	00	00	00	01	00	0
EXTOP	x	x	0	x	x	x
RegDst[1:0]	00	00	01	x	01	x
RegWrite	1	1	1	0	1	0
RegWData[1:0]	00	00	00	x	10	x
ALUSrc	0	0	1	x	x	x
ALUOP[3:0]	0000	0001	0010	x	x	x
MemWrite	0	0	0	0	0	0

序号	信号	位数	方向	描述
1	OP	6	I	OPcode
2	FUNC	6	I	Function code
3	PCOP	2	O	muxPCOP 控制信号
4	RegDst	2	O	muxRegDst 控制信号
5	ExtOP	1	O	ext 控制信号
6	RegWrite	1	O	gpr 写入控制信号
7	RegWData	2	O	muxRegWData 的控制信号
8	ALUSrc	1	O	muxALUSrc 的控制信号
9	ALUOP	4	O	alu 控制信号
10	MemWrite	1	O	dm 写入控制信号

四、冲突模块

信号端口

序号	信号	位数	方向	描述
1	IR_D	1	I	D 级指令
2	IR_E	1	I	E 级指令
3	IR_M	32	I	M 级指令
4	IR_W	32	I	W 级指令
5	MF_RT_E _OP	32	O	MF_RT_E 控制信号
6	MF_RS_E _OP	32	O	MF_RS_E 控制信号

模块功能

序号	功能名称	功能描述
1	转发	输出 MF_RT_E 控制信号，输出 MF_RS_E 控制信号

五、测试代码

测试期望：

34080042	ori \$8,\$0,0x0042
34090058	ori \$9,\$0,0x0058
01095021	addu \$10,\$8,\$9
00095821	addu \$11,\$0,\$9
014b6023	subu \$12,\$10,\$11
016a6823	subu \$13,\$11,\$10
018b7023	subu \$14,\$12,\$11
3c080042	lui \$8,0x0042
3c090058	lui \$9,0x0058
01095021	addu \$10,\$8,\$9
00095821	addu \$11,\$0,\$9
014b6023	subu \$12,\$10,\$11
016a6823	subu \$13,\$11,\$10
018b7023	subu \$14,\$12,\$11
08000c11	j label
3c080042	lui \$8,0x0042
3c090058	lui \$9,0x0058
3c080042	label:
014b8023	lui \$8,0x0042
016a8823	subu \$16,\$10,\$11
02114021	subu \$17,\$11,\$10
02304821	addu \$8,\$16,\$17
02304821	addu \$9,\$17,\$16
34080042	addu \$9,\$17,\$16
34090058	ori \$8,\$0,0x0042
01285021	ori \$9,\$0,0x0058
01205821	addu \$10,\$9,\$8
016a6023	addu \$11,\$9,\$0
014b6823	subu \$12,\$11,\$10
016c7023	subu \$13,\$10,\$11
3c080042	subu \$14,\$11,\$12
3c090058	lui \$8,0x0042
01095021	lui \$9,0x0058
00095821	addu \$10,\$8,\$9
016a6023	addu \$11,\$0,\$9
014b6823	subu \$12,\$11,\$10
016c7023	subu \$13,\$10,\$11
34000042	subu \$14,\$11,\$12
00118821	ori \$0,\$0,0x0042
	addu \$17,\$0,\$17

\$ 8 <= 00000042
\$ 9 <= 00000058
\$10 <= 0000009a
\$11 <= 00000058
\$12 <= 00000042
\$13 <= fffffffbe
\$14 <= fffffffea
\$ 8 <= 00420000
\$ 9 <= 00580000
\$10 <= 009a0000
\$11 <= 00580000
\$12 <= 00420000
\$13 <= ffbe0000
\$14 <= ffea0000
\$ 8 <= 00420000
\$ 8 <= 00420000
\$16 <= 00420000
\$17 <= ffbe0000
\$ 8 <= 00000000
\$ 9 <= 00000000
\$ 9 <= 00000000
\$ 8 <= 00000042
\$ 9 <= 00000058
\$10 <= 0000009a
\$11 <= 00000058
\$12 <= fffffffbe
\$13 <= 00000042
\$14 <= 0000009a
\$ 8 <= 00420000
\$ 9 <= 00580000
\$10 <= 009a0000
\$11 <= 00580000
\$12 <= ffbe0000
\$13 <= 00420000
\$14 <= 009a0000
\$17 <= ffbe0000

六、思考题

令 $CAL_R = \{addu, subu\}$, $CAL_I = \{lui, ori\}$

1、E 级为 CAL_R , M/W 级为 CAL_R , E 级 RS 等于 M/W 级 RD

优先转发 M 级 ALUC_M, 再转发 W 级 WData

样例: `addu $1, $2, $3`

`addu $2, $1, $3`

或

`addu $1, $2, $3`

无关指令

`addu $2, $1, $3`

2、E 级为 CAL_R , M/W 级为 CAL_R , E 级 RT 等于 M/W 级 RD

优先转发 M 级 ALUC_M, 再转发 W 级 WData

样例: `addu $1, $2, $3`

`addu $2, $3, $1`

或

`addu $1, $2, $3`

无关指令

`addu $2, $3, $1`

3、E 级为 CAL_R , M/W 级为 CAL_I , E 级 RS 等于 M/W 级 RT

优先转发 M 级 ALUC_M, 再转发 W 级 WData

样例: `ori $1, $2, 0x0001`

`addu $2, $1, $3`

或

`ori $1, $2, 0x0001`

无关指令

`addu $2, $1, $3`

4、E 级为 CAL_R , M/W 级为 CAL_I , E 级 RT 等于 M/W 级 RT

优先转发 M 级 ALUC_M, 再转发 W 级 WData

样例: ori \$1,\$2,0x0001

addu \$2,\$3,\$1

或

ori \$1,\$2,0x0001

无关指令

addu \$2,\$3,\$1

5、E 级为 CAL_I, M/W 级为 CAL_R, E 级 RT 等于 M 级 RD

优先转发 M 级 ALUC_M, 再转发 W 级 WData

样例: addu \$1,\$2,\$3

ori \$2,\$1,0x00001

或

addu \$1,\$2,\$3

无关指令

ori \$2,\$1,0x00001

6、E 级为 CAL_I, M/W 级为 CAL_I, E 级 RT 等于 M 级 RT

优先转发 M 级 ALUC_M, 再转发 W 级 WData

样例: lui \$1,0x00002

ori \$2,\$1,0x00001

或

lui \$1,0x00002

无关指令

ori \$2,\$1,0x00001

