# CPU设计文档

## 数据通路设计

### 模块规格datapath.v

#### 端口说明

表 1‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| StallF | I |  |
| StallD | I |  |
| FlushE | I |  |
| Forward\_RS\_F[2:0] | I |  |
| Forward\_RS\_D[2:0] | I |  |
| Forward\_RT\_D[2:0] | I |  |
| Forward\_RS\_E[2:0] | I |  |
| Forward\_RT\_E[2:0] | I |  |
| Forward\_RT\_M[2:0] | I |  |
| Clk | I | 时钟信号 |
| Reset | I | 复位信号 |
| IR@D[31:0] | O |  |

#### 功能定义

表 1‑2 功能定义

|  |  |
| --- | --- |
| 功能 | 描述 |
| grf写入 | 若grf需要写入数据，则输出写入的位置及写入的值。 |
| dm写入 | 若dm需要写入数据，则输出写入的位置及写入的值。 |

### 取指

#### 模块规格F.v

表 2‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| NPC[31:0] | I |  |
| NPC\_Sel[1:0] | I | 下一指令地址控制信号 |
| Branch | I | 是否跳转 |
| Clk | I | 时钟信号 |
| Reset | I | 复位信号 |
| StallF | I | 暂停更新PC |
| StallD | I | 暂停更新D级流水线寄存器 |
| IR@D[31:0] | O | 当前指令 |
| PC4@D[31:0] | O |  |

#### 程序计数器pc.v

表 2‑2 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| NPC[31:0] | I | 下一指令地址 |
| Clk | I | 时钟信号 |
| Reset | I | 复位信号 |
| En | I | 使能信号 |
| PC[31:0] | O | 当前指令地址 |

表 2‑3 功能定义

|  |  |
| --- | --- |
| 功能 | 描述 |
| 存储指令地址 | 输出当前指令地址。时钟上升沿到来，且使能端有效时保存下一指令地址 |
| 复位 | 复位信号有效时，PC同步复位于0x0000\_3000 |

#### 指令存储器im.v

表 2‑4 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Addr[11:2] | I | 指令地址 |
| Instr[31:0] | O | 指令 |

表 2‑5 功能定义

|  |  |
| --- | --- |
| 功能 | 描述 |
| 存储指令 | 初始化时，加载指令码。容量为32bit\*1024 |
| 输出指令 | 输出地址信号对应的指令内容 |

### 译码

#### 模块规格D.v

表 3‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| IR@D[31:0] | I | 当前指令 |
| PC4@D[31:0] | I | 下一指令地址 |
| Clk | I | 时钟信号 |
| Reset | I | 复位信号 |
| FlushE | I | 清空流水级寄存器 |
| Forward\_RS\_D\_Sel[2:0] | I | 转发MUX控制信号 |
| Forward\_RT\_D\_Sel[2:0] | I | 转发MUX控制信号 |
| PC4fromE[31:0] | I |  |
| AO[31:0] | I |  |
| PC4fromM[31:0] | I |  |
| MUX\_RF\_WD\_OUT[31:0] | I |  |
| Branch | O | 是否跳转 |
| NPC\_Sel[1:0] | O |  |
| NPC[31:0] | O | 跳转至地址 |
| MF\_RS\_D\_OUT[31:0] | O | D级读寄存器值 |
| IR@E[31:0] | O |  |
| PC4@E[31:0] | O |  |
| RS@E[31:0] | O |  |
| RT@E[31:0] | O |  |
| EXT@E[31:0] | O |  |
| MUX\_RF\_A3\_OUT[4:0] | I |  |
| We | I |  |

#### 寄存器文件grf.v

表 3‑2 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| A1[4:0] | I | 第一操作数地址 |
| A2[4:0] | I | 第二操作数地址 |
| A3[4:0] | I | 第三操作数地址 |
| Wd[31:0] | I | 待写入数据 |
| We | I | 写使能信号 |
| Clk | I | 时钟信号 |
| Reset | I | 复位信号 |
| Rd1[31:0] | O | 第一操作数 |
| Rd2[31:0] | O | 第二操作数 |

表 3‑3 功能定义

|  |  |
| --- | --- |
| 功能 | 描述 |
| 复位 | 复位信号有效时，全部寄存器同步复位为0 |

#### 扩展单元ext.v

表 3‑4 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| ExtOp[1:0] | I | 扩展功能 |
| Imm[15:0] | I | 待扩展立即数 |
| Ext[31:0] | O | 扩展结果 |

表 3‑5 功能定义

|  |  |  |
| --- | --- | --- |
| 功能 | ExtOp[1:0] | 描述 |
| {16’b0, Imm} | 00 | 无符号扩展 |
| {{16{Imm[15]}}, Imm} | 01 | 有符号扩展 |
| {Imm, 16’b0} | 10 | 扩展至低位 |
| 32’b0 | 11 | 复位 |

#### 指令自增单元npc.v

表 3‑6 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| PC4[31:0] | I |  |
| I26[25:0] | I | 指令的低26位 |
| NPCOp | I | 控制信号 |
| NPC[31:0] | O |  |

表 3‑7 功能定义

|  |  |  |
| --- | --- | --- |
| 功能 | NPCOp | 描述 |
| Branch | 0 | NPC = PC+4+{{14{I26[15]}}, I26[15:0], 2’b00} |
| Jump | 1 | NPC = {PC[31:28], I26[25:0], 2’b00} |

#### 比较单元cmp.v

表 3‑8 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Rd1[31:0] | I | 第一个操作数 |
| Rd2[31:0] | I | 第二个操作数 |
| OpCode[5:0] | I | 指令类型 |
| Branch | O | 是否跳转 |

### 执行

#### 模块规格E.v

表 4‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| IR@E[31:0] | I |  |
| PC4@E[31:0] | I |  |
| RS@E[31:0] | I |  |
| RT@E[31:0] | I |  |
| EXT@E[31:0] | I |  |
| Clk | I | 时钟信号 |
| Reset | I | 复位信号 |
| IR@M[31:0] | O |  |
| PC4@M[31:0] | O |  |
| AO@M[31:0] | O |  |
| RT@M[31:0] | O |  |
| Forward\_RS\_E\_Sel[2:0] | I | 转发MUX控制信号 |
| Forward\_RT\_E\_Sel[2:0] | I | 转发MUX控制信号 |
| AO@M[31:0] | I |  |
| PC4fromM[31:0] | I |  |
| MUX\_RF\_WD\_OUT[31:0] | I |  |

#### 计算单元alu.v

表 4‑2 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| A[31:0] | I | A |
| B[31:0] | I | B |
| ALUOp[2:0] | I | 运算类型 |
| C[31:0] | O | C |
| N | O | (C < 0) |
| Z | O | (C == 0) |
| V | O | 运算溢出 |
| Carry | O | 运算进位 |

表 4‑3 功能定义

|  |  |  |
| --- | --- | --- |
| 功能 | ALUOp[2:0] | 描述 |
| Nop | 000 | 0 |
| Add | 001 | A+B |
| Sub | 010 | A-B |
| And | 011 | A&B |
| Or | 100 | A|B |
| Xor | 101 | A^B |
| Le | 110 | A<B |
| B | 111 | B |

### 访存

#### 模块规格M.v

表 5‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| IR@M[31:0] | I |  |
| PC4@M[31:0] | I |  |
| AO@M[31:0] | I |  |
| RT@M[31:0] | I |  |
| Clk | I | 时钟信号 |
| Reset | I | 复位信号 |
| IR@W[31:0] | O |  |
| PC4@W[31:0] | O |  |
| AO@W[31:0] | O |  |
| DR@W[31:0] | O |  |
| Forward\_RT\_M\_Sel[2:0] | I | 转发MUX控制信号 |
| MUX\_RF\_WD\_OUT[31:0] | I |  |

#### 数据寄存器dm.v

表 5‑2 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Addr[11:2] | I | 数据地址 |
| Din[31:0] | I | 待写入数据 |
| We | I | 写使能信号 |
| Clk | I | 时钟信号 |
| Reset | I | 复位信号 |
| Dout[31:0] | O | 数据 |

表 5‑3 功能定义

|  |  |
| --- | --- |
| 功能 | 描述 |
| 复位 | 复位信号有效时，dm同步复位于0x0000\_0000 |

### 写回

#### 模块规格W.v

表 6‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| IR@W[31:0] | I |  |
| PC4@W[31:0] | I |  |
| AO@W[31:0] | I |  |
| DR@W[31:0] | I |  |
| MUX\_RF\_A3\_OUT[4:0] | O | 写回寄存器编号 |
| MUX\_RF\_WD\_OUT[31:0] | O | 写回数据 |
| RegWrite | O |  |

### 功能多选器mux.v

#### MUX\_PC

表 7‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| add4[31:0] | I |  |
| NPC[31:0] | I |  |
| mf[31:0] | I |  |
| NPC\_Sel[1:0] | I |  |
| Branch | I | 是否跳转 |
| MUX\_PC\_OUT[31:0] | O |  |

表 7‑2 MUX\_PC功能定义

|  |  |  |
| --- | --- | --- |
| 信号 | NPC\_Sel[1:0] | 数据来源 |
| add4 | 2’b00 |  |
| NPC | 2’b01 | NPC jump |
| NPC | 2’b10 | (Branch == 1) ? NPC branch : add4 |
| ra | 2’b11 | MF\_RS\_F |

#### MUX\_ALU\_B

表 7‑3 MUX\_ALU\_B功能定义

|  |  |  |
| --- | --- | --- |
| 信号 | ALUSrc\_Sel | 数据来源 |
| Rd2[31:0] | 1’b0 | MF\_RT\_E |
| Ext[31:0] | 1’b1 | EXT@E |

#### MUX\_RF\_A3

表 7‑4 MUX\_RF\_A3功能定义

|  |  |  |
| --- | --- | --- |
| 信号 | RFWA\_Sel[1:0] | 数据来源 |
| rd[4:0] | 2’b00 | IR@W[rd] |
| rt[4:0] | 2’b01 | IR@W[rt] |
| 31 | 2’b10 |  |

#### MUX\_RF\_WD

表 7‑5 MUX\_RF\_WD功能定义

|  |  |  |
| --- | --- | --- |
| 信号 | RFWD\_Sel[1:0] | 数据来源 |
| C[31:0] | 2’b00 | AO@W |
| Dout[31:0] | 2’b01 | DR@W |
| PC4[31:0] | 2’b10 | PC8@W |

### 转发多选器mf.v

表 8‑1 功能定义

|  |  |  |
| --- | --- | --- |
| 信号 | Forward\_Sel[2:0] | 数据来源 |
| PC4@E[31:0] | 3’b100 | PC4@E + 4 |
| AO[31:0] | 3’b011 | AO@M |
| PC4@M[31:0] | 3’b010 | PC4@M + 4 |
| Wd[31:0] | 3’b001 | MUX\_RF\_WD |
| Rd[31:0] | 3’b000 | RF.RD, RD@E, RD@M |

## 控制器设计

### 控制单元ctrl.v

表 1‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Instr[31:0] | I |  |
| NPC\_Sel@F[1:0] | O | 下一指令地址来源 |
| NPCOp@D | O | 指令跳转方式 |
| ExtOp@D[1:0] | O | 立即数扩展方式 |
| ALUSrc@E | O | ALU第二操作数来源 |
| ALUOp@E[2:0] | O | ALU运算方式 |
| MemWrite@M | O | 数据存储器写使能信号 |
| RegDst@W[1:0] | O | 待回写寄存器地址来源 |
| DataSrc@W[1:0] | O | 待回写数据来源 |
| RegWrite@W | O | 寄存器写使能信号 |

表 1‑2 R型指令功能定义

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Funct | 000000 | 001000 | 001001 | 100001 | 100011 | 101010 |
| OpCode | 000000 | 000000 | 000000 | 000000 | 000000 | 000000 |
|  | nop | jr | jalr | addu | subu | slt |
| NPC\_Sel@F[1:0] | 00 | 11 | 11 | 00 | 00 | 00 |
| NPCOp@D | x | x | x | x | x | x |
| ExtOp@D[1:0] | xx | xx | xx | xx | xx | xx |
| ALUSrc@E | x | 0 | 0 | 0 | 0 | 0 |
| ALUOp@E[2:0] | xxx | 100 | 100 | 001 | 010 | 110 |
| MemWrite@M | 0 | 0 | 0 | 0 | 0 | 0 |
| RegDst@W[1:0] | xx | xx | 10 | 00 | 00 | 00 |
| DataSrc@W[1:0] | xx | xx | 10 | 00 | 00 | 00 |
| RegWrite@W | 0 | 0 | 1 | 1 | 1 | 1 |

表 1‑3 I型指令功能定义

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| OpCode | 000100 | 001001 | 001101 | 001111 | 100011 | 101011 |
|  | beq | addiu | ori | lui | lw | sw |
| NPC\_Sel@F[1:0] | 10 | 00 | 00 | 00 | 00 | 00 |
| NPCOp@D | 0 | x | x | x | x | x |
| ExtOp@D[1:0] | xx | 01 | 00 | 10 | 01 | 01 |
| ALUSrc@E | 0 | 1 | 1 | 1 | 1 | 1 |
| ALUOp@E[2:0] | 010 | 001 | 100 | 111 | 001 | 001 |
| MemWrite@M | 0 | 0 | 0 | 0 | 0 | 1 |
| RegDst@W[1:0] | xx | 01 | 01 | 01 | 01 | xx |
| DataSrc@W[1:0] | xx | 00 | 00 | 00 | 01 | xx |
| RegWrite@W | 0 | 1 | 1 | 1 | 1 | 0 |

表 1‑4 J型指令功能定义

|  |  |  |
| --- | --- | --- |
| OpCode | 000010 | 000011 |
|  | j | jal |
| NPC\_Sel@F[1:0] | 01 | 01 |
| NPCOp@D | 1 | 1 |
| ExtOp@D[1:0] | xx | xx |
| ALUSrc@E | x | x |
| ALUOp@E[2:0] | xxx | xxx |
| MemWrite@M | 0 | 0 |
| RegDst@W[1:0] | xx | 10 |
| DataSrc@W[1:0] | xx | 10 |
| RegWrite@W | 0 | 1 |

### 冲突单元hazard.v

表 2‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Instr[31:0] | I | D级流水线寄存器指令 |
| Clk | I | 时钟信号 |
| Reset | I | 复位信号 |
| StallF | O |  |
| StallD | O |  |
| FlushE | O |  |
| Forward\_RS\_D[2:0] | O |  |
| Forward\_RT\_D[2:0] | O |  |
| Forward\_RS\_E[2:0] | O |  |
| Forward\_RT\_E[2:0] | O |  |
| Forward\_RT\_M[2:0] | O |  |

### AT解码器

表 3‑1 端口说明

|  |  |  |
| --- | --- | --- |
| 信号 | 方向 | 描述 |
| Instr[31:0] | I | D级流水线寄存器指令 |
| A1[4:0] | O | 第一个读寄存器编号，如不需要则为0 |
| A2[4:0] | O | 第二个读寄存器编号，如不需要则为0 |
| A3[4:0] | O | 写寄存器编号，如不需要则为0 |
| ic[3:0] | O | 指令类型 |

表 3‑2 ic定义

|  |  |  |
| --- | --- | --- |
| 指令类型 | ic[3:0] | 指令 |
| nop | 4’b0000 | nop |
| cal\_r | 4’b0001 | addu, subu, slt |
| cal\_i | 4’b0010 | addiu, ori, lui |
| load | 4’b0011 | lw |
| store | 4’b0100 | sw |
| b | 4’b0101 | beq |
| j | 4’b0110 | j |
| jr | 4’b0111 | jr |
| jal | 4’b1000 | jal |
| jalr | 4’b1001 | jalr |

## 测试程序

.text

# init

addiu $t0, $0, 10

addiu $t1, $0, 11

# cal\_r

addu $s0, $t0, $t1

addu $s1, $s0, $t0 # r\_e\_rs

addu $s2, $s0, $s1 # r\_e\_rt, r\_m\_rs

addu $s3, $s0, $s1 # r\_m\_rt, r\_w\_rs

addu $s4, $s3, $s1 # r\_w\_rt

addiu $s0, $t0, 11

subu $s1, $s0, $t1 # i\_e\_rs

subu $s2, $s0, $t1 # i\_m\_rs

subu $s3, $s0, $t1 # i\_w\_rs

addiu $s0, $t1, 20

subu $s1, $t1, $s0 # i\_e\_rt

subu $s2, $t1, $s0 # i\_m\_rt

subu $s3, $t1, $s0 # i\_w\_rt

sw $s0, 0x00000000

sw $s1, 0x00000004

sw $s2, 0x00000008

sw $s3, 0x0000000c

lw $s4, 0x00000000

slt $s5, $s4, $t0 # load\_e\_rs

slt $t6, $s4, $t0 # load\_m\_rs

slt $t7, $s4, $t0 # loadw\_rs

lw $s4, 0x00000004

slt $s5, $t0, $s4 # load\_e\_rt

slt $s6, $t0, $s4 # load\_m\_rt

slt $s7, $t0, $s4 # load\_w\_rt

jal mark1

addu $s0, $t0, $ra # jal\_e\_rt

addu $t2, $s1, $ra

mark1:

addu $s1, $t0, $ra # jal\_m\_rt

addu $s2, $t0, $ra # jal\_w\_rt

jal mark2

addu $s0, $t1, $ra # jal\_e\_rs

addu $t3, $s1, $ra

mark2:

addu $s1, $ra, $t1 # jal\_m\_rs

addu $s2, $ra, $t1 # jal\_w\_rs

jal mark3

addu $s1, $t0, $ra # jalr\_m\_rt

addu $s2, $t0, $ra # jalr\_w\_rt

j mark5

nop

mark3:

jalr $ra

addu $s0, $t0, $ra # jalr\_e\_rt

mark5:

jal mark4

addu $s4, $t0, $ra # jalr\_m\_rs

addu $s5, $t0, $ra # jalr\_w\_rs

mark4:

jalr $ra

addu $s3, $t0, $ra # jalr\_e\_rs

ori $s0, $0, 0

ori $s1, $0, 0

ori $s2, $0, 0

ori $s3, $0, 0

ori $s4, $0, 0

ori $s5, $0, 0

ori $s6, $0, 0

ori $s7, $0, 0

ori $t2, $0, 0

# cal\_i

addu $s0, $t0, $t1

addiu $s1, $s0, 1 # r\_e

addiu $s2, $s0, 2 # r\_m

addiu $s3, $s0, 3 # r\_w

lui $s3, 5

addiu $s4, $s3, 4 # i\_e

addiu $s5, $s3, 5 # i\_m

addiu $s6, $s3, 6 # i\_w

lw $s0, 0x00000000

ori $s1, $s0, 0x00000020 # load\_e

ori $s2, $s0, 0x00000020 # load\_m

ori $s3, $s0, 0x00000020 # load\_w

jal mark6

sw $ra, 0x00000010 # jal\_e

addiu $t2, $ra, 100

mark6:

sw $ra, 0x00000004 # jal\_m

sw $ra, 0x00000008 # jal\_w

jal mark7

nop

addiu $s2, $s0, 2 # jalr\_m

addiu $s3, $s0, 3 # jalr\_w

j mark8

nop

mark7:

jalr $s0, $ra

addiu $s1, $s0, 1

mark8:

ori $s0, $0, 0

ori $s1, $0, 0

ori $s2, $0, 0

ori $s3, $0, 0

ori $s4, $0, 0

ori $s5, $0, 0

ori $s6, $0, 0

ori $s7, $0, 0

ori $t2, $0, 0

# b

addu $s0, $t0, $t1

addu $s1, $t0, $s0

beq $s0, $s1, mark9 # r\_m\_rs, r\_e\_rt

nop

ori $t3, $0, 1

mark9:

ori $s0, $0, 0

ori $s1, $0, 0

addu $s0, $t0, $t1

nop

addu $s1, $t0, $s0

beq $s1, $s0, mark10 # r\_e\_rs, r\_w\_rt

nop

ori $t3, $0, 1

mark10:

ori $s0, $0, 0

ori $s1, $0, 1

addu $s0, $t0, $t1

addu $s1, $t0, $t1

nop

beq $s0, $s1, mark11 # r\_w\_rs, r\_m\_rt

nop

ori $t3, $0, 1

mark11:

ori $s0, $0, 0

ori $s1, $0, 0

ori $s0, $0, 1

ori $s1, $0, 2

beq $s0, $s1, mark12 # i\_m\_rs, i\_e\_rt

nop

ori $t3, $0, 1

mark12:

ori $s0, $0, 3

ori $s1, $0, 3

ori $s0, $0, 4

nop

ori $s1, $0, 5

beq $s1, $s0, mark13 # i\_e\_rs, i\_w\_rt

nop

ori $t3, $0, 1

mark13:

ori $s0, $0, 6

ori $s1, $0, 6

ori $s0, $0, 7

ori $s1, $0, 8

nop

beq $s0, $s1, mark14 # i\_w\_rs, i\_m\_rt

nop

ori $t3, $0, 1

mark14:

ori $s0, $0, 0

ori $s1, $0, 0

lw $s0, 0x00000000

lw $s1, 0x00000004

beq $s0, $s1, mark15 # load\_m\_rs, load\_e\_rt

nop

ori $t3, $0, 1

mark15:

ori $s0, $0, 3

ori $s1, $0, 3

lw $s0, 0x00000000

nop

lw $s1, 0x00000004

beq $s1, $s0, mark16 # load\_e\_rs, load\_w\_rt

nop

ori $t3, $0, 1

mark16:

ori $s0, $0, 6

ori $s1, $0, 6

lw $s0, 0x00000004

lw $s1, 0x00000004

nop

beq $s0, $s1, mark17 # load\_w\_rs, load\_m\_rt

nop

ori $t3, $0, 1

mark17:

ori $s0, $0, 0

ori $s1, $0, 0

ori $s2, $0, 4

ori $s3, $0, 0

ori $s4, $0, 0

ori $s5, $0, 0

ori $s6, $0, 0

ori $s7, $0, 0

ori $t2, $0, 0

ori $t2, 0x0000000c

sw $t2, 0x00000010

sw $s2, 0x0000000c

# load

addiu $s0, $0, 0x00000010

lw $s1, 0($s0) # i\_e

subu $s0, $s0, $s2

nop

lw $s1, 4($s0) # i\_m

subu $s0, $s0, $s2

nop

nop

lw $s1, 8($s0) # i\_w

addu $s0, $s0, $0

lw $s1, 8($s0) # r\_e

addu $s0, $s0, $s2

nop

lw $s1, 4($s0) # r\_m

addu $s0, $s0, $s2

nop

nop

lw $s1, 0($s0) # r\_w

lw $s3, 0x00000010

lw $s3, 0($s3) # load\_e

lw $s4, 0x00000010

nop

lw $s4, 0($s3) # load\_m

lw $s5, 0x00000010

nop

nop

lw $s5, 0($s3) # load\_w

ori $s0, $0, 0

ori $s1, $0, 0

ori $s2, $0, 0

ori $s3, $0, 0

ori $s4, $0, 0

ori $s5, $0, 0

ori $s6, $0, 0

ori $s7, $0, 0

ori $t2, $0, 0

# store

addu $s0, $t0, $t1

sw $s0, 0x00000000 # r\_e

sw $s0, 0x00000004 # r\_m

sw $s0, 0x00000008 # r\_w

addiu $s1, $s0, 3

sw $s1, 0x0000000c # i\_e

sw $s1, 0x00000010 # i\_m

sw $s1, 0x00000014 # i\_w

lw $s2, 0x00000000

sw $s2, 0x00000018 # load\_e

sw $s2, 0x0000001c # load\_m

sw $s2, 0x00000020 # load\_w

jal mark18

sw $ra, 0x00000024 # jal\_e

ori $t0, $0, 1

mark18:

sw $ra, 0x00000028 # jal\_m

sw $ra, 0x0000002c # jal\_w

jal mark19

nop

sw $s7, 0x00000034 # jalr\_m

sw $s7, 0x00000038 # jalr\_w

mark19:

jalr $s7, $ra

sw $s7, 0x00000030 #jalr\_e

预期结果：

9@00003000: $ 8 <= 0000000a

11@00003004: $ 9 <= 0000000b

13@00003008: $16 <= 00000015

15@0000300c: $17 <= 0000001f

17@00003010: $18 <= 00000034

19@00003014: $19 <= 00000034

21@00003018: $20 <= 00000053

23@0000301c: $16 <= 00000015

25@00003020: $17 <= 0000000a

27@00003024: $18 <= 0000000a

29@00003028: $19 <= 0000000a

31@0000302c: $16 <= 0000001f

33@00003030: $17 <= ffffffec

35@00003034: $18 <= ffffffec

37@00003038: $19 <= ffffffec

37@0000303c: \*00000000 <= 0000001f

39@00003040: \*00000004 <= ffffffec

41@00003044: \*00000008 <= ffffffec

43@00003048: \*0000000c <= ffffffec

47@0000304c: $20 <= 0000001f

51@00003050: $21 <= 00000000

53@00003054: $14 <= 00000000

55@00003058: $15 <= 00000000

57@0000305c: $20 <= ffffffec

61@00003060: $21 <= 00000001

63@00003064: $22 <= 00000001

65@00003068: $23 <= 00000001

67@0000306c: $31 <= 00003074

69@00003070: $16 <= 0000307e

71@00003078: $17 <= 0000307e

73@0000307c: $18 <= 0000307e

75@00003080: $31 <= 00003088

77@00003084: $16 <= 00003093

79@0000308c: $17 <= 00003093

81@00003090: $18 <= 00003093

83@00003094: $31 <= 0000309c

85@00003098: $17 <= 000030a6

87@000030a8: $31 <= 000030b0

89@000030ac: $16 <= 000030ba

91@0000309c: $18 <= 000030ba

97@000030b0: $31 <= 000030b8

99@000030b4: $20 <= 000030c2

101@000030bc: $31 <= 000030c4

103@000030c0: $19 <= 000030ce

105@000030b8: $21 <= 000030ce

107@000030bc: $31 <= 000030c4

109@000030c0: $19 <= 000030ce

111@000030c4: $16 <= 00000000

113@000030c8: $17 <= 00000000

115@000030cc: $18 <= 00000000

117@000030d0: $19 <= 00000000

119@000030d4: $20 <= 00000000

121@000030d8: $21 <= 00000000

123@000030dc: $22 <= 00000000

125@000030e0: $23 <= 00000000

127@000030e4: $10 <= 00000000

129@000030e8: $16 <= 00000015

131@000030ec: $17 <= 00000016

133@000030f0: $18 <= 00000017

135@000030f4: $19 <= 00000018

137@000030f8: $19 <= 00050000

139@000030fc: $20 <= 00050004

141@00003100: $21 <= 00050005

143@00003104: $22 <= 00050006

145@00003108: $16 <= 0000001f

149@0000310c: $17 <= 0000003f

151@00003110: $18 <= 0000003f

153@00003114: $19 <= 0000003f

155@00003118: $31 <= 00003120

155@0000311c: \*00000010 <= 00003120

157@00003124: \*00000004 <= 00003120

159@00003128: \*00000008 <= 00003120

163@0000312c: $31 <= 00003134

167@00003144: $31 <= 0000314c

169@00003148: $17 <= 0000314d

171@00003134: $18 <= 0000314e

173@00003138: $19 <= 0000314f

179@0000314c: $16 <= 00000000

181@00003150: $17 <= 00000000

183@00003154: $18 <= 00000000

185@00003158: $19 <= 00000000

187@0000315c: $20 <= 00000000

189@00003160: $21 <= 00000000

191@00003164: $22 <= 00000000

193@00003168: $23 <= 00000000

195@0000316c: $10 <= 00000000

197@00003170: $16 <= 00000015

199@00003174: $17 <= 0000001f

207@00003180: $11 <= 00000001

209@00003184: $16 <= 00000000

211@00003188: $17 <= 00000000

213@0000318c: $16 <= 00000015

217@00003194: $17 <= 0000001f

225@000031a0: $11 <= 00000001

227@000031a4: $16 <= 00000000

229@000031a8: $17 <= 00000001

231@000031ac: $16 <= 00000015

233@000031b0: $17 <= 00000015

241@000031c4: $16 <= 00000000

243@000031c8: $17 <= 00000000

245@000031cc: $16 <= 00000001

247@000031d0: $17 <= 00000002

255@000031dc: $11 <= 00000001

257@000031e0: $16 <= 00000003

259@000031e4: $17 <= 00000003

261@000031e8: $16 <= 00000004

265@000031f0: $17 <= 00000005

273@000031fc: $11 <= 00000001

275@00003200: $16 <= 00000006

277@00003204: $17 <= 00000006

279@00003208: $16 <= 00000007

281@0000320c: $17 <= 00000008

289@0000321c: $11 <= 00000001

291@00003220: $16 <= 00000000

293@00003224: $17 <= 00000000

295@00003228: $16 <= 0000001f

297@0000322c: $17 <= 00003120

307@00003238: $11 <= 00000001

309@0000323c: $16 <= 00000003

311@00003240: $17 <= 00000003

313@00003244: $16 <= 0000001f

317@0000324c: $17 <= 00003120

327@00003258: $11 <= 00000001

329@0000325c: $16 <= 00000006

331@00003260: $17 <= 00000006

333@00003264: $16 <= 00003120

335@00003268: $17 <= 00003120

345@0000327c: $16 <= 00000000

347@00003280: $17 <= 00000000

349@00003284: $18 <= 00000004

351@00003288: $19 <= 00000000

353@0000328c: $20 <= 00000000

355@00003290: $21 <= 00000000

357@00003294: $22 <= 00000000

359@00003298: $23 <= 00000000

361@0000329c: $10 <= 00000000

363@000032a0: $10 <= 0000000c

363@000032a4: \*00000010 <= 0000000c

365@000032a8: \*0000000c <= 00000004

369@000032ac: $16 <= 00000010

371@000032b0: $17 <= 0000000c

373@000032b4: $16 <= 0000000c

377@000032bc: $17 <= 0000000c

379@000032c0: $16 <= 00000008

385@000032cc: $17 <= 0000000c

387@000032d0: $16 <= 00000008

389@000032d4: $17 <= 0000000c

391@000032d8: $16 <= 0000000c

395@000032e0: $17 <= 0000000c

397@000032e4: $16 <= 00000010

403@000032f0: $17 <= 0000000c

405@000032f4: $19 <= 0000000c

409@000032f8: $19 <= 00000004

411@000032fc: $20 <= 0000000c

415@00003304: $20 <= 00003120

417@00003308: $21 <= 0000000c

423@00003314: $21 <= 00003120

425@00003318: $16 <= 00000000

427@0000331c: $17 <= 00000000

429@00003320: $18 <= 00000000

431@00003324: $19 <= 00000000

433@00003328: $20 <= 00000000

435@0000332c: $21 <= 00000000

437@00003330: $22 <= 00000000

439@00003334: $23 <= 00000000

441@00003338: $10 <= 00000000

443@0000333c: $16 <= 00000015

443@00003340: \*00000000 <= 00000015

445@00003344: \*00000004 <= 00000015

447@00003348: \*00000008 <= 00000015

451@0000334c: $17 <= 00000018

451@00003350: \*0000000c <= 00000018

453@00003354: \*00000010 <= 00000018

455@00003358: \*00000014 <= 00000018

459@0000335c: $18 <= 00000015

459@00003360: \*00000018 <= 00000015

461@00003364: \*0000001c <= 00000015

463@00003368: \*00000020 <= 00000015

467@0000336c: $31 <= 00003374

467@00003370: \*00000024 <= 00003374

469@00003378: \*00000028 <= 00003374

471@0000337c: \*0000002c <= 00003374

475@00003380: $31 <= 00003388

479@00003390: $31 <= 00003398

479@00003394: \*00000030 <= 00003398

481@00003388: \*00000034 <= 00003398

483@0000338c: \*00000038 <= 00003398

487@00003390: $31 <= 00003398

487@00003394: \*00000030 <= 00003398

# 思考题

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试类型 | | | cal\_r | cal\_i | b | load | store | jr | jalr |
| 前序类型 | 冲突阶段 | 冲突寄存器 | 测试序列 | 测试序列 | 测试序列 | 测试序列 | 测试序列 | 测试序列 | 测试序列 |
| cal\_r | E | rs |  |  |  |  |  |  |  |
| cal\_r | E | rt |  |  |  |  |  |  |  |
| cal\_r | M | rs |  |  |  |  |  |  |  |
| cal\_r | M | rt |  |  |  |  |  |  |  |
| cal\_r | W | rs |  |  |  |  |  |  |  |
| cal\_r | W | rt |  |  |  |  |  |  |  |
| cal\_i | E | rs |  |  |  |  |  |  |  |
| cal\_i | E | rt |  |  |  |  |  |  |  |
| cal\_i | M | rs |  |  |  |  |  |  |  |
| cal\_i | M | rt |  |  |  |  |  |  |  |
| cal\_i | W | rs |  |  |  |  |  |  |  |
| cal\_i | W | rt |  |  |  |  |  |  |  |
| load | E | rs |  |  |  |  |  |  |  |
| load | E | rt |  |  |  |  |  |  |  |
| load | M | rs |  |  |  |  |  |  |  |
| load | M | rt |  |  |  |  |  |  |  |
| load | W | rs |  |  |  |  |  |  |  |
| load | W | rt |  |  |  |  |  |  |  |
| jal | E | rs |  |  |  |  |  |  |  |
| jal | E | rt |  |  |  |  |  |  |  |
| jal | M | rs |  |  |  |  |  |  |  |
| jal | M | rt |  |  |  |  |  |  |  |
| jal | W | rs |  |  |  |  |  |  |  |
| jal | W | rt |  |  |  |  |  |  |  |
| jalr | E | rs |  |  |  |  |  |  |  |
| jalr | E | rt |  |  |  |  |  |  |  |
| jalr | M | rs |  |  |  |  |  |  |  |
| jalr | M | rt |  |  |  |  |  |  |  |
| jalr | W | rs |  |  |  |  |  |  |  |

除了部分不需要读第二个寄存器的指令以外，上表中每格基本都代表一种冲突。比如第一行第一格表示cal\_r指令在D级时，若E级有cal\_r指令，且后者rd值与前者rs值相等则会引发冲突。对于每种冲突类型，笔者尽量设计一种或几种程序进行测试，但限于时间，有些情况还未被测试全面。