

# 一、模块

## 取址阶段(F级)

1. PC

| Port name | Direction | Type   | Description |
|-----------|-----------|--------|-------------|
| clk       | input     |        |             |
| reset     | input     |        |             |
| In        | input     | [31:0] | 下一条指令地址     |
| stall     | input     |        | 暂停信号        |
| Out       | output    | [31:0] | 当前指令地址      |

2. MUX\_PC

| Port name | Direction | Type   | Description       |
|-----------|-----------|--------|-------------------|
| PCPlus4   | input     | [31:0] | PC+4              |
| PCfromNPC | input     | [31:0] | NPC的地址            |
| branch    | input     | [2:0]  | 0：不跳转，PC+4 其他：NPC |
| PCAddr    | output    | [31:0] | 下一指令地址            |

## 流水线寄存器F

| Port name | Direction | Type   | Description |
|-----------|-----------|--------|-------------|
| clk       | input     |        |             |
| reset     | input     |        |             |
| stall     | input     |        | 暂停型号        |
| F_Ins     | input     | [31:0] |             |
| F_PCPlus4 | input     | [31:0] | PC地址加4      |
| F_PCAddr  | input     | [31:0] | PC地址        |
| D_Ins     | output    | [31:0] |             |
| D_PCPlus4 | output    | [31:0] | PC地址加4      |
| D_PCAddr  | output    | [31:0] | PC地址        |

# 译码阶段(D级)

## 1. D\_Control

| Port name | Direction | Type   | Description                              |
|-----------|-----------|--------|--|
| Ins       | input     | [31:0] |  |
| Sign      | output    | [2:0]  | 0: 0扩展 1: 符号扩展 2: 加载到高位                  |
| Branch    | output    | [2:0]  | 0: 不跳转 1: beq 2: jal 3: jr               |
| Rs_T_use  | output    | [2:0]  | 从D级开始, 多少周期后使用Rs(若不用则为3'b111)            |
| Rt_T_use  | output    | [2:0]  | 从D级开始, 多少周期后使用Rt(若不用则为3'b111)            |
| T_new     | output    | [2:0]  | 从D级开始多少周期后更新寄存器(更新的值进入流水线寄存器), 若不更新, 则为0 |

## 2. EXT

| Port name | Direction | Type   | Description |
|-----------|-----------|--------|-------------|
| D_Imme16  | input     | [15:0] |             |
| D_Sign    | input     | [2:0]  |             |
| D_Imme32  | output    | [31:0] |             |

## 3. MUX\_DA

branch时Rs的冲突

| Port name | Direction | Type | Description |
|-----------|-----------|------|-------------|
|-----------|-----------|------|-------------|

| Port name | Direction | Type   | Description  |
|-----------|-----------|--------|--|
| CDRs      | input     | [2:0]  | 1: 与当前处于M阶段的指令写回的寄存器冲突<br>2: 与当前处于W阶段的指令写回的寄存器冲突<br>0:不需转发 |
| D_RsD     | input     | [31:0] | 从寄存器读出的数   |
| M_RegDst  | input     | [2:0]  | M级写哪个寄存器 2: 写31号（用于判断转发值）                                  |
| M_PCPlus8 | input     | [31:0] | M阶段的PC地址+8   |
| M_ALUAns  | input     | [31:0] | M级ALU的结果   |
| W_RegDst  | input     | [2:0]  | W级写哪个寄存器 2: 写31号（用于判断转发值）                                  |
| W_PCPlus8 | input     | [31:0] | W阶段的PC地址+8   |
| W_MemData | input     | [31:0] | W级的准备写回的数  |
| B_Rs      | output    | [31:0] | 用于分支判断的数   |

#### 4. MUX\_DB

branch时Rt的冲突

| Port name | Direction | Type   | Description  |
|-----------|-----------|--------|--|
| CDRt      | input     | [2:0]  | 1: 与当前处于M阶段的指令写回的寄存器冲突<br>2: 与当前处于W阶段的指令写回的寄存器冲突<br>0:不需转发 |
| D_RtD     | input     | [31:0] | 从寄存器读出的数   |
| M_RegDst  | input     | [2:0]  | M级写哪个寄存器 2: 写31号（用于判断转发值）                                  |
| M_PCPlus8 | input     | [31:0] | M阶段的PC地址+8   |
| M_ALUAns  | input     | [31:0] | M级ALU的结果   |
| W_RegDst  | input     | [2:0]  | W级写哪个寄存器 2: 写31号（用于判断转发值）                                  |
| W_PCPlus8 | input     | [31:0] | W阶段的PC地址+8   |
| W_MemData | input     | [31:0] | W级的准备写回的数  |
| B_Rt      | output    | [31:0] | 用于分支判断的数   |

#### 5. NPC

| Port name | Direction | Type | Description |
|-----------|-----------|------|-------------|
|-----------|-----------|------|-------------|

| Port name | Direction | Type   | Description                |
|-----------|-----------|--------|----------------------------|
| Addr      | input     | [31:0] | 当前地址指令                     |
| Branch    | input     | [2:0]  | 0: 不跳转 1: beq 2: jal 3: jr |
| D_Imme    | input     | [31:0] |                            |
| equal     | input     |        |                            |
| D_Ins     | input     | [31:0] |                            |
| JumpAddr  | input     | [25:0] |                            |
| D_RaData  | input     | [31:0] | 31号寄存器的值                   |
| NAddr     | output    | [31:0] | 下一指令地址                     |

6. CMP

| Port name | Direction | Type   | Description |
|-----------|-----------|--------|-------------|
| B_Rs      | input     | [31:0] |             |
| B_Rt      | input     | [31:0] |             |
| equal     | output    |        | Rs和Rt是否相等   |

## 流水线寄存器D

| Port name | Direction | Type   | Description       |
|-----------|-----------|--------|-------------------|
| clk       | input     |        |                   |
| reset     | input     |        |                   |
| stall     | input     |        | 暂停                |
| D_Ins     | input     | [31:0] | D级指令              |
| D_Imme    | input     | [31:0] | 立即数, E阶段使用 (以及扩展) |
| D_T_new   | input     | [2:0]  | 从D级开始多少周期更新寄存器    |
| D_A       | input     | [31:0] | 寄存器第一个数           |
| D_B       | input     | [31:0] | 寄存器第二个数           |
| D_PCAddr  | input     | [31:0] | D级指令地址            |
| D_Rs      | input     | [4:0]  | 寄存器编号             |
| D_Rt      | input     | [4:0]  | 寄存器编号             |
| D_Rd      | input     | [4:0]  | 目标寄存器编号           |
| E_Ins     | output    | [31:0] | E级指令              |
| E_Imme    | output    | [31:0] | E级立即数             |

| Port name | Direction | Type   | Description      |
|-----------|-----------|--------|------------------|
| E_T_new   | output    | [2:0]  | max{D_T_new-1,0} |
| E_A       | output    | [31:0] |                  |
| E_B       | output    | [31:0] |                  |
| E_PCAAddr | output    | [31:0] | E级指令地址           |
| E_Rs      | output    | [4:0]  | 寄存器编号            |
| E_Rt      | output    | [4:0]  | 寄存器编号            |
| E_Rd      | output    | [4:0]  | 目标寄存器编号          |

## 执行阶段(E级)

### 1. E\_Control

| Port name | Direction | Type   | Description               |
|-----------|-----------|--------|---------------------------|
| Ins       | input     | [31:0] | 当前指令                      |
| RegWrite  | output    |        |                           |
| ALUOp     | output    | [3:0]  |                           |
| ALUSrc    | output    |        | 进入ALU的数来自寄存器还是立即数         |
| RegDst    | output    | [2:0]  | 写入的寄存器 0: rt 1: rd 2: 31号 |

### 2. ALU

| Port name | Direction | Type   | Description      |
|-----------|-----------|--------|------------------|
| E_ALUA    | input     | [31:0] |                  |
| E_ALUB    | input     | [31:0] |                  |
| E_ALUOp   | input     | [3:0]  | 执行阶段使用，判断ALU运算种类 |
| E_ALUAns  | output    | [31:0] |                  |

### 3. MULT\_DIV

| Port name | Direction | Type | Description |
|-----------|-----------|------|-------------|
|-----------|-----------|------|-------------|

| Port name | Direction | Type   | Description |
|-----------|-----------|--------|-------------|
| clk       | input     |        |             |
| reset     | input     |        |             |
| Ins       | input     | [31:0] |             |
| ALUA      | input     | [31:0] |             |
| ALUB      | input     | [31:0] |             |
| start     | input     |        |             |
| HI        | output    | [31:0] |             |
| LO        | output    | [31:0] |             |
| busy      | output    |        |             |

#### 4. MUX\_EA

| Port name | Direction | Type   | Description  |
|-----------|-----------|--------|--|
| CEA       | input     | [2:0]  | 1: 与当前处于M阶段的指令写回的寄存器冲突<br>2: 与当前处于W阶段的指令写回的寄存器冲突<br>0:不需转发 |
| E_A       | input     | [31:0] | 寄存器Rs的值  |
| M_RegDst  | input     | [2:0]  | M级的目标寄存器   |
| M_PCPlus8 | input     | [31:0] | M级的指令地址+8  |
| M_ALUAns  | input     | [31:0] | M级的ALU的结果  |
| W_RegDst  | input     | [2:0]  | W级的目标寄存器   |
| W_PCPlus8 | input     | [31:0] | W级的指令地址+8  |
| W_MemData | input     | [31:0] | W级的准备写回的数  |
| E_ALUA    | output    | [31:0] | 进入ALU的数  |

#### 5. MUX\_EB

| Port name | Direction | Type | Description |
|-----------|-----------|------|-------------|
|-----------|-----------|------|-------------|

| Port name | Direction | Type   | Description  |
|-----------|-----------|--------|--|
| CEB       | input     | [2:0]  | 1：与当前处于M阶段的指令写回的寄存器冲突<br>2：与当前处于W阶段的指令写回的寄存器冲突<br>0:不需转发 |
| E_B       | input     | [31:0] | 寄存器Rs的值  |
| M_RegDst  | input     | [2:0]  | M级的目标寄存器   |
| M_PCPlus8 | input     | [31:0] | M级的指令地址+8  |
| M_ALUAns  | input     | [31:0] | M级的ALU的结果  |
| W_RegDst  | input     | [2:0]  | W级的目标寄存器   |
| W_PCPlus8 | input     | [31:0] | W级的指令地址+8  |
| W_MemData | input     | [31:0] | W级的准备写回的数  |
| E_NextB   | output    | [31:0] | 进入下一个多路选择器，选择寄存器值还是立即数的选择器的数，或者写入内存                      |

#### 6. MUX\_ElmeReg

| Port name | Direction | Type   | Description               |
|-----------|-----------|--------|---------------------------|
| E_ALUSrc  | input     |        | 0：来自上一个多路选择器，MUX_EB 1：立即数 |
| E_InB     | input     | [31:0] | 来自上一个多路选择器，MUX_EB         |
| E_Imme    | input     | [31:0] | 立即数                       |
| E_ALUB    | output    | [31:0] | 进入ALU的数                   |
| HI        | input     | [31:0] |                           |
| LO        | input     | [31:0] |                           |

#### 7. MUX\_Target

| Port name   | Direction | Type   | Description        |
|-------------|-----------|--------|--------------------|
| E_RegDst    | input     | [2:0]  | 0：写Rt 1：写Rd 2：写31号 |
| E_Rt        | input     | [31:0] |                    |
| E_Rd        | input     | [31:0] |                    |
| E_TargetReg | output    | [31:0] | 目标寄存器              |

## 流水线寄存器E

| Port name   | Direction | Type   | Description      |
|-------------|-----------|--------|------------------|
| clk         | input     |        |                  |
| reset       | input     |        |                  |
| clr         | input     |        |                  |
| E_Ins       | input     | [31:0] | 指令               |
| E_TargetReg | input     | [4:0]  | 目标寄存器            |
| E_T_new     | input     | [2:0]  |                  |
| E_ALUAns    | input     | [31:0] | ALU的运算结果         |
| E_NextB     | input     | [31:0] | 转发判断后的结果，存入内存的数值 |
| E_PCAddr    | input     | [31:0] | 指令地址             |
| M_Ins       | output    | [31:0] |                  |
| M_TargetReg | output    | [4:0]  | 目标寄存器            |
| M_T_new     | output    | [2:0]  |                  |
| M_ALUAns    | output    | [31:0] | ALU运算结果          |
| M_WriteData | output    | [31:0] | 对应E_NextB        |
| M_PCAddr    | output    | [31:0] | 指令地址             |

## 存储阶段(M级)

### 1. M\_Control

| Port name  | Direction | Type   | Description |
|------------|-----------|--------|-------------|
| Ins        | input     | [31:0] |             |
| RegWrite   | output    |        | 是否写寄存器      |
| MemRead    | output    |        | 是否读寄存器      |
| MemWrite   | output    |        | 是否写内存       |
| MemWriteOp | output    | [2:0]  | 0: sw       |
| MemReadOp  | output    | [2:0]  | 0: lw       |

### 2. BE

| Port name | Direction | Type | Description |
|-----------|-----------|------|-------------|
|-----------|-----------|------|-------------|



| Port name    | Direction | Type   | Description    |
|--------------|-----------|--------|----------------|
| M_MemWrite   | input     | [31:0] | 写入的字，（字节对应的字）  |
| MemRead      | input     | [31:0] |                |
| M_ALUAns     | input     | [31:0] | 地址             |
| M_DataByteen | output    | [3:0]  | 字节使能信号         |
| M_LT         | output    | [31:0] | 读出的最低为位为所需字节的字 |
| M_MemSData   | output    | [31:0] | 真正写入的字         |

### 3. MUX\_DMIN

| Port name   | Direction | Type   | Description      |
|-------------|-----------|--------|------------------|
| CMI         | input     | [2:0]  | 0: 不需要转发 1: 转发结果 |
| W_RegDst    | input     | [2:0]  |                  |
| M_Write     | input     | [31:0] | 不转发写入内存的数        |
| W_PCPlus8   | input     | [31:0] | 指令地址+8           |
| W_Write     | input     | [31:0] | W级写回的数           |
| M_WriteData | output    | [31:0] | 真正写入内存的数         |

## 流水线寄存器M

| Port name   | Direction | Type   | Description |
|-------------|-----------|--------|-------------|
| clk         | input     |        |             |
| reset       | input     |        |             |
| M_TargetReg | input     | [4:0]  | 写回的寄存器地址    |
| M_T_new     | input     | [2:0]  |             |
| M_ReadData  | input     | [31:0] | 内存读出的数      |
| M_WriteData | input     | [31:0] | ALU的结果      |
| M_Ins       | input     | [31:0] | 指令          |
| M_PCAddr    | input     | [31:0] | 指令地址        |
| W_ReadData  | output    | [31:0] | 内存读出的数字     |
| W_ALUData   | output    | [31:0] | ALU算出的数字    |
| W_TargetReg | output    | [4:0]  | 目标寄存器       |
| W_T_new     | output    | [2:0]  |             |
| W_Ins       | output    | [31:0] | 指令          |

| Port name | Direction | Type   | Description |
|-----------|-----------|--------|-------------|
| W_PCAddr  | output    | [31:0] | 指令地址        |
|           |           |        |             |

## 写回阶段(W级)

### 1. W\_Control

| Port name | Direction | Type   | Description   |
|-----------|-----------|--------|---------------|
| Ins       | input     | [31:0] | 指令            |
| RegWrite  | output    |        | 是否写寄存器        |
| MemToReg  | output    | [2:0]  | 写回的数来自内存还是ALU |
| RegDst    | output    | [2:0]  | 目标寄存器         |

### 2. MUX\_BackData

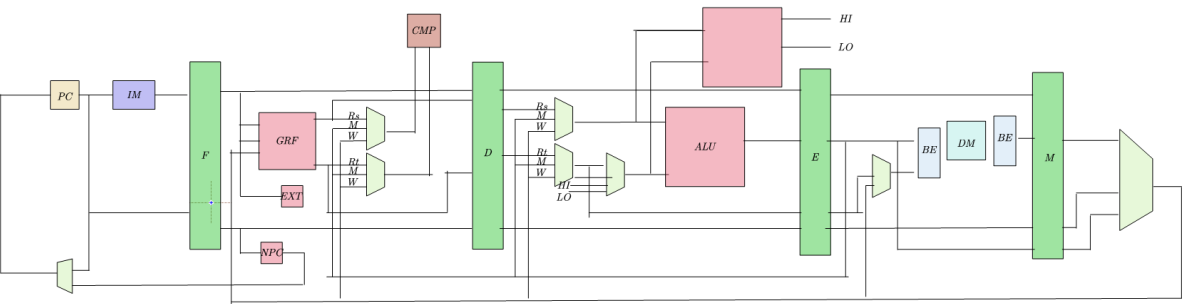
| Port name    | Direction | Type   | Description          |
|--------------|-----------|--------|----------------------|
| W_RegToWrite | input     | [2:0]  | 0: ALU 1: 内存 2: PC+8 |
| W_ReadData   | input     | [31:0] | 内存读出的数               |
| W_ALUData    | input     | [31:0] | ALU的结果               |
| W_PCPlus8    | input     | [31:0] | 地址+8                 |
| W_BackData   | output    | [31:0] | 写回的数                 |

## 冲突处理

| Port name   | Direction | Type  | Description    |
|-------------|-----------|-------|----------------|
| D_Rs_T_use  | input     | [2:0] | D级指令过多少个周期使用Rs |
| D_Rt_T_use  | input     | [2:0] | D级指令过多少个周期使用Rt |
| D_Rs        | input     | [4:0] | D级Rs           |
| D_Rt        | input     | [4:0] | D级Rt           |
| E_Rs        | input     | [4:0] | E级Rs           |
| E_Rt        | input     | [4:0] | E级Rt           |
| M_Rt        | input     | [4:0] | M级Rt           |
| E_TargetReg | input     | [4:0] | 目标寄存器          |
| M_TargetReg | input     | [4:0] | 目标寄存器          |
| W_TargetReg | input     | [4:0] | 目标寄存器          |
| E_T_new     | input     | [2:0] | E级指令过多少周期更新寄存器 |
| M_T_new     | input     | [2:0] | M级指令过多少周期更新寄存器 |

| Port name  | Direction | Type  | Description    |
|------------|-----------|-------|----------------|
| W_t_new    | input     | [2:0] | W级指令过多少周期更新寄存器 |
| E_RegWrite | input     |       | 是否写寄存器         |
| M_RegWrite | input     |       | 是否写寄存器         |
| W_RegWrite | input     |       | 是否写寄存器         |
| CDRs       | output    | [2:0] | D级时分支转发判断      |
| CDRt       | output    | [2:0] | D级时分支转发判断      |
| CEA        | output    | [2:0] | E级时分支转发判断      |
| CEB        | output    | [2:0] | E级时分支转发判断      |
| CMI        | output    | [2:0] | M级时分支判断        |
| Stall      | output    |       | 暂停信号           |
| busy       | input     |       |                |
| start      | input     |       |                |

## 二、电路图



## 三、思考题

### 1. 为什么需要有单独的乘除法部件而不是整合进 ALU？为何需要有独立的 HI、LO 寄存器？

因为乘法需要暂停5个周期，除法需要10个周期，需要的时间比较长。如果乘法后面没有mthi或mtlo指令，则没有必要将流水线阻塞。因为其余指令需要用的是ALU，可以继续流水后面的指令，在乘除法运算的同时继续运行后面的指令，从而获得更高的效率。

### 2.真实的流水线 CPU 是如何使用实现乘除法的？请查阅相关资料进行简单说明。

乘法：通过多次移位和加法来完成。

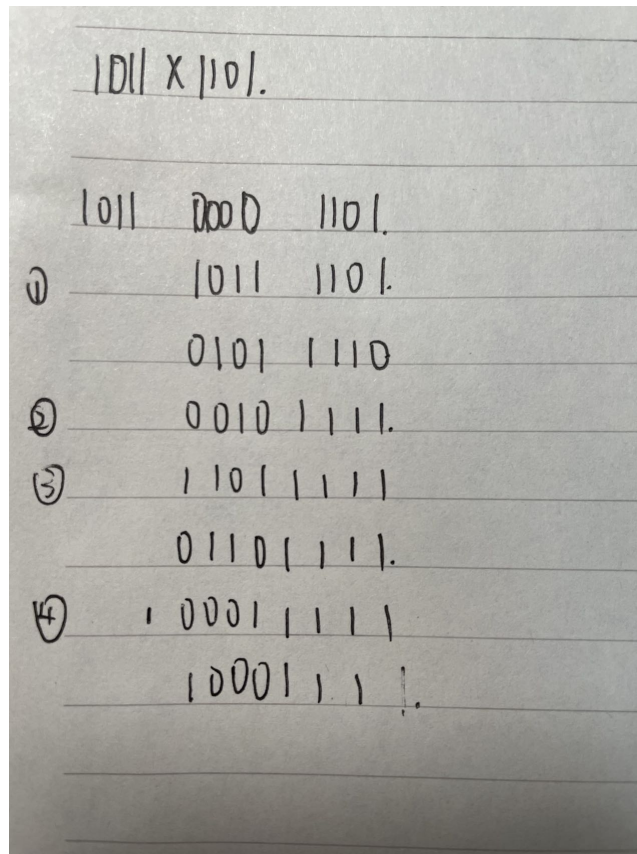
首先CPU会初始化三个通用寄存器用来存放被乘数，乘数，部分积的二进制数，部分积寄存器初始化为0！

然后在判断乘数寄存器的低位是低电平还是高电平（0/1）！

如果为0则将乘数寄存器右移一位，同时将部分积寄存器也右移一位，在位移时遵循计算机位移规则，乘数寄存器低位溢出的一位丢弃，部分积寄存器低位溢出的一位填充到乘数寄存器的高位，同时部分积寄存器高位补0！

如果为1则将部分积寄存器加上被乘数寄存器，在进移位操作。

当所有乘数位处理完成后部分积寄存器做高位乘数寄存器做低位就是最终乘法结果！



除法：通过多次移位和减法来完成。

首先CPU会初始化三个寄存器,用来存放被除数，除数，部分商。余数放到被除数的有效高位上！

CPU做除法时和做乘法时是相反的，乘法是右移，除法是左移，乘法做的是加法，除法做的是减法。

例：1010/10

首先把被除数寄存器左移2位与除数寄存器对齐：

$1010 \ll 2 = 10100$

在用对齐后的被除数与除数做比较：

$10100 - 10 = 00$  刚好等于没有溢出所以上商1，商为0001，在将相减的结果(00)放到被除数有效的高位此时被除数有效的最高位为10100，此时被除数的bit位为00100，

在左移1位：

$00100 \ll 1 = 00100$

在用位移后的被除数与除数做比较：

$00100 - 10 = 11$  符号位为1负数也就是说被除数小于除数所以上商0，商为0010

在左移0位

$0010 \ll 0 = 0010$

在用位移后的被除数与除数做比较：

$0010 - 10 = 00$  两个符号位都为0相等所以上商1，商为101，在将其放到商的有效高位此时被除数的有效bit位为最低的两位(00)10

所有的被除数处理完后商做结果被除数做余数，商为101，余数为00，转换成十进制就是5余0！

### 3.请结合自己的实现分析，你是如何处理 Busy 信号带来的周期阻塞的？

---

考虑的比较简单，如果指令是乘除相关的指令，用到乘除法器或者是读hi和lo，且busy是1或start是1，则阻塞，直至busy复位为0。

### 4.请问采用字节使能信号的方式处理写指令有什么好处？ (提示：从清晰性、统一性等角度考虑)

---

这样对于lb, lh, sb, sh可以统一字节使能信号。把一个字分为4个字节，需要哪个字节，就把字节是信号对应的位置高，哪个地址对应哪个字节或哪几个字节显得更为清晰。

### 5.请思考，我们在按字节读和按字节写时，实际从 DM 获得的数据和向 DM 写入的数据是否是一字节？在什么情况下我们按字节读和按字节写的效率会高于按字读和按字写呢？

---

不是，向DM写入的数据是先读出DM对应位置的字，通过字节控制信号判断，改变相应的字节，最后再存入DM。

读出时，一次性读出的是1整个字，让后再根据字节控制信号去选取需要的字节，扩展再存入寄存器

指令集大量出现的lb指令不读连续的字节时，按字节读和按字节写效率更高

### 6.为了对抗复杂性你采取了哪些抽象和规范手段？这些手段在译码和处理数据冲突的时候有什么样的特点与帮助？

---

分布式译码。

让流水线寄存器流水的数据大大减少，降低连线失误的概率

A-T法

只需要给每一条指令一个tuse和tnew的值，即可用相同的方法去判断阻塞转发，降低了工作量，增强了扩展性

### 7.在本实验中你遇到了哪些不同指令类型组合产生的冲突？你又是如何解决的？相应的测试样例是什么样的？

---

我没有进行指令集的分类，只是给每一条指令一组tuse和tnew的值，然后根据tuse和tnew的值进行判断

通过tuse与tnew的比较

需要阻塞的一共有四种情况：

E\_T\_new = 2, T\_use = 0

lw

beq

E\_T\_new = 1, T\_use = 0

add

beq

E\_T\_new = 2, T\_use = 1

lw

add

M\_T\_new = 1, T\_use = 0

lw

nop

beq

其余情况均为  $T_{\text{new}} < T_{\text{use}}$  的情况，可以靠转发来解决

add

add

add

sw

等等

**8.如果你是手动构造的样例，请说明构造策略，说明你的测试程序如何保证覆盖了所有需要测试的情况；如果你是完全随机生成的测试样例，请思考完全随机的测试程序有何不足之处；如果你在生成测试样例时采用了特殊的策略，比如构造连续数据冒险序列，请你描述一下你使用的策略如何结合了随机性达到强测的效果。**

---

P6的测试除乘除法相关以及HI,LO相关指令，只需测试单条指令的正确性即可。

lb sb 验证是否可以正确的存取字节，以及符号扩展是否正确（字节以及地址）

lh sh 同理验证半字是否正确，符号扩展是否正确（字节以及地址）

bne与beq类似

addi 和andi 主要验证符号扩展

以上指令均可逐条验证，其阻塞与转发可以完全沿用P5的结果。

**mult, multu, div, divu, mfhi, mflo, mthi, mtlo**

mult和div主要验证正与负，正与正，负与负的运算结果是否正确，以及阻塞的周期是否正确

multu和divu主要验证在出现负数时行为是否正确，以及阻塞的周期是否正确

mfhi验证是否可以正确的读出hi寄存器的值，mflo验证是否可以正确的读出lo寄存器的值

mthi

mfhi

mtlo

mflo

mult

mflo

mfhi

验证mfhi，mflo的阻塞是否正确

主要还是用其他同学的自动测试工具进行验证

所有指令均只使用随机生成的三个寄存器，大量测试验证转发阻塞是否正确。

## 四、测试数据

---

其他同学写好的自动测试工具

ori 2,0, 0x132

ori 4,0, 0x234

ori 4,0, 0x47

ori 5,0, 0x914

ori 6,0, 0x42

ori 7,0, 0x4324

ori 8,0, 0x1111

ori 9,0, 0xf73

ori 10,0, 0xa3

ori 11,0, 0x3284

ori 12,0, 0x134c

ori 13,0, 0x231

ori 14,0, 0x2345

ori 15,0, 0x2842

ori 16,0, 0x23c

ori 17,0, 0x81d

ori 18,0, 0x34e

```
ori 19,0, 0x18b
ori 20,0, 0x124e
ori 21,0, 0x1212
ori 22,0, 0x239d
ori 23,0, 0x1384
ori 24,0, 0x172
ori 25,0, 0x1983
ori 26,0, 0x1926
ori 28,0, 0
ori 29,0, 0
ori 14,14, 1
div 29,14
mtlo $14
ori 27,27, 1
div 29,27
mult 29,27
subu 27,14, $14
andi 14,14, 25998
j label1
sltu 14,27, $27
ori 29,29, 1
divu 14,29
label1: mflo $29
ori 14,0, 3225
sb 29, 5(14)
jal label2
ori 29,0, 16
mtlo $29
label2: addu 29,29, $31
jr $29
nop
ori 14,0, 4401
sb 14, 5(14)
```



ori 29,29, 1  
div 29,29  
subu 29,29, \$29  
slt 14,27, \$14  
or 29,29, \$14  
ori 29,0, 8720  
sw 29, 4(29)  
or 14,29, \$29  
j label3  
ori 14,14, 1  
divu 27,14  
ori 14,0, 10568  
lw 27, 12(14)  
label3: ori 14,0, 4726  
lh 27, 14(14)  
lui \$27, 57312  
mult 14,27  
slt 29,29, \$29  
ori 27,0, 404  
lw 29, 4(27)  
bne 29,14, label4  
or 29,14, \$27  
ori 14,0, 2842  
lh 14, 14(14)  
label4: or 27,27, \$29  
lui \$14, 18194  
j label5  
andi 29,29, 65514  
ori 29,29, 24211  
label5: lui \$14, 27936  
addu 27,29, \$14  
bne 29,29, label6  
mflo \$14

ori 29,0, 6092  
lw 29, 12(29)  
label6: mult 14,14  
mult 14,14  
mfhi \$14  
mthi \$29  
ori 27,0, 4530  
lh 29, 14(27)  
subu 27,14, \$14  
sub 27,27, \$27  
subu 14,14, \$14  
ori 14,0, 1953  
lb 14, 5(14)  
andi 27,27, 25262  
ori 27,0, 11524  
sw 27, 4(27)  
multu 14,29  
bne 14,14, label7  
lui \$14, 57856  
lui \$29, 26993  
label7: ori 29,29, 1  
div 29,29  
j label8  
sltu 27,27, \$14  
mflo \$27  
label8: ori 27,14, 23602  
sub 27,29, \$29  
multu 29,29  
mthi \$29  
ori 14,27, 49602  
ori 14,0, 3420  
lh 14, 14(14)  
ori 29,29, 1

```
div 27,29
ori 27,0, 2220
sw 27, 4(27)
subu 14,29, $27
jal label9
ori 14,0, 16
mtlo $27
label9: addu 14,14, $31
jr $14
nop
ori 27,27, 1
div 29,27
ori 27,0, 2040
lw 29, 0(27)
ori 29,29, 1
div 14,29
andi 27,27, 37073
and 27,14, $27
beq 14,27, label10
ori 14,0, 8042
sh 14, 2(14)
sltu 27,27, $14
label10: mflo $14
or 29,14, $14
multu 29,29
addi 29,14, 0
ori 14,0, 4868
lw 14, 8(14)
sltu 27,29, $14
slt 27,29, $29
ori 14,14, 1
div 14,14
mult 14,27
```

```
ori 29,0, 4180
sw 29, 0(29)
lui $14, 57831
ori 14,0, 2540
sh 29, 6(14)
add 14,0, $27
j label11
mtlo $29
or 27,14, $27
label11: addi 27,14, 0
beq 27,14, label12
ori 14,14, 49492
addi 29,27, 0
label12: mult 14,29
bne 29,27, label13
ori 29,29, 1
divu 29,29
addi 14,27, 0
label13: and 27,14, $29
andi 14,14, 62163
slt 29,29, $27
ori 29,0, 5356
lh 29, 14(29)
slt 29,27, $29
lui $27, 39087
ori 27,0, 6689
lbu 29, 11(27)
bne 27,27, label14
multu 27,14
subu 29,14, $29
label14: mult 27,27
ori 14,14, 1
div 29,14
```

or 27,29, \$27  
addu 14,14, \$29  
ori 14,0, 4192  
sh 27, 6(14)  
ori 14,29, 26784  
ori 29,0, 11600  
lh 29, 4(29)  
ori 29,0, 578  
sh 29, 12(29)  
ori 14,0, 636  
lhu 27, 14(14)  
ori 29,0, 424  
lb 29, 7(29)  
ori 14,0, 6496  
lw 14, 4(14)  
ori 14,0, 579  
sb 14, 8(14)  
slt 29,27, \$14  
j label15  
add 14,0, \$14  
ori 14,0, 7468  
sh 27, 4(14)  
label15: ori 14,0, 10414  
lb 29, 10(14)  
ori 27,27, 1  
divu 27,27  
ori 29,0, 2296  
sb 27, 11(29)  
mfhi \$29  
ori 29,0, 11427  
lb 27, 11(29)  
ori 29,0, 6597  
lb 27, 5(29)

ori 14,0, 11352  
sh 29, 8(14)  
mtlo \$27  
lui \$27, 3168  
subu 27,14, \$29  
mtlo \$14  
bne 29,14, label16  
ori 29,0, 6534  
lh 27, 4(29)  
ori 14,0, 10634  
lbu 27, 15(14)  
label16: mthi \$29  
ori 14,0, 5156  
lw 14, 0(14)  
mthi \$14  
ori 29,0, 6379  
lb 27, 14(29)  
ori 14,0, 5433  
sb 14, 8(14)  
mthi \$14  
subu 27,27, \$27  
sltu 14,27, \$29  
and 29,27, \$27  
jal label17  
ori 27,0, 16  
sub 29,29, \$27  
label17: addu 27,27, \$31  
jr \$27  
nop  
ori 29,0, 7208  
lw 29, 4(29)  
mfhi \$27  
ori 29,0, 14

sh 14, 0(29)  
sltu 29, 14, \$14  
addu 29, 27, \$14  
sub 27, 27, \$14  
lui \$14, 9950  
multu 14, 27  
jal label18  
ori 27, 0, 16  
lui \$27, 65045  
label18: addu 27, 27, \$31  
jr \$27  
nop  
andi 29, 29, 46259  
ori 14, 0, 10156  
sh 14, 4(14)  
ori 14, 0, 4628  
sw 29, 12(14)  
ori 14, 14, 1  
div 27, 14  
mtlo \$29  
mult 27, 14  
beq 27, 29, label19  
ori 29, 0, 8346  
lbu 27, 0(29)  
ori 29, 0, 1850  
sh 29, 0(29)  
label19: ori 29, 0, 9363  
lbu 27, 0(29)  
subu 29, 27, \$27  
and 14, 14, \$14  
jal label20  
ori 27, 0, 16  
slt 29, 29, \$14

label20: addu 27,27, \$31

jr \$27

nop

mfhi \$29

multu 14,29

andi 27,27, 41679

subu 27,14, \$29

sub 14,27, \$14

ori 14,0, 10837

lbu 27, 13(14)

and 14,14, \$27

ori 29,29, 1

div 27,29

mflo \$14

add 27,0, \$14

subu 27,29, \$29

lui \$27, 44204

ori 14,0, 5502

lhu 27, 0(14)

mtlo \$14

slt 14,27, \$27

or 27,29, \$29

ori 29,29, 1

divu 14,29

ori 29,0, 5744

sb 29, 4(29)

or 14,27, \$27

ori 14,0, 6623

lb 14, 1(14)

ori 27,0, 700

lw 29, 4(27)

subu 14,14, \$27

ori 14,14, 1



div 14,14  
or 27,29, \$27  
mflo \$27  
mtlo \$27  
slt 27,14, \$14  
ori 29,0, 5974  
lb 14, 1(29)  
mult 29,14  
ori 29,0, 8380  
lw 27, 12(29)  
ori 14,0, 10348  
lw 14, 0(14)  
ori 29,0, 9128  
lhu 14, 12(29)  
multu 14,14  
ori 27,0, 6946  
sh 27, 12(27)  
and 14,29, \$27  
ori 27,0, 1128  
lhu 29, 14(27)  
lui \$14, 2909  
andi 29,14, 1765  
ori 29,0, 3684  
lhu 27, 14(29)  
mfhi \$27  
sltu 27,29, \$29  
mfhi \$29  
ori 27,27, 1  
div 14,27  
ori 27,0, 5152  
lh 14, 8(27)  
sltu 27,29, \$29  
mfhi \$29

or 14,29, \$29  
jal label21  
ori 29,0, 16  
or 27,29, \$27  
label21: addu 29,29, \$31  
jr \$29  
nop  
ori 27,27, 1  
divu 27,27  
bne 29,14, label22  
andi 27,27, 1369  
and 14,27, \$14  
label22: slt 29,14, \$29  
ori 29,29, 1  
divu 29,29  
ori 29,0, 3044  
lw 14, 8(29)  
ori 29,29, 1  
div 29,29  
or 14,14, \$14  
mtlo \$27  
lui \$29, 54148  
andi 29,14, 58154  
mult 29,14  
ori 14,0, 8146  
sb 29, 12(14)  
mthi \$14  
addu 29,29, \$14  
ori 14,14, 1  
div 14,14  
mfhi \$29  
ori 14,14, 1  
divu 29,14

```
ori 27,0, 988
lhu 27, 2(27)
addu 27,27, $29
lui $27, 26566
ori 14,0, 11386
lbu 27, 11(14)
or 14,27, $14
mflo $14
andi 14,14, 2284
addi 14,27, 0
beq 27,27, label23
subu 29,29, $29
andi 29,29, 50100
label23: mflo $27
bne 27,29, label24
lui $29, 28436
andi 27,27, 13106
label24: ori 27,0, 10770
lb 29, 0(27)
subu 14,27, $27
j label25
ori 27,0, 568
lhu 29, 2(27)
multu 29,29
label25: mult 29,27
ori 14,0, 3218
lb 27, 6(14)
slt 27,29, $14
ori 14,0, 11119
lb 29, 8(14)
multu 29,27
subu 27,27, $27
ori 27,0, 9823
```

```
lb 14, 7(27)
ori 29, 0, 9916
lb 29, 10(29)
andi 14, 27, 38000
beq 27, 29, label26
add 29, 0, $14
mflo $27
label26: mflo $29
mfhi $14
mtlo $14
addi 27, 14, 0
slt 29, 14, $27
j label27
ori 27, 0, 5038
lhu 27, 4(27)
lui $14, 22824
label27: ori 27, 0, 2956
lw 14, 12(27)
j label28
andi 14, 29, 19029
mfhi $27
label28: ori 29, 0, 1052
lw 29, 0(29)
ori 14, 0, 6099
sb 14, 2(14)
j label29
or 29, 29, $14
ori 29, 0, 9040
lhu 27, 0(29)
label29: mthi $29
add 27, 0, $14
addu 27, 14, $29
sltu 29, 29, $29
```

addu 14,14, \$27  
ori 14,27, 15879  
lui \$29, 12701  
ori 29,0, 2788  
sw 29, 12(29)  
ori 27,0, 7058  
sb 27, 4(27)  
ori 29,14, 44384  
mfhi \$14  
multu 29,27  
ori 27,0, 7327  
sb 29, 15(27)  
add 14,0, \$27  
addi 27,27, 0  
mflo \$14  
ori 14,0, 6746  
lb 14, 13(14)  
multu 27,27  
ori 29,29, 1  
divu 27,29  
addi 27,14, 0  
mflo \$14  
ori 27,27, 1  
divu 29,27  
ori 14,14, 1  
div 29,14  
addi 14,29, 0  
addu 14,27, \$14  
beq 14,27, label30  
add 29,0, \$27  
ori 29,29, 55342  
label30: addi 14,29, 0  
add 29,0, \$14

```
mflo $27
ori 29,0, 8696
lb 29, 14(29)
beq 14,14, label31
andi 29,27, 2861
sltu 27,14, $27
label31: ori 27,27, 1
div 14,27
or 14,29, $14
andi 14,29, 43792
slt 14,14, $29
ori 29,0, 9589
sb 29, 5(29)
sub 27,29, $14
ori 14,27, 10101
slt 27,14, $27
addu 14,29, $14
jal label32
ori 14,0, 16
mult 14,29
label32: addu 14,14, $31
jr $14
nop
ori 29,0, 5580
sw 29, 12(29)
ori 29,29, 1
div 14,29
mfhi $27
ori 27,0, 773
sb 29, 13(27)
ori 29,0, 11364
lw 29, 12(29)
mfhi $27
```

```
ori 14,0, 922
lh 27, 4(14)
ori 29,0, 7654
sh 14, 12(29)
ori 29,0, 6366
lhu 14, 2(29)
ori 14,14, 3469
bne 14,27, label33
add 14,0, $14
ori 29,0, 160
lw 14, 4(29)
label33: ori 14,14, 31490
ori 27,0, 5568
lw 29, 0(27)
beq 14,14, label34
andi 29,27, 41464
ori 27,0, 6016
sb 14, 4(27)
label34: ori 29,0, 5060
lw 29, 0(29)
j label35
and 14,29, $29
mtlo $27
label35: mflo $14
ori 14,0, 7186
lhu 29, 6(14)
andi 14,29, 34421
ori 27,0, 10908
sw 27, 8(27)
ori 29,0, 7876
lhu 29, 4(29)
mthi $27
and 14,14, $29
```

```
mfhi $27
sub 27,29, $27
ori 14,0, 8480
lw 29, 12(14)
sub 27,27, $29
addi 14,29, 0
ori 27,0, 8617
sb 27, 13(27)
ori 14,0, 10772
lb 29, 13(14)
ori 27,0, 6048
lw 29, 8(27)
ori 27,0, 5662
sh 14, 10(27)
ori 27,0, 2154
sh 14, 10(27)
and 27,27, $27
ori 14,14, 1
div 29,14
addu 14,29, $29
ori 27,27, 1
divu 29,27
mtlo $14
j label36
addu 14,14, $29
lui $29, 50892
label36: lui $27, 51926
mfhi $27
slt 14,14, $29
ori 27,0, 2968
lb 27, 12(27)
or 27,14, $27
add 29,0, $14
```



mult 29,29  
jal label37  
ori 29,0, 16  
multu 14,27  
label37: addu 29,29, \$31  
jr \$29  
nop  
add 29,0, \$14  
ori 27,0, 9216  
lbu 14, 11(27)  
mtlo \$14  
andi 29,14, 14216  
mflo \$14  
addu 29,14, \$14  
ori 14,0, 9440  
sw 27, 4(14)  
sub 27,29, \$29  
add 29,0, \$29  
ori 29,29, 19793  
slt 29,29, \$27  
addu 29,14, \$29  
ori 29,0, 10865  
sb 14, 4(29)  
ori 27,0, 6224  
sh 27, 6(27)  
ori 27,0, 3992  
lw 27, 4(27)  
mthi \$29  
or 14,29, \$14  
ori 27,27, 1  
div 14,27  
ori 27,0, 1885  
sb 14, 5(27)

```
ori 27,0, 2752
sw 29, 12(27)
ori 27,0, 5144
lh 14, 10(27)
ori 14,0, 4807
sb 14, 12(14)
ori 27,0, 7436
lw 27, 12(27)
mthi $27
mult 29,27
ori 14,14, 1
divu 14,14
or 14,29, $29
ori 27,0, 4683
sb 27, 9(27)
mthi $29
ori 14,14, 1266
addi 14,29, 0
ori 27,27, 1
div 27,27
beq 29,14, label38
ori 14,14, 1
div 14,14
and 27,14, $14
label38: ori 27,0, 7780
lw 27, 0(27)
mthi $14
ori 27,27, 1
divu 27,27
ori 14,14, 56954
mthi $29
addu 14,29, $27
mfhi $14
```

ori 27,0, 2918  
lh 29, 0(27)  
mtlo \$29  
mfhi \$29  
mflo \$27  
sub 27,27, \$27  
mfhi \$14  
ori 14,0, 494  
lhu 29, 8(14)  
mfhi \$27  
ori 27,27, 1  
divu 29,27  
addi 14,27, 0  
ori 29,0, 9593  
lbu 14, 15(29)  
lui \$27, 60384  
ori 29,0, 3390  
sb 29, 5(29)  
ori 29,0, 2092  
lhu 27, 4(29)  
bne 27,27, label39  
ori 29,0, 10654  
lb 29, 8(29)  
ori 14,0, 8484  
lhu 27, 8(14)  
label39: slt 14,27, \$14  
addu 29,29, \$27  
lui \$14, 737  
mthi \$14  
mfhi \$27  
mfhi \$29  
j label40  
mtlo \$27

ori 29,0, 3830  
sh 27, 0(29)  
label40: addi 29,27, 0  
mthi \$14  
slt 27,14, \$29  
and 14,14, \$29  
bne 27,14, label41  
slt 14,27, \$27  
and 27,27, \$14  
label41: slt 14,27, \$27  
mult 14,29  
mtlo \$27  
ori 27,0, 6008  
lhu 29, 6(27)  
ori 14,0, 372  
lhu 29, 8(14)  
ori 29,0, 5977  
lb 27, 10(29)  
slt 14,14, \$14  
andi 14,29, 3043  
mfhi \$29  
mfhi \$14  
ori 29,0, 8460  
sb 14, 15(29)  
mtlo \$14  
andi 27,14, 46061  
ori 29,0, 228  
sw 14, 12(29)  
mthi \$29  
subu 29,29, \$29  
sltu 14,27, \$29  
jal label42  
ori 14,0, 16

mfhi \$27  
label42: addu 14,14, \$31  
jr \$14  
nop  
ori 27,27, 1  
div 29,27  
sub 27,29, \$29  
j label43  
ori 27,0, 9713  
sb 29, 1(27)  
mflo \$29  
label43: andi 27,29, 25414  
j label44  
ori 14,0, 6773  
lb 29, 10(14)  
add 14,0, \$29  
label44: ori 29,0, 11148  
lw 27, 4(29)  
and 29,27, \$14  
add 27,0, \$29  
ori 27,0, 7732  
sw 27, 12(27)  
ori 29,0, 807  
lbu 27, 15(29)  
ori 27,27, 1  
divu 14,27  
ori 27,0, 10726  
lb 14, 12(27)  
ori 27,0, 7720  
lw 27, 8(27)  
ori 27,29, 21601  
addu 29,14, \$27  
subu 14,29, \$14

ori 14,14, 1  
div 27,14  
sltu 29,27, \$27  
ori 29,0, 996  
sb 29, 7(29)  
ori 14,0, 4  
lbu 27, 6(14)  
subu 27,14, \$14  
ori 14,0, 11052  
lb 27, 15(14)  
sub 29,29, \$27  
j label45  
ori 14,0, 9852  
lb 14, 15(14)  
ori 14,0, 218  
lh 14, 0(14)  
label45: ori 14,0, 7368  
lh 27, 10(14)  
beq 29,14, label46  
ori 27,0, 8922  
lh 29, 14(27)  
ori 14,0, 4386  
lh 27, 8(14)  
label46: and 29,27, \$27  
or 27,14, \$29  
mflo \$27  
add 27,0, \$27  
mfhi \$27  
addi 27,29, 0  
ori 29,0, 8250  
sh 14, 14(29)  
slt 29,27, \$27  
addu 29,29, \$29

```
addi 27,14, 0
ori 29,14, 40266
and 29,14, $27
lui $29, 56828
ori 14,14, 1
div 14,14
sub 27,29, $29
jal label47
ori 29,0, 16
subu 29,27, $29
label47: addu 29,29, $31
jr $29
nop
andi 27,27, 29901
add 14,0, $29
ori 29,0, 2204
lh 29, 14(29)
add 27,0, $27
lui $27, 40168
andi 27,27, 35443
jal label48
ori 29,0, 16
add 27,0, $27
label48: addu 29,29, $31
jr $29
nop
ori 29,29, 1
div 14,29
and 14,29, $27
beq 14,14, label49
sub 29,14, $29
subu 29,27, $29
label49: ori 29,29, 1
```

```
div 14,29
andi 29,27, 14142
ori 29,0, 2594
lb 27, 1(29)
j label50
ori 29,0, 8264
sh 14, 8(29)
ori 14,0, 3494
lb 29, 9(14)
label50: add 14,0, $14
add 27,0, $14
subu 27,27, $14
jal label51
ori 29,0, 16
mfhi $27
label51: addu 29,29, $31
jr $29
nop
ori 27,0, 8
lw 27, 12(27)
ori 29,29, 1
div 29,29
mflo $14
ori 27,27, 1
divu 29,27
addi 14,29, 0
mthi $14
and 27,14, $14
mfhi $14
ori 27,29, 39100
```



