

# 一、CPU设计方案综述

## (一) 总体设计概述

本CPU为Verilog实现的流水线MIPS - CPU，支持的指令集包含 { LB, LBU, LH, LHU, LW, SB, SH, SW, ADD, ADDU, SUB, SUBU, MULT, MULTU, DIV, DIVU, SLL, SRL, SRA, SLLV, SRLV, SRAV, AND, OR, XOR, NOR, ADDI, ADDIU, ANDI, ORI, XORI, LUI, SLT, SLTI, SLTIU, SLTU, BEQ, BNE, BLEZ, BGTZ, BLTZ, BGEZ, J, JAL, JALR, JR, MFHI, MFLO, MTHI, MTLO }。为了实现这些功能，CPU主要包含了PC、IM、IF\_ID、GRF、ID\_EX、ALU、MDU、EX\_MEM、DM、MEM\_WB这些模块按照流水线五阶段的顶层设计逐级展开。

本设计中，不再设计NPC模块，而使用在mips中构造MUX的方式选择传递给PC的信号。

在本设计中，对模块之间解耦合是设计重点和精要，因此，单个模块的设计将会比较简单，和单周期非常相似，但是在mips.v中，要在这里写全部的mux，采取这样的方案是避免单列mux.v接口过于冗杂。

在本设计中，各级流水线寄存器尽可能多地保留数据，而非仅针对本设计支持的十条指令。

## (二) 关键模块定义

### 1. IM (已外置)

#### 介绍

Instruction Memory，指令存储器，支持初始化时一次性读入全部指令，并根据给定PC读出指令，支持0x3000-0x4000。

#### 外置注意

- 1. 由于本模块外置，因此端口输入输出方向是从CPU视角。
- 2. 在官方tb中，其具体实现为 assign i\_inst\_rdata = inst[(i\_inst\_addr - 32'h3000) >> 2];

端口	输入/输出	位宽	描述
i_inst_addr	I	32	输入指令地址
i_inst_rdata	O	32	输出对应指令

### 2. PC

#### 介绍

获取指令，支持同步复位至0x3000。

#### P6 update

NULL

端口	输入/输出	位宽	描述
clk	I	1	时钟信号
reset	I	1	同步复位信号
F_WE	I	1	是否继续读取下一条指令
NPC	I	32	下一条指令地址
PC	O	32	输出当前指令

hint:本设计中，将MUX模块全部集成在 `mips.v` 中，以便于接线。值得注意的是，本设计交由 `mips.v` 完成转发与选择信号，降低了模块间耦合度，即，`PC` 仅需专注于在每个时钟上升沿给出下一个指令即可。

### 3. GRF

#### 介绍

通用寄存器文件，可以存取32位数据，支持同步复位，寄存器编号为0~31，其中0号寄存器的读出值恒为0。

#### P6 update

NULL

端口	输入/输出	位宽	描述
clk	I	1	时钟信号
reset	I	1	同步复位信号
F_WE(GRF_WE)	I	1	写使能信号
F_PC(GRF_PC)	I	32	当前指令（用于评测姬输出）
D_A1	I	5	A1
D_A2	I	5	A2
D_A3	I	5	A3
D_WD	I	32	WD
D_RD1	O	32	RD1
D_RD2	O	32	RD2

本设计的输入端口中的GRF\_WE和PC同样由 `mips.v` 中实现。

值得注意的是，由于是在流水线中，因此存在一种情况，即，当前 `GRF_WE == 1'b1`（要写入）且 `A3 == A1/A2 != 1'b0`，即，当前要写入的寄存器恰为 `A1` 或 `A2`，这时，如果继续读取原来GRF中的寄存器值将产生错误，需要读取的是WD。在本设计中将这部分实现于GRF。

4. IF\_ID

介绍

I级和D级间的流水线寄存器，同时产生D级的控制信号。

P6 update

对D\_PCsrc针对jalr和b类指令进行判断和增量开发。

端口	方向	位宽	描述
clk	I	1	时钟信号
reset	I	1	同步复位信号
F_WE	I	1	写使能信号（用于处理暂停）
F_instr	I	32	下一个指令
F_PC(pc_PC)	I	32	下一个PC
D_instr	O	32	当前指令
D_PC	O	32	当前PC
D_PCsrc	O	8	8'd0: PC = PC + 4。 8'd1: 如果 Zero = 1, PC 跳转到 beq 指令对应的跳转地址；否则依旧执行 PC = PC + 4。(判断zero是否为0在mips.v中实现) 8'd2: 跳转到 jal 指令和 j 指令对应的跳转地址。 8'd3: 跳转到 jr 指令和jalr对应的跳转地址。 8'd4: 跳转到blez判断后的地址 8'd5: 跳转到bltz判断后的地址 8'd6: 跳转到bgez判断后的地址 8'd7: 跳转到bgtz判断后的地主之 8'd8: 跳转到bne判断后的指令 以上仅根据指令给出D_PCsrc信号，具体判断的MUX在mips.v中实现。

5. GSU (General Splitter Unit)

介绍

通用解码单元，对给定指令分析其各种数据。

P6 update

对所有新增指令进行解码，不再在下表中——列出。

值得注意的是，本次按照JLeo法新增了R类指令判断，这一点在SFU中也有所应用，需要在转跳时加以注意。

端口	方向	位宽	描述
instr	I	32	给定指令
opcode	O	6	当前给定指令opcode
funct	O	6	当前给定指令funct
I25_21	O	5	instr[25:21]
I20_16	O	5	instr[20:16]
I15_11	O	5	instr[15:11]
I10_6	O	5	instr[10:6]
IMM16	O	16	instr[15:0]
IMM26	O	26	instr[25:0]
R	O	1	判断R类指令，即opcode为0的指令
addu	O	1	判断addu
subu	O	1	判断subu
ori	O	1	判断ori
lw	O	1	判断lw
sw	O	1	判断sw
beq	O	1	判断beq
lui	O	1	判断lui
j	O	1	判断j
jal	O	1	判断jal
jr	O	1	判断jr
nop	O	1	判断nop

## 6. ID\_EX

### 介绍

D级、E级间流水线寄存器，同时产生E级的控制信号。

### P6 update

1. 新增了E\_MDUOp信号，用于判断当前指令是否为乘除相关指令，如果是的话具体为哪一个。
2. 对E\_ALUOp依照新增指令新加信号，以满足不同运算需求
3. 对E\_ALU\_Bsrc依照新增指令新加信号，主要用于处理sll,srl,sra,sllv,srlv,srav六条指令

端口	方向	位宽	描述
clk	I	1	时钟信号
reset	I	1	同步复位信号
D_instr	I	32	下一指令
D_PC	I	32	下一PC
D_RD1	I	32	下一RD1
D_RD2	I	32	下一RD2
E_instr	O	32	当前指令
E_PC	O	32	当前PC
E_RD1	O	32	当前RD1
E_RD2	O	32	当前RD2
E_ALUOp	O	8	当前ALU选择信号（详见ALU模块说明）
E_ALU_Bsrc	O	8	选择ALUB的输入端（详见ALU模块说明）

## 7. ALU

### 介绍

处理乘除法之外的算术运算。

### P6 update

依照新增指令新加若干运算类型。

端口	方向	位宽	描述
E_ALU_A	I	32	操作数A
E_ALU_B	I	32	操作数B
E_ALUOp	I	8	操作信号
E_ALUResult	O	32	输出

### E\_ALUOp功能定义

序号	功能名称	功能描述
8'd0	按位与	$E\_ALU\_A \& E\_ALU\_B$
8'd1	按位或	$E\_ALU\_A   E\_ALU\_B$
8'd2	加法	$E\_ALU\_A + E\_ALU\_B$
8'd3	减法	$E\_ALU\_A - E\_ALU\_B$
8'd4	左移16位	$E\_ALU\_B \ll 16$
8'd5	逻辑左移	$E\_ALU\_A \ll E\_ALU\_B[4:0]$
8'd6	逻辑右移	$E\_ALU\_A \gg E\_ALU\_B[4:0]$
8'd7	异或	$E\_ALU\_A \wedge E\_ALU\_B$
8'd8	有符号比较大小	$(E\_ALU\_A < E\_ALU\_B)$
8'd9	无符号比较大小	$(E\_ALU\_A < E\_ALU\_B)$
8'd10	或非	$\sim(E\_ALU\_A   E\_ALU\_B)$
8'd11	算术右移	$E\_ALU\_A \ggg E\_ALU\_B[4:0]$

ALUOp信号表

ALUOp	功能	描述
8'd0	按位与	$E\_ALU\_A \& E\_ALU\_A$
8'd1	按位或	$E\_ALU\_A   E\_ALU\_B$
8'd2	加法	$E\_ALU\_A + E\_ALU\_B$
8'd3	减法	$E\_ALU\_A - E\_ALU\_B$
8'd4	B左移16位	$E\_ALU\_B \ll 16$

8. MDU

介绍

乘除块内，有E\_MDUHi，E\_MDULo两个寄存器，可以进行乘除运算，完成相关乘除指令，并模拟乘除运算的延时。

端口	方向	位宽	描述
clk	I	1	时钟信号
reset	I	1	同步复位信号
E_MDU_A	I	32	参与乘除运算的第一个操作数
E_MDU_B	I	32	参与乘除运算的第二个操作数
E_MDUOp	I	8	MDU功能选择信号，具体见功能定义
Busy	O	1	当前是否正在及逆行乘除运算
E_MDUHi	O	32	E_MDUHi寄存器的值
E_MDULo	O	32	E_MDULo寄存器的值

### E\_MDUOp功能定义

序号	功能名称	功能描述
8'd1	有符号乘法	将A*B的高32位和低32位分别写入E_MDUHi和E_MDULo寄存器，同时设置延迟为5个时钟周期（有符号）
8'd2	无符号乘法	将A*B的高32位和低32位分别写入E_MDUHi和E_MDULo寄存器，同时设置延迟为5个时钟周期（无符号）
8'd3	有符号除法	将A/B的余数和商分别写入E_MDUHi和E_MDULo寄存器，同时设置延迟为10个时钟周期（有符号）
8'd4	无符号除法	将A/B的余数和商分别写入E_MDUHi和E_MDULo寄存器，同时设置延迟为10个时钟周期（无符号）
8'd7	写 E_MDUHi 寄存器	将A写入E_MDUHi寄存器
8'd8	写 E_MDULo 寄存器	将A写入E_MDULo寄存器

## 9. EX\_MEM

### 介绍

E级和M级之间流水线寄存器，同时产生M级的控制信号。

### P6 update

1. 由于DM外置，因此删去M\_DMWE，新增信号m\_data\_byteen[3:0]，支持按字节访存，某位为1即代表该位写使能为1（详见DM说明）。
2. 对M\_MemtoReg依照新增指令添加信号，主要支持jalr,lb,lbu,lh,lhu等。

端口	方向	位宽	描述
clk	I	1	时钟信号
reset	I	1	同步复位信号
E_instr	I	32	E指令
E_PC	I	32	EPC
E_RD1	I	32	ERD1
E_RD2	I	32	ERD2
E_ALUResult	I	32	EALUResult
E_DMWD	I	32	EDMWD
M_instr	O	32	M指令
M_PC	O	32	MPC
M_RD1	O	32	MRD1
M_RD2	O	32	MRD2
M_ALUResult	O	32	MALUResult
M_DMWD	O	32	MDMWD
m_data_byteen	O	4	支持按字节访存
M_MemtoReg	O	8	选择回写到GRF的数据（详见）

## 10. DM（已外置）

### 介绍

数据存储器，数据位宽32位，容量为**12KB**(32bit/word×**3072word**)，支持同步复位，起始地址为0x0000\_0000。

### 外置注意

1. 由于本模块外置，因此**端口输入输出方向是从CPU视角**。
2. 在官方tb中，其具体实现为 `assign i_inst_rdata = inst[(i_inst_addr - 32'h3000) >> 2];`



端口	方向	位宽	描述
clk	I	1	时钟信号
reset	I	1	同步复位信号
m_inst_addr(M_PC)	I	32	当前指令PC
m_data_addr(M_DMA)	I	32	当前读取或写入数据地址
m_data_wdata(M_DMWD)	I	32	写入DM的数据
m_data_byteen	I	4	支持按字节访存
m_data_rdata(M_DMRD)	O	32	读出数据

## 11. MEM\_WB

### 介绍

MEM\_WB级寄存器，由M级向W级流水，同时产生W级的控制信号。

### P6 update

1. 对W\_RegDst依照新增指令添加信号。
2. 对W\_RegWE依照新增指令添加信号。

端口	方向	位宽	描述
clk	I	1	时钟信号
reset	I	1	同步复位信号
M_instr	I	32	M级instr
M_PC	I	32	M级PC
M_RD1	I	32	M级RD1
M_RD2	I	32	M级RD2
M_ALUResult	I	32	M级ALUResult
M_GRFWD	I	32	M级GRFWD
W_instr	O	32	W级instr
W_PC	O	32	W级PC
W_RD1	O	32	W级RD1
W_RD2	O	32	W级RD2
W_ALUResult	O	32	W级ALUResult
W_GRFWD	O	32	W级GRFWD
W_RegDst	O	8	W级RegDst
W_RegWE	O	1	W级RegWE

12.DEC

介绍

指令分类译码器，输入一个指令，返回该指令在数据冒险中对应的类型。

P6 update

- 1. 对新增指令给予判断。

端口	方向	位宽	描述
instr	I	32	需要分类的指令
j	O	1	需要读寄存器的跳转指令，如 beq, jalr, jr（在本CPU中暂不支持 jalr）
r	O	1	除 jalr, jr 外的 R 型指令（这里需要特别注意课上指令）；简单的 R 型算术运算指令：addu、subu。
i	O	1	I 型指令；带有立即数的 I 型算数、逻辑运算指令：ori、lui。
load	O	1	load 型指令，包括 lw
store	O	1	store 型指令，包括 sw；内存写入指令：sw。
jal	O	1	jal, jalr 型指令
rs	O	4	I[25:21]
rt	O	4	I[20:16]
rd	O	4	如果指令为 jal 输出31，如果是课上新添的需要写入特定寄存器的指令则为对应寄存器，否则为I[15:11]

值得注意的是，在 DEC 内部，实例化了 GSU ,值得注意的是，

（三）重要机制实现方法

1. 跳转

NPC放在D级，beq比较器放在D级，jal指令将PC+8的值流水后在W级写入31号寄存器。jjr指令直接跳转，但jr需要rs的值可能需要阻塞。

2. 流水线延迟槽

跳转指令后都有延迟槽，值都进行计算，因此jal需要写入PC+8的值。

3. 转发

转发发生在D,E,M级，D级转发的值来自M级和W级（因为GRF没采用内部转发机制），E级转发的值来自M级和W级，M级转发的值来自W级，主要针对lw,sw指令，如果lw指令后紧跟sw指令且使用的寄存器值相同，则需要转发。

二、测试方案

## (一) 典型测试样例

### 10组测试数据

#### testpoint 0

```
ori $1, $0, 6
ori $2, $0, 5
ori $3, $0, 9
ori $4, $0, 7
sw $1, 0($0)
sw $3, 4($0)
sw $3, 8($0)
sw $1, 12($0)
sw $2, 16($0)
sw $1, 20($0)
sw $3, 24($0)
sw $3, 28($0)
sw $2, 32($0)
sw $1, 36($0)
sw $2, 40($0)
sw $3, 44($0)
sw $2, 48($0)
sw $3, 52($0)
sw $2, 56($0)
sw $1, 60($0)
sw $2, 64($0)
sw $1, 68($0)
sw $2, 72($0)
sw $3, 76($0)
sw $2, 80($0)
sw $4, 84($0)
sw $1, 88($0)
sw $3, 92($0)
sw $3, 96($0)
sw $4, 100($0)
sw $2, 104($0)
sw $4, 108($0)
sw $3, 112($0)
sw $4, 116($0)
sw $1, 120($0)
sw $1, 124($0)
bne $2, $2, TAG1
sb $2, 0($2)
lbu $1, 0($2)
multu $1, $2
TAG1:
bne $1, $1, TAG2
mfhi $2
xor $1, $2, $2
and $4, $1, $1
TAG2:
sw $4, 0($4)
lb $3, 0($4)
beq $4, $3, TAG3
sll $2, $3, 5
TAG3:
```

```

mtlo $2
multu $2, $2
mfhi $4
mthi $4
TAG4:
slti $3, $4, 10
bgez $3, TAG5
mthi $3
lui $4, 3
TAG5:
nor $1, $4, $4
divu $4, $1
bgtz $4, TAG6
mult $1, $1
TAG6:
bltz $1, TAG7
sb $1, 1($1)
addu $4, $1, $1
addi $4, $4, 11
TAG7:
lbu $1, 0($4)
slt $3, $1, $4
lui $3, 8
sra $1, $3, 4
TAG8:
bltz $1, TAG9
mthi $1
bgez $1, TAG9
sll $0, $0, 0
TAG9:
subu $2, $1, $1
mflo $3
mfhi $1
div $3, $1
TAG10:
bltz $1, TAG11
mthi $1
beq $1, $1, TAG11
ori $1, $1, 12
TAG11:
addiu $2, $1, 6
subu $4, $2, $1
sb $2, 0($4)
mtlo $4
TAG12:
bne $4, $4, TAG13
lui $4, 10
beq $4, $4, TAG13
lui $2, 7
TAG13:
mfhi $3
div $3, $3
lui $3, 8
mthi $2
TAG14:
bgez $3, TAG15
lui $1, 13
div $3, $3

```

```
mtlo $3
TAG15:
xori $1, $1, 3
sll $0, $0, 0
beq $1, $1, TAG16
addu $1, $1, $1
TAG16:
bgez $1, TAG17
lui $4, 1
mflo $2
lui $2, 15
TAG17:
bgtz $2, TAG18
multu $2, $2
mflo $4
mflo $2
TAG18:
mthi $2
mtlo $2
sll $0, $0, 0
multu $2, $2
TAG19:
sll $0, $0, 0
mflo $4
divu $1, $2
sll $0, $0, 0
TAG20:
sllv $2, $4, $4
lui $3, 0
lh $1, 0($4)
mult $1, $3
TAG21:
ori $1, $1, 10
ori $2, $1, 5
mult $2, $1
xori $3, $1, 14
TAG22:
lui $4, 12
mfhi $4
mtlo $3
beq $3, $4, TAG23
TAG23:
sb $4, 0($4)
mtlo $4
beq $4, $4, TAG24
srlv $3, $4, $4
TAG24:
mthi $3
lui $1, 2
mtlo $3
lui $1, 3
TAG25:
sll $0, $0, 0
bgez $1, TAG26
mflo $3
beq $1, $1, TAG26
TAG26:
mfhi $4
```

```

mtlo $3
lb $1, 0($3)
mult $3, $1
TAG27:
lui $1, 8
sll $0, $0, 0
beq $1, $1, TAG28
andi $4, $1, 1
TAG28:
addiu $3, $4, 7
bgtz $3, TAG29
sb $3, 0($4)
lui $4, 5
TAG29:
mthi $4
lui $3, 15
mthi $3
div $3, $3
TAG30:
sra $1, $3, 1
mflo $4
lui $3, 10
bltz $1, TAG31
TAG31:
sll $0, $0, 0
bgtz $3, TAG32
mult $3, $3
divu $3, $3
TAG32:
beq $3, $3, TAG33
sll $0, $0, 0
mtlo $3
multu $3, $3
TAG33:
subu $3, $3, $3
mthi $3
multu $3, $3
sb $3, 0($3)
TAG34:
lw $4, 0($3)
mtlo $4
multu $3, $4
sub $1, $4, $3
TAG35:
lb $2, 0($1)
lui $4, 11
mult $1, $1
sll $0, $0, 0
TAG36:
bne $4, $4, TAG37
addiu $3, $4, 14
sll $0, $0, 0
nor $3, $2, $3
TAG37:
sll $0, $0, 0
lui $4, 6
ori $1, $3, 13
lui $1, 8

```

```

TAG38:
sll $0, $0, 0
slti $4, $1, 7
lbu $4, 0($4)
addu $1, $4, $4
TAG39:
srl $4, $1, 11
addu $3, $1, $4
sltiu $1, $4, 6
subu $2, $1, $4
TAG40:
sb $2, 0($2)
mthi $2
srl $1, $2, 12
lui $4, 7
TAG41:
sll $0, $0, 0
xor $3, $2, $4
lui $4, 11
mfhi $2
TAG42:
andi $1, $2, 11
beq $1, $2, TAG43
mfhi $2
mflo $1
TAG43:
sltiu $2, $1, 6
lb $3, 0($1)
lui $2, 12
slti $1, $1, 7
TAG44:
lb $3, 0($1)
sb $1, 0($1)
div $1, $3
mtlo $1
TAG45:
bne $3, $3, TAG46
divu $3, $3
subu $2, $3, $3
bgez $2, TAG46
TAG46:
lw $3, 0($2)
srav $2, $3, $3
sh $2, -256($2)
sllv $1, $2, $2
TAG47:
mfhi $2
blez $1, TAG48
divu $2, $1
bne $1, $1, TAG48
TAG48:
mflo $2
ori $1, $2, 2
bltz $2, TAG49
mfhi $2
TAG49:
multu $2, $2
beq $2, $2, TAG50

```

```

srli $4, $2, 12
lb $3, 0($4)
TAG50:
lui $4, 11
bltz $4, TAG51
sll $0, $0, 0
mthi $1
TAG51:
sh $1, 0($1)
mflo $3
bne $3, $3, TAG52
srli $1, $3, $1
TAG52:
lh $1, 0($1)
multu $1, $1
lbu $1, -256($1)
sw $1, 0($1)
TAG53:
lui $4, 5
mfhi $2
beq $1, $1, TAG54
lui $4, 12
TAG54:
addiu $2, $4, 6
lui $4, 15
srai $4, $4, $2
beq $4, $4, TAG55
TAG55:
sw $4, -15360($4)
sb $4, -15360($4)
bne $4, $4, TAG56
sllv $2, $4, $4
TAG56:
beq $2, $2, TAG57
mtlo $2
addiu $4, $2, 0
ori $3, $4, 5
TAG57:
multu $3, $3
mfhi $3
lbu $4, 0($3)
lui $4, 13
TAG58:
bgtz $4, TAG59
div $4, $4
mtlo $4
sb $4, 0($4)
TAG59:
mtlo $4
sll $0, $0, 0
lui $1, 9
sll $0, $0, 0
TAG60:
lui $4, 12
div $4, $1
mfhi $3
sll $0, $0, 0
TAG61:

```



```

or $2, $3, $3
beq $3, $3, TAG62
div $3, $2
sb $3, 0($2)
TAG62:
addiu $2, $2, 10
or $2, $2, $2
bne $2, $2, TAG63
mflo $3
TAG63:
lui $2, 8
sll $0, $0, 0
addu $3, $2, $2
mthi $3
TAG64:
mthi $3
bltz $3, TAG65
lui $3, 9
mtlo $3
TAG65:
lui $1, 5
sll $0, $0, 0
mfhi $1
multu $3, $3
TAG66:
mfhi $3
mfhi $3
lui $2, 6
sll $0, $0, 0
TAG67:
mfhi $3
lbu $1, 0($3)
lh $4, 0($1)
andi $3, $4, 3
TAG68:
lui $3, 13
mflo $4
mthi $4
mult $3, $4
TAG69:
mthi $4
lh $1, 0($4)
div $1, $1
mfhi $4
TAG70:
mthi $4
mflo $2
ori $2, $2, 10
mthi $4
TAG71:
mfhi $1
addiu $3, $2, 10
sltu $2, $2, $1
lh $4, 0($2)
TAG72:
addiu $4, $4, 9
mtlo $4
lui $3, 13

```

```
addu $1, $4, $4
TAG73:
blez $1, TAG74
lui $4, 7
lui $3, 10
sll $0, $0, 0
TAG74:
sll $0, $0, 0
beq $3, $3, TAG75
ori $3, $3, 4
nor $2, $3, $3
TAG75:
bltz $2, TAG76
sb $2, 0($2)
andi $3, $2, 1
mult $2, $3
TAG76:
lui $3, 15
sll $0, $0, 0
mtlo $3
ori $3, $3, 15
TAG77:
beq $3, $3, TAG78
sll $0, $0, 0
sltiu $1, $1, 14
lui $3, 11
TAG78:
lui $1, 13
mfhi $1
lw $2, 0($1)
sll $0, $0, 0
TAG79:
sll $0, $0, 0
sra $4, $3, 11
sll $0, $0, 0
mthi $4
TAG80:
lui $3, 0
beq $1, $1, TAG81
sw $3, 0($3)
beq $3, $1, TAG81
TAG81:
mflo $3
mflo $4
bne $3, $3, TAG82
mtlo $3
TAG82:
lui $4, 5
lui $4, 11
mthi $4
sll $0, $0, 0
TAG83:
sll $0, $0, 0
mult $1, $1
sh $1, 0($1)
lui $4, 10
TAG84:
slt $3, $4, $4
```

```
addiu $1, $4, 7
bltz $3, TAG85
addu $2, $4, $4
TAG85:
subu $4, $2, $2
div $2, $2
mfhi $3
bne $4, $4, TAG86
TAG86:
mtlo $3
lui $4, 14
or $4, $4, $3
lui $1, 15
TAG87:
bltz $1, TAG88
sll $0, $0, 0
divu $2, $2
sll $0, $0, 0
TAG88:
multu $2, $2
srlv $2, $2, $2
subu $1, $2, $2
bgtz $2, TAG89
TAG89:
multu $1, $1
bne $1, $1, TAG90
multu $1, $1
ori $1, $1, 12
TAG90:
mthi $1
sw $1, 0($1)
divu $1, $1
andi $1, $1, 11
TAG91:
bgtz $1, TAG92
lw $4, 0($1)
lbu $3, 0($4)
bne $4, $3, TAG92
TAG92:
mfhi $3
sllv $1, $3, $3
lui $1, 7
lbu $3, 0($3)
TAG93:
lw $2, 0($3)
blez $3, TAG94
mult $3, $3
lui $3, 5
TAG94:
mthi $3
sw $3, 0($3)
lh $3, 0($3)
sra $1, $3, 2
TAG95:
nor $4, $1, $1
and $4, $4, $1
mflo $4
lw $1, 0($4)
```

```

TAG96:
sllv $1, $1, $1
multu $1, $1
lh $2, 0($1)
bgez $2, TAG97
TAG97:
sltiu $4, $2, 15
mthi $2
sb $2, 0($4)
lbu $4, 0($4)
TAG98:
lh $4, 0($4)
bltz $4, TAG99
nor $2, $4, $4
add $1, $4, $4
TAG99:
lui $3, 11
lb $3, 0($1)
lb $3, 0($1)
mthi $1
TAG100:
mtlo $3
lui $2, 12
lui $1, 14
slti $2, $3, 11
TAG101:
sll $1, $2, 7
lbu $1, 0($2)
sh $1, 0($1)
sub $1, $1, $1
TAG102:
sra $3, $1, 4
multu $1, $3
mthi $3
mflo $3
TAG103:
bne $3, $3, TAG104
lui $3, 11
beq $3, $3, TAG104
sltiu $2, $3, 6
TAG104:
beq $2, $2, TAG105
mthi $2
lui $2, 4
lb $2, 0($2)
TAG105:
lui $1, 6
lh $3, 0($2)
xor $2, $3, $3
mfhi $3
TAG106:
bltz $3, TAG107
nor $2, $3, $3
sw $2, 0($3)
sb $2, 1($2)
TAG107:
sll $0, $0, 0
div $1, $1

```

```
blez $1, TAG108
divu $1, $1
TAG108:
sll $0, $0, 0
div $1, $1
mflo $3
div $1, $3
TAG109:
bltz $3, TAG110
mthi $3
multu $3, $3
blez $3, TAG110
TAG110:
mfhi $3
sltu $4, $3, $3
mtlo $4
srl $2, $3, 10
TAG111:
and $3, $2, $2
lbu $1, 0($2)
lui $2, 15
bne $3, $1, TAG112
TAG112:
addu $3, $2, $2
mthi $2
bgez $2, TAG113
sll $0, $0, 0
TAG113:
divu $3, $3
lui $4, 7
sll $0, $0, 0
lui $2, 8
TAG114:
multu $2, $2
sltiu $2, $2, 1
bne $2, $2, TAG115
mfhi $2
TAG115:
mfhi $2
lui $1, 13
mfhi $4
blez $2, TAG116
TAG116:
lb $3, 0($4)
mfhi $1
mflo $4
beq $4, $4, TAG117
TAG117:
ori $4, $4, 9
bgez $4, TAG118
lui $3, 14
mtlo $3
TAG118:
mtlo $3
mflo $1
subu $1, $1, $1
lui $1, 9
TAG119:
```

```

sll $0, $0, 0
addiu $2, $3, 3
mfhi $2
beq $2, $2, TAG120
TAG120:
mfhi $4
ori $2, $2, 12
mfhi $3
bgez $4, TAG121
TAG121:
mtlo $3
mthi $3
bltz $3, TAG122
mtlo $3
TAG122:
lui $1, 11
bne $3, $1, TAG123
lui $1, 7
mthi $3
TAG123:
bgez $1, TAG124
ori $2, $1, 1
lb $2, 0($2)
mtlo $2
TAG124:
bne $2, $2, TAG125
mthi $2
sll $0, $0, 0
or $4, $2, $2
TAG125:
blez $4, TAG126
sll $0, $0, 0
beq $4, $4, TAG126
srlv $4, $4, $4
TAG126:
mfhi $4
xor $4, $4, $4
nor $3, $4, $4
sb $3, 0($4)
TAG127:
mfhi $3
mfhi $2
lui $2, 7
sll $0, $0, 0
TAG128:
mfhi $1
mflo $4
bne $1, $3, TAG129
mflo $3
TAG129:
lhu $4, 0($3)
blez $4, TAG130
mfhi $2
sb $3, 0($4)
TAG130:
sllv $1, $2, $2
lui $3, 9
sll $0, $0, 0

```

```
bne $1, $2, TAG131
TAG131:
mfhi $1
lui $4, 13
addu $3, $4, $1
blez $1, TAG132
TAG132:
mflo $2
lui $3, 10
bltz $2, TAG133
mtlo $2
TAG133:
srlv $1, $3, $3
sll $0, $0, 0
sll $0, $0, 0
mflo $3
TAG134:
sh $3, 0($3)
bne $3, $3, TAG135
sw $3, 0($3)
bltz $3, TAG135
TAG135:
multu $3, $3
bltz $3, TAG136
multu $3, $3
lui $1, 0
TAG136:
sra $2, $1, 10
lbu $1, 0($2)
divu $2, $1
sb $1, 0($2)
TAG137:
div $1, $1
sltu $3, $1, $1
mtlo $1
beq $3, $3, TAG138
TAG138:
mfhi $3
blez $3, TAG139
lui $1, 12
lui $2, 2
TAG139:
lui $3, 6
lui $2, 9
sll $0, $0, 0
sra $4, $2, 8
TAG140:
mflo $3
sllv $3, $4, $4
lui $1, 2
lui $4, 0
TAG141:
lui $1, 5
xor $4, $4, $1
lui $2, 14
mthi $4
TAG142:
mthi $2
```

```

mthi $2
sll $0, $0, 0
bltz $2, TAG143
TAG143:
sll $0, $0, 0
mult $2, $2
sll $0, $0, 0
mthi $3
TAG144:
lui $3, 9
blez $3, TAG145
sra $4, $3, 9
sw $3, -1152($4)
TAG145:
lui $3, 4
sw $4, -1152($4)
sll $0, $0, 0
subu $3, $3, $2
TAG146:
bgez $3, TAG147
sll $0, $0, 0
lui $1, 15
sll $3, $1, 0
TAG147:
beq $3, $3, TAG148
xori $4, $3, 13
mfhi $1
bltz $3, TAG148
TAG148:
mflo $1
bne $1, $1, TAG149
addu $3, $1, $1
beq $1, $3, TAG149
TAG149:
srlv $4, $3, $3
subu $2, $3, $4
mflo $3
multu $3, $3
TAG150:
mtlo $3
multu $3, $3
xori $3, $3, 2
andi $2, $3, 8
TAG151:
mult $2, $2
mult $2, $2
bgtz $2, TAG152
mult $2, $2
TAG152:
or $2, $2, $2
lui $3, 2
sw $2, 0($2)
lui $2, 3
TAG153:
sltu $4, $2, $2
bne $4, $2, TAG154
slti $4, $2, 8
bne $2, $4, TAG154

```



```

TAG154:
srli $1, $4, 12
addu $3, $4, $1
lh $1, 0($1)
mfhi $4
TAG155:
bne $4, $4, TAG156
sh $4, 0($4)
sh $4, 0($4)
lh $4, 0($4)
TAG156:
sb $4, 0($4)
sw $4, 0($4)
mult $4, $4
sb $4, 0($4)
TAG157:
mthi $4
bgtz $4, TAG158
mflo $3
mtlo $3
TAG158:
slt $1, $3, $3
sra $3, $3, $3
beq $3, $1, TAG159
mflo $2
TAG159:
beq $2, $2, TAG160
lui $2, 14
div $2, $2
lui $4, 8
TAG160:
lui $3, 8
mthi $3
div $3, $3
bne $4, $4, TAG161
TAG161:
divu $3, $3
mtlo $3
sll $0, $0, 0
mtlo $4
TAG162:
blez $4, TAG163
multu $4, $4
mthi $4
divu $4, $4
TAG163:
mtlo $4
sw $4, 0($4)
multu $4, $4
mflo $4
TAG164:
lui $4, 2
bgtz $4, TAG165
lui $3, 13
xor $2, $4, $4
TAG165:
mtlo $2
bne $2, $2, TAG166

```

```
mflo $3
sll $0, $0, 0
TAG166:
mflo $2
lui $4, 4
mfhi $2
bgez $2, TAG167
TAG167:
lw $3, 0($2)
mthi $2
mthi $2
mult $3, $2
TAG168:
xori $2, $3, 13
lb $1, 0($3)
sb $1, 0($3)
multu $2, $1
TAG169:
lw $3, 0($1)
mthi $1
lb $4, 0($1)
sh $3, 0($1)
TAG170:
lui $3, 5
mflo $4
div $4, $3
beq $4, $4, TAG171
TAG171:
lui $1, 8
subu $3, $4, $4
lui $4, 9
lui $1, 0
TAG172:
lhu $2, 0($1)
lui $2, 15
sll $0, $0, 0
lui $1, 15
TAG173:
srl $3, $1, 11
sll $0, $0, 0
lb $2, -480($3)
blez $2, TAG174
TAG174:
sh $2, 0($2)
andi $1, $2, 7
andi $2, $2, 2
lui $3, 3
TAG175:
sll $0, $0, 0
mthi $3
divu $3, $3
sll $0, $0, 0
TAG176:
sll $0, $0, 0
sll $0, $0, 0
mthi $4
divu $4, $4
TAG177:
```

```
lui $1, 15
beq $1, $4, TAG178
sll $0, $0, 0
sll $0, $0, 0
TAG178:
mflo $3
mflo $3
mthi $3
bltz $3, TAG179
TAG179:
lbu $3, 0($3)
bltz $3, TAG180
lbu $3, 0($3)
bltz $3, TAG180
TAG180:
mflo $4
lui $1, 8
mfhi $1
bgez $3, TAG181
TAG181:
lb $2, 0($1)
beq $1, $2, TAG182
multu $1, $2
mfhi $1
TAG182:
mthi $1
multu $1, $1
sltiu $2, $1, 15
sra $3, $1, 10
TAG183:
mthi $3
mfhi $3
mtlo $3
sltiu $4, $3, 6
TAG184:
mfhi $2
lbu $4, 0($4)
mfhi $4
lui $2, 2
TAG185:
sll $0, $0, 0
mult $2, $4
addu $3, $4, $2
bne $3, $2, TAG186
TAG186:
sra $1, $3, $3
mthi $1
sll $0, $0, 0
bltz $3, TAG187
TAG187:
mtlo $1
andi $2, $1, 7
mtlo $2
beq $1, $1, TAG188
TAG188:
sll $2, $2, 1
lb $1, 0($2)
lb $3, 0($1)
```

```
sb $2, 0($2)
TAG189:
mthi $3
lhu $4, 0($3)
mflo $3
beq $4, $4, TAG190
TAG190:
srl $3, $3, 3
mult $3, $3
mthi $3
bgez $3, TAG191
TAG191:
sb $3, 0($3)
beq $3, $3, TAG192
sllv $4, $3, $3
mult $3, $3
TAG192:
lw $4, 0($4)
sb $4, 0($4)
mult $4, $4
sb $4, 0($4)
TAG193:
mfhi $4
blez $4, TAG194
lh $4, 0($4)
lui $3, 9
TAG194:
mult $3, $3
mflo $4
blez $3, TAG195
mthi $3
TAG195:
beq $4, $4, TAG196
xori $2, $4, 6
bne $4, $4, TAG196
slt $1, $4, $4
TAG196:
bne $1, $1, TAG197
sb $1, 0($1)
multu $1, $1
sw $1, 0($1)
TAG197:
multu $1, $1
sb $1, 0($1)
mflo $1
sw $1, 0($1)
TAG198:
and $1, $1, $1
sh $1, 0($1)
lbu $1, 0($1)
multu $1, $1
TAG199:
bltz $1, TAG200
mult $1, $1
multu $1, $1
multu $1, $1
TAG200:
bgtz $1, TAG201
```

```

ori $2, $1, 5
blez $2, TAG201
lui $1, 11
TAG201:
mthi $1
nor $4, $1, $1
bltz $4, TAG202
sll $0, $0, 0
TAG202:
mflo $1
lbu $1, 0($1)
mthi $1
bltz $1, TAG203
TAG203:
mfhi $4
mfhi $1
sb $4, 0($1)
sltu $3, $1, $1
TAG204:
lhu $3, 0($3)
lhu $3, 0($3)
lui $4, 2
sra $1, $3, 10
TAG205:
mult $1, $1
and $2, $1, $1
beq $1, $1, TAG206
sw $2, 0($1)
TAG206:
blez $2, TAG207
mtlo $2
lui $1, 6
lhu $4, 0($1)
TAG207:
multu $4, $4
mtlo $4
bne $4, $4, TAG208
sll $0, $0, 0
TAG208:
divu $4, $4
mfhi $4
mfhi $2
sllv $1, $2, $4
TAG209:
mthi $1
add $4, $1, $1
lh $3, 0($1)
srlv $4, $3, $1
TAG210:
lui $1, 5
sll $0, $0, 0
lui $2, 6
mtlo $2
TAG211:
beq $2, $2, TAG212
sll $0, $0, 0
lb $2, 0($3)
sb $3, 0($2)

```

```

TAG212:
mflo $1
bne $1, $2, TAG213
sll $0, $0, 0
lhu $4, 0($3)
TAG213:
multu $4, $4
lui $4, 8
mflo $4
bne $4, $4, TAG214
TAG214:
lui $4, 12
mflo $3
sb $4, 0($3)
mfhi $2
TAG215:
bne $2, $2, TAG216
multu $2, $2
mthi $2
sb $2, 0($2)
TAG216:
mfhi $1
bne $1, $1, TAG217
lw $4, 0($1)
bgez $1, TAG217
TAG217:
lw $2, 0($4)
mult $4, $2
lui $3, 11
mfhi $3
TAG218:
blez $3, TAG219
mthi $3
lw $3, 0($3)
srl $3, $3, 7
TAG219:
mthi $3
lui $1, 8
lui $1, 15
mtlo $3
TAG220:
mflo $2
mtlo $1
bne $2, $2, TAG221
sll $0, $0, 0
TAG221:
sw $3, 0($3)
bne $3, $3, TAG222
sra $1, $3, 0
blez $1, TAG222
TAG222:
sll $3, $1, 13
and $2, $3, $3
mflo $1
sll $0, $0, 0
TAG223:
sll $0, $0, 0
lui $4, 7

```

```
sra $4, $1, 0
or $4, $1, $4
TAG224:
addiu $4, $4, 15
mthi $4
div $4, $4
addiu $2, $4, 1
TAG225:
sll $0, $0, 0
mthi $1
sll $0, $0, 0
sll $0, $0, 0
TAG226:
bgez $2, TAG227
xori $1, $2, 0
lhu $3, 0($1)
sb $3, 0($1)
TAG227:
mult $3, $3
lb $4, 0($3)
mthi $3
mult $3, $3
TAG228:
mflo $3
lui $1, 2
sh $4, 0($4)
div $1, $1
TAG229:
mthi $1
sll $0, $0, 0
beq $1, $1, TAG230
mthi $1
TAG230:
sll $0, $0, 0
lui $2, 11
sll $2, $2, 5
mfhi $2
TAG231:
lui $4, 7
subu $4, $2, $2
mthi $4
mflo $2
TAG232:
sltiu $2, $2, 14
addiu $2, $2, 13
lhu $1, 0($2)
lb $4, 0($2)
TAG233:
multu $4, $4
slti $4, $4, 11
lbu $3, 0($4)
mthi $4
TAG234:
lhu $2, 0($3)
mfhi $2
lui $2, 1
lui $4, 9
TAG235:
```

```

sll $0, $0, 0
bltz $4, TAG236
sll $0, $0, 0
mflo $4
TAG236:
sh $4, 0($4)
lui $2, 13
mflo $1
lui $4, 12
TAG237:
mfhi $2
lui $3, 11
sll $0, $0, 0
mtlo $4
TAG238:
mtlo $3
sll $0, $0, 0
multu $3, $3
sll $0, $0, 0
TAG239:
mthi $1
beq $1, $1, TAG240
or $4, $1, $1
lb $4, 0($4)
TAG240:
multu $4, $4
bltz $4, TAG241
sh $4, 0($4)
lui $4, 0
TAG241:
andi $2, $4, 2
addu $2, $4, $4
bgtz $4, TAG242
sb $2, 0($2)
TAG242:
bne $2, $2, TAG243
add $4, $2, $2
mthi $2
mfhi $2
TAG243:
lui $2, 5
mthi $2
mflo $2
multu $2, $2
TAG244:
bne $2, $2, TAG245
sra $3, $2, 4
mtlo $2
mult $2, $3
TAG245:
beq $3, $3, TAG246
lh $1, 0($3)
mult $1, $3
divu $3, $3
TAG246:
bgez $1, TAG247
mult $1, $1
multu $1, $1

```



```
mthi $1
TAG247:
ori $2, $1, 9
lb $4, 0($2)
sb $2, 0($1)
bltz $2, TAG248
TAG248:
lhu $4, 0($4)
sb $4, 0($4)
sltiu $2, $4, 0
sb $4, 0($4)
TAG249:
mflo $1
slti $2, $2, 1
lbu $4, 0($2)
mult $2, $1
TAG250:
addiu $1, $4, 8
mthi $4
multu $4, $4
mfhi $4
TAG251:
mtlo $4
sltu $1, $4, $4
blez $4, TAG252
mfhi $3
TAG252:
sh $3, 0($3)
lui $4, 7
mfhi $4
lw $3, 0($4)
TAG253:
sh $3, 0($3)
beq $3, $3, TAG254
mfhi $2
sw $3, 0($3)
TAG254:
srav $2, $2, $2
blez $2, TAG255
subu $3, $2, $2
beq $3, $2, TAG255
TAG255:
and $1, $3, $3
mtlo $3
lui $2, 3
lui $4, 8
TAG256:
xori $2, $4, 15
bltz $2, TAG257
sll $0, $0, 0
mtlo $2
TAG257:
mult $2, $2
divu $2, $2
sll $0, $0, 0
bne $1, $1, TAG258
TAG258:
add $3, $1, $1
```

```
mult $3, $3
lb $3, 0($1)
mflo $1
TAG259:
multu $1, $1
lui $3, 15
mfhi $4
div $1, $3
TAG260:
and $4, $4, $4
mflo $2
bgez $4, TAG261
addi $4, $4, 11
TAG261:
mthi $4
bltz $4, TAG262
slt $2, $4, $4
mflo $1
TAG262:
mflo $4
lbu $3, 0($1)
xor $3, $3, $4
slti $4, $4, 5
TAG263:
sb $4, 0($4)
bltz $4, TAG264
mthi $4
addu $3, $4, $4
TAG264:
beq $3, $3, TAG265
slt $2, $3, $3
lhu $3, 0($3)
bne $2, $3, TAG265
TAG265:
lbu $3, 0($3)
mult $3, $3
sb $3, 0($3)
lui $4, 9
TAG266:
sll $0, $0, 0
mfhi $1
and $2, $4, $1
bgtz $4, TAG267
TAG267:
multu $2, $2
mthi $2
beq $2, $2, TAG268
sh $2, 0($2)
TAG268:
sb $2, 0($2)
sh $2, 0($2)
lui $3, 1
mthi $2
TAG269:
mfhi $1
beq $3, $3, TAG270
mthi $1
sw $1, 0($1)
```

```

TAG270:
beq $1, $1, TAG271
sltiu $2, $1, 1
lui $1, 12
beq $1, $1, TAG271
TAG271:
sw $1, 0($1)
multu $1, $1
sh $1, 0($1)
addiu $3, $1, 12
TAG272:
slt $2, $3, $3
lui $4, 15
slt $1, $4, $3
lui $4, 4
TAG273:
addu $4, $4, $4
mflo $4
beq $4, $4, TAG274
subu $4, $4, $4
TAG274:
srl $1, $4, 8
mfhi $4
beq $4, $4, TAG275
sb $4, 0($4)
TAG275:
bltz $4, TAG276
lb $1, 0($4)
multu $1, $4
bne $1, $4, TAG276
TAG276:
mfhi $3
sub $3, $1, $1
mthi $1
mfhi $3
TAG277:
lbu $3, 0($3)
lui $3, 0
multu $3, $3
sb $3, 0($3)
TAG278:
mult $3, $3
bne $3, $3, TAG279
addiu $2, $3, 9
mflo $1
TAG279:
slti $2, $1, 12
sb $2, 0($2)
lui $4, 0
lui $4, 13
TAG280:
sll $0, $0, 0
sltiu $3, $3, 14
mfhi $3
slt $3, $3, $3
TAG281:
sll $2, $3, 12
lbu $3, 0($2)

```

```
and $3, $3, $3
sr1 $3, $3, 14
TAG282:
addi $1, $3, 6
lui $2, 10
sw $3, 0($3)
lui $3, 2
TAG283:
mflo $1
bne $3, $3, TAG284
mfhi $1
lui $2, 3
TAG284:
divu $2, $2
mflo $2
bne $2, $2, TAG285
mtlo $2
TAG285:
xori $2, $2, 7
bgez $2, TAG286
sltiu $3, $2, 14
lbu $4, 0($3)
TAG286:
mtlo $4
mtlo $4
mfhi $2
mflo $2
TAG287:
sll $0, $0, 0
mflo $2
mtlo $2
sll $0, $0, 0
TAG288:
sll $0, $0, 0
bgtz $4, TAG289
sll $0, $0, 0
addi $2, $4, 11
TAG289:
sll $0, $0, 0
bgtz $2, TAG290
sll $0, $0, 0
lhu $4, 0($2)
TAG290:
lui $3, 5
div $3, $4
beq $3, $3, TAG291
mtlo $4
TAG291:
lui $3, 3
sll $0, $0, 0
bltz $3, TAG292
ori $3, $3, 10
TAG292:
bltz $3, TAG293
lui $1, 11
sll $0, $0, 0
multu $1, $1
TAG293:
```

```

srlv $3, $1, $1
divu $1, $1
lui $3, 8
beq $3, $3, TAG294
TAG294:
sllv $2, $3, $3
mflo $3
beq $3, $3, TAG295
lb $3, 0($3)
TAG295:
bgez $3, TAG296
mult $3, $3
bne $3, $3, TAG296
lb $4, 0($3)
TAG296:
lui $3, 8
mtlo $3
or $2, $3, $4
srlv $3, $2, $2
TAG297:
ori $3, $3, 1
lui $1, 14
bne $1, $1, TAG298
subu $2, $3, $1
TAG298:
bne $2, $2, TAG299
sllv $2, $2, $2
sltu $3, $2, $2
sll $0, $0, 0
TAG299:
mult $3, $3
mtlo $3
andi $3, $3, 13
slti $4, $3, 13
TAG300:
bgtz $4, TAG301
lbu $4, 0($4)
bgtz $4, TAG301
lbu $2, 0($4)
TAG301:
addu $3, $2, $2
mflo $2
sll $0, $0, 0
beq $2, $3, TAG302
TAG302:
mult $2, $2
bltz $2, TAG303
add $1, $2, $2
lw $1, 0($1)
TAG303:
mtlo $1
mtlo $1
lui $4, 14
sll $0, $0, 0
TAG304:
sll $0, $0, 0
bgtz $3, TAG305
lui $4, 2

```

```

mflo $4
TAG305:
lw $2, 0($4)
xor $2, $2, $2
lui $1, 8
mult $4, $4
TAG306:
multu $1, $1
sll $0, $0, 0
lui $3, 7
bltz $1, TAG307
TAG307:
lui $4, 2
lui $1, 13
sll $0, $0, 0
bltz $1, TAG308
TAG308:
mthi $1
sll $0, $0, 0
sll $0, $0, 0
mthi $1
TAG309:
lui $4, 3
divu $1, $1
sll $0, $0, 0
divu $4, $4
TAG310:
bne $4, $4, TAG311
srl $2, $4, 3
divu $4, $2
sw $2, -24576($2)
TAG311:
bgez $2, TAG312
lw $4, -24576($2)
multu $4, $2
sw $2, 0($4)
TAG312:
bltz $4, TAG313
slt $4, $4, $4
bgtz $4, TAG313
or $1, $4, $4
TAG313:
lui $3, 15
mthi $3
beq $1, $3, TAG314
lhu $4, 0($1)
TAG314:
subu $3, $4, $4
lb $2, -24576($4)
multu $4, $2
mtlo $3
TAG315:
lui $1, 6
mtlo $2
andi $2, $2, 14
lui $3, 8
TAG316:
beq $3, $3, TAG317

```

```
div $3, $3
lui $2, 11
bltz $3, TAG317
TAG317:
nor $1, $2, $2
srl $1, $2, 10
sb $1, 0($1)
lh $1, 0($1)
TAG318:
bne $1, $1, TAG319
nor $2, $1, $1
bne $1, $1, TAG319
andi $4, $2, 12
TAG319:
sw $4, 0($4)
xor $1, $4, $4
lbu $1, 0($4)
sb $1, 0($4)
TAG320:
divu $1, $1
lui $4, 4
divu $4, $4
bltz $1, TAG321
TAG321:
mflo $4
sb $4, 0($4)
srlv $1, $4, $4
sw $1, 0($1)
TAG322:
mtlo $1
bne $1, $1, TAG323
slti $3, $1, 9
lbu $4, 0($1)
TAG323:
lw $2, 0($4)
multu $4, $4
mfhi $3
mflo $3
TAG324:
bgtz $3, TAG325
multu $3, $3
multu $3, $3
lh $1, 0($3)
TAG325:
lb $1, 0($1)
mthi $1
mflo $2
bgez $2, TAG326
TAG326:
xor $4, $2, $2
bne $4, $2, TAG327
lbu $4, 0($4)
mtlo $4
TAG327:
mflo $2
bgez $4, TAG328
addiu $1, $2, 3
blez $2, TAG328
```

```
TAG328:
subu $2, $1, $1
andi $2, $2, 14
lui $2, 13
bgtz $1, TAG329
TAG329:
mflo $3
sub $2, $2, $3
mfhi $3
sh $2, 0($3)
TAG330:
multu $3, $3
beq $3, $3, TAG331
sh $3, 0($3)
mflo $3
TAG331:
sh $3, 0($3)
bltz $3, TAG332
sltiu $1, $3, 3
blez $3, TAG332
TAG332:
lui $3, 9
bltz $3, TAG333
sll $0, $0, 0
blez $3, TAG333
TAG333:
sll $0, $0, 0
lb $3, 0($4)
sra $1, $3, $3
lb $1, 0($3)
TAG334:
lh $3, 0($1)
mfhi $1
mult $1, $1
mult $3, $1
TAG335:
mult $1, $1
mtlo $1
mfhi $4
bgtz $4, TAG336
TAG336:
sw $4, 0($4)
mtlo $4
mult $4, $4
mflo $1
TAG337:
bgez $1, TAG338
sll $2, $1, 11
mflo $3
sb $2, 0($1)
TAG338:
bltz $3, TAG339
lui $1, 2
xori $2, $3, 5
slti $3, $3, 8
TAG339:
sb $3, 0($3)
xor $2, $3, $3
```



```

srl $2, $3, 3
addi $3, $2, 14
TAG340:
or $2, $3, $3
lui $4, 6
mthi $3
addiu $2, $3, 8
TAG341:
mult $2, $2
mthi $2
lbu $3, 0($2)
bgtz $2, TAG342
TAG342:
mflo $3
srl $3, $3, 13
bltz $3, TAG343
sh $3, 0($3)
TAG343:
beq $3, $3, TAG344
lui $1, 7
sb $3, 0($1)
lui $2, 1
TAG344:
mtlo $2
lui $4, 13
addiu $4, $2, 8
sllv $2, $2, $4
TAG345:
bne $2, $2, TAG346
multu $2, $2
blez $2, TAG346
mthi $2
TAG346:
mflo $3
mthi $3
addiu $4, $2, 5
multu $2, $3
TAG347:
sltiu $1, $4, 1
mflo $2
addu $4, $1, $1
mfhi $2
TAG348:
lui $4, 6
bltz $2, TAG349
nor $1, $2, $4
and $4, $1, $4
TAG349:
beq $4, $4, TAG350
nor $1, $4, $4
and $2, $1, $1
lui $1, 2
TAG350:
bne $1, $1, TAG351
div $1, $1
blez $1, TAG351
lui $2, 0
TAG351:

```

```

mflo $1
lui $1, 5
blez $2, TAG352
sll $0, $0, 0
TAG352:
mult $3, $3
mthi $3
lui $4, 15
lui $1, 8
TAG353:
beq $1, $1, TAG354
divu $1, $1
div $1, $1
div $1, $1
TAG354:
subu $3, $1, $1
blez $3, TAG355
mflo $4
beq $3, $1, TAG355
TAG355:
mult $4, $4
lb $4, 0($4)
mflo $2
lb $4, 0($2)
TAG356:
bne $4, $4, TAG357
mflo $3
lbu $2, 0($3)
lb $2, 0($3)
TAG357:
bne $2, $2, TAG358
lui $4, 2
sh $4, 0($2)
sb $4, 0($2)
TAG358:
addiu $1, $4, 11
mthi $1
multu $1, $1
sll $0, $0, 0
TAG359:
addiu $3, $1, 15
sra $2, $3, 10
sll $0, $0, 0
mfhi $4
TAG360:
or $4, $4, $4
nor $2, $4, $4
ori $3, $2, 5
mthi $4
TAG361:
addu $3, $3, $3
mflo $3
sll $0, $0, 0
mflo $2
TAG362:
xor $4, $2, $2
sh $2, 0($4)
sw $2, 0($4)

```

```

lui $4, 9
TAG363:
srl $2, $4, 1
sltu $1, $4, $4
divu $4, $4
multu $2, $1
TAG364:
mtlo $1
bltz $1, TAG365
slti $2, $1, 8
bne $1, $2, TAG365
TAG365:
sltu $2, $2, $2
slti $2, $2, 4
sb $2, 0($2)
divu $2, $2
TAG366:
beq $2, $2, TAG367
sb $2, 0($2)
sra $2, $2, 11
blez $2, TAG367
TAG367:
addu $3, $2, $2
bgtz $2, TAG368
lb $2, 0($2)
mflo $1
TAG368:
mtlo $1
bgtz $1, TAG369
lui $1, 7
lui $4, 2
TAG369:
lui $2, 2
lui $3, 15
sll $0, $0, 0
sll $0, $0, 0
TAG370:
ori $3, $4, 6
multu $3, $4
addiu $3, $4, 7
mult $3, $4
TAG371:
bgez $3, TAG372
sll $0, $0, 0
lui $3, 3
and $2, $3, $3
TAG372:
mfhi $1
bne $2, $2, TAG373
mfhi $4
subu $2, $2, $1
TAG373:
sll $0, $0, 0
mflo $1
sltu $4, $2, $2
addiu $4, $1, 9
TAG374:
addiu $4, $4, 12

```

```

bne $4, $4, TAG375
sll $0, $0, 0
xor $2, $4, $4
TAG375:
blez $2, TAG376
andi $4, $2, 14
srav $1, $4, $4
divu $1, $1
TAG376:
sll $0, $0, 0
multu $1, $3
lui $4, 7
divu $1, $4
TAG377:
div $4, $4
lui $1, 11
div $1, $1
bltz $4, TAG378
TAG378:
srav $2, $1, $1
mflo $3
mthi $1
lui $3, 13
TAG379:
beq $3, $3, TAG380
mult $3, $3
bgtz $3, TAG380
sub $2, $3, $3
TAG380:
sll $0, $0, 0
mflo $1
sltu $3, $2, $1
sub $2, $1, $3
TAG381:
sb $2, 0($2)
mflo $4
mtlo $4
mthi $2
TAG382:
sw $4, 0($4)
sh $4, 0($4)
lui $4, 1
lui $3, 7
TAG383:
sll $0, $0, 0
andi $4, $3, 14
bltz $4, TAG384
subu $3, $4, $3
TAG384:
sll $0, $0, 0
bne $3, $3, TAG385
mtlo $3
slt $2, $3, $3
TAG385:
lui $3, 15
sll $0, $0, 0
bltz $2, TAG386
mtlo $2

```

```

TAG386:
lw $3, 0($2)
mtlo $2
sb $2, 0($3)
sltu $3, $2, $2
TAG387:
mthi $3
mthi $3
bne $3, $3, TAG388
andi $2, $3, 15
TAG388:
sh $2, 0($2)
srl $1, $2, 2
bltz $2, TAG389
sllv $1, $2, $1
TAG389:
sra $2, $1, 3
lw $1, 0($1)
beq $2, $1, TAG390
or $4, $1, $2
TAG390:
bltz $4, TAG391
slti $4, $4, 8
sra $1, $4, 6
bgez $4, TAG391
TAG391:
mult $1, $1
sh $1, 0($1)
bne $1, $1, TAG392
sh $1, 0($1)
TAG392:
sll $1, $1, 5
bne $1, $1, TAG393
lui $4, 4
lh $1, 0($1)
TAG393:
lui $3, 5
mfhi $2
lbu $4, 0($1)
sb $4, 0($1)
TAG394:
sltiu $4, $4, 15
bne $4, $4, TAG395
lbu $2, 0($4)
mult $2, $4
TAG395:
mflo $3
sw $3, 0($3)
subu $1, $2, $3
sh $3, 0($1)
TAG396:
beq $1, $1, TAG397
lui $3, 5
lbu $3, 0($1)
slt $3, $3, $1
TAG397:
nor $4, $3, $3
blez $4, TAG398

```

```

sll $4, $4, 8
mult $4, $3
TAG398:
subu $3, $4, $4
mflo $4
bne $4, $4, TAG399
add $2, $4, $4
TAG399:
beq $2, $2, TAG400
mtlo $2
add $1, $2, $2
blez $2, TAG400
TAG400:
multu $1, $1
lbu $3, 0($1)
bne $1, $3, TAG401
lh $4, 0($1)
TAG401:
mflo $1
sub $2, $4, $1
multu $1, $1
sh $2, 0($1)
TAG402:
mtlo $2
sh $2, 0($2)
sb $2, 0($2)
mflo $2
TAG403:
srl $2, $2, 11
multu $2, $2
bne $2, $2, TAG404
mflo $3
TAG404:
sub $4, $3, $3
bne $4, $3, TAG405
mflo $2
sra $1, $2, 1
TAG405:
lbu $3, 0($1)
sb $1, 0($1)
lui $3, 14
bne $1, $3, TAG406
TAG406:
mflo $2
sll $0, $0, 0
beq $3, $3, TAG407
lui $1, 15
TAG407:
sll $0, $0, 0
sll $0, $0, 0
slt $4, $3, $3
sra $2, $3, 1
TAG408:
addu $3, $2, $2
multu $3, $3
mflo $4
mthi $4
TAG409:

```

```
addi $1, $4, 13
bgtz $1, TAG410
slti $2, $4, 11
addu $1, $2, $1
TAG410:
lui $1, 7
subu $2, $1, $1
div $1, $1
sllv $1, $1, $1
TAG411:
srl $4, $1, 2
bgtz $1, TAG412
sll $0, $0, 0
mflo $4
TAG412:
divu $4, $4
sll $0, $0, 0
mtlo $4
bgtz $4, TAG413
TAG413:
sll $0, $0, 0
beq $4, $4, TAG414
sll $0, $0, 0
beq $4, $4, TAG414
TAG414:
sll $0, $0, 0
beq $4, $4, TAG415
xori $2, $4, 6
mult $4, $4
TAG415:
mult $2, $2
ori $2, $2, 4
sll $0, $0, 0
blez $2, TAG416
TAG416:
mfhi $4
beq $4, $2, TAG417
sll $0, $0, 0
mfhi $2
TAG417:
mflo $4
bne $2, $2, TAG418
srl $3, $2, 15
lui $4, 10
TAG418:
lui $1, 14
sll $0, $0, 0
mult $1, $2
ori $1, $2, 4
TAG419:
mtlo $1
beq $1, $1, TAG420
mfhi $2
lui $4, 2
TAG420:
bltz $4, TAG421
mult $4, $4
and $2, $4, $4
```

```

mfhi $4
TAG421:
divu $4, $4
bgez $4, TAG422
sw $4, 0($4)
sll $4, $4, 12
TAG422:
mtlo $4
subu $4, $4, $4
mfhi $4
addu $1, $4, $4
TAG423:
lbu $1, 0($1)
blez $1, TAG424
slti $4, $1, 15
bne $1, $1, TAG424
TAG424:
sb $4, 0($4)
mflo $4
sra $1, $4, 1
sw $4, 0($4)
TAG425:
beq $1, $1, TAG426
sh $1, 0($1)
lb $1, 0($1)
bne $1, $1, TAG426
TAG426:
slti $3, $1, 3
andi $1, $3, 8
lui $2, 10
xor $1, $1, $2
TAG427:
mflo $4
sltiu $1, $4, 4
addu $1, $4, $1
lbu $1, 0($1)
TAG428:
divu $1, $1
mfhi $2
beq $2, $1, TAG429
div $1, $1
TAG429:
lb $3, 0($2)
lui $2, 9
sll $0, $0, 0
bltz $2, TAG430
TAG430:
sllv $2, $4, $4
bgtz $4, TAG431
addiu $4, $4, 2
lhu $4, 0($4)
TAG431:
lui $1, 11
sll $0, $0, 0
lui $1, 11
subu $1, $4, $4
TAG432:
mtlo $1

```



```

lb $1, 0($1)
or $3, $1, $1
srl $4, $1, 10
TAG433:
lui $3, 3
mflo $4
mthi $4
blez $4, TAG434
TAG434:
sltiu $1, $4, 10
bne $4, $4, TAG435
sra $3, $1, 13
sb $4, 0($1)
TAG435:
bgtz $3, TAG436
sh $3, 0($3)
lw $3, 0($3)
lui $2, 3
TAG436:
bgtz $2, TAG437
lui $4, 4
blez $2, TAG437
mtlo $2
TAG437:
sll $0, $0, 0
lui $4, 11
lui $1, 4
sll $0, $0, 0
TAG438:
div $1, $1
sll $0, $0, 0
ori $4, $1, 3
bltz $4, TAG439
TAG439:
mthi $4
srav $1, $4, $4
beq $1, $1, TAG440
mult $4, $1
TAG440:
addiu $3, $1, 3
sll $0, $0, 0
sll $0, $0, 0
lui $4, 4
TAG441:
xori $1, $4, 7
mflo $1
srl $3, $1, 3
mfhi $3
TAG442:
lui $1, 1
mflo $1
sh $3, 0($3)
div $1, $1
TAG443:
bltz $1, TAG444
mult $1, $1
blez $1, TAG444
subu $2, $1, $1

```

```
TAG444:
srlv $1, $2, $2
mflo $2
mfhi $1
lui $4, 13
TAG445:
bgtz $4, TAG446
lui $1, 11
lui $2, 12
lbu $4, 0($4)
TAG446:
subu $1, $4, $4
mflo $1
mthi $4
mfhi $1
TAG447:
div $1, $1
nor $1, $1, $1
sll $0, $0, 0
bne $2, $2, TAG448
TAG448:
sll $0, $0, 0
srl $4, $2, 2
bne $4, $2, TAG449
mfhi $3
TAG449:
mfhi $1
mtlo $3
mfhi $2
sh $2, 0($2)
TAG450:
beq $2, $2, TAG451
sh $2, 0($2)
sltu $3, $2, $2
bgez $3, TAG451
TAG451:
sb $3, 0($3)
blez $3, TAG452
add $1, $3, $3
bgez $3, TAG452
TAG452:
multu $1, $1
beq $1, $1, TAG453
lui $3, 0
mthi $3
TAG453:
multu $3, $3
lui $1, 14
mult $1, $1
or $1, $3, $3
TAG454:
lui $3, 3
sll $0, $0, 0
multu $1, $3
multu $4, $4
TAG455:
mflo $4
mfhi $1
```

```

mfhi $2
addu $4, $2, $4
TAG456:
bne $4, $4, TAG457
lui $3, 15
bne $4, $3, TAG457
sll $0, $0, 0
TAG457:
bne $3, $3, TAG458
sll $0, $0, 0
sll $0, $0, 0
mthi $3
TAG458:
sllv $2, $1, $1
sll $0, $0, 0
bne $4, $2, TAG459
sra $1, $4, $4
TAG459:
sll $0, $0, 0
sll $0, $0, 0
bgtz $1, TAG460
sll $0, $0, 0
TAG460:
divu $4, $4
sll $0, $0, 0
blez $3, TAG461
mfhi $1
TAG461:
bne $1, $1, TAG462
lui $1, 0
and $2, $1, $1
lui $1, 13
TAG462:
mult $1, $1
slt $2, $1, $1
mflo $4
sh $4, 0($2)
TAG463:
bne $4, $4, TAG464
mtlo $4
mtlo $4
sub $1, $4, $4
TAG464:
bgez $1, TAG465
mult $1, $1
lui $3, 7
bgtz $3, TAG465
TAG465:
mtlo $3
sll $0, $0, 0
mtlo $3
sll $0, $0, 0
TAG466:
bltz $3, TAG467
sll $0, $0, 0
lbu $2, 0($4)
lui $4, 10
TAG467:

```

```

sll $0, $0, 0
mflo $2
beq $3, $2, TAG468
multu $3, $2
TAG468:
lui $4, 10
ori $2, $4, 11
sll $0, $0, 0
subu $3, $2, $2
TAG469:
lui $1, 2
mflo $1
sh $1, 0($3)
mult $1, $1
TAG470:
lb $3, 0($1)
mult $1, $1
beq $3, $3, TAG471
mtlo $1
TAG471:
beq $3, $3, TAG472
mfhi $3
beq $3, $3, TAG472
lbu $1, 0($3)
TAG472:
sh $1, 0($1)
addi $4, $1, 15
mflo $4
lui $4, 10
TAG473:
slt $4, $4, $4
mflo $2
bne $4, $4, TAG474
mult $2, $4
TAG474:
mflo $3
beq $3, $3, TAG475
sw $3, 0($2)
sh $3, 0($2)
TAG475:
and $2, $3, $3
lui $2, 1
mthi $2
sll $0, $0, 0
TAG476:
bgtz $4, TAG477
srl $1, $4, 8
bltz $4, TAG477
mtlo $1
TAG477:
sb $1, 0($1)
sub $2, $1, $1
sb $1, 0($2)
mflo $3
TAG478:
bgez $3, TAG479
sb $3, 0($3)
sh $3, 0($3)

```

```
lui $2, 13
TAG479:
multu $2, $2
lui $2, 5
sll $0, $0, 0
bne $2, $2, TAG480
TAG480:
mflo $1
bne $1, $2, TAG481
mtlo $2
blez $1, TAG481
TAG481:
mtlo $1
bne $1, $1, TAG482
sw $1, 0($1)
sw $1, 0($1)
TAG482:
mult $1, $1
mfhi $1
sltu $4, $1, $1
mfhi $3
TAG483:
xori $4, $3, 4
addu $1, $4, $3
bne $3, $1, TAG484
sh $3, 0($1)
TAG484:
slt $4, $1, $1
lh $1, 0($4)
lui $1, 7
lui $4, 2
TAG485:
bltz $4, TAG486
sll $0, $0, 0
sll $0, $0, 0
lui $4, 7
TAG486:
bne $4, $4, TAG487
lui $2, 1
and $1, $2, $4
mfhi $2
TAG487:
lui $1, 3
mfhi $1
bltz $2, TAG488
mfhi $4
TAG488:
lui $1, 6
mflo $4
add $3, $4, $4
lui $3, 11
TAG489:
slti $3, $3, 9
mfhi $2
lw $1, 0($3)
mult $1, $3
TAG490:
mthi $1
```

```

addi $1, $1, 14
divu $1, $1
lh $1, 0($1)
TAG491:
or $4, $1, $1
lh $4, 0($1)
sra $4, $1, 12
srlv $1, $1, $1
TAG492:
sb $1, 0($1)
mflo $1
mult $1, $1
lui $1, 4
TAG493:
mfhi $1
addi $3, $1, 2
sw $3, 0($1)
ori $1, $1, 12
TAG494:
sll $4, $1, 6
bltz $1, TAG495
sra $4, $4, 13
bltz $1, TAG495
TAG495:
lh $4, 0($4)
srlv $4, $4, $4
multu $4, $4
mthi $4
TAG496:
add $4, $4, $4
mtlo $4
lui $3, 9
lbu $1, 0($4)
TAG497:
mfhi $1
bne $1, $1, TAG498
lh $3, 0($1)
sw $3, 0($1)
TAG498:
sh $3, 0($3)
lb $4, 0($3)
lui $4, 12
mfhi $3
TAG499:
beq $3, $3, TAG500
mult $3, $3
mflo $4
mfhi $1
TAG500:
multu $1, $1
sltiu $3, $1, 5
sb $1, 0($3)
lb $3, 0($3)
TAG501:
mult $3, $3
mfhi $1
lw $2, 0($1)
mfhi $1

```

```
TAG502:
addiu $2, $1, 14
bltz $1, TAG503
sra $4, $1, 5
beq $4, $2, TAG503
TAG503:
sb $4, 0($4)
lbu $3, 0($4)
and $1, $3, $3
ori $2, $3, 0
TAG504:
bne $2, $2, TAG505
multu $2, $2
lui $2, 7
mfhi $1
TAG505:
slt $3, $1, $1
sltu $2, $1, $3
multu $1, $1
mult $1, $3
TAG506:
mthi $2
bne $2, $2, TAG507
lui $4, 0
bgez $2, TAG507
TAG507:
mtlo $4
beq $4, $4, TAG508
lhu $4, 0($4)
addi $4, $4, 7
TAG508:
blez $4, TAG509
sllv $4, $4, $4
mthi $4
addi $4, $4, 14
TAG509:
mflo $1
mfhi $4
sltu $2, $4, $4
addiu $4, $4, 4
TAG510:
lui $1, 10
sll $0, $0, 0
srl $1, $1, 14
bgez $4, TAG511
TAG511:
mtlo $1
bgtz $1, TAG512
slti $3, $1, 0
multu $1, $1
TAG512:
lb $1, 0($3)
multu $3, $1
multu $1, $1
subu $4, $1, $3
TAG513:
mthi $4
mult $4, $4
```

```
beq $4, $4, TAG514
lb $2, 0($4)
TAG514:
sh $2, 0($2)
bgtz $2, TAG515
mult $2, $2
lui $3, 6
TAG515:
blez $3, TAG516
multu $3, $3
mfhi $4
addu $4, $4, $3
TAG516:
beq $4, $4, TAG517
slt $2, $4, $4
mfhi $1
sltu $1, $1, $2
TAG517:
lui $2, 2
mflo $2
lh $2, 0($1)
bne $1, $1, TAG518
TAG518:
mult $2, $2
sltu $3, $2, $2
sub $2, $3, $2
multu $2, $2
TAG519:
mult $2, $2
mult $2, $2
mflo $4
lui $4, 10
TAG520:
sll $0, $0, 0
mthi $4
mtlo $4
addiu $2, $4, 9
TAG521:
lui $4, 14
xor $3, $2, $4
lui $4, 1
andi $4, $4, 0
TAG522:
mflo $3
multu $3, $4
slti $3, $3, 0
bne $3, $3, TAG523
TAG523:
sllv $2, $3, $3
lbu $3, 0($3)
mult $3, $3
sh $2, 0($3)
TAG524:
lui $1, 15
mthi $1
mult $1, $1
lw $1, 0($3)
TAG525:
```



```

mthi $1
bne $1, $1, TAG526
or $1, $1, $1
bltz $1, TAG526
TAG526:
lui $3, 1
mthi $3
bne $3, $3, TAG527
sltu $3, $3, $3
TAG527:
bltz $3, TAG528
lui $2, 13
lui $2, 15
bne $2, $3, TAG528
TAG528:
lui $1, 0
lui $1, 13
mflo $1
sh $1, 0($1)
TAG529:
mflo $3
multu $3, $1
xori $4, $1, 5
lb $3, 0($3)
TAG530:
addi $2, $3, 1
sb $3, 0($2)
sb $3, 0($2)
sb $3, 0($2)
TAG531:
bgez $2, TAG532
mflo $2
lb $3, 0($2)
xor $3, $3, $2
TAG532:
lb $1, 0($3)
lui $4, 6
lw $2, 0($1)
bne $2, $3, TAG533
TAG533:
mult $2, $2
blez $2, TAG534
srl $1, $2, 11
mfhi $4
TAG534:
lui $2, 6
bgtz $4, TAG535
mfhi $2
div $4, $2
TAG535:
mfhi $2
divu $2, $2
ori $2, $2, 11
sb $2, 0($2)
TAG536:
sra $3, $2, 13
mult $2, $2
lb $2, 0($2)

```

```

lui $4, 4
TAG537:
andi $4, $4, 11
sb $4, 0($4)
xori $1, $4, 2
lh $4, 0($4)
TAG538:
mthi $4
beq $4, $4, TAG539
mult $4, $4
mtlo $4
TAG539:
sltu $2, $4, $4
slti $1, $2, 8
lui $2, 7
sb $2, 0($4)
TAG540:
bne $2, $2, TAG541
lui $3, 15
beq $2, $2, TAG541
mfhi $3
TAG541:
bgtz $3, TAG542
lui $1, 6
lui $1, 12
blez $3, TAG542
TAG542:
subu $3, $1, $1
mult $3, $3
mult $3, $3
bgtz $3, TAG543
TAG543:
multu $3, $3
lh $4, 0($3)
sltiu $3, $4, 13
mflo $2
TAG544:
mtlo $2
sw $2, 0($2)
mtlo $2
bgez $2, TAG545
TAG545:
mflo $2
bgez $2, TAG546
sll $2, $2, 3
sltu $4, $2, $2
TAG546:
mthi $4
mtlo $4
multu $4, $4
mthi $4
TAG547:
sw $4, 0($4)
or $2, $4, $4
srl $1, $2, 4
lui $1, 6
TAG548:
bgtz $1, TAG549

```

```

sll $3, $1, 7
beq $1, $1, TAG549
mult $1, $3
TAG549:
multu $3, $3
sll $0, $0, 0
multu $1, $3
andi $4, $3, 3
TAG550:
sw $4, 0($4)
add $1, $4, $4
sra $3, $4, 14
mult $3, $3
TAG551:
sh $3, 0($3)
xori $2, $3, 10
mfhi $2
mflo $3
TAG552:
beq $3, $3, TAG553
mflo $3
mfhi $1
sra $4, $3, $3
TAG553:
addu $4, $4, $4
mflo $3
lh $2, 0($4)
bne $4, $2, TAG554
TAG554:
lhu $2, 0($2)
beq $2, $2, TAG555
multu $2, $2
lui $3, 13
TAG555:
bne $3, $3, TAG556
lui $3, 3
slt $3, $3, $3
srlv $2, $3, $3
TAG556:
srlv $3, $2, $2
mflo $3
bgtz $3, TAG557
lui $1, 6
TAG557:
div $1, $1
lui $1, 14
mflo $4
sll $0, $0, 0
TAG558:
bgtz $4, TAG559
mfhi $3
bgtz $4, TAG559
addu $2, $4, $3
TAG559:
srl $2, $2, 3
mthi $2
bgtz $2, TAG560
mthi $2

```

```

TAG560:
lhu $4, 0($2)
mult $4, $2
multu $2, $4
sra $3, $2, 10
TAG561:
and $1, $3, $3
lui $3, 13
mult $3, $1
slti $4, $3, 1
TAG562:
sh $4, 0($4)
sb $4, 0($4)
lui $4, 15
mthi $4
TAG563:
beq $4, $4, TAG564
sll $0, $0, 0
div $4, $4
div $4, $3
TAG564:
mtlo $3
sll $0, $0, 0
bgtz $3, TAG565
sll $0, $0, 0
TAG565:
sll $0, $0, 0
nor $4, $3, $4
blez $4, TAG566
srl $3, $3, 11
TAG566:
bne $3, $3, TAG567
addiu $2, $3, 8
mthi $3
or $1, $2, $3
TAG567:
mult $1, $1
sra $1, $1, 10
blez $1, TAG568
mthi $1
TAG568:
mthi $1
mult $1, $1
blez $1, TAG569
sh $1, 0($1)
TAG569:
lui $2, 8
mfhi $4
nor $3, $2, $4
lui $3, 9
TAG570:
sll $0, $0, 0
slti $1, $3, 7
sh $3, 0($4)
beq $4, $1, TAG571
TAG571:
mtlo $1
mtlo $1

```

```
lb $4, 0($1)
lui $4, 11
TAG572:
mflo $1
beq $4, $1, TAG573
slti $4, $4, 15
sw $4, 0($4)
TAG573:
mult $4, $4
sb $4, 0($4)
mult $4, $4
lhu $1, 0($4)
TAG574:
lbu $3, 0($1)
mthi $3
xor $2, $1, $1
ori $3, $2, 8
TAG575:
mthi $3
bgez $3, TAG576
addiu $1, $3, 13
lbu $2, 0($3)
TAG576:
lw $1, 0($2)
sw $1, 0($1)
mult $1, $2
lui $4, 1
TAG577:
bne $4, $4, TAG578
addu $2, $4, $4
nor $2, $2, $4
bltz $2, TAG578
TAG578:
sll $0, $0, 0
sll $0, $0, 0
sll $3, $2, 12
sll $0, $0, 0
TAG579:
multu $3, $3
mthi $3
divu $3, $3
sltiu $1, $3, 10
TAG580:
mult $1, $1
sh $1, 0($1)
mfhi $4
sra $1, $1, 9
TAG581:
mfhi $4
beq $1, $4, TAG582
slt $4, $4, $4
mfhi $4
TAG582:
mult $4, $4
mtlo $4
mflo $1
addi $3, $1, 4
TAG583:
```

```
beq $3, $3, TAG584
sra $2, $3, 12
multu $3, $2
slt $3, $3, $3
TAG584:
lh $1, 0($3)
blez $3, TAG585
multu $3, $3
bne $1, $3, TAG585
TAG585:
slti $2, $1, 4
mfhi $1
bltz $1, TAG586
div $1, $2
TAG586:
sll $4, $1, 11
sra $4, $4, $4
lw $1, 0($1)
mfhi $4
TAG587:
sll $3, $4, 7
and $2, $4, $4
andi $3, $4, 9
sb $3, 0($2)
TAG588:
mfhi $1
mfhi $2
mtlo $2
subu $2, $3, $1
TAG589:
lui $3, 5
lbu $3, 0($2)
sh $3, 0($3)
addiu $1, $3, 15
TAG590:
lbu $3, 0($1)
mthi $3
mult $1, $3
div $1, $3
TAG591:
sra $2, $3, $3
bne $2, $2, TAG592
lui $3, 8
lui $4, 7
TAG592:
mflo $4
beq $4, $4, TAG593
mthi $4
lb $3, 0($4)
TAG593:
bgez $3, TAG594
mfhi $1
mtlo $1
lhu $1, 0($3)
TAG594:
lui $4, 15
sllv $3, $4, $1
sll $0, $0, 0
```

```

sll $0, $0, 0
TAG595:
sh $2, 0($2)
addi $3, $2, 8
sb $3, 0($2)
mult $3, $2
TAG596:
mtlo $3
lui $1, 15
mthi $1
mtlo $3
TAG597:
mfhi $4
mfhi $3
mthi $4
mfhi $1
TAG598:
mflo $3
lui $2, 1
and $3, $1, $1
sll $0, $0, 0
TAG599:
lui $3, 3
blez $1, TAG600
lui $2, 13
sll $0, $0, 0
TAG600:
sll $0, $0, 0
mult $1, $1
multu $1, $1
srl $1, $1, 0
TAG601:
blez $1, TAG602
multu $1, $1
mtlo $1
div $1, $1
TAG602:
sll $0, $0, 0
sll $0, $0, 0
or $2, $2, $1
lui $2, 10
TAG603:
div $2, $2
mflo $1
lui $4, 12
lui $2, 14
TAG604:
mflo $3
sb $3, 0($3)
mfhi $2
beq $2, $2, TAG605
TAG605:
mtlo $2
mult $2, $2
sh $2, 0($2)
subu $3, $2, $2
TAG606:
mfhi $1

```

```
sh $3, 0($3)
lui $1, 14
ori $4, $1, 14
TAG607:
beq $4, $4, TAG608
div $4, $4
lbu $1, 0($4)
beq $1, $1, TAG608
TAG608:
divu $1, $1
sll $0, $0, 0
div $1, $1
mfhi $4
TAG609:
lbu $1, 0($4)
mult $4, $1
mult $1, $1
sw $1, 0($1)
TAG610:
sub $3, $1, $1
beq $1, $1, TAG611
slti $2, $1, 10
bne $2, $3, TAG611
TAG611:
lb $4, 0($2)
sh $2, 0($4)
bltz $4, TAG612
mtlo $2
TAG612:
andi $2, $4, 2
mflo $2
mflo $1
bgez $4, TAG613
TAG613:
mtlo $1
lui $4, 14
blez $4, TAG614
lui $4, 4
TAG614:
mflo $2
beq $2, $2, TAG615
mult $2, $2
lh $2, 0($2)
TAG615:
mtlo $2
div $2, $2
nor $4, $2, $2
mfhi $2
TAG616:
lui $1, 7
sh $1, 0($2)
mtlo $1
sra $2, $1, $1
TAG617:
mfhi $2
srl $4, $2, 14
lhu $3, 0($2)
lui $1, 3
```



```

TAG618:
blez $1, TAG619
sll $0, $0, 0
blez $1, TAG619
sll $0, $0, 0
TAG619:
xori $4, $2, 15
bne $4, $4, TAG620
lbu $1, 0($4)
mthi $4
TAG620:
lb $4, 0($1)
mflo $4
mfhi $1
sll $2, $1, 5
TAG621:
lui $3, 0
mflo $3
bgez $2, TAG622
div $2, $3
TAG622:
lui $2, 9
lui $2, 8
nor $3, $2, $3
mtlo $3
TAG623:
mflo $3
lui $4, 12
sll $0, $0, 0
mtlo $3
TAG624:
slt $4, $4, $4
lui $3, 15
lhu $3, 0($4)
lb $2, 0($4)
TAG625:
add $4, $2, $2
beq $4, $2, TAG626
sltu $3, $4, $4
bgtz $4, TAG626
TAG626:
lhu $3, 0($3)
mflo $1
bgez $3, TAG627
div $3, $1
TAG627:
bgez $1, TAG628
sll $0, $0, 0
multu $1, $2
blez $1, TAG628
TAG628:
lui $3, 9
mult $2, $3
sh $3, 0($2)
bgez $2, TAG629
TAG629:
sll $0, $0, 0
sll $0, $0, 0

```

```

sll $1, $4, 8
mult $3, $4
TAG630:
sb $1, 0($1)
bltz $1, TAG631
mtlo $1
mflo $4
TAG631:
sb $4, 0($4)
sub $3, $4, $4
andi $2, $3, 3
sw $4, 0($2)
TAG632:
blez $2, TAG633
sh $2, 0($2)
bne $2, $2, TAG633
lhu $4, 0($2)
TAG633:
bgtz $4, TAG634
sb $4, 0($4)
beq $4, $4, TAG634
sw $4, 0($4)
TAG634:
sb $4, 0($4)
mfhi $4
mthi $4
lui $4, 11
TAG635:
mult $4, $4
bne $4, $4, TAG636
div $4, $4
blez $4, TAG636
TAG636:
sll $0, $0, 0
sra $1, $4, 10
addiu $1, $1, 3
nor $4, $4, $1
TAG637:
beq $4, $4, TAG638
lui $1, 5
bgtz $4, TAG638
mflo $1
TAG638:
divu $1, $1
beq $1, $1, TAG639
sllv $4, $1, $1
slt $1, $1, $4
TAG639:
sll $0, $0, 0
bne $1, $1, TAG640
sll $0, $0, 0
mthi $1
TAG640:
lui $1, 0
mfhi $3
sb $3, 0($2)
subu $2, $2, $3
TAG641:

```

```
xor $3, $2, $2
sb $2, 0($3)
bltz $2, TAG642
mtlo $2
TAG642:
subu $3, $3, $3
mtlo $3
beq $3, $3, TAG643
sh $3, 0($3)
TAG643:
bne $3, $3, TAG644
mthi $3
mflo $3
mthi $3
TAG644:
srlv $1, $3, $3
bgez $3, TAG645
mult $3, $1
mfhi $3
TAG645:
addi $1, $3, 4
lui $3, 12
mtlo $3
bgtz $3, TAG646
TAG646:
multu $3, $3
mfhi $1
mult $1, $1
mult $3, $3
TAG647:
mtlo $1
mthi $1
bltz $1, TAG648
mflo $2
TAG648:
mtlo $2
lbu $3, -144($2)
bltz $3, TAG649
sub $1, $2, $3
TAG649:
divu $1, $1
sltiu $1, $1, 4
lh $4, 0($1)
lui $1, 2
TAG650:
mflo $4
mult $1, $4
sll $0, $0, 0
addiu $1, $1, 10
TAG651:
mfhi $3
multu $1, $3
beq $3, $1, TAG652
lui $2, 4
TAG652:
or $1, $2, $2
sll $0, $0, 0
lui $1, 14
```

```
lui $4, 1
TAG653:
mfhi $4
sb $4, 0($4)
multu $4, $4
lbu $4, 0($4)
TAG654:
lui $3, 15
andi $4, $4, 1
multu $3, $3
lb $1, 0($4)
TAG655:
beq $1, $1, TAG656
mflo $1
div $1, $1
mult $1, $1
TAG656:
bltz $1, TAG657
xor $4, $1, $1
mflo $2
lbu $1, 0($2)
TAG657:
lui $3, 13
bgez $1, TAG658
sw $3, 0($1)
ori $4, $3, 5
TAG658:
slt $1, $4, $4
lui $4, 0
lui $1, 4
div $1, $1
TAG659:
subu $1, $1, $1
lui $2, 1
mtlo $1
xori $4, $1, 13
TAG660:
mfhi $1
sb $1, 0($4)
sb $4, 0($4)
srl $4, $4, 10
TAG661:
lui $1, 4
sll $0, $0, 0
mtlo $1
mult $1, $1
TAG662:
sll $0, $0, 0
srl $4, $1, 8
sra $1, $2, 14
lui $4, 5
TAG663:
sll $0, $0, 0
sll $0, $0, 0
lui $2, 3
or $2, $4, $2
TAG664:
mflo $3
```

```
bltz $3, TAG665
multu $2, $2
mthi $2
TAG665:
mult $3, $3
multu $3, $3
bgez $3, TAG666
lh $4, 0($3)
TAG666:
xor $1, $4, $4
bne $4, $4, TAG667
mthi $4
mult $1, $1
TAG667:
mtlo $1
mtlo $1
bgtz $1, TAG668
mthi $1
TAG668:
beq $1, $1, TAG669
sb $1, 0($1)
sh $1, 0($1)
srlv $3, $1, $1
TAG669:
add $4, $3, $3
mtlo $3
lui $1, 10
mflo $4
TAG670:
multu $4, $4
mthi $4
bgez $4, TAG671
mtlo $4
TAG671:
bltz $4, TAG672
lui $4, 14
bltz $4, TAG672
lui $1, 2
TAG672:
srav $3, $1, $1
slti $3, $3, 1
mtlo $1
beq $1, $3, TAG673
TAG673:
mfhi $4
mult $3, $3
mfhi $3
mtlo $4
TAG674:
add $3, $3, $3
mthi $3
bgez $3, TAG675
mult $3, $3
TAG675:
mflo $3
beq $3, $3, TAG676
addu $1, $3, $3
sh $1, 0($1)
```

```
TAG676:
lh $2, 0($1)
lb $2, 0($1)
bltz $2, TAG677
lui $1, 11
TAG677:
bltz $1, TAG678
mfhi $2
sb $1, 0($2)
mflo $2
TAG678:
sw $2, 0($2)
bgtz $2, TAG679
mflo $1
nor $2, $2, $1
TAG679:
beq $2, $2, TAG680
srl $4, $2, 2
xori $2, $2, 7
lw $2, 0($4)
TAG680:
mfhi $4
mtlo $4
sh $2, 1($2)
bgez $2, TAG681
TAG681:
sb $4, 0($4)
xor $2, $4, $4
mfhi $1
addi $4, $1, 2
TAG682:
lhu $4, 0($4)
mtlo $4
mult $4, $4
mthi $4
TAG683:
mfhi $4
sw $4, 0($4)
lhu $4, 0($4)
blez $4, TAG684
TAG684:
nor $4, $4, $4
bgez $4, TAG685
sb $4, 0($4)
blez $4, TAG685
TAG685:
sra $3, $4, 7
multu $3, $3
multu $3, $4
lhu $4, 0($4)
TAG686:
bne $4, $4, TAG687
addi $3, $4, 2
mthi $4
mfhi $3
TAG687:
mult $3, $3
beq $3, $3, TAG688
```

```
or $3, $3, $3
sb $3, 0($3)
TAG688:
bgtz $3, TAG689
addiu $4, $3, 6
lui $3, 12
divu $4, $3
TAG689:
beq $3, $3, TAG690
lui $3, 7
mflo $3
blez $3, TAG690
TAG690:
sll $0, $0, 0
addiu $2, $3, 11
nor $2, $2, $3
mult $2, $3
TAG691:
xori $2, $2, 3
beq $2, $2, TAG692
sll $0, $0, 0
lhu $2, 0($2)
TAG692:
mtlo $2
bne $2, $2, TAG693
multu $2, $2
sll $0, $0, 0
TAG693:
mfhi $1
beq $1, $2, TAG694
lui $3, 3
lui $1, 12
TAG694:
xori $1, $1, 0
mtlo $1
sll $0, $0, 0
lui $3, 13
TAG695:
bne $3, $3, TAG696
sra $2, $3, 13
srav $2, $3, $2
addu $4, $2, $2
TAG696:
blez $4, TAG697
mfhi $2
sra $3, $4, 15
mthi $3
TAG697:
lui $2, 4
xori $1, $3, 15
sb $2, 0($1)
bltz $3, TAG698
TAG698:
lb $1, 0($1)
srl $3, $1, 10
addu $3, $1, $3
bne $3, $3, TAG699
TAG699:
```

```

and $3, $3, $3
lh $4, 0($3)
beq $4, $4, TAG700
and $4, $4, $4
TAG700:
beq $4, $4, TAG701
nor $1, $4, $4
sllv $4, $1, $1
bltz $4, TAG701
TAG701:
slt $4, $4, $4
andi $1, $4, 14
mflo $3
sw $1, 0($1)
TAG702:
sra $2, $3, $3
sll $0, $0, 0
multu $2, $3
sll $0, $0, 0
TAG703:
sll $0, $0, 0
bgtz $2, TAG704
lui $1, 4
slt $2, $2, $2
TAG704:
slt $2, $2, $2
add $3, $2, $2
lhu $3, 0($3)
bltz $3, TAG705
TAG705:
lui $3, 13
sll $0, $0, 0
sll $0, $0, 0
bltz $3, TAG706
TAG706:
mflo $1
beq $3, $3, TAG707
sw $3, 0($1)
bgtz $1, TAG707
TAG707:
lb $3, 0($1)
lb $1, 0($1)
mfhi $1
beq $1, $1, TAG708
TAG708:
multu $1, $1
sra $1, $1, 2
mthi $1
divu $1, $1
TAG709:
lui $1, 3
beq $1, $1, TAG710
lui $1, 5
bgtz $1, TAG710
TAG710:
lui $1, 8
mult $1, $1
mthi $1

```



```
addiu $2, $1, 2
TAG711:
sll $0, $0, 0
bgez $2, TAG712
andi $3, $2, 0
add $2, $2, $3
TAG712:
sllv $3, $2, $2
bgtz $2, TAG713
mthi $3
lb $1, 0($2)
TAG713:
lui $2, 4
mfhi $1
mult $1, $2
mthi $2
TAG714:
sll $0, $0, 0
mthi $4
mult $1, $4
multu $4, $1
TAG715:
addu $4, $4, $4
mfhi $1
sb $4, 0($4)
lui $3, 6
TAG716:
bne $3, $3, TAG717
mfhi $3
mfhi $1
lhu $2, 0($3)
TAG717:
bgez $2, TAG718
lw $4, 0($2)
mtlo $4
sh $2, 0($2)
TAG718:
lui $3, 1
sll $0, $0, 0
bltz $4, TAG719
mflo $1
TAG719:
bltz $1, TAG720
mthi $1
srav $2, $1, $1
and $2, $1, $1
TAG720:
mthi $2
bgtz $2, TAG721
addiu $3, $2, 4
lb $1, 0($2)
TAG721:
sb $1, 0($1)
sh $1, 0($1)
bltz $1, TAG722
multu $1, $1
TAG722:
blez $1, TAG723
```

```

add $2, $1, $1
div $1, $1
beq $1, $1, TAG723
TAG723:
multu $2, $2
mthi $2
lhu $4, 0($2)
mflo $2
TAG724:
lui $4, 6
mult $2, $2
nor $3, $2, $4
sll $0, $0, 0
TAG725:
mflo $4
sll $0, $0, 0
mult $1, $1
lui $2, 2
TAG726:
mfhi $3
bgez $2, TAG727
multu $3, $3
lui $2, 6
TAG727:
slt $3, $2, $2
lui $2, 7
mfhi $1
sb $2, 0($1)
TAG728:
bltz $1, TAG729
mtlo $1
lbu $3, 0($1)
lbu $3, 0($1)
TAG729:
lh $4, 0($3)
blez $4, TAG730
sh $3, 0($4)
lbu $2, 0($4)
TAG730:
blez $2, TAG731
sll $0, $0, 0
blez $2, TAG731
div $2, $2
TAG731:
lui $3, 14
sll $0, $0, 0
mtlo $2
divu $3, $3
TAG732:
sll $2, $3, 8
sll $0, $0, 0
bne $2, $2, TAG733
lui $1, 13
TAG733:
mfhi $4
sw $4, 0($4)
lhu $1, 0($4)
slti $4, $1, 14

```

```

TAG734:
nor $1, $4, $4
sltu $2, $4, $4
mfhi $1
bgez $2, TAG735
TAG735:
sw $1, 0($1)
sh $1, 0($1)
sb $1, 0($1)
mfhi $1
TAG736:
subu $3, $1, $1
lui $3, 10
lui $1, 9
lui $1, 6
TAG737:
mfhi $2
lhu $3, 0($2)
sb $2, 0($3)
sll $0, $0, 0
TAG738:
lui $1, 4
sll $0, $0, 0
lui $1, 7
lbu $1, 0($3)
TAG739:
beq $1, $1, TAG740
mthi $1
blez $1, TAG740
sw $1, 0($1)
TAG740:
blez $1, TAG741
sra $1, $1, 13
lbu $3, 0($1)
bltz $1, TAG741
TAG741:
mtlo $3
lui $4, 12
subu $2, $4, $4
div $2, $4
TAG742:
lui $1, 15
lui $1, 4
sll $0, $0, 0
mthi $1
TAG743:
sltiu $3, $2, 15
addu $3, $3, $3
lui $4, 0
mtlo $3
TAG744:
mfhi $4
multu $4, $4
beq $4, $4, TAG745
lui $2, 11
TAG745:
mflo $2
mult $2, $2

```

```

lui $2, 15
sll $0, $0, 0
TAG746:
sll $0, $0, 0
bgez $4, TAG747
divu $4, $4
bne $4, $4, TAG747
TAG747:
sllv $3, $4, $4
sll $0, $0, 0
mfhi $4
add $3, $4, $4
TAG748:
sb $3, 0($3)
beq $3, $3, TAG749
lui $1, 1
bgtz $3, TAG749
TAG749:
mflo $1
mtlo $1
lui $2, 1
lbu $1, 0($1)
TAG750:
nop
nop
test_end:
beq $0, $0, test_end
nop

```

```

code.txt
3408007b
01002021
8c84ff85
3c0b007b
3c0b0000
8d6c0000
34083028
01002021
00800008
34083038
ac080000
8c040000
00800008
00000000
0c000c12
00000000
08000c14
00000000
03e00008
00000000
0c000c16
00000000
8fe8cfa8
8d10cfc8
8e11cfc8
8e32cfc8
ae48cfc8

```

```
test_ans
@00003000: $ 8 <= 0000007b
@00003004: $ 4 <= 0000007b
@00003008: $ 4 <= 00000000
@0000300c: $11 <= 007b0000
@00003010: $11 <= 00000000
@00003014: $12 <= 00000000
@00003018: $ 8 <= 00003028
@0000301c: $ 4 <= 00003028
@00003024: $ 8 <= 00003038
@00003028: *00000000 <= 00003038
@0000302c: $ 4 <= 00003038
@00003038: $31 <= 00003040
@00003050: $31 <= 00003058
@00003058: $ 8 <= 00003038
@0000305c: $16 <= 00003038
@00003060: $17 <= 00003038
@00003064: $18 <= 00003038
@00003068: *00000000 <= 00003038
```

## (二) 自动测试工具

### 1. 测试样例生成器

### 2. 自动执行脚本

### 3. 正确性判定脚本

## 三、思考题

### 流水线冒险

1. 在采用本节所述的控制冒险处理方式下，PC 的值应当如何被更新？请从数据通路和控制信号两方面进行说明。
  - 将 IFU 的 nPC 输入端和 D 级的一个 MUX 输出相连，该 MUX 的输入为正常执行命令的下一个 PC 地址以及各种跳转指令对应的下一个 PC 地址。
  - ID/EX 流水线寄存器直接译码产生 PCSrc 控制信号，配合前移至 D 级的比较器产生的相等信号，用来控制上述的 MUX。
2. 对于 jal 等需要将指令地址写入寄存器的指令，为什么需要回写 PC+8？

PC + 4 的位置是延迟槽，跳回时应该跳到延迟槽下一条指令，即 PC + 8。

### 数据冒险的分析

为什么所有的供给者都是存储了上一级传来的各种数据的**流水级寄存器**，而不是由 ALU 或者 DM 等部件来提供数据？

如果从非流水线寄存器部件转发，那么某一级的总延迟就会增加，从而根据木桶效应，时钟周期就会增加，总效率反而降低，得不偿失。

# AT 法处理流水线数据冒险

## 1. “转发（旁路）机制的构造”中的Thinking 1-4;

1. 如果不采用已经转发过的数据，而采用上一级中的原始数据，会出现怎样的问题？试列举指令序列说明这个问题。

- 计算过程或存储过程中会用到还未更改过的寄存器值，从而出错。例如：

```
ori $1, $0, 1
nop
nop
nop
nop
lw $1, 0($0)
sw $1, 4($0)
```

这样 `sw` 指令就会把 1 存到 DM 中。

2. 我们为什么要对GPR采用内部转发机制？如果不采用内部转发机制，我们要怎样才能解决这种情况下的转发需求呢？

- GPR采用内部转发机制相当于 MEM/WB 流水线寄存器的值直接实时反馈到 GPR 的输出端，从而当前处于 D 级的指令可以直接用到对应寄存器的值。
- 如果不采用内部转发机制，需要额外建立从 MEM/WB 流水线寄存器转发到 D 级的数据通路。

3. 为什么0号寄存器需要特殊处理？

- 因为指令可以对 0 号寄存器赋值，只是不会造成实际作用，但是转发过程中如果不特判就默认 0 号寄存器的值被更改了，从而造成错误。

4. 什么是“最新产生的数据”？

- 根据指令的执行顺序，越后执行的指令更改的寄存器的值越新，按照 ID/EX、EX/MEM、MEM/WB 的顺序，越靠前所转发出的信息越新，因此优先级更高。

2. 在 AT 方法讨论转发条件的时候，只提到了“供给者需求者的A相同，且不为 0”，但在 CPU 写入 GRF 的时候，是有一个 `we` 信号来控制是否要写入的。为何在 AT 方法中不需要特判 `we` 呢？为了**用且仅用** A 和 T 完成转发，在翻译出 A 的时候，要结合 `we` 做什么操作呢？

- AT 方法如是说：

只要**当前位点的读取寄存器地址和某转发输入来源的写入寄存器地址相等且不为0**

那么既然是要写入的，`WE` 必然为 1，因此不用特判。

- 对于确实要写入的寄存器，不做改变；对于条件判断后不要写，将寄存器地址赋为0。

## P6思考题

- 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？
  - 乘除法延迟远大于 ALU，整合进 ALU 那么根据木桶原理 CPU 整体周期将大幅增加。增加 HI 和 LO 寄存器可以让乘除法指令和其它指令并行执行，需要结果时再取出即可。
- 参照你对延迟槽的理解，试解释“乘除槽”。
  - 当乘除法进行或即将开始时，乘除有关指令会被阻塞在 IF/ID 流水线寄存器，相当于处于“乘除槽”。
- 举例说明并分析何时按字节访问内存相对于按字访问内存性能上更有优势。（Hint：考虑C语言中字符串的情况）
  - 当访问类型只占一个字节时，比如 `char`。

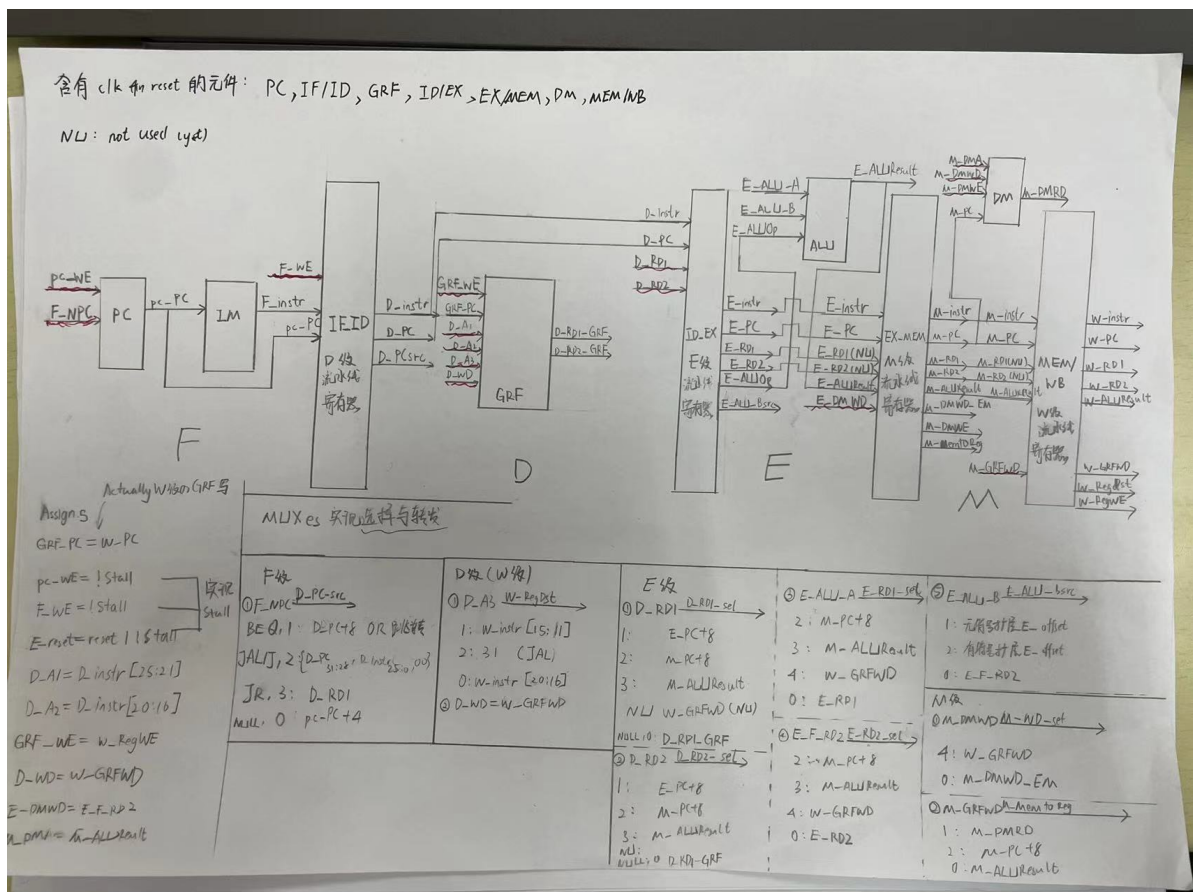
- 在本实验中你遇到了哪些不同指令类型组合产生的冲突？你又是如何解决的？相应的测试样例是什么样的？

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。

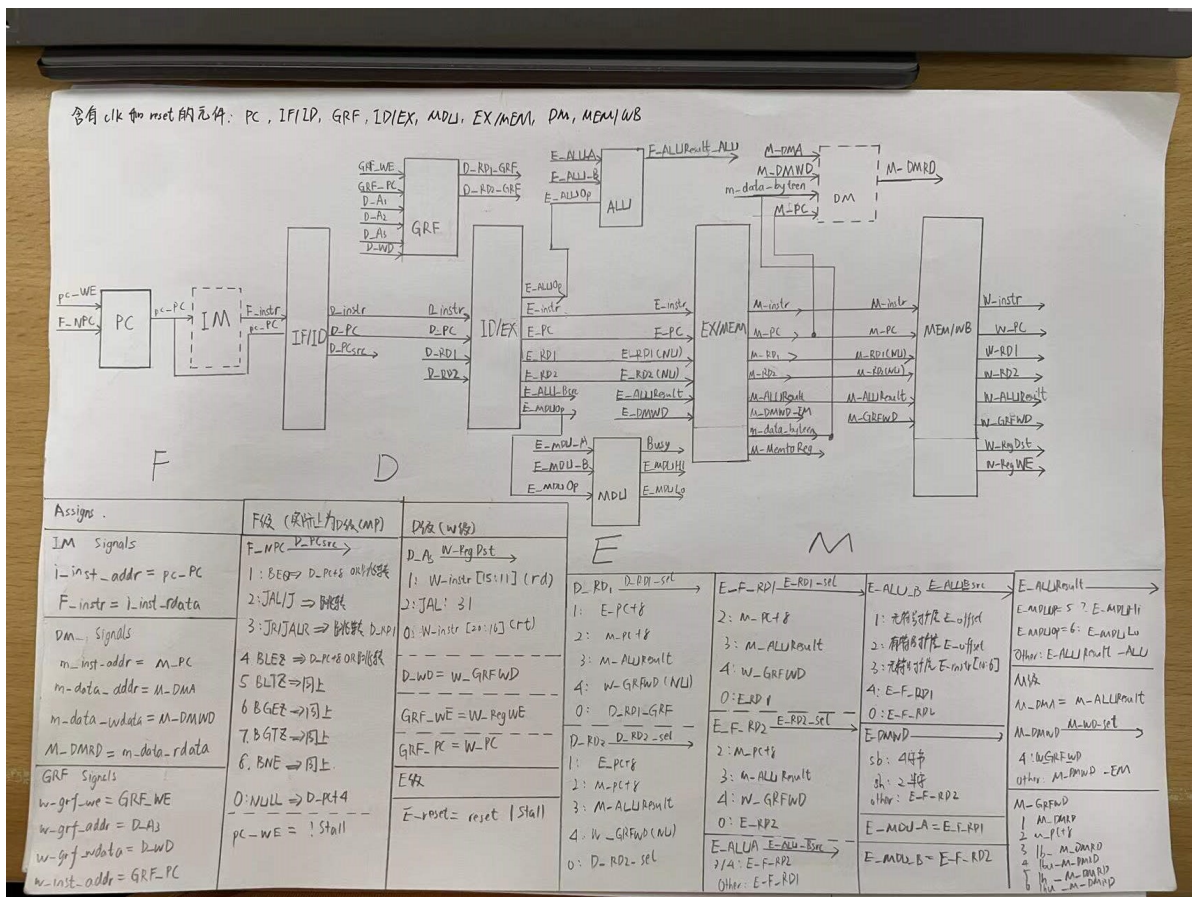
此思考题请同学们结合自己测试CPU使用的具体手段，按照自己的实际情况进行回答。

- 主要是数据冒险和控制冒险，分别通过暂停转发以及比较前移+延迟槽解决。
- 数据生成器采用了特殊策略：单组数据中除了 0 和 31 号寄存器外，至多涉及 3 个寄存器。一方面，这样产生的代码中，邻近的指令几乎全部都存在数据冒险，可以充分测试转发和暂停；另一方面，当测试数据的组数一定多，几乎涉及了每个寄存器，避免了只测试部分寄存器。此外，所有跳转指令都是特殊构造的，不会进入死循环的同时如果跳转出错可以输出中体现。
- 对于一些会产生异常的指令，为防止 MARS 报错，进行了一定的规避。
- 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？
  - 主要采用了指令分类的方法，P6 完全沿用了 P5 的分类方法，新增的指令对应的特点都没有脱离这些分类，因此对于每条指令而言，只需译码后将其加入对应的分类，数据通路部分和 P5 完全类似，转发部分完全不用改，暂停部分只需添加一个因乘除块而导致的暂停。

## 后记









端口	输入/输出	位宽	描述
nPC	I	32	设置下一个 PC 值。
WE	I	1	使能端。
clk	I	1	时钟信号。
reset	I	1	同步复位信号。
I	O	32	当前指令。
PC	O	32	当前 PC 值。

### 功能定义

序号	功能名称	功能描述
1	同步复位	当时钟上升沿到来且同步复位信号有效时，将 PC 设置为 0x00003000。
2	取指令	当时钟上升沿到来时，I 输出当前 PC 对应的指令。
3	取PC值	当时钟上升沿到来时，PC 输出当前 PC 值。
4	设置PC值	当时钟上升沿到来且使能端有效时，将 nPC 设置为当前 PC 值。

## 2.GRF

### 介绍

通用寄存器组，也称为寄存器文件、寄存器堆，可以存取 32 位数据，具有同步复位功能。寄存器标号为 0 到 31，其中 0 号寄存器读取的结果恒为 0。

### 端口定义

端口	输入/输出	位宽	描述
PC	I	32	当前指令的 PC 值。
A1	I	5	指定 32 个寄存器中的一个，将其存储的数据读出到 RD1。
A2	I	5	指定 32 个寄存器中的一个，将其存储的数据读出到 RD2。
A3	I	5	指定 32 个寄存器中的一个，作为写入的目标寄存器。
WD	I	32	写入寄存器的数据信号。
WE	I	1	写使能信号，高电平有效。
clk	I	1	时钟信号。
reset	I	1	同步复位信号。
RD1	O	32	输出 A1 指定的寄存器中的数据。
RD2	O	32	输出 A2 指定的寄存器中的数据。

## 功能定义

序号	功能名称	功能描述
1	同步复位	当时钟上升沿到来且同步复位信号有效时，将所有寄存器的值设置为 0x00000000。
2	读数据	读出 A1 和 A2 地址对应寄存器中存储的数据到 RD1 和 RD2；当 WE 有效时会把 WD 的值会实时反馈到对应的 RD1 或 RD2，即内部转发。
3	写数据	当 WE 有效且时钟上升沿到来时，将 WD 的数据写入 A3 对应的寄存器中。

## 3.ALU

### 介绍

算术逻辑单元，提供 32 位的多种运算功能。

### 端口定义

端口	输入/输出	位宽	描述
A	I	32	参与 ALU 计算的第一个值。
B	I	32	参与 ALU 计算的第二个值。
Op	I	4	ALU 功能的选择信号，具体见功能定义。
AO	O	32	ALU 的计算结果。

### 功能定义

序号	功能名称	功能描述
1	按位与	当 Op = 0 时, $AO = A \& B$ 。
2	按位或	当 Op = 1 时, $AO = A   B$ 。
3	加法	当 Op = 2 时, $AO = A + B$ 。
4	减法	当 Op = 3 时, $AO = A - B$ 。
5	左移 16 位	当 Op = 4 时, $AO = A \ll 16$ 。
6	有符号比大小	当 Op = 5 时, $AO = A < B$ 。
7	无符号比大小	当 Op = 6 时, $AO = A < B$ 。（有符号）
8	逻辑左移	当 Op = 7 时, $AO = A \ll B[4:0]$ 。
9	逻辑右移	当 Op = 8 时, $AO = A \gg B[4:0]$ 。
10	算术右移	当 Op = 9 时, $AO = A \ggg B[4:0]$ 。
11	异或	当 Op = 10 时, $AO = A \wedge B$ 。
12	或非	当 Op = 11 时, $AO = \sim(A   B)$ 。

## 4.MDU

### 介绍

乘除块，内有 hi 和 lo 两个寄存器，可以进行乘除运算，完成相关乘除指令，并模拟乘除运算的延时。

### 端口定义

端口	输入/输出	位宽	描述
A	I	32	参与乘除运算的第一个值。
B	I	32	参与乘除运算的第二个值。
Op	I	4	MDU 功能的选择信号，具体见功能定义。
clk	I	1	时钟信号。
reset	I	1	同步复位信号。
Busy	O	1	当前是否正在进行乘除运算。
hi	O	32	hi 寄存器的值。
lo	O	32	lo 寄存器的值。

### 功能定义

序号	功能名称	功能描述
1	同步复位	当时钟上升沿到来且同步复位信号有效时，将 MDU 中所有值设为 0x00000000。
2	有符号乘	当 Op = 1 时，将 A * B 的高 32 位和低 32 位分别写入 hi 和 lo 寄存器，延迟为 5 个时钟周期。（有符号）
3	无符号乘	当 Op = 2 时，将 A * B 的高 32 位和低 32 位分别写入 hi 和 lo 寄存器，延迟为 5 个时钟周期。（无符号）
4	有符号除	当 Op = 3 时，将 A / B 的余数和商分别写入 hi 和 lo 寄存器，延迟为 10 个时钟周期。（有符号）
5	无符号除	当 Op = 4 时，将 A / B 的余数和商分别写入 hi 和 lo 寄存器，延迟为 10 个时钟周期。（无符号）
6	写 hi 寄存器	当 Op = 7 时，将 A 写入 hi 寄存器。
7	写 lo 寄存器	当 Op = 8 时，将 A 写入 lo 寄存器。

## 5.DM

### 介绍

数据存储器，可以存取 32 位数据，容量为 32bit \* 4096，具有同步复位功能，起始地址为 0x00000000。

### 端口定义

端口	输入/输出	位宽	描述
PC	I	32	当前指令的 PC 值。
A	I	5	读取或写入数据的地址。
WD	I	32	写入 DM 中的数据。
Op	I	2	DM 功能的选择信号，具体见功能定义。
clk	I	1	时钟信号。
reset	I	1	同步复位信号。
RD	O	32	根据 A 和 Op 输出对应的数据。

### 功能定义

序号	功能名称	功能描述
1	同步复位	当时钟上升沿到来且同步复位信号有效时，将 DM 中所有值设为 0x00000000。
2	有符号读字节	当 Op = 0 时，输出对应字节的 32 位有符号扩展。
3	无符号读字节	当 Op = 1 时，输出对应字节的 32 位无符号扩展。
4	有符号读半字	当 Op = 2 时，输出对应半字的 32 位有符号扩展。
5	无符号读半字	当 Op = 3 时，输出对应半字的 32 位无符号扩展。
6	读字	当 Op = 4 时，输出对应字。
7	写字节	当 Op = 5 且时钟上升沿来临时，将 WD[7:0] 写入 A 对应的地址。
8	写半字	当 Op = 6 且时钟上升沿来临时，将 WD[15:0] 写入 A 对应的地址。
9	写字	当 Op = 7 且时钟上升沿来临时，将 WD 写入 A 对应的地址。

## 6. IF\_ID

### 介绍

I 级和 D 级间的流水线寄存器，同时产生 D 级的控制信号。

### 端口定义

端口	输入/输出	位宽	描述
nl	I	32	下一个指令。
nPC	I	32	下一个 PC 值。
WE	I	1	使能端
clk	I	1	时钟信号。
reset	I	1	同步复位信号。
I	O	32	当前指令。
PC	O	32	当前 PC 值。
PCSrc	O	4	0: $PC = PC + 4$ 。 1: 跳转到 beq 指令对应的跳转地址。 2: 跳转到 jal 指令和 j 指令对应的跳转地址。 3: 跳转到 jalr 指令和 jr 指令对应的跳转地址。 4: 跳转到 bne 指令对应的跳转地址。 5: 跳转到 blez 指令对应的跳转地址。 6: 跳转到 bgtz 指令对应的跳转地址。 7: 跳转到 bltz 指令对应的跳转地址。 8: 跳转到 bgez 指令对应的跳转地址。

## 功能定义

序号	功能名称	功能描述
1	同步复位	当时钟上升沿到来且同步复位信号有效时，将所有值设为 0x00000000。
2	读数据	读出各个当前寄存器对应的值。
3	写数据	当 WE 有效且时钟上升沿到来时，将各个对应的值写入寄存器中。

## 7. ID\_EX

### 介绍

D 级和 E 级间的流水线寄存器，同时产生 E 级的控制信号。

### 端口定义

端口	输入/输出	位宽	描述
nl	I	32	下一个指令。
nRD1	I	32	下一个 RD1 值。
nRD2	I	32	下一个 RD2 值。
nPC	I	32	下一个 PC 值。
clk	I	1	时钟信号。
reset	I	1	同步复位信号。
I	O	32	当前指令。
RD1	O	32	当前 RD1 值。
RD2	O	32	当前 RD2 值。
PC	O	32	当前 PC 值。
ALUOp	O	4	ALU 功能的选择信号，具体见 ALU 模块的功能定义。
ALUSrc	O	3	0: ALU 的 A 输入端选择 RD1 输出端，ALU 的 B 输入端选择 RD2 输出端。 1: ALU 的 A 输入端选择 RD1 输出端，ALU 的 B 输入端选择 I[15:0] 的 32 位无符号扩展。 2: ALU 的 A 输入端选择 RD1 输出端，ALU 的 B 输入端选择 I[15:0] 的 32 位有符号扩展。 3: ALU 的 A 输入端选择 RD2 输出端，ALU 的 B 输入端选择 I[10:6]。 4: ALU 的 A 输入端选择 RD2 输出端，ALU 的 B 输入端选择 RD1 输出端。
MDUOp	O	3	MDU 功能的选择信号，具体见 MDU 模块的功能定义。

### 功能定义

序号	功能名称	功能描述
1	同步复位	当时钟上升沿到来且同步复位信号有效时，将所有值设为 0x00000000。
2	读数据	读出各个当前寄存器对应的值。
3	写数据	当时钟上升沿到来时，将各个对应的值写入寄存器中。

## 8. EX\_MEM

### 介绍

E 级和 M 级间的流水线寄存器，同时产生 M 级的控制信号。

## 端口定义

端口	输入/输出	位宽	描述
nl	I	32	下一个指令。
nAO	I	32	下一个 ALU 计算结果。
nWD	I	32	下一个写入 DM 的值。
nPC	I	32	下一个 PC 值。
clk	I	1	时钟信号。
reset	I	1	同步复位信号。
I	O	32	当前指令。
AO	O	32	当前 ALU 计算结果。
WD	O	32	当前写入 DM 的值。
PC	O	32	当前 PC 值。
MemOp	O	3	DM 功能的选择信号，具体见 DM 模块的功能定义。
MemtoReg	O	2	0: 准备写回 GRF 的 WD 输入端选择 AO 输出端。 1: 准备写回 GRF 的 WD 输入端选择 DM 的 RD 输出端。 2: 准备写回 GRF 的 WD 输入端选择 PC + 8。

## 功能定义

序号	功能名称	功能描述
1	同步复位	当时钟上升沿到来且同步复位信号有效时，将所有值设为 0x00000000。
2	读数据	读出各个当前寄存器对应的值。
3	写数据	当时钟上升沿到来时，将各个对应的值写入寄存器中。

## 9. MEM\_WB

### 介绍

M 级和 W 级间的流水线寄存器，同时产生 W 级的控制信号。

### 端口定义



nl	I	32	下一个指令。
nPC	I	32	下一个 PC 值。
nWD	I	32	下一个写入 GRF 的值。
clk	I	1	时钟信号。
reset	I	1	同步复位信号。
I	O	32	当前指令。
PC	O	32	当前 PC 值。
WD	O	32	当前写入 GRF 的值。
RegWrite	O	1	0: GRF 的写使能端 WE 无效。 1: GRF 的写使能端 WE 有效。
RegDst	O	2	0: 准备写回 GRF 的 A3 输入端选择 I[20:16]。 1: 准备写回 GRF 的 A3 输入端选择 I[15:11]。 2: 准备写回 GRF 的 A3 输入端选择 31。
RegWrite	O	1	0: GRF 的写使能端 WE 无效。 1: GRF 的写使能端 WE 有效。

### 功能定义

序号	功能名称	功能描述
1	同步复位	当时钟上升沿到来且同步复位信号有效时，将所有值设为 0x00000000。
2	读数据	读出各个当前寄存器对应的值。
3	写数据	当时钟上升沿到来时，将各个对应的值写入寄存器中。

## 10. SFU

### 介绍

暂停 (Stall) 和 转发 (Forward) 的控制单元，产生两者的控制信号。

### 端口与功能定义

端口	输入/输出	位宽	描述
if_id_l	I	32	IF/ID 流水线寄存器当前的指令。
id_ex_l	I	32	ID/EX 流水线寄存器当前的指令。
ex_mem_l	I	32	EX/MEM 流水线寄存器当前的指令。
mem_wb_l	I	32	MEM/WB 流水线寄存器当前的指令。
Stall	O	1	暂停信号。
Busy	I	1	乘除块的 Busy 信号。
Start	I	1	乘除块的 Start 信号。
F_if_id_rs	O	3	0: D 级 Rs 选择 GRF 的 RD1 输出端。 1: D 级 Rs 选择 ID/EX 流水线寄存器的 PC + 8 输出端。 2: D 级 Rs 选择 EX/MEM 流水线寄存器的 PC + 8 输出端。 3: D 级 Rs 选择 EX/MEM 流水线寄存器的 AO 输出端。
F_if_id_rt	O	3	0: D 级 Rt 选择 GRF 的 RD2 输出端。 1: D 级 Rt 选择 ID/EX 流水线寄存器的 PC + 8 输出端。 2: D 级 Rt 选择 EX/MEM 流水线寄存器的 PC + 8 输出端。 3: D 级 Rt 选择 EX/MEM 流水线寄存器的 AO 输出端。
F_id_ex_rs	O	3	0: E 级 Rs 选择 ID/EX 流水线寄存器的 RD1 输出端。 2: E 级 Rs 选择 EX/MEM 流水线寄存器的 PC + 8 输出端。 3: E 级 Rs 选择 EX/MEM 流水线寄存器的 AO 输出端。 4: E 级 Rs 选择 MEM/WB 流水线寄存器的 WD 输出端。
F_id_ex_rt	O	3	0: E 级 Rt 选择 ID/EX 流水线寄存器的 RD2 输出端。 2: E 级 Rt 选择 EX/MEM 流水线寄存器的 PC + 8 输出端。 3: E 级 Rt 选择 EX/MEM 流水线寄存器的 AO 输出端。 4: E 级 Rt 选择 MEM/WB 流水线寄存器的 WD 输出端。
F_ex_mem_rt	O	3	0: M 级 Rt 选择 EX/MEM 流水线寄存器的 WD 输出端。 4: M 级 Rt 选择 MEM/WB 流水线寄存器的 WD 输出端。

## 数据冒险分析表

## 11.DEC

## 介绍

指令分类译码器，输入一个指令，返回该指令在数据冒险中对应的类型。

## 端口与功能定义

端口	输入/输出	位宽	描述
l	I	32	需要分类的指令。
j	O	1	需要读寄存器的跳转指令。
r	O	1	除 jalr, jr 外的 R 型指令。
i	O	1	I 型指令。
ld	O	1	load 型指令。
st	O	1	store 型指令。
jal	O	1	jal, jalr 指令。
rs	O	5	I[25:21]。
rt	O	5	I[20:16]。
rd	O	5	如果指令为 jal 输出 31，否则输出 I[15:11]。

## 12.CTRL

## 介绍

指令译码器，输入 Op 和 Func，输出该指令是哪个指令。本 CPU 采用分布式译码，对应控制信号的功能已在上述四个流水线寄存器中详细描述。

## 端口与功能定义

端口	输入/输出	位宽	描述
Op	I	6	所有指令的操作码，对应 I[31:26]。
Func	I	6	R 指令中辅助识别的操作码，对应 I[5:0]。
每种指令	O	1	判断是否是每种指令。

## 重要机制实现方法

### 通过指令分类和 Tuse-Tnew 法解决数据冒险

- 首先通过 Tuse-Tnew 的定义对指令进行分类。
- 再列出对应的 Tuse-Tnew 表，已在 SFU 部分展示。
- 再通过 Tuse-Tnew 表实现暂停信号与转发信号的生成。
- 最后，将暂停信号和转发信号连接到对应的部件上。

# 测试方案

## 自动测试工具

### 全自动化对拍器

- 重复下述过程无数次：
  - 使用 C++ 随机化生成一段 MIPS 汇编代码。
  - 使用魔改版的 Mars，执行汇编指令时会按照格式输出评测机需要的信息（即需要 `$display` 的信息）。
  - 使用 Mars 的命令行代码编译并执行，将输出导入到 `m.out`。
  - 使用 Mars 的命令行代码编译并将机器码导出到 `code.txt`。
  - 使用 IVerilog 命令行代码将 testbench 文件转换为可进行仿真的 `tb.out` 文件。
  - 使用 IVerilog 命令行代码执行仿真，并将输出导入到 `v.out`。
  - 使用 C++ 删掉 `v.out` 中多余的信息（例如一些警告和时间），将 `m.out` 和 `v.out` 分别排序。
  - 使用 `fc` 将 `v.out` 和 `m.out` 进行比对，在第一组 `WA` 的数据点停下。
- 代码如下：
  - 数据生成器：

```
#include <bits/stdc++.h>

using namespace std;

vector<int> r;
mt19937 mt(time(0));
uniform_int_distribution<int>
    u16(0, (1 << 16) - 1),
    s16(-(1 << 15), (1 << 15) - 1),
    siz(0, 15),
    reg(0, 2),
    grf(1, 30),
    shift(0, 31),
    I(1, 40),
    J(41, 49),
    IJ(1, 49),
    one(11, 40);

int cnt, tot;

int getR(){
    return r[reg(mt)];
}

void solve(int i){
    int x, X;
    switch(i){
        case 1:
            x = getR();
            printf("ori %d, $0, 0\n", x);
            printf("lb %d, %d($%d)\n", getR(), siz(mt), x);
            tot += 2;
            break;
        case 2:
```

```

    x = getR();
    printf("ori %d, $0, 0\n", x);
    printf("lbu %d, %d(%d)\n", getR(), siz(mt), x);
    tot += 2;
    break;
case 3:
    x = getR();
    printf("ori %d, $0, 0\n", x);
    printf("lhu %d, %d(%d)\n", getR(), siz(mt) >> 1 << 1, x);
    tot += 2;
    break;
case 4:
    x = getR();
    printf("ori %d, $0, 0\n", x);
    printf("lhu %d, %d(%d)\n", getR(), siz(mt) >> 1 << 1, x);
    tot += 2;
    break;
case 5:
    x = getR();
    printf("ori %d, $0, 0\n", x);
    printf("lw %d, %d(%d)\n", getR(), siz(mt) >> 2 << 2, x);
    tot += 2;
    break;
case 6:
    x = getR();
    printf("ori %d, $0, 0\n", x);
    printf("sb %d, %d(%d)\n", getR(), siz(mt), x);
    tot += 2;
    break;
case 7:
    x = getR();
    printf("ori %d, $0, 0\n", x);
    printf("sh %d, %d(%d)\n", getR(), siz(mt) >> 1 << 1, x);
    tot += 2;
    break;
case 8:
    x = getR();
    printf("ori %d, $0, 0\n", x);
    printf("sw %d, %d(%d)\n", getR(), siz(mt) >> 2 << 2, x);
    tot += 2;
    break;
case 9:
    x = getR();
    printf("ori %d, %d, 1\n", x, x);
    printf("div %d, %d\n", getR(), x);
    tot += 2;
    break;
case 10:
    x = getR();
    printf("ori %d, %d, 1\n", x, x);
    printf("divu %d, %d\n", getR(), x);
    tot += 2;
    break;
case 11:
    printf("add %d, $0, %d\n", getR(), getR());
    tot++;
    break;
case 12:

```

```

        printf("addu %d, %d, %d\n", getR(), getR(), getR());
        tot++;
        break;
case 13:
    x = getR();
    printf("sub %d, %d, %d\n", getR(), x, x);
    tot++;
    break;
case 14:
    printf("subu %d, %d, %d\n", getR(), getR(), getR());
    tot++;
    break;
case 15:
    printf("mult %d, %d\n", getR(), getR());
    tot++;
    break;
case 16:
    printf("multu %d, %d\n", getR(), getR());
    tot++;
    break;
case 17:
    printf("slt %d, %d, %d\n", getR(), getR(), getR());
    tot++;
    break;
case 18:
    printf("sltu %d, %d, %d\n", getR(), getR(), getR());
    tot++;
    break;
case 19:
    printf("sll %d, %d, %d\n", getR(), getR(), shift(mt));
    tot++;
    break;
case 20:
    printf("srl %d, %d, %d\n", getR(), getR(), shift(mt));
    tot++;
    break;
case 21:
    printf("sra %d, %d, %d\n", getR(), getR(), shift(mt));
    tot++;
    break;
case 22:
    printf("sllv %d, %d, %d\n", getR(), getR(), getR());
    tot++;
    break;
case 23:
    printf("srlv %d, %d, %d\n", getR(), getR(), getR());
    tot++;
    break;
case 24:
    printf("srav %d, %d, %d\n", getR(), getR(), getR());
    tot++;
    break;
case 25:
    printf("and %d, %d, %d\n", getR(), getR(), getR());
    tot++;
    break;
case 26:
    printf("or %d, %d, %d\n", getR(), getR(), getR());

```

```

        tot++;
        break;
case 27:
    printf("xor %d, %d, %d\n", getR(), getR(), getR());
    tot++;
    break;
case 28:
    printf("nor %d, %d, %d\n", getR(), getR(), getR());
    tot++;
    break;
case 29:
    printf("addi %d, %d, %d\n", getR(), getR(), 0);
    tot++;
    break;
case 30:
    printf("addiu %d, %d, %d\n", getR(), getR(), s16(mt));
    tot++;
    break;
case 31:
    printf("andi %d, %d, %d\n", getR(), getR(), u16(mt));
    tot++;
    break;
case 32:
    printf("ori %d, %d, %d\n", getR(), getR(), u16(mt));
    tot++;
    break;
case 33:
    printf("xori %d, %d, %d\n", getR(), getR(), u16(mt));
    tot++;
    break;
case 34:
    printf("lui %d, %d\n", getR(), u16(mt));
    tot++;
    break;
case 35:
    printf("slti %d, %d, %d\n", getR(), getR(), s16(mt));
    tot++;
    break;
case 36:
    printf("sltiu %d, %d, %d\n", getR(), getR(), s16(mt));
    tot++;
    break;
case 37:
    printf("mfhi %d\n", getR());
    tot++;
    break;
case 38:
    printf("mflo %d\n", getR());
    tot++;
    break;
case 39:
    printf("mthi %d\n", getR());
    tot++;
    break;
case 40:
    printf("mtlo %d\n", getR());
    tot++;
    break;

```

```
case 41:
    printf("beq %d, %d, label%d\n", getR(), getR(), ++cnt);
    solve(I(mt));
    solve(I(mt));
    printf("label%d: ", cnt);
    solve(I(mt));
    tot++;
    break;
case 42:
    printf("bne %d, %d, label%d\n", getR(), getR(), ++cnt);
    solve(I(mt));
    solve(I(mt));
    printf("label%d: ", cnt);
    solve(I(mt));
    tot++;
    break;
case 43:
    printf("blez %d, label%d\n", getR(), ++cnt);
    solve(I(mt));
    solve(I(mt));
    printf("label%d: ", cnt);
    solve(I(mt));
    tot++;
    break;
case 44:
    printf("bgtz %d, label%d\n", getR(), ++cnt);
    solve(I(mt));
    solve(I(mt));
    printf("label%d: ", cnt);
    solve(I(mt));
    tot++;
    break;
case 45:
    printf("bltz %d, label%d\n", getR(), ++cnt);
    solve(I(mt));
    solve(I(mt));
    printf("label%d: ", cnt);
    solve(I(mt));
    tot++;
    break;
case 46:
    printf("bgez %d, label%d\n", getR(), ++cnt);
    solve(I(mt));
    solve(I(mt));
    printf("label%d: ", cnt);
    solve(I(mt));
    tot++;
    break;
case 47:
    printf("j label%d\n", ++cnt);
    solve(I(mt));
    solve(I(mt));
    printf("label%d: ", cnt);
    solve(I(mt));
    tot++;
    break;
case 48:
    printf("jal label%d\n", ++cnt);
```



```

        X = getR();
        printf("ori %d, $0, 16\n", X);
        solve(one(mt));
        printf("label%d: addu %d, %d, $31\n", cnt, X, X);
        printf("jr %d\n", X);
        puts("nop");//solve(I(mt));
        tot += 4;
        break;
    case 49:
        printf("jal label%d\n", ++cnt);
        X = getR();
        printf("ori %d, $0, 16\n", X);
        solve(one(mt));
        printf("label%d: addu %d, %d, $31\n", cnt, X, X);
        printf("jalr %d, %d\n", getR(), X);
        puts("nop");//solve(I(mt));
        tot += 4;
        break;
    }
}

int main(){
    r.push_back(grf(mt)), r.push_back(grf(mt)), r.push_back(grf(mt));
    freopen("test.asm", "w", stdout);
    puts("ori $28, $0, 0");
    puts("ori $29, $0, 0");
    while(tot < 900) solve(IJ(mt));
}

```

- 格式处理器:

```

#include <bits/stdc++.h>
#define maxn 100086

using namespace std;

vector<string> v, w;
char s[maxn];

int main(){
    freopen("v.out", "r", stdin);
    while(gets(s) != NULL){
        string S = s;
        v.push_back(S);
    }
    freopen("v.out", "w", stdout);

    for(int i = 0; i < v.size(); i++){
        if(v[i].length() <= 20 || v[i][20] != '@') continue;
        w.push_back(v[i].substr(20));
    }
    sort(w.begin(), w.end());
    for(int i = 0; i < w.size(); i++) printf("%s\n", w[i].c_str());

    v.clear();
    freopen("m.out", "r", stdin);
    while(gets(s) != NULL){

```

```

        string s = s;
        v.push_back(s);
    }
    freopen("m.out", "w", stdout);
    sort(v.begin(), v.end());
    for(int i = 0; i < v.size(); i++){
        if(v[i][0] == '@') printf("%s\n", v[i].c_str());
    }
}

```

- 评测代码:

```

#include <bits/stdc++.h>

using namespace std;

char s[10086];
int cnt = 0;

int main(){
    while(1){
        system("gen.exe");
        system("java -jar Mars_Changed.jar db mc CompactDataAtZero nc
test.asm > m.out");
        system("java -jar Mars_Changed.jar mc CompactDataAtZero a dump .text
HexText code.txt test.asm > log.txt");
        system("iverilog -o tb.out -y D:\\coding\\CO\\Verilog\\P6
D:\\coding\\CO\\Verilog\\P6\\tb.v");
        system("vvp tb.out > v.out");
        system("del.exe");
        system("fc v.out m.out > log.txt");
        freopen("log.txt", "r", stdin);
        gets(s), gets(s);
        printf("test%d:", ++cnt);
        if(s[0] != 'F'){
            puts("Wrong Answer!");
            break;
        }
        puts("Accepted!");
    }
}

```

- 效果图如下:

o

## 思考题

- 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？
  - 乘除法延迟远大于 ALU，整合进 ALU 那么根据木桶原理 CPU 整体周期将大幅增加。增加 HI 和 LO 寄存器可以让乘除法指令和其它指令并行执行，需要结果时再取出即可。
- 参照你对延迟槽的理解，试解释“乘除槽”。
  - 当乘除法进行或即将开始时，乘除有关指令会被阻塞在 IF/ID 流水线寄存器，相当于处于“乘除槽”。

- 举例说明并分析何时按字节访问内存相对于按字访问内存性能上更有优势。（Hint：考虑C语言中字符串的情况）
  - 当访问类型只占一个字节时，比如 `char`。
- 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？

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。

此思考题请同学们结合自己测试CPU使用的具体手段，按照自己的实际情况进行回答。

- 主要是数据冒险和控制冒险，分别通过暂停转发以及比较前移+延迟槽解决。
- 数据生成器采用了特殊策略：单组数据中除了 0 和 31 号寄存器外，至多涉及 3 个寄存器。一方面，这样产生的代码中，邻近的指令几乎全部都存在数据冒险，可以充分测试转发和暂停；另一方面，当测试数据的组数一定多，几乎涉及了每个寄存器，避免了只测试部分寄存器。此外，所有跳转指令都是特殊构造的，不会进入死循环的同时如果跳转出错可以输出中体现。
- 对于一些会产生异常的指令，为防止 MARS 报错，进行了一定的规避。
- 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？
  - 主要采用了指令分类的方法，P6 完全沿用了 P5 的分类方法，新增的指令对应的特点都没有脱离这些分类，因此对于每条指令而言，只需译码后将其加入对应的分类，数据通路部分和 P5 完全类似，转发部分完全不用改，暂停部分只需添加一个因乘除块而导致的暂停。