# 计算机组成原理实验报告参考模板

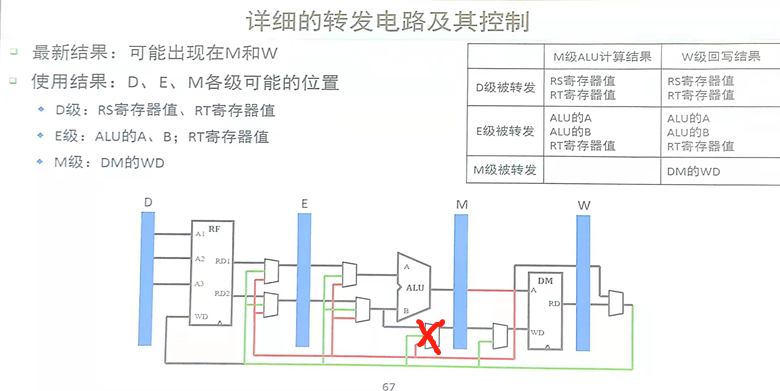
## 一、CPU设计方案综述

### （一）总体设计概述

我们设计的 CPU 将包含 Controller（控制器）、IFU（取指令单元，我使用了三个单元PC, NPC，IM来实现）、GRF（通用寄存器组，也称为寄存器文件、寄存器堆）、ALU（算术逻辑单元）、DM（数据存储器）、 EXT（位扩展器）等基本部件，通过 MUX、 Splitter 等内置器件组合连接成数据通路。并采用分布式的Control Unit, stall ctrl, forward ctrl对数据通路进行控制

### （二）关键模块定义

### 数据通路总览&示意图



#### F级

PC,IM

1. D级
2. GRF

课程功能要求：

1.用具有写使能的寄存器实现，寄存器总数为 32 个，应具有**异步复位**功能。

**2. 0 号寄存器**的值始终保持为 0。其他寄存器**初始值（复位后）均为 0**，无需专门设置。

|  |  |  |
| --- | --- | --- |
| GRF | IO | Description |
| A1[4:0] | I | 通常A1=rs(instr[25:21])  RD1= GRF[A1] |
| A2[4:0] | I | 通常A2=rt(instr[20:16])  RD2= GRF[A2] |
| A3[4:0] | I | A3 写入寄存器的编号  A3 当其为I指令时多为rt, R指令时多为rd, jal等link指令时为31  写入GRF[A3] <= WD3 |
| WD3[31:0] | I | 写入GRF[A3] <= WD3 |
| clk | I | 时钟 |
| reset | I | 异步复位 |
| RD1 | O | RD1= GRF[A1] |
| RD2 | O | RD2= GRF[A2] |

（可使用表格进行端口说明）

#### NPC

|  |  |  |
| --- | --- | --- |
| NPC | IO | Description |
| F\_PC[31:0] | I |  |
| D\_PC[31:0] | I |  |
| NPCop[2:0] | I | PC4: NPC = F\_PC + 4  J: NPC = {D\_PC[31:28], imm26, 2'b00}  B: NPC = D\_PC + 4 + {{14{imm26[15]}}, imm26[15:0], 2'b00};  JR:NPC=jreg(可能会转发) |
| Imm26[25:0] | I |  |
| Jreg[31:0] | I |  |
| **NPC** | IO | Description |
| F\_PC[31:0] | I |  |
| D\_PC[31:0] | I |  |
| NPCop[2:0] | I | PC4: NPC = F\_PC + 4  J: NPC = {D\_PC[31:28], imm26, 2'b00}  B: NPC = D\_PC + 4 + {{14{imm26[15]}}, imm26[15:0], 2'b00};  JR:NPC=jreg(可能会转发) |

1. E级

#### ALU

提供 32 位加、减、或运算及大小比较功能。

可以不支持溢出（不检测溢出）。

|  |  |  |
| --- | --- | --- |
| ALU | IO | Description |
| A[31:0] | I | Operand1 通常RD1 |
| B[31:0] | I | Operand2 ALUsrc进行RD2, 和 (sign/zero)ext\_imm的选择 |
| ALUop[3:0] | I | 自定义的ALUcontrol |
| overflow | O | 溢出检测 |
| zero | O | zero = (A == B) |
| ALUresult[4:0] | O | 计算结果 |

#### M级

#### DM

课程功能要求：

使用 RAM 实现，容量为 32bit \* 32，应具有异步复位功能，复位值为 0x00000000。

起始地址：0x00000000。

RAM 应使用双端口模式，即设置 RAM 的 Data Interface 属性为 Separate load and store ports。

|  |  |  |
| --- | --- | --- |
| DM | IO | Description |
| A[31:0] | I | 读地址/写地址 |
| WD[31:0] | I | 写入数据 |
| clk | I | 时钟 |
| WE | I | 写使能 |
| RD | O | 读数据 |

1. W级

#### Control Unit

使用与或门阵列构造控制信号，

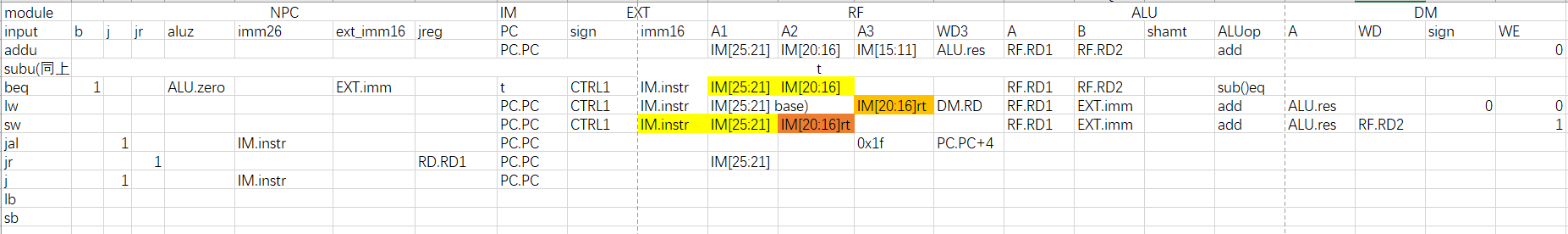
|  |  |  |
| --- | --- | --- |
| ALU | IO | Description |
| opcode[5:0] | I | Instr[31:26] |
| func[5:0] | I | Instr[5:0] |
| RegWrite | O | GRF的写入控制 |
| RegDst[1:0] | O | GRF A3 控制  00: rt (Itype)  01: rd(Rtype)  10: 31(Jal, link) |
| Branch | O | 分支指令和zero一同控制NPC |
| MemWrite | O | 计算结果 |
| ALUSrc | O | 0: B = RD2  1: B = ext\_imm32 |
| WhichtoReg[1:0] | O | GRF WD选择  00: WD = ALU\_result  01: WD = DM RD  10: WD = lui\_imm  11: WD = PC+4 |
| jr | O | jr 指令 控制 NPC |
| j | O | Jtype/ jal… 指令 控制 NPC |
| sign | O | 0：zero\_ext(imm16)  当Itype逻辑运算ori andi  1: sign\_ext(imm16)  当Itype分支指令、算术运算 |
| ALU\_op | O | 见ALU |

### （三）真值表与数据通路控制表

最初版真值表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | op | func | Regwrite | RegDst | ALUsrc | Branch | MemWrite | MemtoReg | sllv? |
| add | 0 | 100000 | 1 | b01 | 0 | 0 | 0 | 0 |  |
| sub | 0 | 100010 | 1 | b01 | 0 | 0 | 0 | 0 |  |
| and | 0 | 100100 | 1 | b01 | 0 | 0 | 0 | 0 |  |
| or | 0 | 100101 | 1 | b01 | 0 | 0 | 0 | 0 |  |
| sll | 0 | 0 | 1 | b01 | 0 | 0 | 0 | 0 | 0use instr shamt |
| sllv | 0 | 100 | 1 | b01 | 0 | 0 | 0 | 0 | 1use RD1 |
| slt | 0 | 101010 | 1 | b01 | 0 | 0 | 0 | 0 |  |
| jr | 0 | 1000 | 0 | x1 | 0 | 0 | 0 | 0 |  |
| jalr(x) | 0 | 1001 | 1 | b10($31) |  |  |  |  |  |
| j | 10 |  | 0 | x0 | 0 | 0 | 0 | 0 |  |
| jal | 11 | 0 | 1 | b10($31) | 0 | 0 | 0 | 0 |  |
| addi | 1000 |  | 1 | b00 | 1 | 0 | 0 | 0 |  |
| addiu(x) | 1001 |  | 1 | b00 | 1 | 0 | 0 | 0 |  |
| ori | 1101 |  | 1 | b00 | 1 | 0 | 0 | 0 |  |
| lui | 1111 |  | 1 | b00 | 1 | 0 | 0 | 0 |  |
| slti | 1010 |  | 1 | b00 | 1 | 0 | 0 | 0 |  |
| sw | 101011 |  | 0 | x0 | 1 | 0 | 1 | 0 |  |
| lw | 100011 |  | 1 | b00 | 1 | 0 | 0 | 1 |  |
| beq | 100 |  | 0 | x | 0 | 1 |  |  |  |
| blez(x) | 110 |  |  |  |  |  |  |  |  |

重新改版：



转发表



暂停表



## 二、测试方案

### （一）典型测试样例

覆盖测试：

ori $t0,11

jal eea

addu $t0,$ra,$0

eea:

ori $t0,11

jal eeb

ori $t0,$ra,11

eeb:

ori $t0,11

addu $t1,$t2,$t0

nop

addu $t2,$t1,$t0

ori $t0,11

addu $t1,$t2,$t0

nop

ori $t0,$t1,1

addu $a0,$0,$0

nop

lw $t0,0($a0)

ori $t0,111

addu $a0,$0,$0

nop

sw $t0,0($a0)

ori $t1,1

addu $t2,$t1,$0

nop

beq $t2,$t1,outa

nop

outa:

nop

ori $t1,$0,0x00003080

addu $t2,$t1,$0

nop

jr $t2

nop

ori $t0,11

ori $t1,$t2,0

nop

addu $t2,$t1,$t0

ori $t0,11

ori $t1,$t2,0

nop

ori $t0,$t1,1

ori $a0,$0,0

nop

lw $t0,0($a0)

ori $t0,111

ori $a0,$0,0

nop

sw $t0,0($a0)

ori $t1,1

ori $t2,$t1,0

nop

beq $t2,$t1,outb

nop

outb:

nop

ori $t1,$0,0x000030e8

ori $t2,$t1,0

nop

jr $t2

nop

ori $t0,11

jal eec

nop

addu $t0,$ra,$0

eec:

ori $t0,11

jal eed

nop

ori $t0,$ra,11

eed:

ori $a0,$0,0x00003000

jal eee

subu $ra,$ra,$a0

eee:

lw $t1,0($ra)

ori $t1,1

ori $a0,$0,0x00003000

jal eef

subu $ra,$ra,$a0

eef:

sw $t1,0($ra)

ori $t0,11

addu $t1,$t2,$t0

nop

nop

addu $t2,$t1,$t0

ori $t0,11

addu $t1,$t2,$t0

nop

nop

ori $t0,$t1,1

addu $a0,$0,$0

nop

nop

lw $t0,0($a0)

ori $t0,111

addu $a0,$0,$0

nop

nop

sw $t0,0($a0)

ori $t1,1

addu $t2,$t1,$0

nop

nop

beq $t2,$t1,outc

nop

outc:

nop

ori $t1,$0,0x000031ac

addu $t2,$t1,$0

nop

nop

jr $t2

nop

ori $t0,11

ori $t1,$t2,0

nop

nop

addu $t2,$t1,$t0

ori $t0,11

ori $t1,$t2,0

nop

nop

ori $t0,$t1,1

ori $a0,$0,0

nop

nop

lw $t0,0($a0)

ori $t0,111

ori $a0,$0,0

nop

nop

sw $t0,0($a0)

ori $t1,1

ori $t2,$t1,0

nop

nop

beq $t2,$t1,outd

nop

outd:

nop

ori $t1,$0,0x0000322c

ori $t2,$t1,0

nop

nop

jr $t2

nop

ori $t1,1

ori $t0,$0,0x00000000

lw $t1,0($t0)

nop

nop

addu $t2,$t1,$t0

ori $t1,1

ori $t0,$0,0x00000000

lw $t1,0($t0)

nop

nop

ori $t0,$t1,1

ori $t0,$0,0x00000000

lw $t1,0($t0)

nop

nop

ori $t1,$0,0x00000000

lw $t0,0($t1)

ori $t0,$0,0x00000000

lw $t1,0($t0)

nop

nop

ori $t1,$0,0x00000000

sw $t0,0($t1)

ori $t1,1

ori $t0,$0,0x00000000

lw $t1,0($t0)

ori $t2,$t1,0

nop

beq $t1,$t2,oute

nop

oute:

nop

ori $t0,$0,0x00000000

ori $t2,$0,0x000032cc

sw $t2,0($t0)

lw $t1,0($t0)

nop

nop

jr $t1

nop

ori $t0,11

jal eeg

nop

nop

addu $t0,$ra,$0

eeg:

ori $t0,11

jal eeh

nop

nop

ori $t0,$ra,11

eeh:

ori $a0,$0,0x00003000

jal eei

subu $ra,$ra,$a0

eei:

nop

lw $t1,0($ra)

ori $t1,1

ori $a0,$0,0x00003000

jal eej

subu $ra,$ra,$a0

eej:

nop

sw $t1,0($ra)

ori $t0,$0,11

jal eek

addu $t0,$0,$ra

eek:

ori $t0,$0,11

addu $t1,$t2,$t0

nop

addu $t2,$t0,$t1

ori $t0,$0,111

addu $a0,$t0,$0

nop

sw $a0,0($0)

ori $t1,$0,1

addu $t2,$t1,$0

nop

beq $t1,$t2,outf

nop

outf:

nop

ori $t0,$0,11

ori $t1,$t2,0

nop

addu $t2,$t0,$t1

ori $a0,$0,111

nop

sw $a0,0($0)

ori $t1,$0,1

ori $t2,$t1,0

nop

beq $t1,$t2,outg

nop

outg:

nop

ori $t0,$0,11

jal eel

nop

addu $t0,$0,$ra

eel:

ori $t1,$0,1

ori $a0,$0,0x00003000

jal eem

subu $ra,$ra,$a0

eem:

sw $ra,0($0)

ori $t0,$0,11

addu $t1,$t2,$t0

nop

nop

addu $t2,$t0,$t1

ori $t0,$0,111

addu $a0,$0,$0

nop

nop

sw $a0,0($0)

ori $t1,$0,1

addu $t2,$t1,$0

nop

nop

beq $t1,$t2,outh

nop

outh:

nop

ori $t0,$0,11

ori $t1,$t2,0

nop

nop

addu $t2,$t0,$t1

ori $t0,$0,111

ori $a0,$0,0

nop

nop

sw $a0,0($0)

ori $t1,$0,1

ori $t2,$t1,0

nop

nop

beq $t1,$t2,outi

nop

outi:

nop

ori $t1,$0,1

ori $t0,$0,0x00000000

lw $t1,0($t0)

nop

nop

addu $t2,$t0,$t1

ori $t0,$0,0x00000000

lw $t1,0($t0)

nop

nop

sw $t1,0($0)

ori $t1,$0,1

ori $t0,$0,0x00000000

lw $t1,0($t0)

ori $t2,$t1,0

nop

beq $t2,$t1,outj

nop

outj:

nop

ori $t0,$0,11

jal een

nop

nop

addu $t0,$0,$ra

een:

ori $t1,$0,1

ori $a0,$0,0x00003000

jal eeo

subu $ra,$ra,$a0

eeo:

nop

sw $ra,0($0)

ori $t2,$0,111

addu $t1,$t2,$t3

subu $t4,$t1,$0

ori $t2,$0,111

subu $t4,$t2,$0

ori $t2,$t4,111

addu $t1,$t2,$t3

ori $t1,$0,0x00000000

lw $t2,0($t1)

addu $t1,$t2,$t3

ori $t1,$0,0x00000000

sw $t2,0($t1)

ori $t2,$0,111

ori $t1,$t2,0

subu $t4,$t1,$0

ori $t2,$0,111

ori $t4,$t2,0

ori $t1,$t2,3#####

ori $t1,$0,0x00000000

lw $t2,0($t1)

ori $t1,$t2,3#####

ori $t1,$0,0x00000000

sw $t2,0($t1)

jal eep

subu $t1,$ra,$t2

eep:

jal eeq

lui $t8,11

eeq:

jal eer

lw $0,0($0)

eer:

jal ees

sw $0,0($0)

ees:

ori $t2,$0,111

addu $t1,$t2,$t3

nop

subu $t4,$t1,$0

ori $t2,$0,111

subu $t4,$t2,$0

nop

ori $t2,$t4,111

addu $t1,$t2,$t3

nop

ori $t1,$0,0x00000000

lw $t2,0($t1)

addu $t1,$t2,$t3

nop

sw $t2,0($t1)

ori $t2,$0,111

ori $t1,$t2,0

nop

subu $t4,$t1,$0

ori $t2,$0,111

nop

ori $t4,$t2,0

ori $t1,$t2,3#####

nop

ori $t1,$0,0x00000000

lw $t2,0($t1)

ori $t1,$t2,3#####

nop

ori $t1,$0,0x00000000

sw $t2,0($t1)

ori $t1,$0,1

ori $t0,$0,0x00000000

lw $t1,0($t0)

nop

addu $t2,$t1,$t0

ori $t1,$0,1

ori $t0,$0,0x00000000

lw $t1,0($t0)

nop

ori $t0,$t1,1

ori $t0,$0,0x00000000

lw $t1,0($t0)

nop

lw $t0,0($t1)

ori $t0,$0,0x00000000

lw $t1,0($t0)

nop

sw $t0,0($t1)

jal eet

nop

eet:

subu $t1,$ra,$t2

jal eeu

nop

eeu:

lui $ra,111

ori $t1,$0,0x00003000

jal eev

eev:

subu $ra,$ra,$t1

lw $0,0($0)

ori $t1,$0,0x00003000

jal eew

eew:

subu $ra,$ra,$t1

sw $0,0($0)

ori $t2,$0,111

addu $t1,$t2,$t3

subu $t4,$0,$t1

addu $t1,$t2,$t3

ori $t2,$0,0x00000000

sw $t1,0($t2)

ori $t2,$0,111

ori $t1,$t2,0

subu $t4,$0,$t1

ori $t1,$t2,100

ori $t1,$0,0x00000000

sw $t1,0($t1)

jal eex

subu $t1,$t2,$ra

eex:

jal eey

sw $ra,0($0)

eey:

ori $t2,$0,111

addu $t1,$t2,$t3

nop

subu $t4,$0,$t1

addu $t1,$t2,$t3

nop

ori $t2,$0,0x00000000

sw $t1,0($t2)

ori $t2,$0,111

ori $t1,$t2,0

nop

subu $t4,$0,$t1

ori $t1,$t2,3#####

nop

ori $t2,$0,0x00000000

sw $t1,0($t2)

ori $t1,$0,1

ori $t0,$0,0x00000000

lw $t1,0($t0)

nop

addu $t2,$t0,$t1

ori $t0,$0,0x00000000

lw $t1,0($t0)

nop

sw $t1,0($t0)

jal eez

nop

eez:

subu $t1,$t2,$ra

ori $t1,$0,0x00003000

jal eeea

subu $ra,$ra,$t1

eeea:

sw $ra,0($0)

ori $t2,$0,10

addu $t1,$t2,$t3

sw $t1,0($0)

ori $t2,$0,10

ori $t1,$t2,10

sw $t1,0($0)

ori $t2,$0,10

lw $t2,0($0)

sw $t2,0($0)

ori $t2,$0,10

jal eeeb

sw $ra,0($0)

eeeb:

ori $s0,$0,1

addu $s1,$s0,$0

beq $s0,$s1,eeec

nop

eeec:

nop

ori $s0,$0,1

ori $s1,$s0,2

beq $s1,$s0,eeed

nop

eeed:

nop

ori $s0,$0,10

lw $s1,0($0)

beq $s1,$s0,eeee

nop

eeee:

nop

ori $s0,$0,10

lw $s1,0($0)

nop

beq $s1,$s0,eeef

nop

eeef:

addu $t0,$t0,$t0

lw $t2,0($0)

addu $t2,$t2,$t2

lw $t2,0($0)

ori $t2,$t2,100

lw $t2,0($0)

ori $t2,$0,0x00000000

lw $t3,0($t2)

lw $t2,0($0)

ori $t2,$0,0x00000000

sw $t3,0($t2)

ori $t3,$0,0x0000300c

addu $t2,$0,0x37b4

jr $t2

nop

ori $t3,$0,0x000037c4

ori $t2,$t3,0

jr $t2

nop

ori $t3,$0,0x000037d8

sw $t3,0($0)

lw $t2,0($0)

jr $t2

nop

ori $t3,$0,0x000037f0

sw $t3,0($0)

lw $t2,0($0)

nop

jr $t2

nop

ori $t1,$t1,1

强测代码：

.text

init\_1:j init\_44

lui $0, 58479

init\_2:j init\_61

lui $26, 40699

init\_3:nop

j init\_26

ori $29, 12340

init\_4:j init\_21

lui $9, 18793

init\_5:nop

j init\_9

ori $13, 19610

init\_6:j init\_38

ori $28, 17819

init\_7:nop

j init\_37

ori $27, 17810

init\_8:j init\_48

ori $14, 1324

init\_9:j init\_8

lui $14, 20958

init\_10:j init\_31

lui $4, 28505

init\_11:nop

j init\_13

ori $23, 24263

init\_12:j init\_16

ori $12, 2525

init\_13:j init\_62

lui $24, 49213

init\_14:j init\_59

ori $6, 27235

init\_15:j init\_52

ori $16, 28030

init\_16:j init\_5

lui $13, 39021

init\_17:j init\_23

lui $20, 45636

init\_18:nop

j init\_57

ori $7, 34738

init\_19:j init\_56

lui $22, 48232

init\_20:j init\_7

lui $27, 21269

init\_21:nop

j init\_24

ori $9, 60940

init\_22:j init\_47

lui $19, 24579

init\_23:j init\_53

ori $20, 23617

init\_24:j init\_27

lui $10, 4700

init\_25:nop

j init\_60

ori $5, 25135

init\_26:j init\_34

lui $30, 14559

init\_27:j init\_29

ori $10, 45253

init\_28:j init\_12

lui $12, 55820

init\_29:j init\_36

lui $11, 49875

init\_30:j init\_25

lui $5, 35220

init\_31:j init\_30

ori $4, 7615

init\_32:j begin

lui $31, 60984

init\_33:j init\_43

ori $2, 18084

init\_34:j init\_63

ori $30, 41019

init\_35:j init\_15

lui $16, 54272

init\_36:nop

j init\_28

ori $11, 2111

init\_37:j init\_6

lui $28, 33755

init\_38:j init\_3

lui $29, 26291

init\_39:nop

j init\_2

ori $25, 48740

init\_40:j init\_42

lui $1, 43965

init\_41:nop

j init\_19

ori $21, 27953

init\_42:nop

j init\_50

ori $1, 18337

init\_43:j init\_54

lui $3, 36555

init\_44:j init\_40

ori $0, 39840

init\_45:j init\_22

ori $18, 28396

init\_46:j init\_4

ori $8, 13173

init\_47:nop

j init\_17

ori $19, 60189

init\_48:j init\_49

lui $15, 28446

init\_49:nop

j init\_35

ori $15, 43996

init\_50:j init\_33

lui $2, 50534

init\_51:j init\_45

lui $18, 47692

init\_52:j init\_58

lui $17, 18098

init\_53:j init\_41

lui $21, 23125

init\_54:nop

j init\_10

ori $3, 34935

init\_55:j init\_39

lui $25, 37200

init\_56:j init\_64

ori $22, 41245

init\_57:j init\_46

lui $8, 16690

init\_58:nop

j init\_51

ori $17, 23659

init\_59:j init\_18

lui $7, 16431

init\_60:j init\_14

lui $6, 20586

init\_61:j init\_20

ori $26, 21724

init\_62:j init\_55

ori $24, 64211

init\_63:j init\_32

lui $31, 32491

init\_64:j init\_11

lui $23, 36591

begin:

ori $11, $11, 39941

sw $11, 0($0)

ori $22, $22, 13378

nop

sw $22, 4($0)

addu $8, $31, $16

nop

nop

sw $8, 8($0)

subu $20, $16, $6

sw $20, 12($0)

addu $9, $15, $7

nop

sw $9, 16($0)

addu $19, $23, $27

nop

nop

sw $19, 20($0)

subu $5, $1, $31

sw $5, 24($0)

addu $22, $6, $26

nop

sw $22, 28($0)

addu $10, $4, $18

nop

nop

sw $10, 32($0)

subu $19, $3, $0

sw $19, 36($0)

subu $22, $24, $1

nop

sw $22, 40($0)

subu $19, $10, $14

nop

nop

sw $19, 44($0)

subu $0, $19, $15

sw $0, 48($0)

ori $21, $21, 728

nop

sw $21, 52($0)

subu $13, $29, $4

nop

nop

sw $13, 56($0)

addu $11, $3, $1

sw $11, 60($0)

addu $27, $19, $11

nop

sw $27, 64($0)

addu $27, $16, $28

nop

nop

sw $27, 68($0)

ori $25, $25, 7272

sw $25, 72($0)

ori $31, $31, 65375

nop

sw $31, 76($0)

ori $13, $13, 65318

nop

nop

sw $13, 80($0)

ori $5, $5, 27677

sw $5, 84($0)

ori $13, $13, 30209

nop

sw $13, 88($0)

subu $16, $28, $20

nop

nop

sw $16, 92($0)

addu $6, $16, $21

sw $6, 96($0)

subu $22, $11, $31

nop

sw $22, 100($0)

subu $25, $23, $19

nop

nop

sw $25, 104($0)

ori $0, $0, 8927

sw $0, 108($0)

ori $24, $24, 21563

nop

sw $24, 112($0)

addu $13, $16, $11

nop

nop

sw $13, 116($0)

subu $14, $12, $27

sw $14, 120($0)

addu $15, $24, $27

nop

sw $15, 124($0)

subu $20, $1, $24

nop

nop

sw $20, 128($0)

subu $13, $28, $15

sw $13, 132($0)

ori $28, $28, 56842

nop

sw $28, 136($0)

addu $31, $6, $23

nop

nop

sw $31, 140($0)

ori $10, $10, 11112

sw $10, 144($0)

addu $21, $23, $9

nop

sw $21, 148($0)

subu $22, $16, $29

nop

nop

sw $22, 152($0)

addu $16, $1, $31

sw $16, 156($0)

subu $12, $15, $28

nop

sw $12, 160($0)

addu $6, $10, $22

nop

nop

sw $6, 164($0)

ori $12, $12, 14691

sw $12, 168($0)

ori $16, $16, 34145

nop

sw $16, 172($0)

ori $11, $11, 18551

nop

nop

sw $11, 176($0)

sw $sp, 180($0)

sw $ra, 184($0)

sw $at, 188($0)

ori $sp, $0, 4060

ori $1, $0, 32

jal foo1

nop

lui $1, 0

ori $1, 0

beq $1, $0, skip\_manual1

nop

j dl

nop

skip\_manual1:

lui $1, 10994

lui $2, 10994

beq $1, $2, skip\_manual2

nop

j dl

nop

skip\_manual2:

lui $3, 10995

nop

nop

beq $1, $3, dl

addu $4, $4, $3

lui $1, 0x6183

addu $2, $2, $1

lui $5, 0x8124

addu $4, $5, $1

subu $6, $6, $5

jal skip\_manual3

nop

sw $0, 4($0)

skip\_manual3:

sw $7, -0x3000($ra)

lw $ra, -0x3000($ra)

ori $ra, $0, 0

jal skip\_manual4

nop

sw $0, 8($0)

skip\_manual4:

beq $ra, $0, dl

nop

ori $ra, $0, 0

jal skip\_manual5

nop

sw $ra, 12($0)

skip\_manual5:

nop

beq $ra, $0, dl

nop

ori $4, $0, 4

ori $5, $0, 5

ori $1, $0, 1

addu $4, $4, $1

nop

nop

beq $4, $5, skip\_manual6

nop

sw $0, 16($0)

skip\_manual6:

ori $1, $0, 1

ori $2, $0, 2

ori $3, $0, 3

ori $4, $0, 4

ori $5, $0, 6

ori $6, $0, 5

subu $5, $5, $1

addu $6, $2, $1

beq $5, $6, dl

nop

jal skip\_manual8

nop

skip\_manual8:

addu $3, $3, $ra

subu $4, $4, $ra

jal foo

nop

jal fooo

nop

jal foooo

nop

jal fooooo

nop

sw $0, 192($0)

sw $1, 196($0)

sw $2, 200($0)

sw $3, 204($0)

sw $4, 208($0)

sw $5, 212($0)

sw $6, 216($0)

sw $7, 220($0)

sw $8, 224($0)

sw $9, 228($0)

sw $10, 232($0)

sw $11, 236($0)

sw $12, 240($0)

sw $13, 244($0)

sw $14, 248($0)

sw $15, 252($0)

sw $16, 256($0)

sw $17, 260($0)

sw $18, 264($0)

sw $19, 268($0)

sw $20, 272($0)

sw $21, 276($0)

sw $22, 280($0)

sw $23, 284($0)

sw $24, 288($0)

sw $25, 292($0)

sw $26, 296($0)

sw $27, 300($0)

sw $28, 304($0)

sw $29, 308($0)

sw $30, 312($0)

sw $31, 316($0)

lui $31, 63605

jal tag\_0

nop

ori $ra, $0, 2

tag\_0:sw $ra, 320($0)

addu $14, $13, $7

jal tag\_1

nop

ori $ra, $0, 2

tag\_1:sw $ra, 324($0)

lw $25, 12($0)

jal tag\_2

nop

ori $ra, $0, 2

tag\_2:sw $ra, 328($0)

lui $29, 14949

nop

jal tag\_3

nop

ori $ra, $0, 2

tag\_3:sw $ra, 332($0)

addu $10, $24, $1

nop

jal tag\_4

nop

ori $ra, $0, 2

tag\_4:sw $ra, 336($0)

lw $12, 164($0)

nop

jal tag\_5

nop

ori $ra, $0, 2

tag\_5:sw $ra, 340($0)

lui $31, 58593

nop

nop

jal tag\_6

nop

ori $ra, $0, 2

tag\_6:sw $ra, 344($0)

addu $1, $7, $12

nop

nop

jal tag\_7

nop

ori $ra, $0, 2

tag\_7:sw $ra, 348($0)

lw $20, 256($0)

nop

nop

jal tag\_8

nop

ori $ra, $0, 2

tag\_8:sw $ra, 352($0)

lui $17, 19367

jal tag\_9

nop

ori $ra, $0, 2

tag\_9:sw $ra, 356($0)

addu $8, $11, $11

jal tag\_10

nop

ori $ra, $0, 2

tag\_10:sw $ra, 360($0)

lw $1, 132($0)

jal tag\_11

nop

ori $ra, $0, 2

tag\_11:sw $ra, 364($0)

lui $18, 56313

nop

jal tag\_12

nop

ori $ra, $0, 2

tag\_12:sw $ra, 368($0)

addu $23, $25, $24

nop

jal tag\_13

nop

ori $ra, $0, 2

tag\_13:sw $ra, 372($0)

lw $20, 368($0)

nop

jal tag\_14

nop

ori $ra, $0, 2

tag\_14:sw $ra, 376($0)

lui $21, 22951

nop

nop

jal tag\_15

nop

ori $ra, $0, 2

tag\_15:sw $ra, 380($0)

addu $8, $10, $1

nop

nop

jal tag\_16

nop

ori $ra, $0, 2

tag\_16:sw $ra, 384($0)

lw $4, 88($0)

nop

nop

jal tag\_17

nop

ori $ra, $0, 2

tag\_17:sw $ra, 388($0)

ori $4, $0, 12

jal skip\_manual7

nop

skip\_manual7:

addu $ra, $ra, $4

jr $ra

ori $4, $0, 8

addu $ra, $ra, $4

nop

jr $ra

nop

dl:addu $ra, $0, $0

beq $0, $0, dl

nop

foo: jr $ra

ori $ra, $ra, 0xff

fooo: ori $6, $ra, 0xa

jr $ra

nop

foooo: jr $ra

ori $t8, $ra, 0xff

fooooo: ori $t9, $ra, 0xa

jr $ra

nop

foo1:

sw $a0, 0($sp)

sw $ra, 16($sp)

sw $a1, 4($sp)

sw $a2, 8($sp)

sw $a3, 12($sp)

sw $t0, 20($sp)

sw $t1, 24($sp)

sw $t2, 28($sp)

ori $s1, $0, 84

ori $s2, $0, 220

lw $t0, -40($s1)

lw $t1, -88($s2)

addu $a3, $t0, $t1

addu $a0, $a0, $t0

addu $a1, $a1, $t1

addu $t2, $a0, $a1

ori $a0, $t2, 11072

addu $t2, $t2, $t2

addu $t2, $t2, $t2

ori $a1, $t2, 30986

ori $a2, $a1, 0xf0

nop

beq $a2, $a1, skip1

nop

jal foo2

subu $sp, $sp, $1

skip1: lw $a3, 16($sp)

addu $sp, $sp, $1

nop

jr $a3

nop

foo2:

sw $a0, 0($sp)

sw $a1, 4($sp)

sw $ra, 16($sp)

sw $a2, 8($sp)

sw $a3, 12($sp)

sw $t0, 20($sp)

sw $t1, 24($sp)

sw $t2, 28($sp)

ori $s1, $0, 152

ori $s2, $0, 54

lw $t1, 326($s2)

lw $t0, 160($s1)

addu $a3, $t0, $t1

addu $a0, $a0, $t0

addu $a1, $a1, $t1

addu $t2, $a0, $a1

ori $a0, $t2, 21109

addu $t2, $t2, $t2

addu $t2, $t2, $t2

ori $a1, $t2, 60683

ori $a2, $a1, 0xf0

beq $a2, $a1, skip2

nop

jal foo3

subu $sp, $sp, $1

skip2: lw $a3, 16($sp)

jr $a3

addu $sp, $sp, $1

foo3:

sw $ra, 16($sp)

sw $a0, 0($sp)

sw $a1, 4($sp)

sw $a2, 8($sp)

sw $a3, 12($sp)

sw $t0, 20($sp)

sw $t1, 24($sp)

sw $t2, 28($sp)

ori $s1, $0, 139

ori $s2, $0, 302

lw $t0, -35($s1)

lw $t1, -34($s2)

addu $a3, $t0, $t1

addu $a0, $a0, $t0

addu $a1, $a1, $t1

addu $t2, $a0, $a1

ori $a0, $t2, 29202

addu $t2, $t2, $t2

addu $t2, $t2, $t2

ori $a1, $t2, 43269

ori $a2, $a1, 0xf0

nop

beq $a2, $a1, skip3

nop

jal foo13

subu $sp, $sp, $1

skip3: lw $a3, 16($sp)

addu $sp, $sp, $1

jr $a3

nop

foo4:

sw $a0, 0($sp)

sw $ra, 16($sp)

sw $a1, 4($sp)

sw $a2, 8($sp)

sw $a3, 12($sp)

sw $t0, 20($sp)

sw $t1, 24($sp)

sw $t2, 28($sp)

ori $s1, $0, 297

ori $s2, $0, 96

lw $t1, 104($s2)

lw $t0, -197($s1)

addu $a3, $t0, $t1

addu $a0, $a0, $t0

addu $a1, $a1, $t1

addu $t2, $a0, $a1

ori $a0, $t2, 14171

addu $t2, $t2, $t2

addu $t2, $t2, $t2

ori $a1, $t2, 18179

ori $a2, $a1, 0xf0

beq $a2, $a1, skip4

nop

jal foo11

subu $sp, $sp, $1

skip4: lw $a3, 16($sp)

addu $sp, $sp, $1

nop

jr $a3

nop

foo5:

sw $a0, 0($sp)

sw $a1, 4($sp)

sw $ra, 16($sp)

sw $a2, 8($sp)

sw $a3, 12($sp)

sw $t0, 20($sp)

sw $t1, 24($sp)

sw $t2, 28($sp)

ori $s1, $0, 106

ori $s2, $0, 190

lw $t0, -26($s1)

lw $t1, -82($s2)

addu $a3, $t0, $t1

addu $a0, $a0, $t0

addu $a1, $a1, $t1

addu $t2, $a0, $a1

ori $a0, $t2, 49045

addu $t2, $t2, $t2

addu $t2, $t2, $t2

ori $a1, $t2, 36619

ori $a2, $a1, 0xf0

nop

beq $a2, $a1, skip5

nop

jal foo1

subu $sp, $sp, $1

skip5: lw $a3, 16($sp)

jr $a3

addu $sp, $sp, $1

foo6:

sw $ra, 16($sp)

sw $a0, 0($sp)

sw $a1, 4($sp)

sw $a2, 8($sp)

sw $a3, 12($sp)

sw $t0, 20($sp)

sw $t1, 24($sp)

sw $t2, 28($sp)

ori $s1, $0, 345

ori $s2, $0, 52

lw $t1, -32($s2)

lw $t0, 15($s1)

addu $a3, $t0, $t1

addu $a0, $a0, $t0

addu $a1, $a1, $t1

addu $t2, $a0, $a1

ori $a0, $t2, 25874

addu $t2, $t2, $t2

addu $t2, $t2, $t2

ori $a1, $t2, 45316

ori $a2, $a1, 0xf0

beq $a2, $a1, skip6

nop

jal foo4

subu $sp, $sp, $1

skip6: lw $a3, 16($sp)

addu $sp, $sp, $1

jr $a3

nop

foo7:

sw $a0, 0($sp)

sw $ra, 16($sp)

sw $a1, 4($sp)

sw $a2, 8($sp)

sw $a3, 12($sp)

sw $t0, 20($sp)

sw $t1, 24($sp)

sw $t2, 28($sp)

ori $s1, $0, 319

ori $s2, $0, 212

lw $t0, -167($s1)

lw $t1, 120($s2)

addu $a3, $t0, $t1

addu $a0, $a0, $t0

addu $a1, $a1, $t1

addu $t2, $a0, $a1

ori $a0, $t2, 44079

addu $t2, $t2, $t2

addu $t2, $t2, $t2

ori $a1, $t2, 38407

ori $a2, $a1, 0xf0

nop

beq $a2, $a1, skip7

nop

jal foo10

subu $sp, $sp, $1

skip7: lw $a3, 16($sp)

addu $sp, $sp, $1

nop

jr $a3

nop

foo8:

sw $a0, 0($sp)

sw $a1, 4($sp)

sw $ra, 16($sp)

sw $a2, 8($sp)

sw $a3, 12($sp)

sw $t0, 20($sp)

sw $t1, 24($sp)

sw $t2, 28($sp)

ori $s1, $0, 366

ori $s2, $0, 244

lw $t1, -116($s2)

lw $t0, -290($s1)

addu $a3, $t0, $t1

addu $a0, $a0, $t0

addu $a1, $a1, $t1

addu $t2, $a0, $a1

ori $a0, $t2, 20552

addu $t2, $t2, $t2

addu $t2, $t2, $t2

ori $a1, $t2, 22025

ori $a2, $a1, 0xf0

beq $a2, $a1, skip8

nop

jal foo5

subu $sp, $sp, $1

skip8: lw $a3, 16($sp)

jr $a3

addu $sp, $sp, $1

foo9:

sw $ra, 16($sp)

sw $a0, 0($sp)

sw $a1, 4($sp)

sw $a2, 8($sp)

sw $a3, 12($sp)

sw $t0, 20($sp)

sw $t1, 24($sp)

sw $t2, 28($sp)

ori $s1, $0, 223

ori $s2, $0, 283

lw $t0, 13($s1)

lw $t1, -27($s2)

addu $a3, $t0, $t1

addu $a0, $a0, $t0

addu $a1, $a1, $t1

addu $t2, $a0, $a1

ori $a0, $t2, 28872

addu $t2, $t2, $t2

addu $t2, $t2, $t2

ori $a1, $t2, 52993

ori $a2, $a1, 0xf0

nop

beq $a2, $a1, skip9

nop

jal foo15

subu $sp, $sp, $1

skip9: lw $a3, 16($sp)

addu $sp, $sp, $1

jr $a3

nop

foo10:

sw $a0, 0($sp)

sw $ra, 16($sp)

sw $a1, 4($sp)

sw $a2, 8($sp)

sw $a3, 12($sp)

sw $t0, 20($sp)

sw $t1, 24($sp)

sw $t2, 28($sp)

ori $s1, $0, 228

ori $s2, $0, 255

lw $t1, -155($s2)

lw $t0, 12($s1)

addu $a3, $t0, $t1

addu $a0, $a0, $t0

addu $a1, $a1, $t1

addu $t2, $a0, $a1

ori $a0, $t2, 56866

addu $t2, $t2, $t2

addu $t2, $t2, $t2

ori $a1, $t2, 21770

ori $a2, $a1, 0xf0

beq $a2, $a1, skip10

nop

jal foo12

subu $sp, $sp, $1

skip10: lw $a3, 16($sp)

addu $sp, $sp, $1

nop

jr $a3

nop

foo11:

sw $a0, 0($sp)

sw $a1, 4($sp)

sw $ra, 16($sp)

sw $a2, 8($sp)

sw $a3, 12($sp)

sw $t0, 20($sp)

sw $t1, 24($sp)

sw $t2, 28($sp)

ori $s1, $0, 54

ori $s2, $0, 306

lw $t0, 278($s1)

lw $t1, -238($s2)

addu $a3, $t0, $t1

addu $a0, $a0, $t0

addu $a1, $a1, $t1

addu $t2, $a0, $a1

ori $a0, $t2, 34513

addu $t2, $t2, $t2

addu $t2, $t2, $t2

ori $a1, $t2, 36103

ori $a2, $a1, 0xf0

nop

beq $a2, $a1, skip11

nop

jal foo8

subu $sp, $sp, $1

skip11: lw $a3, 16($sp)

jr $a3

addu $sp, $sp, $1

foo12:

sw $ra, 16($sp)

sw $a0, 0($sp)

sw $a1, 4($sp)

sw $a2, 8($sp)

sw $a3, 12($sp)

sw $t0, 20($sp)

sw $t1, 24($sp)

sw $t2, 28($sp)

ori $s1, $0, 248

ori $s2, $0, 72

lw $t1, 152($s2)

lw $t0, 76($s1)

addu $a3, $t0, $t1

addu $a0, $a0, $t0

addu $a1, $a1, $t1

addu $t2, $a0, $a1

ori $a0, $t2, 1104

addu $t2, $t2, $t2

addu $t2, $t2, $t2

ori $a1, $t2, 24322

ori $a2, $a1, 0xf0

beq $a2, $a1, skip12

nop

jal foo9

subu $sp, $sp, $1

skip12: lw $a3, 16($sp)

addu $sp, $sp, $1

jr $a3

nop

foo13:

sw $a0, 0($sp)

sw $ra, 16($sp)

sw $a1, 4($sp)

sw $a2, 8($sp)

sw $a3, 12($sp)

sw $t0, 20($sp)

sw $t1, 24($sp)

sw $t2, 28($sp)

ori $s1, $0, 386

ori $s2, $0, 194

lw $t0, -82($s1)

lw $t1, -142($s2)

addu $a3, $t0, $t1

addu $a0, $a0, $t0

addu $a1, $a1, $t1

addu $t2, $a0, $a1

ori $a0, $t2, 18477

addu $t2, $t2, $t2

addu $t2, $t2, $t2

ori $a1, $t2, 2816

ori $a2, $a1, 0xf0

nop

beq $a2, $a1, skip13

nop

jal foo7

subu $sp, $sp, $1

skip13: lw $a3, 16($sp)

addu $sp, $sp, $1

nop

jr $a3

nop

foo14:

sw $a0, 0($sp)

sw $a1, 4($sp)

sw $ra, 16($sp)

sw $a2, 8($sp)

sw $a3, 12($sp)

sw $t0, 20($sp)

sw $t1, 24($sp)

sw $t2, 28($sp)

ori $s1, $0, 184

ori $s2, $0, 255

lw $t1, 13($s2)

lw $t0, -88($s1)

addu $a3, $t0, $t1

addu $a0, $a0, $t0

addu $a1, $a1, $t1

addu $t2, $a0, $a1

ori $a0, $t2, 26871

addu $t2, $t2, $t2

addu $t2, $t2, $t2

ori $a1, $t2, 23821

ori $a2, $a1, 0xf0

beq $a2, $a1, skip14

nop

jal foo16

subu $sp, $sp, $1

skip14: lw $a3, 16($sp)

jr $a3

addu $sp, $sp, $1

foo15:

sw $ra, 16($sp)

sw $a0, 0($sp)

sw $a1, 4($sp)

sw $a2, 8($sp)

sw $a3, 12($sp)

sw $t0, 20($sp)

sw $t1, 24($sp)

sw $t2, 28($sp)

ori $s1, $0, 264

ori $s2, $0, 366

lw $t0, -212($s1)

lw $t1, -2($s2)

addu $a3, $t0, $t1

addu $a0, $a0, $t0

addu $a1, $a1, $t1

addu $t2, $a0, $a1

ori $a0, $t2, 36680

addu $t2, $t2, $t2

addu $t2, $t2, $t2

ori $a1, $t2, 1289

ori $a2, $a1, 0xf0

nop

beq $a2, $a1, skip15

nop

jal foo14

subu $sp, $sp, $1

skip15: lw $a3, 16($sp)

addu $sp, $sp, $1

jr $a3

nop

foo16:

sw $a0, 0($sp)

sw $ra, 16($sp)

sw $a1, 4($sp)

sw $a2, 8($sp)

sw $a3, 12($sp)

sw $t0, 20($sp)

sw $t1, 24($sp)

sw $t2, 28($sp)

ori $s1, $0, 39

ori $s2, $0, 301

lw $t1, -17($s2)

lw $t0, 317($s1)

addu $a3, $t0, $t1

addu $a0, $a0, $t0

addu $a1, $a1, $t1

addu $t2, $a0, $a1

ori $a0, $t2, 706

addu $t2, $t2, $t2

addu $t2, $t2, $t2

ori $a1, $t2, 2561

ori $a2, $a1, 0xf0

beq $a2, $a1, skip16

nop

jal foo6

subu $sp, $sp, $1

skip16: lw $a3, 16($sp)

addu $sp, $sp, $1

nop

jr $a3

nop

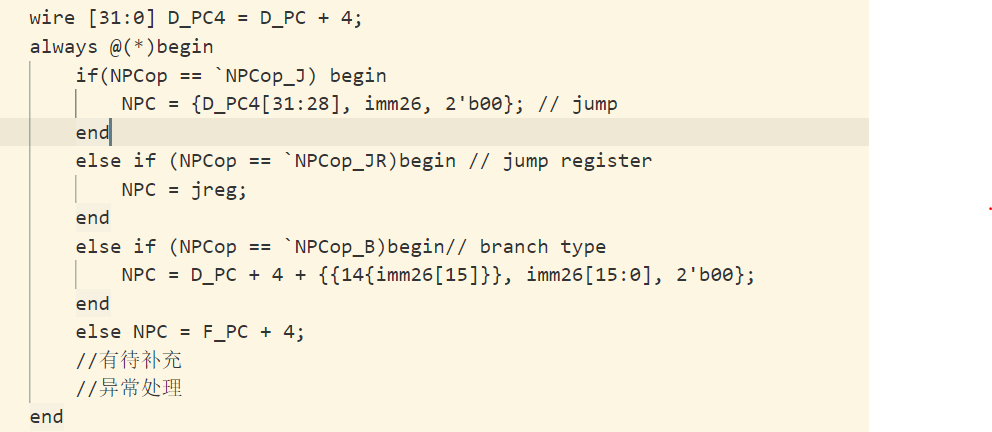
### （二）自动测试工具

文件夹里

## 三、思考题

本节思考题

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。



数据通路：NPC处于D级应当以DPC4为基地址， 同时还要获取FPC以供PC+4的运行

PC：PC+4、branch型指令、JumpRegister指令和J型指令。

控制信号：NPCop

controller解码出应当选择的NPC值即可。

1. 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？

因为要跳过延迟槽，否则jr就要回到延迟槽上了。

1. 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？

寄存器更加稳定是确定的值， ruo 采用组合逻辑计算的值，转发可能存在不稳定

1. 为了实现转发机制，我们对这些输入前加上一个 MUX。这些 MUX 的默认输入来源是上一级中**已经转发过**的数据。（ **Thinking 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。）

若采用上一级的原始数据，则前面的转发没有意义， D:add $t0, $t0, 1 E:add $t0, $t0, 1 M:$t0, $t0, 1, 当E级数据向D转发时，若采用原始数据（没有经历过M的t0+1）,则最终结果就是少加了一个1，无法连续转发。

1. （**Thinking 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？）

内部转发可以省掉 W->D的转发， 不采用的话就得加上W->D的转发了

1. 选择信号的生成规则是：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**（ **Thinking 3**：为什么 0 号寄存器需要特殊处理？），就选择该转发输入来源；在有多个转发输入来源都满足条件时，**最新产生的数据优先级最高**。（ **Thinking 4**：什么是“最新产生的数据”？）为了获取生成选择信号所需的信息，我们需要**对指令的读取寄存器和写入寄存器在 D 级进行译码并流水**（指令的“ A 信息”）

因为若仅仅读取地址相同，则当其都为0时，会发生冲突然后转发，然鹅根本不需要，因为0寄存器恒为0， 相当于写使能为0不冲突

最新就是最贴近指令的右端各级寄存器

1. 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？

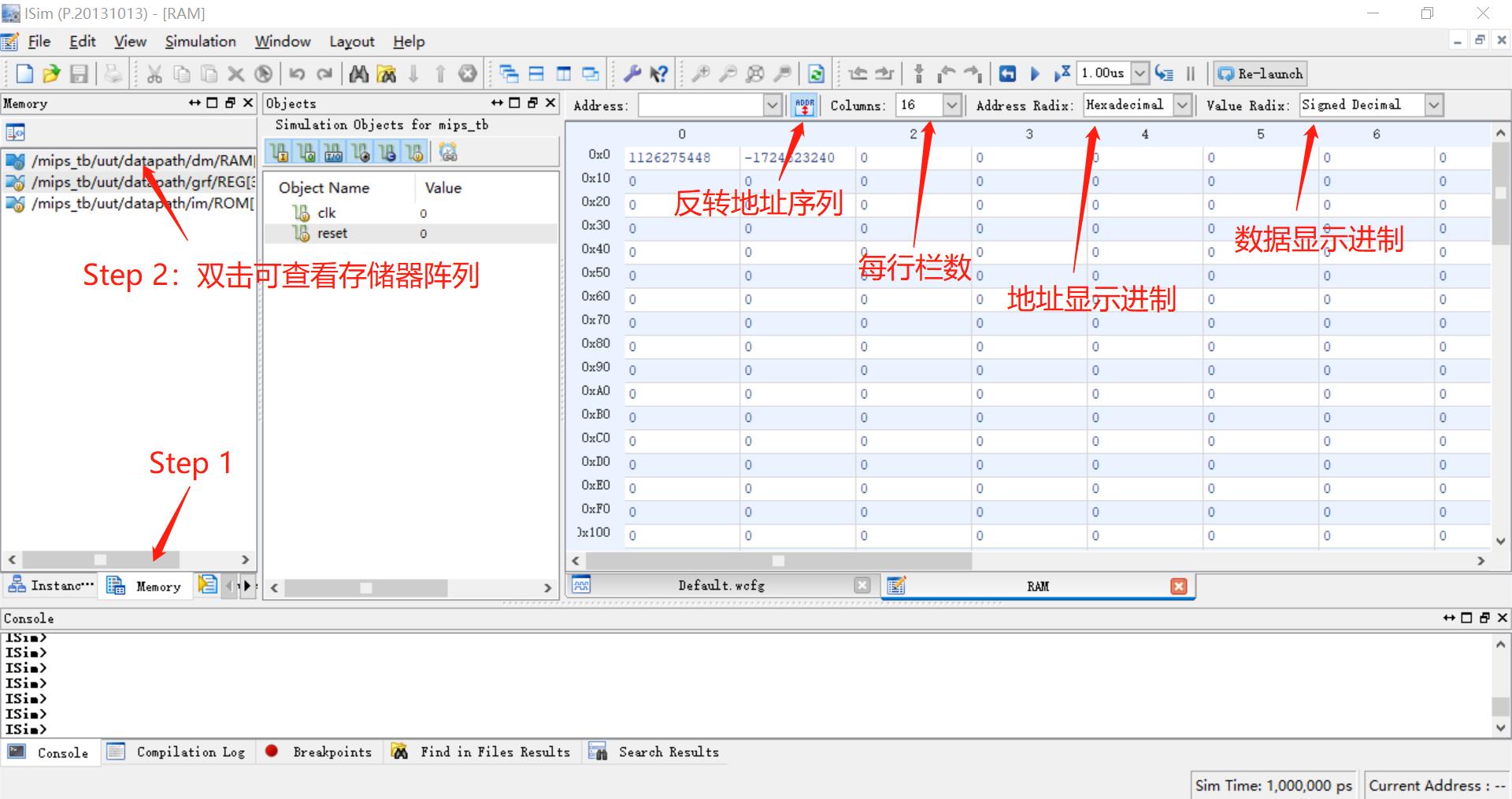
如果we是0的话把A3变成0，这样子的话， 后面的A3=0, 就不会被转发了

附件：

查看存储器数据方法

在后续的工程 Project 中，我们需要对指令存储器、内存与寄存器阵列进行仿真。要查看内存中的取值，首先要选择 Memory，双击目标存储器，可以得到下图所示的显示。

当然，这可能与刚打开的格式有所差异，我们可以使用地址反转工具，将地址由高到低变为由低到高显示，另外可以改变每一行显示数据的数目，以及显示的进制格式等。



计算类指令功能测试

* 寄存器数据方面，可以考虑以下情况：
  + 00 及附近的数：-2, -1, 0, 1, 2−2,−1,0,1,2
  + 3232 位数边界附近的数： -2147483648, -2147483647, 2147483646, 2147483647−2147483648,−2147483647,2147483646,2147483647
  + 3232 位数范围内的一些随机数：-1000786109, 1919156834, ...−1000786109,1919156834,...
* 无符号立即数方面，可以考虑以下情况：
  + 00 及附近的数：0, 1, 2, 30,1,2,3
  + 1616 位无符号数边界附近的数：65533, 65534, 6553565533,65534,65535
  + 1616 位无符号数范围内的一些随机数：25779, 42528, ...25779,42528,...
* 符号立即数 (P3 不涉及) 方面，可以考虑以下情况：
  + 00 及附近的数：-2, -1, 0, 1, 2−2,−1,0,1,2
  + 1616 位符号数边界附近的数：-32768, -32767, 32766, 32767−32768,−32767,32766,32767
  + 1616 位符号数范围内的一些随机数：-5329, 25299, ...−5329,25299,...
* 特别的，可注意测试目标寄存器是 \$0$0 的情况。

存取类指令功能测试

* offset 方面，可以考虑以下情况：
  + offset 是正数
  + offset 是零
  + offset 是负数
* $base 寄存器方面，可以考虑以下情况：
  + $base 寄存器中的值是正数
  + $base 寄存器中的值是零
  + $base 寄存器中的值是负数
* 特别的，对于 sw 指令，建议存入的 word 中，每个 byte 都不是零。
* 特别的，对于 lw 指令，可注意测试目标寄存器是 $0 的情况。

跳转类指令功能测试

* 对于非比较相关的部分，可以考虑以下情况：
  + 跳转，且目标在此跳转指令之前
  + 跳转，且目标是此跳转指令
  + 跳转，且目标在此跳转指令之后
  + 不跳转，且目标在此跳转指令之前
  + 不跳转，且目标是此跳转指令
  + 不跳转，且目标在此跳转指令之后
* 对于比较相关的部分，本质上依旧是构造寄存器数据，处理类似 “计算类指令功能测试”。