# CPU设计文档

1. 数据通路设计
2. 模块规格
3. IFU

|  |  |  |
| --- | --- | --- |
| 功能描述 | 输出PC所在位置指令，并根据控制信号计算NPC，将NPC值在下一时钟上升沿赋值给PC | |
| 信号名称 | 方向 | 描述 |
| clk | input | 时钟信号 |
| reset | input | 复位信号 |
| [25:0]imm | input | 立即数输入 |
| ifequal | input | beq指令时判断是否跳转 |
| [31:0] BUSA | input | rs寄存器的值 |
| [1:0] PCsel | input | 选择NPC的计算方式 |
| [31:0] instruction | output | PC当前指令输出 |
| [31:0] PCadd4 | output | PC+4输出 |

1. GRF

|  |  |  |
| --- | --- | --- |
| 功能描述 | 寄存器堆，分别根据A1,A2地址输出相应编号的寄存器的数据到BUSA,BUSB中；  当使能信号有效时，将WD的值输入到A3对应编号的寄存器中 | |
| 信号名称 | 方向 | 描述 |
| clk | input | 时钟信号 |
| reset | input | 复位信号 |
| [4:0] A1 | input | 第一个输出寄存器编号 |
| [4:0] A2 | input | 第二个输出寄存器编号 |
| [4:0] A3 | input | 带输入数据寄存器编号 |
| [31:0] WD | input | 待输入32位数据 |
| En | input | 使能信号 |
| [31:0] PCadd4 | input | 方便display输出 |
| [31:0] BUSA | input | 第一个32位输出 |
| [31:0] BUSB | input | 第二个32位输出 |

1. EXT

|  |  |  |
| --- | --- | --- |
| 功能描述 | 根据EXTsel选择拓展方式，将输入的的立即数进行拓展 | |
| 信号名称 | 方向 | 描述 |
| [15:0] imm | input | 16位立即数输入 |
| [1:0] EXTsel | input | 选择拓展方式 |
| [31:0] EXTout | output | 拓展结果输出 |

1. compare

|  |  |  |
| --- | --- | --- |
| 功能描述 | 若输入值相同，输出1，反之输出0 | |
| 信号名称 | 方向 | 描述 |
| [31:0] num1 | input | 第一个数据 |
| [31:0] num2 | input | 第二个数据 |
| ifequal | output | if(num1==num2)ifequal=1;  else ifequal=0; |

1. ALU

|  |  |  |
| --- | --- | --- |
| 功能描述 | 根据ALUsel,选择计算方式 | |
| 信号名称 | 方向 | 描述 |
| [31:0] A | input | 第一个32位操作数 |
| [31:0] B | input | 第二个32位操作数 |
| [2:0] ALUsel | input | 000:A+B  001:A-B  010:A|B |
| [31:0] C | output | 32位输出 |

1. DM

|  |  |  |
| --- | --- | --- |
| 功能描述 | 4K存储模块。  使能信号有效时，将DI输入Address所对应的地址中。并将Address所对应的数据输出至DMout | |
| 信号名称 | 方向 | 描述 |
| clk | input | 时钟信号 |
| reset | input | 复位信号 |
| [31:0] Address | input | 地址 |
| [31:0] DI | input | 32位带输入数据 |
| DMEn | input | 使能信号 |
| [31:0] PCadd4 | input | 方便display输出 |
| [31:0] DMout | output | 32位输出数据 |

1. D\_register

|  |  |  |
| --- | --- | --- |
| 功能描述 | D级流水寄存器,使能信号有效时，下一时钟上升沿读入信息。清零信号有效时，同步清零。  输出下一级信息 | |
| 信号名称 | 方向 | 描述 |
| clk | input | 时钟信号 |
| reset | input | 复位信号 |
| [31:0] IF | input | 上一级信息 |
| [31:0] PCadd4 | input | 上一级信息 |
| En | input | 使能信号 |
| clear | input | 清零信号 |
| [31:0] D\_IF | output | 下一级信息 |
| [31:0] D\_PCadd4 | output | 下一级信息 |

1. E\_register

|  |  |  |
| --- | --- | --- |
| 功能描述 | E级流水线寄存器。使能信号有效是读入上一级信息。  输出下一级信息 | |
| 信号名称 | 方向 | 描述 |
| clk | input | 时钟信号 |
| reset | input | 复位信号 |
| clear | input | 清零信号 |
| [31:0] IF | input | 上一级信息 |
| [31:0] PCadd4 | input | 上一级信息 |
| [31:0] BUSA | input | 上一级信息 |
| [31:0] BUSB | input | 上一级信息 |
| [31:0] EXTout | input | 上一级信息 |
| [2:0] ALUsel | input | 上一级信息 |
| Bsel | input | 上一级信息 |
| DMEn | input | 上一级信息 |
| [1:0] A3sel | input | 上一级信息 |
| [1:0] WDsel | input | 上一级信息 |
| GRFEn | input | 上一级信息 |
|  |  |  |
| [31:0] E\_IF | output | 下一级信息 |
| [31:0] E\_PCadd4 | output | 下一级信息 |
| [31:0] E\_ BUSA | output | 下一级信息 |
| [31:0] E\_BUSB | output | 下一级信息 |
| [31:0] E\_EXTout | output | 下一级信息 |
| [2:0] E\_ALUsel | output | 下一级信息 |
| E\_Bsel | output | 下一级信息 |
| E\_DMEn | output | 下一级信息 |
| [1:0] E\_A3sel | output | 下一级信息 |
| [1:0] E\_WDsel | output | 下一级信息 |
| E\_GRFEn | output | 下一级信息 |

1. M\_register

|  |  |  |
| --- | --- | --- |
| 功能描述 | M级流水线寄存器。使能信号有效是读入上一级信息。  输出下一级信息 | |
| 信号名称 | 方向 | 描述 |
| clk | input | 时钟信号 |
| reset | input | 复位信号 |
| [31:0] IF | input | 上一级信息 |
| [31:0] PCadd4 | input | 上一级信息 |
| [31:0] BUSB | input | 上一级信息 |
| [31:0] ALUout | input | 上一级信息 |
| DMEn | input | 上一级信息 |
| [1:0] A3sel | input | 上一级信息 |
| [1:0] WDsel | input | 上一级信息 |
| GRFEn | input | 上一级信息 |
|  |  |  |
| [31:0] M\_IF | output | 下一级信息 |
| [31:0] M\_PCadd4 | output | 下一级信息 |
| [31:0] M\_BUSB | output | 下一级信息 |
| [31:0] M\_ALUout | output | 下一级信息 |
| M\_DMEn | output | 下一级信息 |
| [1:0] M\_A3sel | output | 下一级信息 |
| [1:0] M\_WDsel | output | 下一级信息 |
| M\_GRFEn | output | 下一级信息 |

1. W\_register

|  |  |  |
| --- | --- | --- |
| 功能描述 | M级流水线寄存器。使能信号有效是读入上一级信息。  输出下一级信息 | |
| 信号名称 | 方向 | 描述 |
| clk | input | 时钟信号 |
| reset | input | 复位信号 |
| [31:0] IF | input | 上一级信息 |
| [31:0] PCadd4 | input | 上一级信息 |
| [31:0] ALUout | input | 上一级信息 |
| [31:0] DMout | input | 上一级信息 |
| [1:0] A3sel | input | 上一级信息 |
| [1:0] WDsel | input | 上一级信息 |
| GRFEn | input | 上一级信息 |
|  |  |  |
| [31:0] W\_IF | output | 下一级信息 |
| [31:0] W\_PCadd4 | output | 下一级信息 |
| [31:0] W\_ALUout | output | 下一级信息 |
| [31:0] W\_DMout | output | 下一级信息 |
| W\_DMEn | output | 下一级信息 |
| [1:0] W\_A3sel | output | 下一级信息 |
| [1:0] W\_WDsel | output | 下一级信息 |
| W\_GRFEn | output | 下一级信息 |

1. MUX\_B

|  |  |  |
| --- | --- | --- |
| 功能描述 | 多路选择器 | |
| 信号名称 | 方向 | 描述 |
| [31:0] E\_EXTout | input |  |
| [31:0] MFRTE | input |  |
| E\_Bsel | input | 0:MFRTE  1:E\_EXTout |
| [31:0] B | output | ALU的第二个操作数 |

1. MUX\_A3

|  |  |  |
| --- | --- | --- |
| 功能描述 | 多路选择器 | |
| 信号名称 | 方向 | 描述 |
| [4:0] W\_IF[`rd] | input |  |
| [4:0] W\_IF[`rt] | input |  |
| 5’b11111 | input |  |
| W\_A3sel | input | 00:[4:0] W\_IF[`rd]  01:[4:0] W\_IF[`rt]  10:5’b11111 |

1. MUX\_WD

|  |  |  |
| --- | --- | --- |
| 功能描述 | 多路选择器 | |
| 信号名称 | 方向 | 描述 |
| [31:0] W\_ALUout | input |  |
| [31;0] W\_DMout | input |  |
| [31:0] PCadd4 | input |  |
| W\_WDsel | input | 00:[31:0] W\_ALUout  01:[31;0] W\_DMout  10:[31:0] PCadd4+4 |

1. MFRSD

|  |  |  |
| --- | --- | --- |
| 功能描述 | D级中rs转发器 | |
| 信号名称 | 方向 | 描述 |
| [31:0] E\_PCadd4 | input |  |
| [31:0] M\_PCadd4 | input |  |
| [31:0] M\_ALUout | input |  |
| [31:0] MUX\_WD | input |  |
| [31:0 ] BUSA | input |  |
| [2:0] RSDsel | input | 000:BUSA  001:MUX\_WD  010:M\_ALUout  011:M\_PCadd4  100:E\_PCadd4 |
| [31:0] MFRSD | output |  |

1. MFRTD

|  |  |  |
| --- | --- | --- |
| 功能描述 | D级中rt转发器 | |
| 信号名称 | 方向 | 描述 |
| [31:0] E\_PCadd4 | input |  |
| [31:0] M\_PCadd4 | input |  |
| [31:0] M\_ALUout | input |  |
| [31:0] MUX\_WD | input |  |
| [31:0 ] BUSB | input |  |
| [2:0] RSDsel | input | 000:BUSB  001:MUX\_WD  010:M\_ALUout  011:M\_PCadd4  100:E\_PCadd4 |
| [31:0] MFRTD | output |  |

1. MFRSE

|  |  |  |
| --- | --- | --- |
| 功能描述 | E级中rs转发器 | |
| 信号名称 | 方向 | 描述 |
| [31:0] M\_PCadd4 | input |  |
| [31:0] M\_ALUout | input |  |
| [31:0] MUX\_WD | input |  |
| [31:0 ] E\_BUSA | input |  |
| [2:0] RSEsel | input | 000:E\_BUSA  001:MUX\_WD  010:M\_ALUout  011:M\_PCadd4 |
| [31:0] MFRSE | output |  |

1. MFRTE

|  |  |  |
| --- | --- | --- |
| 功能描述 | E级中rt转发器 | |
| 信号名称 | 方向 | 描述 |
| [31:0] M\_PCadd4 | input |  |
| [31:0] M\_ALUout | input |  |
| [31:0] MUX\_WD | input |  |
| [31:0 ] E\_BUSB | input |  |
| [2:0] RTEsel | input | 000:E\_BUSB  001:MUX\_WD  010:M\_ALUout  011:M\_PCadd4 |
| [31:0] MFRTE | output |  |

1. MFRTM

|  |  |  |
| --- | --- | --- |
| 功能描述 | M级中rs转发器 | |
| 信号名称 | 方向 | 描述 |
| [31:0] MUX\_WD | input |  |
| [31:0 ] M\_BUSB | input |  |
| [2:0] RTMsel | input | 000:M\_BUSB  001:MUX\_WD |
| [31:0] MFRTM | output |  |

CP0设计

|  |  |  |
| --- | --- | --- |
| 功能描述 | 进行异常信号的处理 | |
| 信号名称 | 方向 | 描述 |
| Clk | In | 时钟信号 |
| Reset | In | 复位信号 |
| En | In | 使能信号 |
| [4:0] Address | In | 写入地址 |
| [31:0] DI | In | 写入数据 |
| [31:0] PC | In | 写入EPC的PC值 if\_delay==1:  EPC🡨 PC-4;  Else  EPC🡨 PC; |
| IFU\_Exc | In | 取指异常信号 |
| Undefined\_code | In | 未命名指令信号 |
| Read\_Exc | In | 取数异常 |
| Save\_Exc | In | 存数异常 |
| ALU\_Exc | In | 溢出异常 |
| If\_delay | In | 延时槽标志 |
| [7:2] HWInt | Output | 六位中断信号 |
| [31:0] CP0\_Data\_out | Ouput | 根据Address输出CP0相应寄存器的值 |
| [31:0] CP0\_EPC\_out | Output | EPC的值 |
| IntReq | Output | 是否进入异常处理模式 |

桥与IO设计

|  |  |  |
| --- | --- | --- |
| 功能描述 | 根据地址以及使能信号生成计时器的使能信号，并根据地址向CPU输出正确计时器的值 | |
| 信号名称 | 方向 | 描述 |
| [31:0] Addr | In | 计时器地址 |
| En | In | 计时器使能 |
| TC1\_En | Output | 一号计时器使能 |
| TC2\_En | Output | 二号计时器使能 |
| [31:0] TC1\_out | In | 一号计时器数据输出 |
| [31:0] TC2\_out | In | 二号计时器数据输出 |
| [31:0] PrRD | In | CPU读入的计时器数据 |

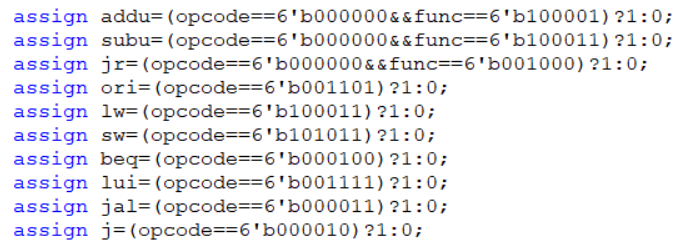
1. 数据通路

|  |  |  |  |
| --- | --- | --- | --- |
| 模块 | 部件 | 输入 | 输入来源 |
|
| F级功能部件 | IFU | clk |  |
| reset |  |
| imm | D.IF[25:0] |
| ifequal | ifequal |
| BUSA | MFRSD |
| PCsel | controller\_PCsel |
| PCEn | RFcontroller\_PCEn |
| D级流水线寄存器 | D\_register | clk |  |
| reset |  |
| IF | IF |
| PCadd4 | PCadd4 |
| En | RFcontroller\_D\_En |
| clear | RFcontroller\_D\_clear |
| D级功能部件 | GRF | clk |  |
| reset |  |
| A1 | D\_IF[25:21] |
| A2 | D\_IF[20:16] |
| EXT | imm | D.IF[15:0] |
| EXTsel | controller\_EXTsel |
| Compare | num1 | MFRSD |
| num2 | MFRTD |
| E级流水线寄存器 | E\_register | clk |  |
| reset |  |
| clear | RFcontroller\_E\_clear |
| E\_IF | D.IF |
| E\_PCadd4 | D.PCadd4 |
| E\_BUSA | BUSA |
| E\_BUSB | BUSB |
| E\_EXTout | EXTout |
| E\_ALUsel | controller\_ALUsel |
| E\_Bsel | controller\_Bsel |
| E\_DMEn | controller\_DMEn |
| E\_A3sel | controller\_A3sel |
| E\_WDsel | controller\_WDsel |
| E\_GRFEn | controller\_GRFEn |
| E级功能部件 | ALU | A | MFRSE |
| B | MUX\_B |
| ALUsel | E\_ALUsel |
| M级流水线寄存器 | M\_registre | clk |  |
| reset |  |
| M\_IF | E\_IF |
| M\_PCadd4 | E\_PCadd4 |
| M\_BUSB | E\_BUSB |
| M\_ALUout | ALUout |
| M\_DMEn | E\_DMEn |
| M\_A3sel | E\_A3sel |
| M\_WDsel | E\_WDsel |
| M\_GRFEn | E\_GRFEn |
| M级功能部件 | DM | clk |  |
| reset |  |
| Address | M\_ALUout |
| DI | MFRTM |
| DMEn | M\_DMEn |
| PCadd4 | M\_PCadd4 |
| W级流水线寄存器 | W\_register | clk |  |
| reset |  |
| W\_IF | M\_IF |
| W\_PCadd4 | M\_PCadd4 |
| W\_ALUout | M\_ALUout |
| W\_DMout | DMout |
| W\_A3sel | M\_A3sel |
| W\_WDsel | M\_WDsel |
| W\_GRFEn | M\_GRFEn |
| W级功能部件 | GRF | A3 | MUX\_A3 |
| WD | MUX\_WD |
| En | W\_GRFEn |
| PCadd4 | W\_PCadd4 |
| 转发选择器 | MFRSD | BUSA | BUSA |
| E\_PCadd4 | E\_PCadd4 |
| M\_PCadd4 | M\_PCadd4 |
| M\_ALUout | M\_ALUout |
| MUX\_WD | MUX\_WD |
| MFRTD | BUSB | BUSB |
| E\_PCadd4 | E\_PCadd4 |
| M\_PCadd4 | M\_PCadd4 |
| M\_ALUout | M\_ALUout |
| MUX\_WD | MUX\_WD |
| MFRSE | E\_BUSA | E\_BUSA |
| M\_PCadd4 | M\_PCadd4 |
| M\_ALUout | M\_ALUout |
| MUX\_WD | MUX\_WD |
| MFRTE | E\_BUSB | E\_BUSA |
| M\_PCadd4 | M\_PCadd4 |
| M\_ALUout | M\_ALUout |
| MUX\_WD | MUX\_WD |
| MFRTM | M\_BUSB | M\_BUSB |
| MUX\_WD | MUX\_WD |
| 多路选择器 | MUX\_B | MFRTE | MFRTE |
| E\_EXTout | E\_EXTout |
| MUX\_A3 | W\_IF[`rd] | W\_IF[`rd] |
| W\_IF[`rt] | W\_IF[`rt] |
| 1f | 5'b11111 |
| MUX\_WD | W\_ALUout | W\_ALUout |
| W\_DMout | W\_DMout |
| W\_PCadd4+4 | W\_PCadd4+32'h00000004 |

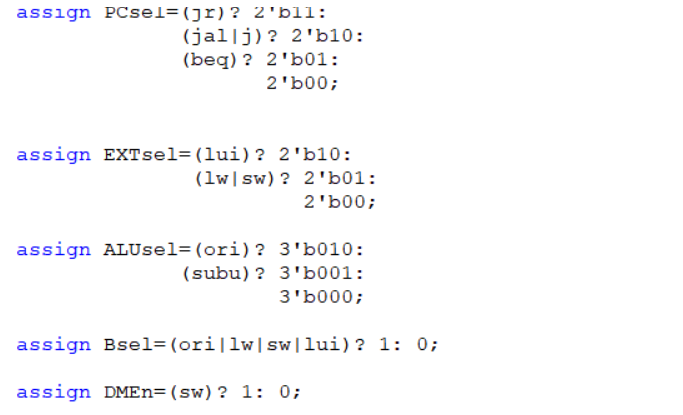
1. 控制器设计

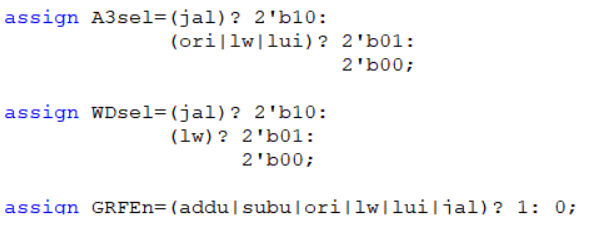
1、主控制器

与阵列：



或阵列：

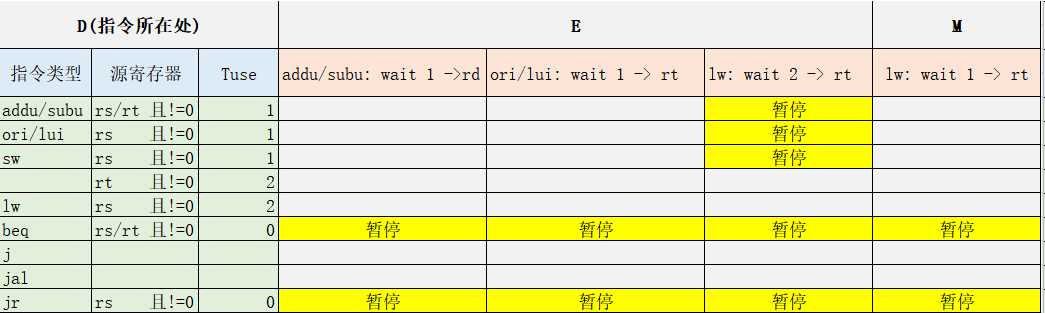




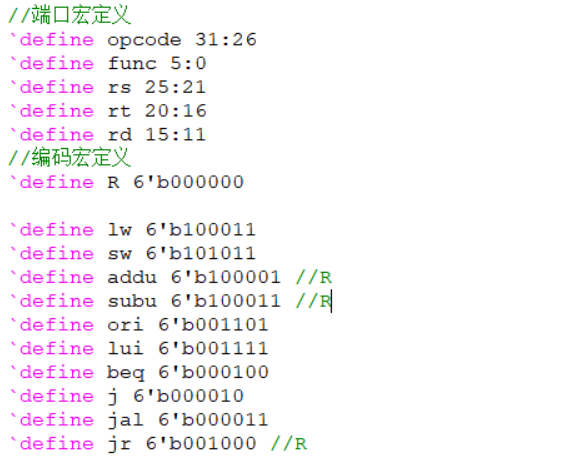
2、冒险控制器

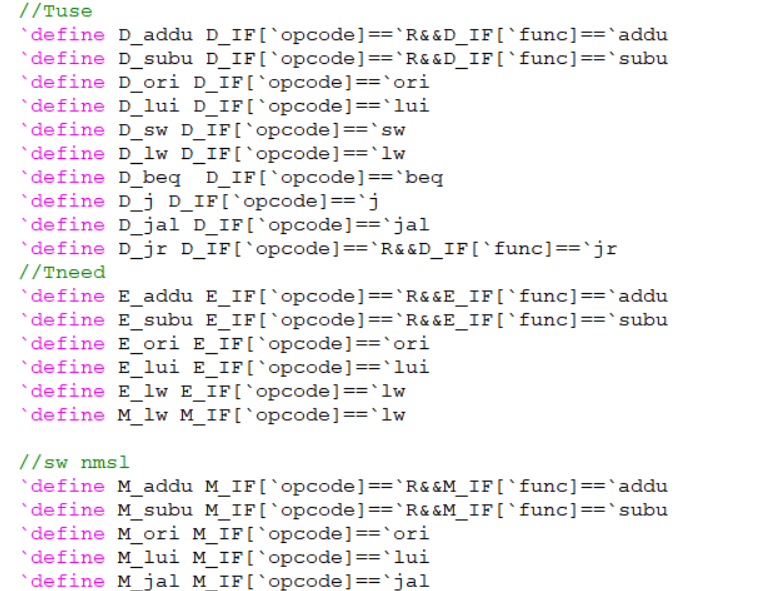
(1)暂停控制

构建暂停表格：

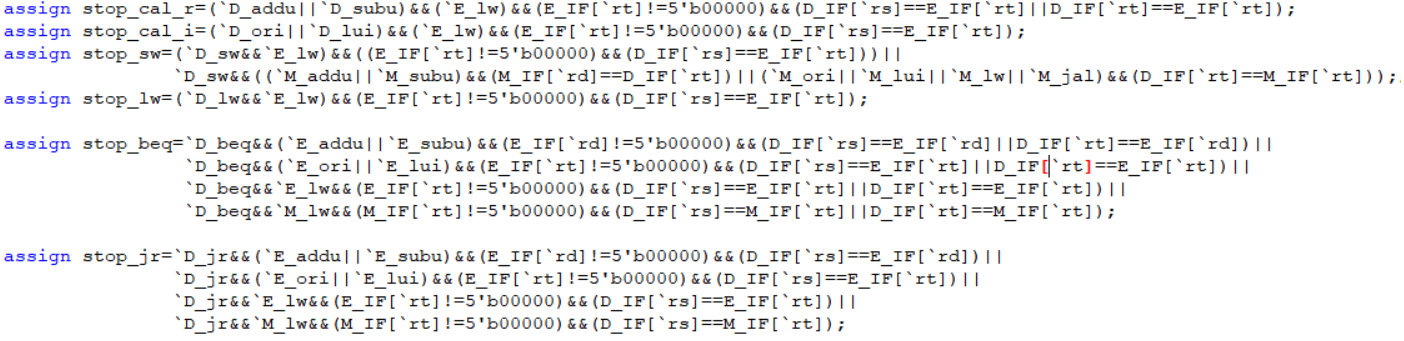


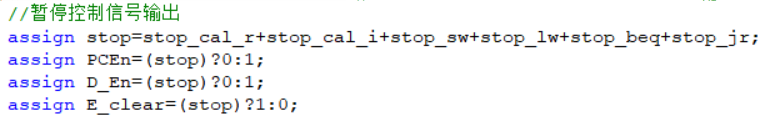
宏定义：





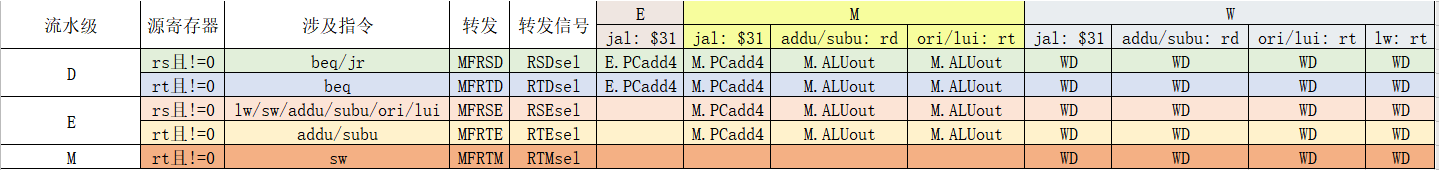




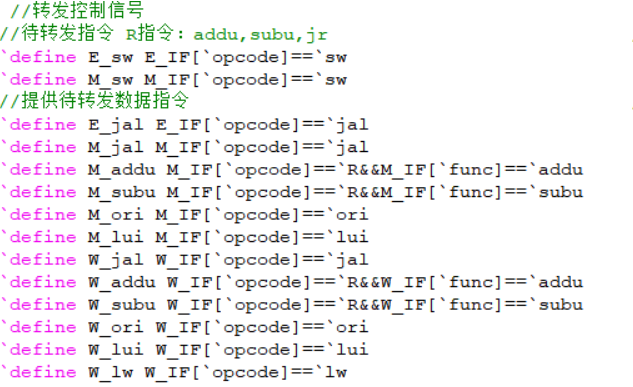


1. 转发控制

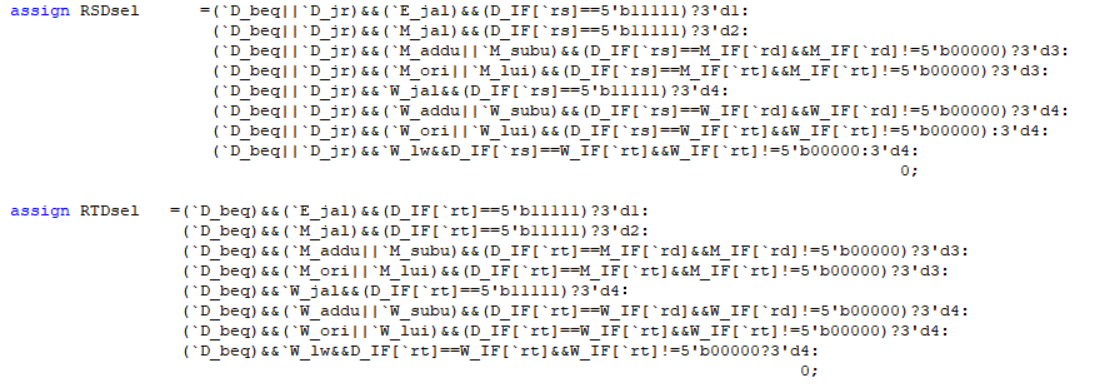
构建转发表格：

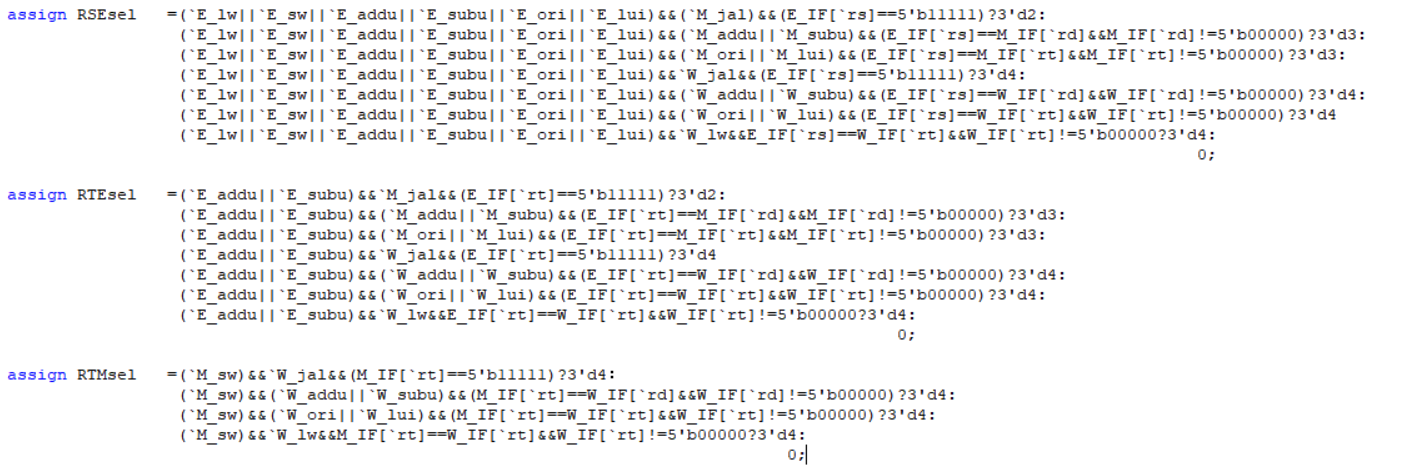


宏定义：









三、测试程序

1、

Cal\_r:

li $4,0x98765432

li $5,0x23456789

li $6,0xfedcba98

li $10,1

li $9,0

add $3,$4,$5

addu $2,$4,$6

and $11,$4,$5

or $12,$5,$4

nor $13,$4,$5

xor $14,$4,$5

move $31,$0

mult $4,$6

mfhi $15

mflo $16

multu $4,$6

mfhi $17

mflo $18

div $4,$6

mfhi $19

mflo $20

divu $4,$6

mfhi $21

mflo $22

nop

nop

nop

nop

nop

#start

#add/sub,beq/bne

add $7,$5,$4

beq $7,$3,label1

nop

ori $1,$0,1

label1:

sub $7,$7,$5

beq $7,$4,label2

nop

ori $1,$1,2

label2:

#addu/subu,beq/bne

addu $7,$6,$4

beq $7,$2,label3

nop

ori $1,$0,4

label3:

subu $7,$7,$6

bne $7,$4,label4

nop

ori $1,$1,8

label4:

#slt/sltu,beq/bne

slt $7,$6,$5#1

bne $7,$10,label5

nop

ori $1,$0,4

label5:

slt $7,$5,$4#0

beq $7,$10,label6

nop

ori $1,$1,8

label6:

sltu $7,$5,$4#1

bne $7,$10,label7

nop

ori $1,$0,4

label7:

sltu $7,$6,$5#0

bne $7,$10,label8

nop

ori $1,$1,8

label8:

#slt/sltu,blez/bltz/bgez/bgtz

slt $7,$6,$5#1

blez $7,label9

nop

ori $1,$0,4

label9:

slt $7,$5,$4#0

blez $7,label10

nop

ori $1,$1,8

label10:

sltu $7,$5,$4#1

bgtz $7,label11

nop

ori $1,$0,4

label11:

sltu $7,$6,$5#0

bgtz $7,label12

nop

ori $1,$1,8

label12:

#sll,srl,sra,sllv,srlv,srav

sll $7,$4,3#<0

bgez $7,label13

nop

ori $1,$0,1

label13:

srl $7,$4,1

bgez $7,label14#>0

nop

ori $1,$1,2

label14:

sra $7,$6,14#<0

blez $7,label15

nop

ori $1,$0,1

label15:

sllv $7,$4,$4#>0

blez $7,label16

nop

ori $1,$0,1

label16:

srav $7,$6,$6#>0

bgtz $7,label17

nop

ori $1,$1,2

label17:

srlv $7,$6,$6

bgtz $7,label18#<0

nop

ori $1,$0,1

label18:

srlv $7,$5,$4

bltz $7,label19#>0

nop

ori $1,$1,2

label19:

srav $7,$4,$5

bltz $7,label20

nop

ori $1,$0,1

label20:

#and,or,nor,xor

and $7,$4,$5

beq $7,$11,label21

nop

ori $1,$0,1

label21:

or $7,$4,$5

beq $7,$12,label22

nop

ori $1,$0,1

label22:

nor $7,$4,$5

beq $7,$13,label23

nop

ori $1,$0,1

label23:

xor $7,$4,$5

beq $7,$14,label24

nop

ori $1,$0,1

label24:

#mflo,mfhi

mult $4,$6

mfhi $7

beq $7,$15,mul1

nop

xori $1,$1,1

mul1:

mflo $7

beq $7,$16,mul2

nop

xori $1,$1,1

mul2:

multu $4,$6

mfhi $7

beq $7,$17,mul3

nop

xori $1,$1,1

mul3:

mflo $7

beq $7,$18,mul4

nop

xori $1,$1,1

mul4:

div $4,$6

mfhi $7

beq $7,$19,div1

nop

xori $1,$1,1

div1:

mflo $7

beq $7,$20,div2

nop

xori $1,$1,1

div2:

divu $4,$6

mfhi $7

beq $7,$21,div3

nop

xori $1,$1,1

div3:

mflo $7

beq $7,$18,div4

nop

xori $1,$1,1

div4:

beq $0,$0,div4

nop

cal\_i:

#initialize

li $4,0x98765432

li $5,0x23456789

li $6,0xfedcba98

addi $8,$4,0x7654

addi $9,$4,0xfedc

addiu $10,$4,0x7654

addiu $11,$4,0xfedc

andi $12,$5,0x7654

ori $13,$5,0x7654

xori $14,$5,0x7654

lui $15,0x7654

slti $16,$4,-32765

sltiu $17,$4,-32765

nop

nop

nop

nop

nop

#start

#addi/addiu

addi $7,$4,0x7654

beq $7,$8,label1

nop

ori $1,$0,1

label1:

addi $7,$4,0xfedc

beq $7,$9,label2

nop

ori $1,$1,2

label2:

addiu $7,$4,0x7654

beq $7,$10,label3

nop

ori $1,$0,4

label3:

addiu $7,$4,0xfedc

bne $7,$11,label4

nop

ori $1,$1,8

label4:

#andi/ori/xori

andi $7,$5,0x7654

bne $7,$12,label5

nop

ori $1,$0,4

label5:

ori $7,$5,0x7654

beq $7,$13,label6

nop

ori $1,$1,8

label6:

xori $7,$5,0x7654

bne $7,$14,label7

nop

ori $1,$0,4

label7:

#lui,slti,sltiu

slti $7,$4,-32765

beq $7,$16,label8

nop

ori $1,$0,4

label8:

sltiu $7,$4,-32765

beq $7,$17,label9

nop

label9:

beq $0,$0,label9

nop

load:

li $3,0xfedcba98

li $4,0

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

sw $3,0($4)

addiu $3,$3,0x01010101

addu $4,$4,4

#90 sw

#start

#lb,lbu,lh,lhu,lw

lbu $3,0($0)

lb $3,-15($3)

lbu $3,1($0)

lbu $3,-15($3)

lbu $3,1($0)

lh $3,-12($3)

lbu $3,1($0)

lhu $3,-4($3)

lbu $3,1($0)

lw $3,2($3)

#sb,sh,sw

lbu $3,0($0)

sb $3,-15($3)

lbu $3,1($0)

sb $3,-15($3)

lbu $3,1($0)

sh $3,-12($3)

lbu $3,1($0)

sw $3,2($3)

#add,addu,sub,subu,slt,sltu,sll,srl,sra,sllv,srlv,srav,and,or,xor,nor

lbu $3,0($0)

add $3,$3,$3

lbu $3,1($0)

addu $3,$3,$3

lbu $3,1($0)

sub $3,$3,$0

lbu $3,1($0)

sub $3,$0,$3

lbu $3,1($0)

subu $3,$3,$0

lbu $3,1($0)

subu $3,$0,$3

#slt: set $3, $4=$3+x,$3=$3+2

lw $3,4($0)

addiu $4,$3,0x10101010

slt $5,$3,$4

addiu $3,$3,2

slt $5,$3,$4

lw $3,4($0)

slt $3,$3,$4

lw $3,4($0)

addiu $4,$3,2

slt $5,$3,$4

addiu $3,$3,0x3f3f3f3f

slt $5,$3,$4

lw $3,4($0)

slt $3,$3,$4

#sltu

lw $3,4($0)

addiu $4,$3,2

sltu $5,$3,$4

addiu $3,$3,2

slt $5,$3,$4

lw $3,4($0)

sltu $3,$3,$4

lw $3,4($0)

addiu $4,$3,2

slt $5,$3,$4

addiu $3,$3,0x3f3f3f3f

sltu $5,$3,$4

lw $3,4($0)

sltu $3,$3,$4

#sll,srl

lw $3,4($0)

sll $3,$3,1

lw $3,4($0)

sll $3,$3,2

lw $3,4($0)

sll $3,$3,3

lw $3,4($0)

sll $3,$3,4

lw $3,4($0)

sll $3,$3,5

lw $3,4($0)

sll $3,$3,6

lw $3,4($0)

sll $3,$3,7

lw $3,4($0)

sll $3,$3,8

lw $3,4($0)

sll $3,$3,9

lw $3,4($0)

sll $3,$3,10

lw $3,4($0)

sll $3,$3,11

lw $3,4($0)

sll $3,$3,12

lw $3,4($0)

sll $3,$3,13

lw $3,4($0)

sll $3,$3,14

lw $3,4($0)

sll $3,$3,15

lw $3,4($0)

sll $3,$3,16

lw $3,4($0)

sll $3,$3,17

lw $3,4($0)

sll $3,$3,18

lw $3,4($0)

sll $3,$3,19

lw $3,4($0)

sll $3,$3,20

lw $3,4($0)

sll $3,$3,21

lw $3,4($0)

sll $3,$3,22

lw $3,4($0)

sll $3,$3,23

lw $3,4($0)

sll $3,$3,24

lw $3,4($0)

sll $3,$3,25

lw $3,4($0)

sll $3,$3,26

lw $3,4($0)

sll $3,$3,27

lw $3,4($0)

sll $3,$3,28

lw $3,4($0)

sll $3,$3,29

lw $3,4($0)

sll $3,$3,30

lw $3,4($0)

sll $3,$3,31

lw $3,4($0)

srl $3,$3,1

lw $3,4($0)

srl $3,$3,2

lw $3,4($0)

srl $3,$3,3

lw $3,4($0)

srl $3,$3,4

lw $3,4($0)

srl $3,$3,5

lw $3,4($0)

srl $3,$3,6

lw $3,4($0)

srl $3,$3,7

lw $3,4($0)

srl $3,$3,8

lw $3,4($0)

srl $3,$3,9

lw $3,4($0)

srl $3,$3,10

lw $3,4($0)

srl $3,$3,11

lw $3,4($0)

srl $3,$3,12

lw $3,4($0)

srl $3,$3,13

lw $3,4($0)

srl $3,$3,14

lw $3,4($0)

srl $3,$3,15

lw $3,4($0)

srl $3,$3,16

lw $3,4($0)

srl $3,$3,17

lw $3,4($0)

srl $3,$3,18

lw $3,4($0)

srl $3,$3,19

lw $3,4($0)

srl $3,$3,20

lw $3,4($0)

srl $3,$3,21

lw $3,4($0)

srl $3,$3,22

lw $3,4($0)

srl $3,$3,23

lw $3,4($0)

srl $3,$3,24

lw $3,4($0)

srl $3,$3,25

lw $3,4($0)

srl $3,$3,26

lw $3,4($0)

srl $3,$3,27

lw $3,4($0)

srl $3,$3,28

lw $3,4($0)

srl $3,$3,29

lw $3,4($0)

srl $3,$3,30

lw $3,4($0)

srl $3,$3,31

lw $3,4($0)

#sra,sllv,srlv,srav

lw $3,4($0)

sra $3,$3,1

lw $3,4($0)

sra $3,$3,2

lw $3,4($0)

sra $3,$3,3

lw $3,4($0)

sra $3,$3,4

lw $3,4($0)

sra $3,$3,5

lw $3,4($0)

sra $3,$3,6

lw $3,4($0)

sra $3,$3,7

lw $3,4($0)

sra $3,$3,8

lw $3,4($0)

sra $3,$3,9

lw $3,4($0)

sra $3,$3,10

lw $3,4($0)

sra $3,$3,11

lw $3,4($0)

sra $3,$3,12

lw $3,4($0)

sra $3,$3,13

lw $3,4($0)

sra $3,$3,14

lw $3,4($0)

sra $3,$3,15

lw $3,4($0)

sra $3,$3,16

lw $3,4($0)

sra $3,$3,17

lw $3,4($0)

sra $3,$3,18

lw $3,4($0)

sra $3,$3,19

lw $3,4($0)

sra $3,$3,20

lw $3,4($0)

sra $3,$3,21

lw $3,4($0)

sra $3,$3,22

lw $3,4($0)

sra $3,$3,23

lw $3,4($0)

sra $3,$3,24

lw $3,4($0)

sra $3,$3,25

lw $3,4($0)

sra $3,$3,26

lw $3,4($0)

sra $3,$3,27

lw $3,4($0)

sra $3,$3,28

lw $3,4($0)

sra $3,$3,29

lw $3,4($0)

sra $3,$3,30

lw $3,4($0)

sra $3,$3,31

#sllv,srlv,srav

lw $3,4($0)

srav $3,$3,$3

lw $3,4($0)

srlv $3,$3,$3

lw $3,4($0)

sllv $3,$3,$3

#and,or,xor,nor

li $4,0x98765432

lw $3,4($0)

or $3,$3,$4

lw $3,4($0)

or $3,$4,$3

lw $3,4($0)

xor $3,$3,$4

lw $3,4($0)

xor $3,$4,$3

lw $3,4($0)

nor $3,$3,$4

lw $3,4($0)

nor $3,$4,$3

#addi,addiu,andi,ori,xori,lui,slti,sltiu

lw $3,4($0)

addi $3,$3,0xff3f

lw $3,4($0)

addi $3,$3,0x3f3f

lw $3,4($0)

addiu $3,$3,0xff3f

lw $3,4($0)

addiu $3,$3,0x3f3f

lw $3,4($0)

andi $3,$3,0xff3f

lw $3,4($0)

andi $3,$3,0x3f3f

lw $3,4($0)

ori $3,$3,0xff3f

lw $3,4($0)

ori $3,$3,0x3f3f

lw $3,4($0)

xori $3,$3,0xff3f

lw $3,4($0)

xori $3,$3,0x3f3f

lw $3,4($0)

lui $3,0xff3f

lw $3,4($0)

lui $3,0x3f3f

#slti,sltiu

lw $3,4($0)

slti $3,$3,-32000

lw $3,4($0)

sltiu $3,$3,-32000

lw $3,4($0)

slti $3,$3,32000

lw $3,4($0)

sltiu $3,$3,32000

#b

lw $4,4($0)

lw $3,4($0)

beq $3,$4,next

next:

xori $1,$1,1

#jr,jalr

li $4,0x3ed0 #jump to line 722

sw $4,0($0)

lw $3,0($0)

jr $3

xori $1,$1,1

li $4,0x3ee4

sw $4,0($0)

lw $3,0($0)

jalr $8,$3

xori $1,$1,1

#mfhi,mflo,mthi,mtlo

lw $3,0($0)

mthi $3

mfhi $4

lw $3,4($0)

mtlo $3

mflo $4

#mult,multu,div,divu

lw $3,4($0)

mult $3,$3

mfhi $3

mflo $3

lw $3,4($0)

multu $3,$3

mfhi $3

mflo $3

lw $3,4($0)

div $3,$3

mfhi $3

mflo $3

lw $3,4($0)

divu $3,$3

mfhi $3

mflo $3

#b - MEM->ID

lw $4,4($0)

lw $3,4($0)

nop

beq $3,$4,next2

next2:

xori $1,$1,1

#jr,jalr - MEM->ID

li $4,0x3f68 #jump to line 722

sw $4,0($0)

lw $3,0($0)

nop

jr $3

xori $1,$1,1

li $4,0x3f80

sw $4,0($0)

lw $3,0($0)

nop

jalr $8,$3

xori $1,$1,1

end:

beq $0,$0,end

nop

save:

#initialize

li $2,0x3000

li $3,0x0004

#start

#addi/addiu

add $4,$2,$3

addiu $4,$4,16#line 10

jr $4

xori $1,$1,1

addiu $4,$4,12#line 13

jr $4

xori $1,$1,1

#andi/ori/xori

ori $4,$4,12#line 18 3024->302c

jr $4

xori $1,$1,1

li $4,0xffffffff

andi $4,$4,0x303c#line 21 3034->303c

jr $4

xori $1,$1,1

xori $4,$4,0x70#line 24 3044->304c

jalr $10,$4

xori $1,$1,1

#lui,slti,sltiu

#这个测不了

#add/sub

li $12,16

add $4,$4,$12

jalr $10,$4

xori $1,$1,1

li $12,-16

sub $4,$4,$12

jalr $10,$4

xori $1,$1,1

#addu/subu

li $12,16

addu $4,$4,$12

jalr $10,$4

xori $1,$1,1

li $12,-16

subu $4,$4,$12

jalr $10,$4

xori $1,$1,1

#slt/sltu

#没法测

#sll,srl,sra,sllv,srlv,srav

#srl,sra

li $12,0x309c0

srl $4,$12,4

jalr $10,$4

xori $1,$1,1

li $12,0x30b00

sra $4,$12,4

jalr $10,$4

xori $1,$1,1

#sll,sllv

li $12,0x30c

sll $4,$12,4

jalr $10,$4

xori $1,$1,1

li $12,0xc38

li $13,0x77777702

nop

nop

sllv $4,$12,$13

jalr $10,$4

xori $1,$1,1

#srlv,srav

li $12,0xc3c0

srlv $4,$12,$13

jalr $10,$4

xori $1,$1,1

li $12,0xc400

srav $4,$12,$13

jalr $10,$4

xori $1,$1,1

#now:3100(line 83)

#and,or,nor,xor

li $12,0x3110

li $13,4

or $4,$12,$13

jalr $10,$4

xori $1,$1,1

li $14,0x34

xor $4,$12,$14

jalr $10,$4

xori $1,$1,1

#mflo,mfhi

li $15,0x313c

li $16,0x1

mult $15,$16

mflo $4

jalr $10,$4

xori $1,$1,1

li $15,0x315c000

li $16,0x100000

mult $15,$16

mfhi $4

jalr $10,$4

xori $1,$1,1

end:

beq $0,$0,end

nop

2、异常行为

.ktext 0x4180

mfc0 $a1,$14

addi $a1,$a1,4

mtc0 $a1,$14

eret

.text

ori $s1,0x300d

jr $s1

nop

sw $s1,0($zero)

sw $s1,4($zero)

and $s1,0

ori $s1,0x2fff

jr $s1

nop

sw $s1,0($zero)

sw $s1,4($zero)

and $s1,0

ori $s1,0x5000

jr $s1

nop

sw $s1,0($zero)

sw $s1,4($zero)

and $s1,0

ori $s1,0x4321

sw $s1,0($zero)

lw $s1,2($zero)

sw $s1,4($zero)

and $s1,0

lh $s1,1($zero)

sw $s1,4($zero)

sw $s1,8($zero)

and $s1,0

lhu $s1,1($zero)

sw $s1,4($zero)

sw $s1,8($zero)

and $t0,0

and $s1,0

ori $t0,0x7018

lh $s1,0($t0)

sw $t0,4($zero)

nop

and $t0,0

and $s1,0

ori $s1,0x4321

sw $s1,1000($zero)

ori $t0,0xf000

lw $s1,2000($t0)

sw $s1,4($zero)

nop

and $t0,0

and $s1,0

ori $t0,0x3000

lw $s1,0($t0)

sw $s1,4($zero)

nop

#存数异常

and $t0,0

and $s1,0

sw $0,$0($zero)

sw $0,1($zero)

sw $0,4($zero)

nop

sw $0,$0($zero)

sh $0,1($zero)

sw $0,4($zero)

nop

ori $t0,0x7010

sh $0,0($t0)

sw $0,4($zero)

nop

sb $0,0($t0)

sw $0,4($zero)

nop

and $t0,0

ori $t0,0xf000

sw $0,2000($t0)

and $t0,0

ori $t0,0x7f18

sw $t0,0($t0)

sw $t0,4($zero)

nop

and $t0,0

ori $t0,0x3000

sw $0,0($t0)

sw $t0,4($zero)

nop

#未知指令

break

sw $s1,4($zero)

nop

#溢出异常

and $s1,0

and $s2,0

ori $s1,0xf000

ori $s2,0xf000

add $s3,$s1,$s2

sw $s1,0($zero)

nop

1. 中断

.ktext 0x4180

\_entry:

mfc0 $1, $13

ori $k0, $0, 0x1000

sw $sp, -4($k0)

addiu $k0, $k0, -256

move $sp, $k0

j \_save\_context

nop

\_quick\_handle:

mfc0 $1, $13

mfc0 $k0, $13

andi $k0, $k0, 0x00ff

srl $k0, $k0, 2

ori $k1, $0, 0x0004

beq $k0, $k1, adel\_handler\_quick

nop

ori $k1, $0, 0x000a

beq $k0, $k1, ri\_handler\_quick

nop

j \_entry

nop

adel\_handler\_quick:

mfc0 $t8,$14

andi $t9,$t8,3

bne $t9,$0,adel\_type\_1

nop

addi $t9,$t8,-0x3000

lui $s7,0xffff

ori $s7,$s7,0xe000

and $t9,$t9,$s7

bne $t9,$0,adel\_type\_2

nop

j \_entry

nop

adel\_type\_1:

ori $10,$0,0x3230

mtc0 $10,$14

eret

adel\_type\_2:

ori $10,$0,0x3240

mtc0 $10,$14

eret

ri\_handler\_quick:

ori $10,$0,0x3220

mtc0 $10,$14

eret

\_main\_handler:

mfc0 $k0, $13

andi $k0, $k0, 0x00ff

srl $k0, $k0, 2

ori $k1, $0, 0x0000

beq $k0, $k1, int\_handler

nop

ori $k1, $0, 0x0004

beq $k0, $k1, adel\_handler

nop

ori $k1, $0, 0x0005

beq $k0, $k1, ades\_handler

nop

ori $k1, $0, 0x000a

beq $k0, $k1, ri\_handler

nop

ori $k1, $0, 0x000c

beq $k0, $k1, ov\_handler

nop

int\_handler:

sw $ra, 0($sp)

addiu $sp, $sp, -16

mfc0 $v0, $12

sw $v0, 0($sp)

mfc0 $v0, $13

sw $v0, 4($sp)

# check INT[3]

lw $v0, 0($sp)

lw $v1, 4($sp)

and $v0, $v1, $v0

andi $v0, $v0, 0x800

bne $v0, $0, timer1\_handler

nop

# check INT[2]

lw $v0, 0($sp)

lw $v1, 4($sp)

and $v0, $v1, $v0

andi $v0, $v0, 0x400

bne $v0, $0, timer0\_handler

nop

timer0\_handler:

# first we load the global variable cnt0:

# ++cnt0, then save to global variable cnt0

li $fp, 0x8

lw $t0, 0($fp) # get cnt0

addi $s6, $0 , 1

beq $t0, $s6, skip0

nop

addiu $t0, $t0, 1 # add cnt0

skip0: sw $t0, 0($fp) # update cnt0

jal restart\_timer

nop

# mask INT[2]

mfc0 $t0, $12

andi $t0, $t0, 0x03ff

ori $t0, $t0, 0x800

mtc0 $t0, $12

j \_restore\_context

nop

timer1\_handler:

# first we load the global variable cnt1:

# ++cnt1, then save to global variable cnt1

li $fp, 0xc

lw $t0, 0($fp) # get cnt1

addi $s6, $0 , 1

beq $t0, $s6, skip1

nop

addiu $t0, $t0, 1 # add cnt1

skip1: sw $t0, 0($fp) # update cnt1

jal restart\_timer

nop

# mask INT[3]

mfc0 $t0, $12

andi $t0, $t0, 0x03ff

ori $t0, $t0, 0x400

mtc0 $t0, $12

j \_restore\_context

nop

restart\_timer:

# swap two PRESET

li $t0, 0x0

lw $t0, 0($t0)

lw $t5, 4($t0)

li $t2, 0x4

lw $t2, 0($t2)

lw $t6, 4($t2)

# restart Timer 0

li $t1, 0x0

lw $t1, 0($t1)

lw $t0, 0($t1)

sw $0, 0($t1)

li $t2, 0x8

lw $t2, 0($t2)

addi $s6, $0 , 5

beq $t2, $s6, f0 nop

sw $t6, 4($t1)

addiu $t0, $0, 9

sw $t0, 0($t1) f0:

# restart Timer 1

li $t1, 0x4

lw $t1, 0($t1)

lw $t0, 0($t1) # $t0 is the CTRL Reg of Timer 1

sw $0, 0($t1) # disable Timer 1

li $t2, 0xc

lw $t2, 0($t2)

addi $s6, $0 , 5

beq $t2, $s6, f1

nop

sw $t5, 4($t1)

addiu $t0, $0, 9

sw $t0, 0($t1)

f1:

jr $ra

nop

adel\_handler:

mfc0 $t0, $14

mfc0 $k0, $13

lui $t2, 0x8000

and $t3, $k0, $t2

addi $t0, $t0, 4

bne $t3, $t2, adel\_nxt

nop

addi $t0, $t0, 4

adel\_nxt:

mtc0 $t0, $14

j \_restore\_context

nop

ades\_handler:

mfc0 $t0, $14

mfc0 $k0, $13

lui $t2, 0x8000

and $t3, $k0, $t2

addi $t0, $t0, 4

bne $t3, $t2, ades\_nxt

nop

addi $t0, $t0, 4

ades\_nxt:

mtc0 $t0, $14

j \_restore\_context

nop

ri\_handler:

mfc0 $t0, $14

mfc0 $k0, $13

lui $t2, 0x8000

and $t3, $k0, $t2

addi $t0, $t0, 4

bne $t3, $t2, ri\_nxt

nop

addi $t0, $t0, 4

ri\_nxt:

mtc0 $t0, $14

j \_restore\_context

nop

ov\_handler:

mfc0 $t0, $14

mfc0 $k0, $13

lui $t2, 0x8000

and $t3, $k0, $t2

addi $t0, $t0, 4

bne $t3, $t2, ov\_nxt

nop

addi $t0, $t0, 4

ov\_nxt:

mtc0 $t0, $14

j \_restore\_context

nop

\_restore:

eret

\_save\_context:

sw $2, 8($sp)

sw $3, 12($sp)

sw $4, 16($sp)

sw $5, 20($sp)

sw $6, 24($sp)

sw $7, 28($sp)

sw $8, 32($sp)

sw $9, 36($sp)

sw $10, 40($sp)

sw $11, 44($sp)

sw $12, 48($sp)

sw $13, 52($sp)

sw $14, 56($sp)

sw $15, 60($sp)

sw $16, 64($sp)

sw $17, 68($sp)

sw $18, 72($sp)

sw $19, 76($sp)

sw $20, 80($sp)

sw $21, 84($sp)

sw $22, 88($sp)

sw $23, 92($sp)

sw $24, 96($sp)

sw $25, 100($sp)

sw $28, 112($sp)

sw $29, 116($sp)

sw $30, 120($sp)

sw $31, 124($sp)

mfhi $k0

sw $k0, 128($sp)

mflo $k0

sw $k0, 132($sp)

j \_main\_handler

nop

\_restore\_context:

li $sp, 0x1000

addi $sp, $sp, -256

lw $2, 8($sp)

lw $3, 12($sp)

lw $4, 16($sp)

lw $5, 20($sp)

lw $6, 24($sp)

lw $7, 28($sp)

lw $8, 32($sp)

lw $9, 36($sp)

lw $10, 40($sp)

lw $11, 44($sp)

lw $12, 48($sp)

lw $13, 52($sp)

lw $14, 56($sp)

lw $15, 60($sp)

lw $16, 64($sp)

lw $17, 68($sp)

lw $18, 72($sp)

lw $19, 76($sp)

lw $20, 80($sp)

lw $21, 84($sp)

lw $22, 88($sp)

lw $23, 92($sp)

lw $24, 96($sp)

lw $25, 100($sp)

lw $28, 112($sp)

lw $30, 120($sp)

lw $31, 124($sp)

lw $k0, 128($sp)

mthi $k0

lw $k0, 132($sp)

mtlo $k0

lw $29, 116($sp)

ori $1,$0,1

j \_restore

nop

.data

.globl TC0\_BASE TC1\_BASE cnt0 cnt1 cnt0\_double cnt1\_double

TC0\_BASE: .word 0x7f00

TC1\_BASE: .word 0x7f10

cnt0: .word 1

cnt1: .word 1

cnt0\_double: .word 0

cnt1\_double: .word 0

.text

ori $28, $0, 0x0000

ori $29, $0, 0x0f00

mtc0 $0, $12

#save start address

ori $t0, $0, 0x7f00

sw $t0, 0($0)

ori $t1, $0, 0x7f10

sw $t1, 4($0)

#set SR included IM, IE, EXL

ori $t0,$0, 0x0c01

mtc0 $t0,$12

#set Timer0

la $t1, TC0\_BASE

lw $t1, 0($t1)

sw $0, 0($t1) # disable Timer0.CTRL

addiu $t0, $0, 0x80 # set Timer0.PRESET

sw $t0, 4($t1)

addiu $t0, $0, 9 # set Timer0.CTRL

sw $t0, 0($t1)

#set Timer1

la $t1, TC1\_BASE

lw $t1, 0($t1)

sw $0, 0($t1) # disable Timer1.CTRL

addiu $t0, $0, 0x40 # set Timer1.PRESET

sw $t0, 4($t1)

addiu $t0, $0, 9 # set Timer1.CTRL

sw $t0, 0($t1)

ori $t0, $0, 0x0001

wait:

lw $k0, 8($0)

lw $k1,12($0)

bne $k0, $t0, wait

nop

bne $k1, $t0, wait

nop

ori $t0, $0, 0xffff

ori $t1, $0, 0xffff

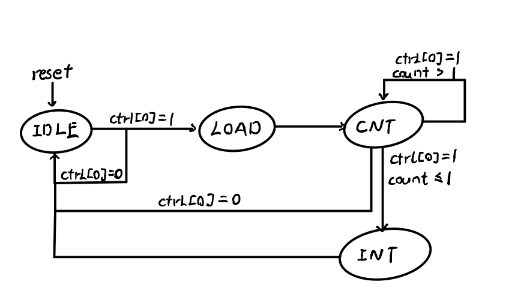
dead\_loop:

j dead\_loop

nop

思考题：

1、



2、计时器使用说明

状态说明：

(1)待定状态

若允许计数，则进入加载状态。

(2)加载状态

将preset寄存器里的初始值赋值给counter寄存器。

进入计数状态。

(3)计数状态

若允许计数，且计时器数值大于1，则每过一个时钟周期，计时器数值减一

若允许计数期间，计时器数值到0，则产生中断信号，并进入中断状态。

若不允许计数，则返回待定状态。

(4)中断状态

若计数模式为方式0，则修改计数器使能为0，并且下一周期返回待定状态。并且今后中断信号持续有效。

若计数模式为方式1，则仅在中断状态期间中断信号有效。下一周期直接返回待定状态。

操作说明：

1. 复位操作

更改TC状态为待定状态。更改TC内部寄存器均为0。

1. 写入操作

将Din的值写入指定的某个内部寄存器。

错误操作：

若向计时器写入值，则会异常。

3、在我们设计的流水线中，DM 处于 CPU 内部，请你考虑现代计算机中它的位置应该在何处。

答：应该位于CPU外部。

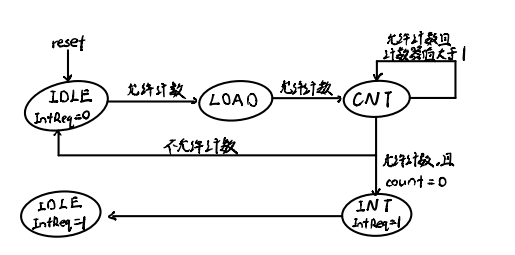
4、BE 部件对所有的外设都是必要的吗？

答：不是，如本次实验所用的timer其实可以不用BE实现。

5、请阅读官方提供的定时器源代码，阐述两种中断模式的异同，并分别针对每一种模式绘制状态转移图

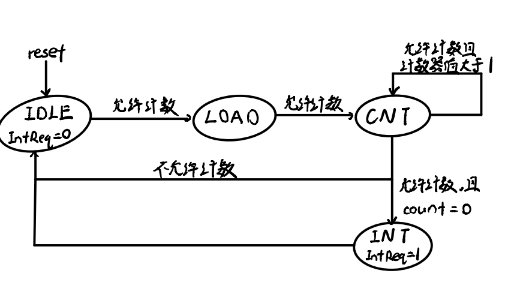
模式0：

用于产生持续有效的中断信号，状态图如下：



模式1：

用于产生周期性间断的中断信号，状态图如下：



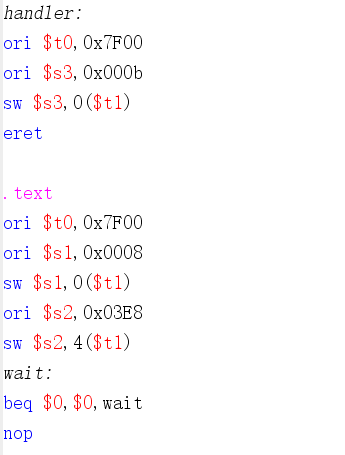
6、请开发一个主程序以及定时器的exception handler。整个系统完成如下功能：

(1)、定时器在主程序中被初始化为模式0；

(2)、定时器倒计数至0产生中断；

(3)、handler设置使能Enable为1从而再次启动定时器的计数器。2及3被无限重复。

(4)、主程序在初始化时将定时器初始化为模式0，设定初值寄存器的初值为某个值，如100或1000。（注意，主程序可能需要涉及对CP0.SR的编程，推荐阅读过后文后再进行。）



7、请查阅相关资料，说明鼠标和键盘的输入信号是如何被CPU知晓的？

答：设备实际上包括两部分接口控制器（也称为接口芯片）和设备主体，设备主体不直接与主机连接，而是通过接口控制器与主机连接。鼠标和键盘的输入信号相当于中断，当键盘、鼠标有信息时，产生一个中断然后中断例程会从端口读入数据到寄存器。CPU接收到中断请求之后进入中断处理程序获得鼠标键盘的信息。