基于VerilogHDL语言以及MIPS32-lite2体系结构的

流水线CPU设计文档

一、数据通路

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | addu | subu | ori | lui | lw | sw | beq | j | jal | jr | MUX | 0 | 1 | MF | 0 | 1 | 2 |
| Fetch | PC | ADD4 | ADD4 | ADD4 | ADD4 | ADD4 | ADD4 | NPC|ADD4 | NPC | NPC | NPC | MUX\_PC | ADD4 | NPC |  |  |  |  |
| Decode | NPC |  |  |  |  |  |  | offset | index | index | V1D |  |  |  |  |  |  |  |
| RF | V1D | 1 | 1 | 1 | 1 | 1 | 1 | 1 |  |  | 1 |  |  |  | MF\_V1D | V1D | DMoutW | AOM |
| V2D | 1 | 1 |  |  |  | 1 | 1 |  |  |  |  |  |  | MF\_V2D | V2D | DMoutW | AOM |
| CMP | RD1 |  |  |  |  |  |  | 1 |  |  |  |  |  |  | MF\_V1D | V1D | DMoutW | AOM |
| RD2 |  |  |  |  |  |  | V2D |  |  |  |  |  |  | MF\_V2D | V2D | DMoutW | AOM |
| CMPsel |  |  |  |  |  |  | EQ |  |  |  |  |  |  |  |  |  |  |
| EXT | EXTOp |  |  | ZERO | UPPER | SIGN | SIGN |  |  |  |  |  |  |  |  |  |  |  |
| NPC | NPCsel |  |  |  |  |  |  | BRANCH | JUMP | JUMP | REGI |  |  |  |  |  |  |  |
| Execute | V1E | 1 | 1 | 1 | 1 | 1 | 1 | FLUSH | FLUSH | FLUSH | FLUSH |  |  |  |  |  |  |  |
| V2E | 1 | 1 |  |  |  | 1 | FLUSH | FLUSH | FLUSH | FLUSH |  |  |  |  |  |  |  |
| ImmOutE |  |  | 1 | 1 | 1 | 1 | FLUSH | FLUSH | FLUSH | FLUSH |  |  |  |  |  |  |  |
| ShiftE |  |  |  |  |  |  | FLUSH | FLUSH | FLUSH | FLUSH |  |  |  |  |  |  |  |
| ALU | SrcAE | 1 | 1 | 1 | 1 | 1 | 1 |  |  |  |  |  |  |  | MF\_V1E | V1E | DMoutW | AOM |
| SrcBE | V2E | V2E | imm | imm | imm | imm |  |  |  |  | MUX\_SrcB | V2E | imm | MF\_V2E | V2E | DMoutW | AOM |
| Shift |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| ALUCtrl | ADD | SUB | OR | ADD | ADD | ADD |  |  |  |  |  |  |  |  |  |  |  |
| M | WRM | rd | rd | A2rt | A2rt | A2rt |  |  |  |  |  | MUX\_WR | rd | A2rt |  |  |  |  |
| AOM | value | value | value | value | addr | addr |  |  |  |  |  |  |  |  |  |  |  |
| WDMM |  |  |  |  |  | V2E |  |  |  |  |  |  |  | MF\_WDMM | V2E | DMoutW |  |
| DM | DMWE |  |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |
| Load |  |  |  |  | WORD | WORD |  |  |  |  |  |  |  |  |  |  |  |
| DMoutM |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |
| WriteBack | DMoutW |  |  |  |  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |
| AOW | value | value | value | value |  |  |  |  |  |  |  |  |  |  |  |  |  |
| PC | PC | PC | PC | PC | PC |  |  | PC+8 | PC+8 |  |  |  |  |  |  |  |  |
| WRW | rd | rd | A2rt | A2rt | A2rt |  |  |  | 31 |  |  |  |  |  |  |  |  |

二、各个模块以及控制器设计

|  |  |  |
| --- | --- | --- |
| reg[31:0] PC |  | Fetch 取指令 |
| 指示下一条指令的地址 |  | 初始化：PC置0x00003000 |
| NPC[31:0] | I | 下一次的PC值输入 |
| EN | I | 冲突单元控制PC使能端，暂停用 |
| reset | I | 同步复位，复位值为0x00003000 |
| clk | I | 时钟 |
| PC\_F[31:0] | O | F级PC输出 |
|  |  |  |
| reg[31:0] IM[0:1023] |  |  |
| 指令存储器 | 0x00003000-0x00003ffc |  |
| PC[31:0] | I | PC输入 |
| Instr\_F[31:0] | O | 指令输出 |
|  |  |  |
|  |  |  |
| ADD4 |  |  |
| PC[31:0] | I | PC输入 |
| PCADD4F[31:0] | O | F级PC+4输出 |

|  |  |  |
| --- | --- | --- |
| Decode | 译码 |  |
| PC\_F[31:0] | I | PC输入 |
| PCADD4\_F[31:0] | I | PC+4输入 |
| Instr\_F[31:0] | I | 指令输入 |
| EN | I | 冲突单元控制D级流水使能端，暂停用 |
| clk | I | 时钟 |
| reset | I | 同步复位置0 |
| PC\_D[31:0] | O | D级PC输出 |
| PCADD4\_D[31:0] | O | D级PC+4输出 |
| InstrD[31:0] | O | D级指令输出 |
|  |  |  |
| RF |  |  |
| clk | I | 时钟 |
| reset | I | 同步复位,全部置0 |
| A1rs[25:21] | I | rs |
| A2rt[20:16] | I | rt |
| A3[4:0] | I | 写回寄存器的地址 |
| RegWE | I | 寄存器写使能 |
| WPC[31:0] | I | 写寄存器的指令的PC |
| RWD[31:0] | I | 写寄存器的数据RegWriteData |
| V1D[31:0] | O | D级读出值1 |
| V2D[31:0] | O | D级读出值2 |
|  |  |  |
| CMP |  | 有符号 |
| Instr[31:0] | I | 整个指令做比较器选择信号(统一用A2做func的情况\_ |
| RD1[31:0] | I | 待比较的寄存器值1 |
| RD2[31:0] | I | 待比较的寄存器值2 |
| CMPout | O | 比较结果输出 |
|  |  |  |
| EXT |  |  |
| Immin[15:0] | I | 立即数输入 |
| EXTOp[1:0] | I | 扩展选择信号 |
| ImmOut\_D[15:0] | O | D级扩展立即数输出信号 |
|  |  |  |
| NPC |  |  |
| RegRead | I | 寄存器的地址值，jr,jalr用 |
| Offset[15:0] | I | 偏移量 |
| InstrIndex[25:0] | I | j与jal用的跳转目标 |
| PC+4 | I | PC+4输入 |
| PC[31:0] | I | PC输入 |
| NPCsel[1:0] | I | NPC选择信号 |
| Instr[31:0] | I |  |
| NPCout[31:0] | O | npc输出 |
| PC+8[31:0] | O |  |

|  |  |  |
| --- | --- | --- |
| Execute | 执行 |  |
| PC[31:0] | I | PC输入 |
| V1[31:0] | I | 寄存器读出值1 |
| V2[31:0] | I | 寄存器读出值2 |
| ExtImm[31:0] | I | 扩展后的立即数 |
| Shift[10:6] | I | 移位运算用的数 |
| reset | I | 指令清除(暂停&bgezal等有条件写入) |
|  |  |  |
| clk | I |  |
| PCAdd8[31:0] | I | PC+8输入 |
| PCE[31:0] | O | E级PC输出 |
| V1E[31:0] | O | E级寄存器读出值1 |
| V2E[31:0] | O | E级寄存器读出值2 |
| ImmOutE[31:0] | O | E级扩展后的立即数 |
| ShiftE[4:0] | O | E级移位运算用的数 |
| PCAdd8\_E[31:0] | O | PC+8输出 |
|  |  |  |
| ALU |  |  |
| SrcA\_E[31:0] | I | 运算数A |
| SrcB\_E[31:0] | I | 运算数B |
| Shift\_E[4:0] | I | 移位数 |
| ALUCtrl[3:0] | I | ALU控制信号 |
| AO\_E[31:0] | O | E级ALU运算结果 |

|  |  |  |
| --- | --- | --- |
| Memory | 存储 |  |
| clk | I | 时钟 |
| reset | I | 同步复位,全部置0 |
| PC\_E[31:0] | I |  |
| AO\_E[31:0] | I | E级ALU运算结果 |
| WDM\_E[31:0] | I | E级写内存的数据WriteDataMemoryE |
| PCAdd8\_E[31:0] | I |  |
| AO\_M[31:0] | O | M级ALU运算结果 |
| WDM\_M[31:0] | O | M级写内存的数据 |
| PC\_M[31:0] | O |  |
| PCAdd8\_M[31:0] | O |  |
|  |  |  |
| DM |  |  |
| clk | I | 时钟 |
| reset | I | 同步复位置0 |
| WE | I | 内存写使能 |
| Load[3:0] | I | 读写位控制信号 |
| AO\_M[31:0] | I | 内存写入地址 |
| WDM\_M[31:0] | I | 内存写入数据 |
|  |  |  |
| PCM | I | M级PC |
| DMout\_M | O | M级DM读取数据 |
|  |  |  |
| WriteBack | 写回 |  |
| clk | I | 时钟 |
| reset | I | 同步复位置0 |
| DMout\_M | I | M级DM读取数据 |
| AO\_M[31:0] | I | M级ALU运算结果 |
| PCAdd8\_M[31:0] | I |  |
| PC\_M | I | M级PC |
|  |  |  |
| DMout\_W | O | W级DM读取数据 |
| AO\_W[31:0] | O | W级ALU运算结果 |
| PCAdd8\_W[31:0] | O |  |
| PC\_W | O | W级PC |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Instr[31:0] | I | 指令信息输入 |  |  |
|  |  |  |  |  |
| ControlUnitD |  |  |  |  |
| CMPrst | I | 比较结果输入 |  |  |
| MUX\_PC\_sel | O | MUX\_PC选择信号(+4or跳) |  |  |
| EXTCtrl[1:0] | O | 扩展单元控制信号 |  |  |
|  |  |  |  |  |
| ControlUnitE |  |  |  |  |
| ALUSrcB | O | 选择ALU的SrcB来源 |  |  |
| ALUCtrl[3:0] | O | ALU控制信号 |  |  |
|  |  |  |  |  |
| ControlUnitM |  |  |  |  |
| MemWrite | O | 内存写使能 |  |  |
| Load[2:0] | O | 读写位控制信号 |  |  |
|  |  |  |  |  |
| ControlUnitW |  |  |  |  |
| Mem2Reg[1:0] | O | 寄存器堆写使能， |  |  |
| RegWE |  | 有的还取决于CMPout |  |  |
|  |  |  |  |  |
| HazardUnit |  |  |  |  |
| StallUnit:算出当前指令的rs,rt,A3，用Res寄存器传下去 |  |  |  |  |
| Instr[31:0] | I | 指令信息输入 |  |  |
| Tnew\_E | I |  |  |  |
| Tnew\_M | I |  |  |  |
| Tnew\_W | I |  |  |  |
| Stall | O |  |  |  |
| rs\_D | O |  |  |  |
| rt\_D | O |  |  |  |
| Tnew\_rs\_D | O | ResE | ResM | ResW |
| Tnew\_rt\_D | O | Tnew | Tnew | Tnew |
|  |  | rsE | A3M | A3W(输出到RF |
|  |  | rtE |  |  |
| ForwardUnit |  | A3E |  |  |
| rsD | I | reset(暂停orbgezal失败时，清空rsE,rtE,A3E) |  |  |
| rtD | I |  |  |  |
| rsE | I |  |  |  |
| rtE | I |  |  |  |
| A3E | I |  |  |  |
| A3M | I |  |  |  |
| A3W | I |  |  |  |
| sel\_V1D | O |  |  |  |
| sel\_V2D | O |  |  |  |
| sel\_V1E | O |  |  |  |
| sel\_V2E | O |  |  |  |
| sel\_WDMM | O |  |  |  |

三、测试程序：

暂停系列

0, E1:

lui $t0 0xf65b

ori $t1 $0 0xabfe

ori $t2 $0 0x3000

lui $t3 0x49bf

addu $t5 $t1 $t3

addu $t4 $t1 $t3

beq $t4 $t5 end #cal-beq

subu $t6 $t2 $t3

func:

ori $1 $0 4

subu $31 $31 $1

jr $31 #cal-jr

sw $t4 0($0)

jr $31

end: subu $t6 $t3 $t1

jal func

lui $s0 0x462b

ori $s1 $0 0x336c

subu $s2 $s0 $s1

nop

运行结果：

$ 8 <= f65b0000

$ 9 <= 0000abfe

$10 <= 00003000

$11 <= 49bf0000

$13 <= 49bfabfe

$12 <= 49bfabfe

$14 <= b6413000

$14 <= 49be5402

$31 <= 00003040

$16 <= 462b0000

$ 1 <= 00000004

$31 <= 0000303c

\*00000000 <= 49bfabfe

$16 <= 462b0000

$17 <= 0000336c

$18 <= 462acc94

0,E2:

lui $t0 0xf65b

ori $t1 $0 0xabfe

ori $t2 $0 0x3000

lui $t3 0x49bf

addu $t5 $t1 $t3

sw $t5 0($0)

#rs Tuse=0, Tnew\_E=`DM

lw $t4 0($0) #sw-lw

beq $t4 $t5 end #lw-beq

func:

sw $31 4($0)

subu $31 $31 4

beq $t4 $t5 end

lui $t4 0x83ba

nop

lw $31 4($0) #lw-jr

jr $31

addu $t4 $t0 $t2

end:subu $t4 $t3 $t1

jal func

addu $t6 $t4 $t5

ori $s0 $0 0xfa6b

addu $s1 $t4 $s0

运行结果：

$ 8 <= f65b0000

$ 9 <= 0000abfe

$10 <= 00003000

$11 <= 49bf0000

$13 <= 49bfabfe

\*00000000 <= 49bfabfe

$12 <= 49bfabfe

\*00000004 <= 00000000

$12 <= 49be5402

$31 <= 00003054

$14 <= 937e0000

\*00000004 <= 00003054

$ 1 <= 00000000

$ 1 <= 00000004

$31 <= 00003050

$12 <= 83ba0000

$31 <= 00003054

$12 <= f65b3000

$16 <= 0000fa6b

$17 <= f65c2a6b

0,M1

lui $t0 0xf65b

ori $t1 $0 0xabfe

ori $t2 $0 0x3000

lui $t3 0x49bf

addu $t5 $t1 $t3

sw $t5 0($0)

lw $t4 0($0)

addu $t1 $t1 $t3

beq $t4 $t5 end #lw-cal-beq

func:

sw $31 4($0)

subu $31 $31 4

beq $t4 $t5 end

lui $t4 0x83ba

nop

lw $31 4($0)

addu $t1 $t1 $t2

jr $31 #lw-cal-jr

addu $t4 $t0 $t2

end:subu $t4 $t3 $t1

jal func

addu $t6 $t4 $t5

ori $s0 $0 0xfa6b

addu $s1 $t4 $s0

运行结果：

$ 8 <= f65b0000

$ 9 <= 0000abfe

$10 <= 00003000

$11 <= 49bf0000

$13 <= 49bfabfe

\*00000000 <= 49bfabfe

$12 <= 49bfabfe

$ 9 <= 49bfabfe

\*00000004 <= 00000000

$12 <= ffff5402

$31 <= 0000305c

$14 <= 49bf0000

\*00000004 <= 0000305c

$ 1 <= 00000000

$ 1 <= 00000004

$31 <= 00003058

$12 <= 83ba0000

$31 <= 0000305c

$ 9 <= 49bfdbfe

$12 <= f65b3000

$16 <= 0000fa6b

$17 <= f65c2a6b

1,E2

ori $t2 $0 0x3000

lui $t3 0x49bf

addu $t5 $t1 $t3

sw $t5 0($0)

ori $t1 $0 0xabfe

lw $t4 0($0)

addu $t1 $t1 $t4 #lw-cal

ori $t0 $0 0x0008

sw $t0 4($t0)

lw $t0 4($t0) #lw sw

sw $t1 8($t0)

运行结果：

$10 <= 00003000

$11 <= 49bf0000

$13 <= 49bf0000

\*00000000 <= 49bf0000

$ 9 <= 0000abfe

$12 <= 49bf0000

$ 9 <= 49bfabfe

$ 8 <= 00000008

\*0000000c <= 00000008

$ 8 <= 00000008

\*00000010 <= 49bfabfe

1,E2

ori $t1 $0 0xabfe

lw $t4 0($0)

addu $t1 $t4 $t1 #lw-cal

ori $t0 $0 0x0008

运行结果：

$10 <= 00003000

$11 <= 49bf0000

$13 <= 49bf0000

\*00000000 <= 49bf0000

$ 9 <= 0000abfe

$12 <= 49bf0000

$ 9 <= 49bfabfe

$ 8 <= 00000008

Rs 0,E1,E2,M1

lui $t0 0xf65b

ori $t1 $0 0xabfe

ori $t2 $0 0x3000

lui $t3 0x49bf

addu $t5 $t1 $t3

addu $t4 $t1 $t3

beq $t5 $t4 end #cal-beq

subu $t6 $t2 $t3

ori $1 $0 4

subu $31 $31 $1

end: subu $t6 $t3 $t1

sw $t6 0($0)

lw $s0 0($0)

end1:lw $s1 0($0)

beq $s0 $s1 end1

ori $s0 $s1 0x619f

运行结果：

$ 8 <= f65b0000

$ 9 <= 0000abfe

$10 <= 00003000

$11 <= 49bf0000

$13 <= 49bfabfe

$12 <= 49bfabfe

$14 <= b6413000

$14 <= 49be5402

\*00000000 <= 49be5402

$16 <= 49be5402

$17 <= 49be5402

$16 <= 49be759f

$17 <= 49be5402

$16 <= 49be759f

转发系列

lui $t0 0xf65b

ori $t1 $0 0xabfe

lui $t4 0xaf78

tiao:

ori $t2 $0 0x3018

addu $t5 $t1 $t3

beq $t4 $t5 end

lui $t3 0x49bf

j tiao

addu $t4 $t1 $t3

addu $t4 $t4 $t3

end:

subu $t5 $t1 $t2

#addu $t4 $t4 $t3

ori $t3 $0 0x3044

func:

beq $t3 $31 ha

addu $t4 $t4 $t2

ori $t5 $t3 0x6349

jal func

addu $t5 $t2 $t3

subu $t1 $t2 $t5

ha:

subu $t4 $t4 $0

addu $t5 $0 $t4

beq $t4 $t5 end6

subu $t4 $t5 $t3

addu $t6 $t2 $t3

j ha

end6:

ori $t3 $t4 0x26cb

ori $1 $0 48

addu $ra $ra, $1

lui $2 0x41ba

jr $ra

addu $1 $0 $t3

subu $t4 $t4 $t1

ori $s0 $0 24

addu $ra $ra $s0

stein:jr $ra

addu $t4 $t5 $t1

subu $t2 $t4 $t3

ori $1 $0 4

jal stein

addu $ra $ra $1

lui $s1 0x6249

subu $s1 $s1 $31

jal stein

subu $t0 $t3 $t5

lui $t0 0xf65b

ori $t1 $0 0xabfe

lui $t4 0xaf78

tiao:

ori $t2 $0 0x3018

addu $t5 $t1 $t3

beq $t4 $t5 end

lui $t3 0x49bf

j tiao

addu $t4 $t1 $t3

addu $t4 $t4 $t3

end:

subu $t5 $t1 $t2

#addu $t4 $t4 $t3

ori $t3 $0 0x3044

func:

beq $t3 $31 ha

addu $t4 $t4 $t2

ori $t5 $t3 0x6349

jal func

addu $t5 $t2 $t3

subu $t1 $t2 $t5

ha:

subu $t4 $t4 $0

addu $t5 $0 $t4

beq $t4 $t5 end6

subu $t4 $t5 $t3

addu $t6 $t2 $t3

j ha

end6:

ori $t3 $t4 0x26cb

ori $1 $0 48

addu $ra $ra, $1

lui $2 0x41ba

jr $ra

addu $1 $0 $t3

subu $t4 $t4 $t1

ori $s0 $0 24

addu $ra $ra $s0

stein:jr $ra

addu $t4 $t5 $t1

subu $t2 $t4 $t3

ori $1 $0 4

jal stein

addu $ra $ra $1

lui $s1 0x6249

subu $s1 $s1 $31

jal stein

subu $t0 $t3 $t5

ori $t2 $0 0x3000

lui $t3 0x49bf

addu $t5 $t1 $t3

sw $t5 0($0)

lw $t4 0($0)

addu $t1 $t1 $t3

beq $t4 $t5 end

func:

sw $31 4($0)

subu $31 $31 4

beq $t4 $t5 end

lui $t4 0x83ba

nop

lw $31 4($0)

addu $t1 $t1 $t2

jr $31

addu $t4 $t0 $t2

end:subu $t4 $t3 $t1

jal func

addu $t6 $t4 $t5

ori $s0 $0 0xfa6b

addu $s1 $t4 $s0

ori $t2 $0 0x3000

lui $t3 0x49bf

addu $t5 $t1 $t3

addu $t4 $t1 $t3

beq $t4 $t5 end

subu $t6 $t2 $t3

func:

ori $1 $0 4

subu $31 $31 $1

jr $31

sw $t4 0($0)

jr $31

end: subu $t6 $t3 $t1

jal func

lui $s0 0x462b

ori $s1 $0 0x336c

subu $s2 $s0 $s1

nop

运行结果：

$ 8 <= f65b0000

$ 9 <= 0000abfe

$12 <= af780000

$10 <= 00003018

$13 <= 0000abfe

$11 <= 49bf0000

$12 <= 49bfabfe

$10 <= 00003018

$13 <= 49bfabfe

$11 <= 49bf0000

$13 <= 00007be6

$11 <= 00003044

$12 <= 49bfdc16

$13 <= 0000734d

$31 <= 00003044

$13 <= 0000605c

$12 <= 49c00c2e

$12 <= 49c00c2e

$13 <= 49c00c2e

$12 <= 49bfdbea

$11 <= 49bfffeb

$ 1 <= 00000030

$31 <= 00003074

$ 2 <= 41ba0000

$ 1 <= 49bfffeb

$ 1 <= 49bfffeb

$12 <= 49bf2fec

$16 <= 00000018

$31 <= 0000308c

$12 <= 49c0b82c

$10 <= 0000b841

$ 1 <= 00000004

$31 <= 0000309c

$31 <= 000030a0

$12 <= 49c0b82c

$17 <= ffffcf60

$31 <= 000030ac

$ 8 <= fffff3bd

$12 <= 49c0b82c

$ 8 <= f65b0000

$ 9 <= 0000abfe

$12 <= af780000

$10 <= 00003018

$13 <= 49c0abe9

$11 <= 49bf0000

$12 <= 49bfabfe

$10 <= 00003018

$13 <= 49bfabfe

$11 <= 49bf0000

$13 <= 00007be6

$11 <= 00003044

$12 <= 49bfdc16

$13 <= 0000734d

$31 <= 00003044

$13 <= 0000605c

$12 <= 49c00c2e

$12 <= 49c00c2e

$13 <= 49c00c2e

$12 <= 49bfdbea

$11 <= 49bfffeb

$ 1 <= 00000030

$31 <= 00003074

$ 2 <= 41ba0000

$ 1 <= 49bfffeb

$ 1 <= 49bfffeb

$12 <= 49bf2fec

$16 <= 00000018

$31 <= 0000308c

$12 <= 49c0b82c

$10 <= 0000b841

$ 1 <= 00000004

$31 <= 0000309c

$31 <= 000030a0

$12 <= 49c0b82c

$17 <= ffff9ec0

$31 <= 000030ac

$ 8 <= fffff3bd

$12 <= 49c0b82c

$ 8 <= f65b0000

$ 9 <= 0000abfe

$12 <= af780000

$10 <= 00003018

$13 <= 49c0abe9

$11 <= 49bf0000

$12 <= 49bfabfe

$10 <= 00003018

$13 <= 49bfabfe

$11 <= 49bf0000

$13 <= 00007be6

$11 <= 00003044

$12 <= 49bfdc16

$13 <= 0000734d

$31 <= 00003044

$13 <= 0000605c

$12 <= 49c00c2e

$12 <= 49c00c2e

$13 <= 49c00c2e

$29 <= 3cbccd80

$30 <= 2a790000

$30 <= 2a79845e

$31 <= b8100000

$31 <= 8e8b0000

$ 5 <= 06bab9db

\*00000000 <= 06bab9db

$ 1 <= ab1053c7

\*00000004 <= ab1053c7

$13 <= 2411dc29

\*00000008 <= 2411dc29

$24 <= bca89fef

\*0000000c <= bca89fef

$30 <= f3ffd7e9

\*00000010 <= f3ffd7e9

$ 3 <= d51e6c67

\*00000014 <= d51e6c67

$ 2 <= e71176c6

\*00000018 <= e71176c6

$ 7 <= 56a89ecc

\*0000001c <= 56a89ecc

$26 <= fb1af48f

\*00000020 <= fb1af48f

$29 <= f8091e3a

\*00000024 <= f8091e3a

$23 <= 3bbdf53f

\*00000028 <= 3bbdf53f

$19 <= c1351f3f

\*0000002c <= c1351f3f

$ 2 <= c0befca1

\*00000030 <= c0befca1

$ 8 <= 00000000

\*00000034 <= 00000000

$22 <= 823690a5

\*00000038 <= 823690a5

$19 <= 05dfe452

\*0000003c <= 05dfe452

$ 0 <= 00fad8e1

\*00000040 <= 00000000

$28 <= d3b5c932

\*00000044 <= d3b5c932

$ 6 <= d9ad8fdb

\*00000048 <= d9ad8fdb

$ 1 <= 2c08bdef

\*0000004c <= 2c08bdef

$29 <= c880cd9e

\*00000050 <= c880cd9e

$19 <= 69466d5f

\*00000054 <= 69466d5f

$17 <= cd5b6089

\*00000058 <= cd5b6089

$23 <= b6885c2a

\*0000005c <= b6885c2a

$ 1 <= 132ad16c

\*00000060 <= 132ad16c

$ 6 <= bc51db57

\*00000064 <= bc51db57

$ 0 <= 00007e8e

\*00000068 <= 00000000

$13 <= ccce5db0

\*0000006c <= ccce5db0

$21 <= 28c8a16b

\*00000070 <= 28c8a16b

$13 <= 132ad16c

\*00000074 <= 132ad16c

$28 <= d3b5f97e

\*00000078 <= d3b5f97e

$28 <= d3b5fd7f

\*0000007c <= d3b5fd7f

$ 3 <= 2ea885bd

\*00000080 <= 2ea885bd

$ 2 <= c0beffa3

\*00000084 <= c0beffa3

$14 <= 660bf7cf

\*00000088 <= 660bf7cf

$29 <= c880cd9e

\*0000008c <= c880cd9e

$19 <= 42230a3d

\*00000090 <= 42230a3d

$30 <= f3ffd7ed

\*00000094 <= f3ffd7ed

$31 <= 8e8becde

\*00000098 <= 8e8becde

$10 <= bad6f67b

\*0000009c <= bad6f67b

$15 <= b0a87fdd

\*000000a0 <= b0a87fdd

$26 <= fb1af7ff

\*000000a4 <= fb1af7ff

$23 <= b6887f2a

\*000000a8 <= b6887f2a

$27 <= 2ea885bd

\*000000ac <= 2ea885bd

$31 <= 8e8becdf

\*000000b0 <= 8e8becdf

\*000000b4 <= c880cd9e

\*000000b8 <= 8e8becdf

\*000000bc <= 132ad16c

$29 <= 00000fdc

$ 1 <= 00000020

$31 <= 00003474

\*00000fdc <= 76b2c1bd

\*00000fec <= 00003474

\*00000fe0 <= 06bab9db

\*00000fe4 <= bc51db57

\*00000fe8 <= 56a89ecc

\*00000ff0 <= 00000000

\*00000ff4 <= 91fd296d

\*00000ff8 <= bad6f67b

$17 <= 0000006c

$18 <= 00000064

$ 8 <= 00000000

$ 9 <= 132ad16c

$ 7 <= 132ad16c

$ 4 <= 76b2c1bd

$ 5 <= 19e58b47

$10 <= 90984d04

$ 4 <= 9098ef87

$10 <= 21309a08

$10 <= 42613410

$ 5 <= 42617712

$ 6 <= 426177f2

$31 <= 000037dc

$29 <= 00000fbc

\*00000fcc <= 000037dc

\*00000fbc <= 9098ef87

\*00000fc0 <= 42617712

\*00000fc4 <= 426177f2

\*00000fc8 <= 132ad16c

\*00000fd0 <= 00000000

\*00000fd4 <= 132ad16c

\*00000fd8 <= 42613410

$17 <= 000000bb

$18 <= 00000154

$ 9 <= 2ea885bd

$ 8 <= 132ad16c

$ 7 <= 41d35729

$ 4 <= a3c3c0f3

$ 5 <= 7109fccf

$10 <= 14cdbdc2

$ 4 <= 14cdfdc6

$10 <= 299b7b84

$10 <= 5336f708

$ 5 <= 5336f70d

$ 6 <= 5336f7fd

$31 <= 00003cec

$29 <= 00000f9c

\*00000f9c <= 14cdfdc6

\*00000fa0 <= 5336f70d

\*00000fac <= 00003cec

\*00000fa4 <= 5336f7fd

\*00000fa8 <= 41d35729

\*00000fb0 <= 132ad16c

\*00000fb4 <= 2ea885bd

\*00000fb8 <= 5336f708

$17 <= 000000d6

$18 <= 0000004b

$ 8 <= c0befca1

$ 9 <= bad6f67b

$ 7 <= 7b95f31c

$ 4 <= d58cfa67

$ 5 <= 0e0ded88

$10 <= e39ae7ef

$ 4 <= e39af7ef

$10 <= c735cfde

$10 <= 8e6b9fbc

$ 5 <= 8e6bdfbe

$ 6 <= 8e6bdffe

$31 <= 00003c7c

$29 <= 00000f7c

\*00000f7c <= e39af7ef

\*00000f8c <= 00003c7c

\*00000f80 <= 8e6bdfbe

\*00000f84 <= 8e6bdffe

\*00000f88 <= 7b95f31c

\*00000f90 <= c0befca1

\*00000f94 <= bad6f67b

\*00000f98 <= 8e6b9fbc

$17 <= 000000c6

$18 <= 00000119

$ 9 <= 2ea885bd

$ 8 <= f8091e3a

$ 7 <= 26b1a3f7

$ 4 <= dba41629

$ 5 <= bd14657b

$10 <= 98b87ba4

1. 冲突处理

|  |  |
| --- | --- |
| rs暂停 |  |
| 0,E1 | calc + beq/jr |
| 0,E2 | lw + beq/jr |
| 0,M1 | lw + x + beq/jr |
| 1,E2 | lw + cal |
|  | lw + sw |
|  |  |
| rt暂停 |  |
| 0,E1 | calc + beq |
| 0,E2 | lw + beq |
| 0,M1 | lw + x + beq |
| 1,E2 | lw + cal |

|  |  |
| --- | --- |
| rs转发 |  |
| 0,E\_PC | 暂无 |
| 0,M\_ALU | cal + x + beq/jr |
| 0,M\_PC | jal + delay + beq |
|  | jal + delay + jr |
| 0,W\_ALU | jal + addu$31,$31,4 + jr |
|  |  |
| 0,W\_DM | lw + x + beq/jr |
| 0,W\_PC | jal + delay + beq/jr |
| 1,E\_ALU | cal/lw + cal/lw |
| 1,E\_PC | jal + cal/lw/sw |
| 1,M\_ALU | cal + cal |
| 1,M\_DM | lw + cal |
| 1,M\_PC | jal + lw/sw/cal |
| 1,W\_ALU | cal + x + x + cal |
| 1,W\_DM | lw + x + x + cal |
| 1,W\_PC | jal + x + x + cal |
|  |  |
| rt转发(仅列出Tuse=2的情况，Tuse=1的情况与rs类似) | |
| 2,E\_ALU | cal + sw |
| 2,E\_DM | lw + sw |
| 2,E\_PC | jal + sw |
| 2,M\_ALU | cal + x + sw |
| 2,M\_DM | lw + x + sw |
| 2,M\_PC | jal + x + sw |
| 2,W\_ALU | cal + x + x + sw |
| 2,W\_DM | lw + x + x + sw |
| 2,W\_PC | jal + x + x + sw |