

Comparch Lab01

PB18111757 陈金宝

1

XOR属于R类型指令。

在IF段根据PC从指令内存中取出对应的指令。

在ID段将取出的指令的OP(instr[0:6]),func3(instr[12:14]),func7(instr[25:31])送入control unit,产生控制信号: RegWriteD, RegReadD, AluControlD, Alusrc1D, Alusrc2D。寄存器根据 RegRead 信号以及指令中的 rs 和 rt, 读出对应的寄存器数据(RegOut1D, RegOut2D)。根据 rd 字段将写入的寄存器号读出。rs2为inst[24:20], rs1为inst[19:15]。RegWrite=0。

之后将信号以及读出的寄存器数据向后传到EX段。在EX段根据传来的 AluControlE, alu决定进行xor操作。根据 AluSrc1E=0, AluSrc2E=0, RegOut1E, RegOut2E 选取操作数, alu对指定的数据进行xor操作。

计算后的结果 AluOutE 以及 Regwrite, rdE 信号继续向后传到M段,M段 MemWrite =0。

M段继续将信息向后传到W段。W段有MemTORegW=0, RegWriteW=1。W段将 rdw, Resultw,以及 RegWrite 信号传给寄存器, 向指定寄存器写入数据。

2

beq属于B类指令。只有IF,ID和EX段。

在IF段取指后进入ID段。

ID段指令送入Control unit。产生RegWrite, MemtoRegD, MemWriteD, LoadReadD, RegReadD, BranchTypeD, AluContrlD, AluSrc1D, AluSrc2D, ImmType, RegWriteW信号。rs2为inst[24:20], rs1为inst[19:15], RegWrite=0。同时寄存器取出对应的源寄存器值以及offset并进行符号扩展并计算出跳转地址(JalNpc)。

这些值和信号送入EX段。在EX段,Branch Decision接收从ID段接受的BranchType以及REG1,REG2信号。决定是否跳转。跳转则产生BrE信号送入NPC GEN。BrNPC送入NPC GEN的BrT口。

3

LHU属于I类指令。

IF段取指后进入ID段。

ID段从寄存器取出RS1, 对offset进行扩展, control unit产生RegWrite, MemtoRegD, MemWriteD, LoadReadD, RegReadD, BranchTypeD, AluContrlD, AluSrc1D, AluSrc2D, ImmType信号。rs1为inst[19:15], rd为inst[11:7]。RegWrite=0。计算imm。之后rd,rs1,offset和相应信号被送入EX段。

EX段ALU根据AluControl和送入的操作数计算出目标地址ALuOutE(rs1 + sext(offset))。之后将该值以及rd,RegWrite,MemtoReg信号送入MEM段。

MEM段MemWriteM=0, 根据地址取出数据RD, 将其以及信号rd,RegWrite,MemtoReg送入WB段。

WB段MemTORegW=1, RegWriteW=1。根据Data Ext的Out写入寄存器

4

部件:

增加CSR寄存器组。

数据通路:

增加CSR寄存器的读写信号, 读地址信号, 写地址信号, 寄存器输入、输出选择信号 (选择CSR或寄存器组),

立即数扩展后与CSR寄存器组选择信号相连。

在EX, MEM, WB等段增加MUX, 从而控制对CSR的读写。

同时扩展ALUControl信号, 以便支持对应的操作。

类似于寄存器文件的读写操作, ID段读出对应寄存器值, EX段进行操作, WB进行写回。

5

伪代码 (k是要扩展的位数,top为要扩展的数的最高位, 即符号位)

```
case opcode
  零扩展: assign out = {k{0},offset}
  符号扩展: assign out = {k{offset[top]},offset}
endcase
```

6

数据线依旧为32位, 此时:

load:若以地址对齐的方式load byte和half word, 则进行符号扩展到32位。若地址是非对齐的。则可分多次读取后进行拼接。

store:若以地址对齐的方式存储byte和half word则增加控制信号, 控制memory写入的位数(如00:byte 01:half word,10:word)。若地址是非对齐的, 则类似load的处理, 拆开分多次存储。

7

默认无符号

8

表示branch是否命中。此时若命中NPC GEN会进行跳转同时清空对应流水段。

9

branch和jalr是EX段跳转, 而jal是ID段。所以必须设置优先级, 使得在后的指令先跳转。同时, 若修改数据通路, 使得br,jal,jalr均在EX段跳转, 则不会有冲突, 此时也就不需要设置优先级。

10

load后若立即使用, 此时会有冲突。需要插入1个气泡。

if id ex me wb

if id ex me wb

if id ex me wb

11

branch在EX段跳转。未命中则flush信号为0，不进行清空。branch命中后需要控制flush信号为1来清空跳转后的ID和EX段。并在之后flush信号回到0

12

会产生影响。涉及到x0寄存器时就不需要forward (x0恒为0)