

Lab 1 report

PB21020485 吴敌

1

在IF段根据PC从指令cache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` 等等。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。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` ,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` 信号继续向后传到 `MemWrite` =0, 所以不进行写入MEM。

继续将传到REG_FILE。有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

在IF段取指。

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并进行符号扩展并计算出跳转地址。

这些值送入EX段。Branch Decision接收 `BranchTypeD` 以及REG1,REG2。决定是否跳转。跳转则产生BR信号送入NPC GEN。

3

在IF段取指。

ID段从寄存器取出RS1，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` 。之后信号被送入EX段。

EX段ALU根据AluControl和操作数计算出目标地址 `ALUOutE(rs1 + sext(offset))` 。之后信号送入MEM段。

MEM段 `MemWriteM=0` , 根据地址取出RD, 将其送入WB段。

WB段 `MemT0RegW=1` , `RegWriteW=1` 。写入寄存器。

4

增加CSR寄存器组。

增加CSR寄存器的读写信号，读地址信号，写地址信号，寄存器/CSR选择信号，立即数扩展与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。

5

立即数扩展：比如20位立即数：无符号扩展： `{12'b0, imm}`; 有符号扩展： `{12{imm[19]}}, imm}` 。

6

load: 可分多次读取后进行拼接。

store: 先读取原有数值，拆分开多次存储。

7

ALU默认无符号数运算。

8

表示branch是否命中。若命中会进行跳转，并清空对应流水段。

9

branch和jalr是EX段跳转，而jal是在ID段。所以需要设置优先级，使得在后的指令先跳转。

10

load后若立即使用，此时会有冲突，需要停顿一个周期。

11

branch命中后需要控制flush信号为1来清空跳转后的ID和EX段。

同时stall信号为1表示流水线停顿。

12

会产生影响。涉及到 `x0` 寄存器时就不需要forwarding。