

Lab1

1.

IF: 从 IF PC 寄存器中取出指令, 通过指令缓存写入 IR 中

ID: 控制器根据指令产生控制信号。GRF 根据寄存器 rs1 的地址将 rs1 的值写入 Op1 寄存器, 12 位立即数经过拓展后和 Reg2 一起进入一个多选器, 控制多选器的 Op2 Src 信号选择 imm 作为输出写入到 Op2 寄存器中。同时, 指令中目的寄存器 rd 的地址直接写入 addr 寄存器中。

EX: 控制信号 Op1 set 和 Op2 set 选择 Reg1 和 Reg2OrImm 作为 ALU 运算单元的输入, 控制信号 ALU Func 使 ALU 将输入两个数相加并输出结果 ALU Out。信号 Load NPC 控制多选器将 ALU Out 写入 Result 寄存器中。寄存器 rd 的地址从 ID 段的 Addr 寄存器直接发送到 EX 段的 Addr 寄存器。

MEM: 控制信号 Wb set 控制多选器将 Result 写入 WBData 寄存器中。寄存器 rd 的地址从 EX 段的 Addr 寄存器直接发送到 MEM 段的 Addr 寄存器。

WB: WBData 寄存器和 Addr 寄存器将要写回的数据和要写回的地址发送到 GRF, 将运算结果写回寄存器 rd 中

2.

IF: 从 IF PC 寄存器中取出指令, 通过指令缓存写入 IR 中。

ID: 控制器根据指令产生控制信号。GRF 根据寄存器 rs1 的地址将 rs1 的值写入 Op1 寄存器, 12 位立即数经过拓展后和 Reg2 一起进入一个多选器, 控制多选器的 Op2 Src 信号选择 imm 作为输出写入到 Op2 寄存器中。同时, 指令中目的寄存器 rd 的地址直接写入 addr 寄存器中。控制器产生的 Jalr 信号。

EX: 控制信号 Op1 set 和 Op2 set 选择 Reg1 和 Reg2OrImm 作为 ALU 运算单元的输入, 控制信号 ALU Func 使 ALU 将输入两个数相加并输出结果 ALU Out。ALU Out 作为下一次跳转的地址写回 PC 多选器。控制器产生的 Jalr 信号使得 PC 多选器选择 Jalr Target 作为下一个 PC。Load Npc 信号选择 PCE(即 pc+4)写入 Result 寄存器中。寄存器 rd 的地址从 ID 段的 Addr 寄存器直接发送到 EX 段的 Addr 寄存器。

MEM: 控制信号 Wb set 控制多选器将 Result 写入 WBData 寄存器中。寄存器 rd 的地址从 EX 段的 Addr 寄存器直接发送到 MEM 段的 Addr 寄存器。

WB: WBData 寄存器和 Addr 寄存器将要写回的数据和要写回的地址发送到 GRF, 将 pc+4 写回寄存器 rd 中。

3.

IF: 从 IF PC 寄存器中取出指令, 通过指令缓存写入 IR 中。

ID: 控制器根据指令产生控制信号。GRF 根据寄存器 rs1 的地址将 rs1 的值写入 Op1 寄存器, 12 位立即数经过拓展后和 Reg2 一起进入一个多选器, 控制多选器的 Op2 Src 信号选择 imm 作为输出写入到 Op2 寄存器中。同时, 指令中目的寄存器 rd 的地址直接写入 addr 寄存器中。

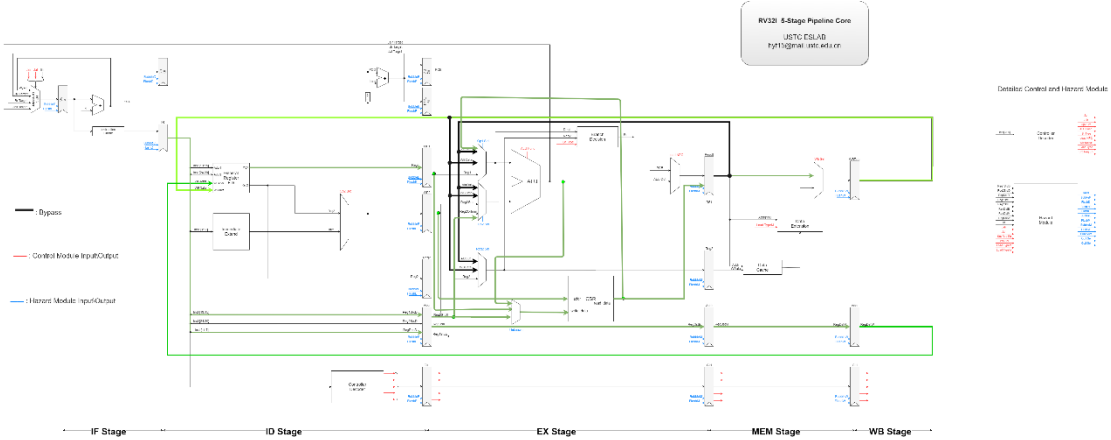
EX: 控制信号 Op1 set 和 Op2 set 选择 Reg1 和 Reg2OrImm 作为 ALU 运算单元的输入, 控制信号 ALU Func 使 ALU 将输入两个数相加并输出结果 ALU Out。ALU Out 作为下一次跳转

的地址写回 PC 多选器。控制器产生的 Jalr 信号使得 PC 多选器选择 Jalr Target 作为下一个 PC。Load Npc 信号选择 ALU Out(即 memory 的地址)写入 Result 寄存器中。寄存器 rd 的地址从 ID 段的 Addr 寄存器直接发送到 EX 段的 Addr 寄存器。

MEM: Data Cache 根据 Result 寄存器中的地址取出数据，控制信号 Load TypeM 控制读取出的数据为 32 位数。Wb set 信号控制多选器将 mem 中读出的数据写入 WBData 寄存器中。寄存器 rd 的地址从 EX 段的 Addr 寄存器直接发送到 MEM 段的 Addr 寄存器。

WB: WBData 寄存器和 Addr 寄存器将要写回的数据和要写回的地址发送到 GRF，将 pc+4 写回寄存器 rd 中。

4. 需要在 EX 段中增加一个 CSR（控制与状态寄存器），有一个数据写端口，有一个地址写端口，一个数据读端口。输入的地址总是 csr，要写入的数据根据指令不同可能为寄存器 rs 的值，rs 对应的立即数，rs 与 csr 中的值进行或/异或运算。读出的 csr 的值写回 rd。



5. I-type: ADDI
S-type: SW
B-type BEQ
U-type: LUI
J-type: JAL
将其拓展为 32 位的方法如下图所示

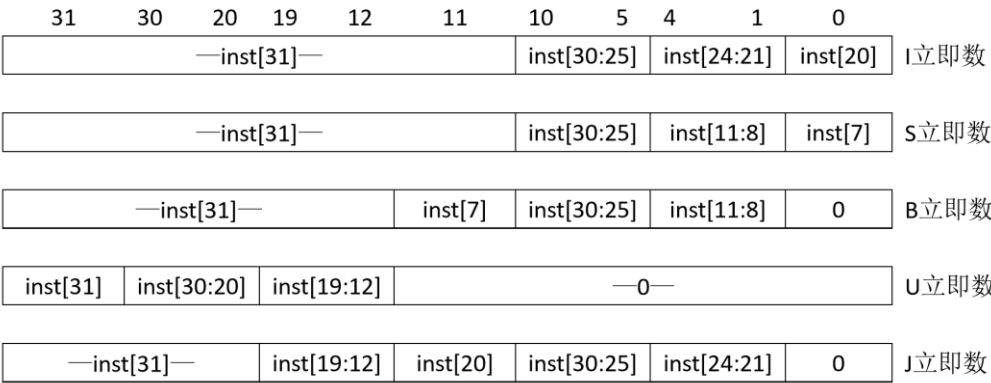


图 2-4 RISC-V 指令生成的立即数，图中各的位标注了用于构成立即数的位段，红色标注为

6.

1.访问半字时，若数据在同一个字中，我们可以用地址的最后两位来指定访问开始的位置

2 访问半字或字时，若数据在不同的字中，我们可以分两次访问取出字对齐的数据再根据需要拼接

7.

默认 wire 为无符号数

8.

当执行 Jalr 指令时，写入寄存器 rd 的为 PCE，此时 Load Npc=1 使多选器选择 PCE 作为输出。

9.

Br 的优先级应该高于 Jal 和 Jalr。

因为 Jal 和 Jalr 在 ID 阶段就由控制器产生了，而 Br 信号在 EX 段才产生。如果 Br 信号和 Jal/Jalr 同时为高，说明 Jal/Jalr 指令前有一个 Br 指令需要跳转，这时候优先处理 Br。

10.

数据冒险：RAW 冲突需要插入气泡

控制冒险：Jump 和条件转移指令在某些情况下（如没有延迟槽或者采取了 stall 直到分支确定的策略）会产生气泡

11.

Hazard 模块在 EX 段产生关于 Branch 的清空流水线信号，此时 Br Target 已经将下一个 PC 写入 IF PC,所以我们只需要使 FlushD 和 FlushE 有效,其他的 Flush 和 Bubble 信号置为无效。

12.

指令对 0 号寄存器的写入会被丢弃，从 0 号寄存器中读出的数据始终是 0，所以我们在处理前一个指令对 0 号寄存器进行写入，后一个指令从 0 号寄存器读数据的情况下不需要 forward。