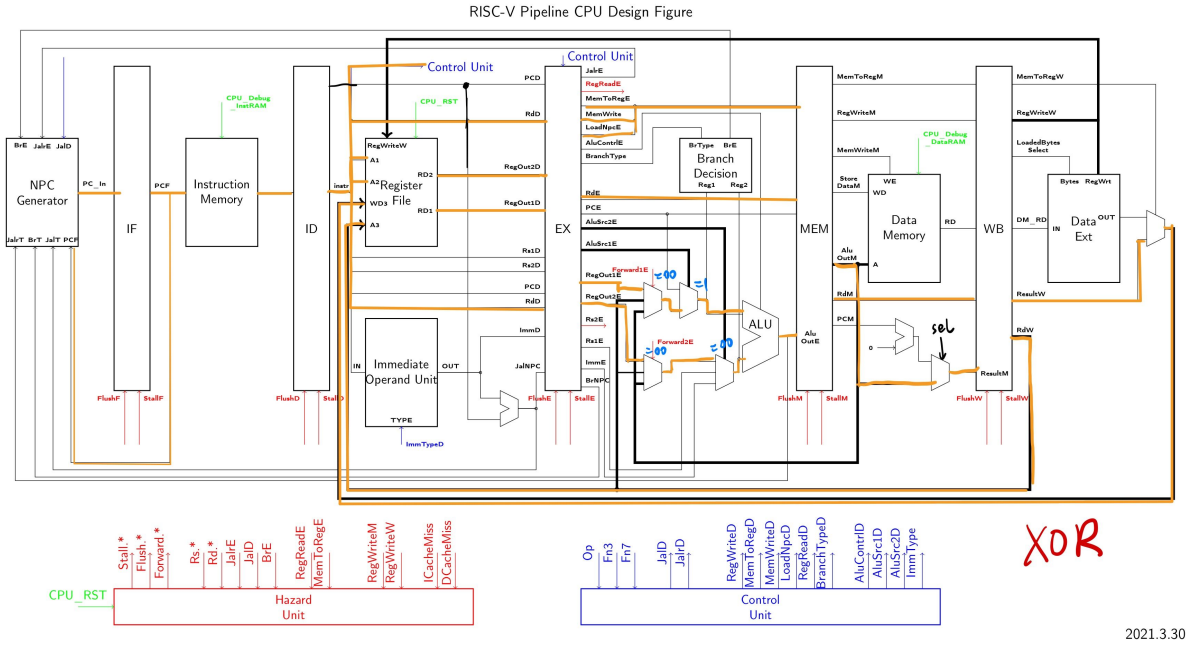
Lab1 实验报告

雷雨轩 PB18111791 计算机学院

> 一律认为选择器输入从上到下依次为0,1 或 00,01,10,11

1. 描述执行一条 XOR 指令的过程（数据通路、控制信号等）。

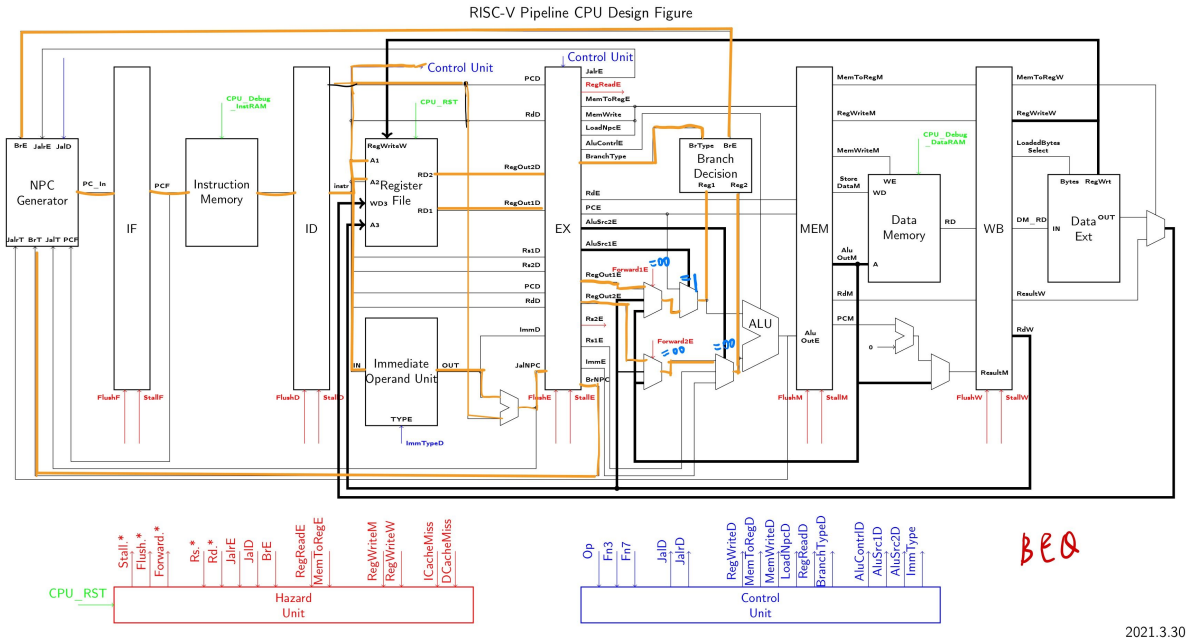


- XOR为一条R型指令

0000000	rs2	rs1	100	rd	0110011	xor

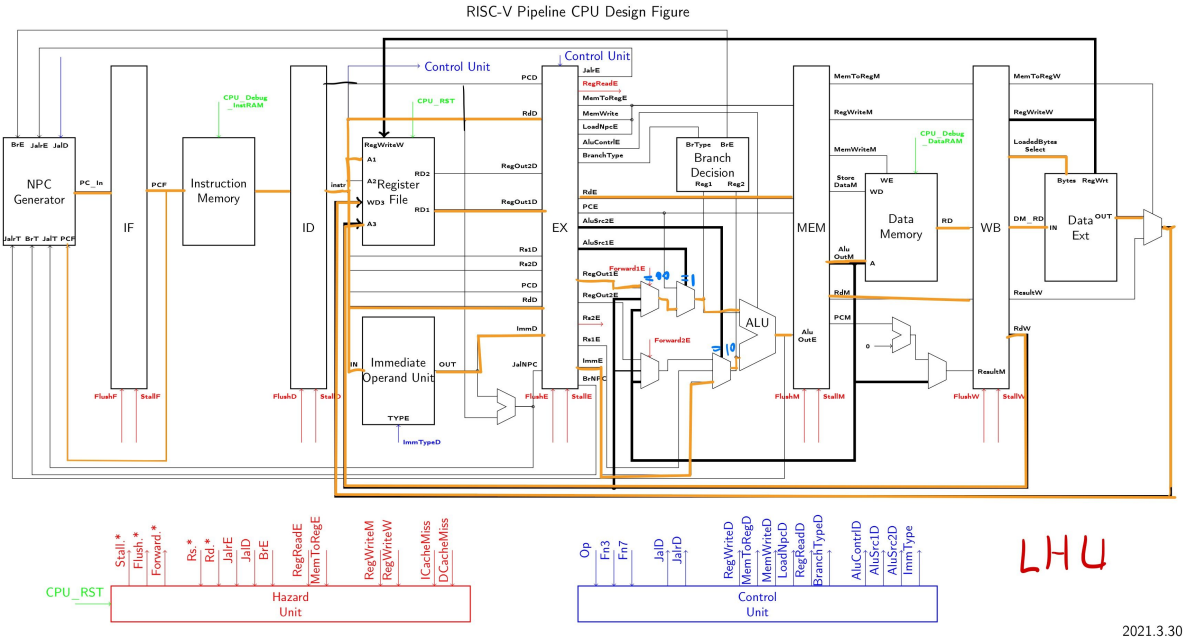
- IF：IR=Instruction Memory[PC]
- ID:
  - opcode = 0110011，funct3 = 100，funct7=0000000
  - A1=Rs1D = IR [19:15]
  - A2=Rs2D = IR[24:20]
  - RdD=IR[11:7]
- EX:
  - Forward1E=00, Forward2E=00, AluSrc1E=1, AluSrc2E=00
  - MemToRegE = 1, MemWrite=0, LoadNpcE=0,
  - AluContrlE=xor
  - RdE直传
  - AluOutE=RegOut1E xor RegOut2E
- MEM:
  - sel=1
  - MemToRegM=1, RegWriteM=1, MemWriteM=0
  - RdM直传
- WB:
  - MemToRegW=1
  - RegWriteW=1
  - A3=RdW

2. 描述执行一条 BEQ 指令的过程·（数据通路、控制信号等）。



	funct3			opcode	
imm[12|10:5]	rs2	rs1	000	imm[4:1|11]	1100011	BEQ
- IF：IR=Instruction Memory[PC]
- ID:
  - opcode = 1100011， funct3 = 000
  - IN=imm[12] | imm[11] | imm[10:5] | imm[4:1] （|表示拼接）
  - immD = sign_extend(IN|0)
  - A1 = Rs1D=IR [19:15]
  - A2 = Rs2D=IR[24:20]
  - jalNPC = immD+PCD
- EX:
  - Forward1E=00, Forward2E=00, AluSrc1E=1, AluSrc2E=00
  - Reg1=RegOut1E, Reg2=RegOut2E
  - BrT=BrNPC, 为跳转的目标地址，BrNPC=jalNPC
  - BrE根据Reg1和Reg2的比较，若相等则跳转，BrE=1; 否则BrE=0
  - BranchType=BEQ

3. 描述执行一条 LHU 指令的过程（数据通路、控制信号等）。



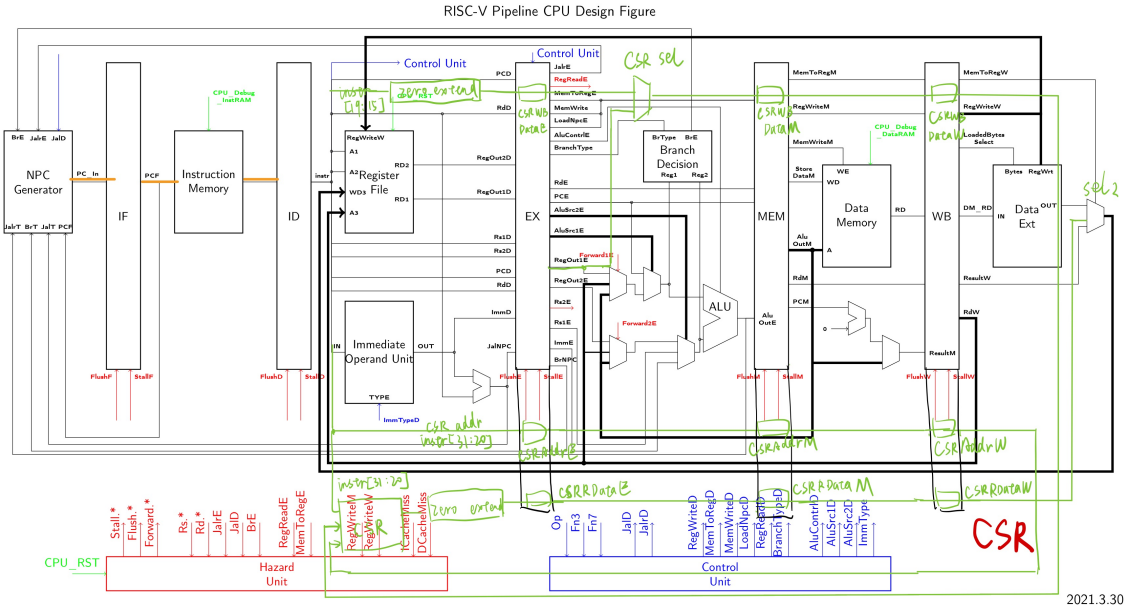
- | imm[11:0] | rs1 | 110 | rd | 0000011 | lhu |

  LHU指令读取一个16位无符号数值，零扩展到32位

- IF：IR=Instruction Memory[PC]
- ID:
  - opcode = 0000011，funct3 = 110
  - A1= IR [19:15]
  - RdD=IR[11:7]
  - IN=sign_extend(Imm[11:0])
- EX:
  - Forward1E=00, AluSrc1E=1, AluSrc2E=10
  - MemToRegE = 0, MemWrite=0, LoadNpcE=0,
  - AluContrlE=Add
  - RdE直传
  - AluOutE=RegOut1E + ImmE
- MEM:
  - MemToRegM=0, RegWriteM=1, MemWriteM=0
  - 从Data memory里读取数据， 地址A=AluOutM
  - RdM直传
- WB:
  - MemToRegW=0
  - RegWriteW=1
  - A3=RdW
  - LoadedBytesSelect=lhu (读取一个16位无符号数值，零扩展到32位)

4. 如果要实现 CSR 指令（csrrw，csrrs，csrrc，csrrwi，csrrsi，csrrci），设计图中还需要增加什么部件和数据通路？给出详细说明。

-



-

31 20	19 15	14 12	11 7	6 0
csr	rs1	funct3	rd	opcode
12	5	3	5	7

source/dest	source	CSRRW	dest	SYSTEM
source/dest	source	CSRRS	dest	SYSTEM
source/dest	source	CSRRC	dest	SYSTEM
source/dest	zimm[4:0]	CSRRWI	dest	SYSTEM
source/dest	zimm[4:0]	CSRRSI	dest	SYSTEM
source/dest	zimm[4:0]	CSRRCI	dest	SYSTEM

- 6条指令的详细描述参考`RISC-V_指令集卷2-特权级指令-中文版.pdf` 8-9页
- 需要增加的部件：
  - ID段：增加CSR寄存器、对Instr[19:15]进行0扩展、对从CSR读出的数据进行0扩展
  - 在ID/EX、EX/MEM、MEM/WB段 均加入3个段间寄存器，存放CSR写回地址（CSRAddr）、CSR写回数据(CSRWBData)、CSR读出数据(CSRRData)
  - 在EX段增加CSR SEL的选择器，用来选择CSR写回数据是rs1内容还是zimm[4:0]立即数扩展的内容
  - 将CSR读出数据在WB段接入选择器sel2，用来选择并将其写入rd表示的寄存器
- 增加的数据通路
  - ID：
    - 从instr[19:15]进行零扩展，存入ID/EX段的CSRWBDataE寄存器；
    - instr[31:20]存入CSRAddrE寄存器； 并利用instr[31:20]从CSR寄存器读出数据，零扩展后存入CSRRDataE寄存器
  - EX：
    - 经过CSR sel选择器，选择CSRDataE寄存器以及RegOut1E的值之一，存入CSRWBDataM寄存器
    - CSRAddr和CSRRData寄存器直传
  - MEM：所有相关寄存器直传
  - WB：
    - 由CSRAddrW寄存器里存放的CSR写回地址，把CSRWBDataW寄存器中的值写入CSR寄存器；
    - 利用sel2选择器把CSRRDataW的值写入RdW对应的寄存器

5. Verilog 如何实现立即数的扩展?

- I-type:

  `assign Imm[31:0]={20{IR[31]}, IR[31:20]}`

- S-type:

  `assign Imm[31:0]={20{IR[31]},IR[31:25],IR[11:7]}`

- B-type:

  `assign Imm[31:0]={20{IR[31]},IR[7],IR[30:25],IR[11:8],1'b0}`

- U-type:

  `assign Imm[31:0]={IR[31:12],12'b0}`

- J-type:

  `assign Imm[31:0]={12{IR[31]},IR[19:12],IR[20],IR[30:21],1'b0}`

6. 如何实现 Data Memory 的非字对齐的 Load 和 Store?

- 以1 word=4 Bytes为例
- Load：分两次进行读取。
  - 先找到低位两字节对应的字地址，读取该字并分离出需要的两字节数据，存入寄存器A的低位;
  - 然后增加addr，找到高位两字节所在字地址，读取该字并分离出需要的两字节数据，并写入寄存器A的高位
- Store指令：同样分两次写入，第一次从字里分离出低位字节，写入其对应的字地址；第二次分离出高位字节，写入对应的字地址

7. ALU 模块中，默认 wire 变量是有符号数还是无符号数?

- 无符号数

8. 简述BranchE信号的作用。

- 指示当前B类指令是否跳转
- 根据BrType和Reg1，Reg2的值来判断当前Br指令是否跳转成功。若条件满足，则PC会更新为跳转地址，BranchE=1（指示成功跳转）； 若条件不满足，则PC=PC+4，BranchE=0

9. NPC Generator 中对于不同跳转 target 的选择有没有优先级?

- Br Target和Jalr Target的优先级更高， Jal target的优先级较低， PC+4优先级最低。
- 因为Br、Jalr的指令执行到EX段才会发出Br/Jalr跳转信号，而Jal指令在ID段就会发出Jal信号。如果同时遇到Br/Jalr和Jal的信号，显然Br和Jalr所在的指令更靠前（是先执行的指令）。
- 显然，只有Br/Jalr/Jal信号均无效时，才会选择PC+4，即不跳转的正常情况。

10. Harzard 模块中，有哪几类冲突需要插入气泡，分别使流水线停顿几个周期?

- 主要是RAW相关

  在lw指令后紧接着下一条指令要读取lw指令的目的寄存器中的数据，此时需要插入气泡，需使流水线停顿一个周期

11. Harzard 模块中采用静态分支预测器，即默认不跳转，遇到 branch 指令时，如何控制 flush 和 stall 信号?

- 以该branch指令的各段运行情况来讨论，pc记为Branch指令对应的PC值
  - IF段：取指，此时还不知道是Branch指令
  - ID段：此时发现是Branch指令，并会计算出对应的跳转地址；由于默认不跳转，所以仍正常取指取出下一条指令(PC=pc+4)
  - EX段：此段结束时可以计算出Branch指令是否会跳转，但是仍会继续取下一条指令，即PC=pc+8的指令进入流水线IF段

    当EX段结束时，若判断发生跳转，那么branch指令后紧接着的两条指令就不应该继续执行下去，所以需要清空对应段的内容。所以flushD=1,flushE=1，其余flush和所有stall信号均为0。等EX段结束后，PC地址值(IF段寄存器)正确置为PC=br target。

    反之，若判断不发生跳转，则不需要清空，只需保持流水线继续进行下去即可，flush信号、stall信号全为0。

12. 0 号寄存器值始终为 0，是否会对 forward 的处理产生影响?

- 有影响。0号寄存器不需要forward，所以转发前需要判断转发源的dest register是否是0号寄存器