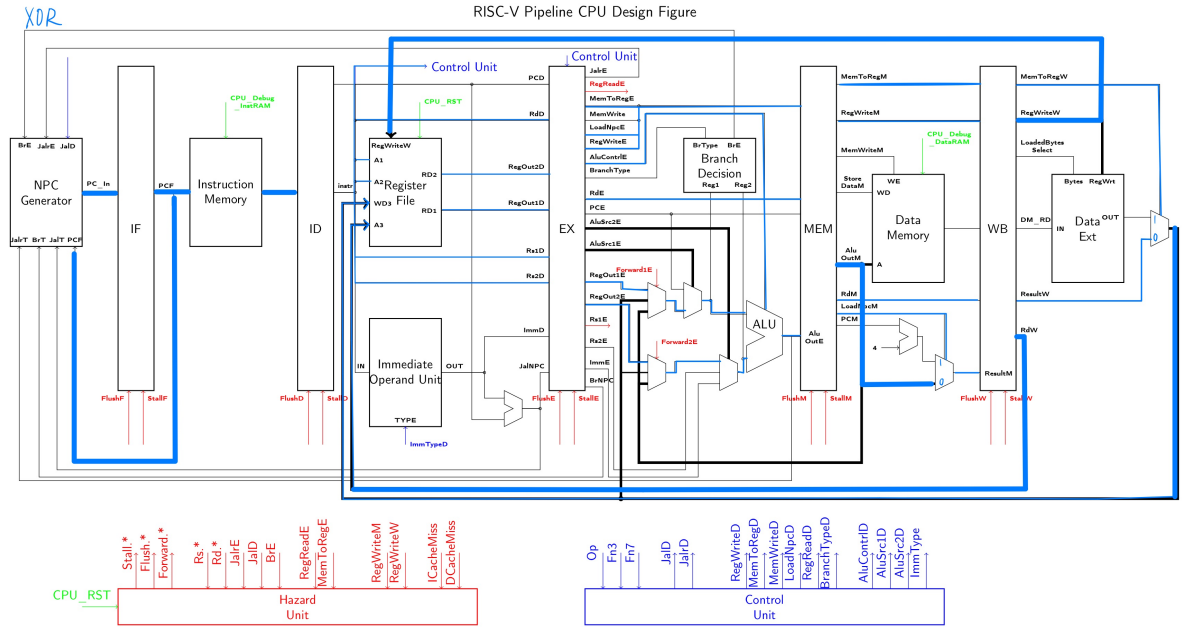


Lab1 实验报告

1、描述执行一条 XOR 指令的过程（数据通路、控制信号等）。

数据通路如下路所示：



2021.3.30

执行过程：

IF: NPC Generator输出PC_in通过IF计算出新的PC值，通过指令存储器读出XOR指令输入寄存器ID中；

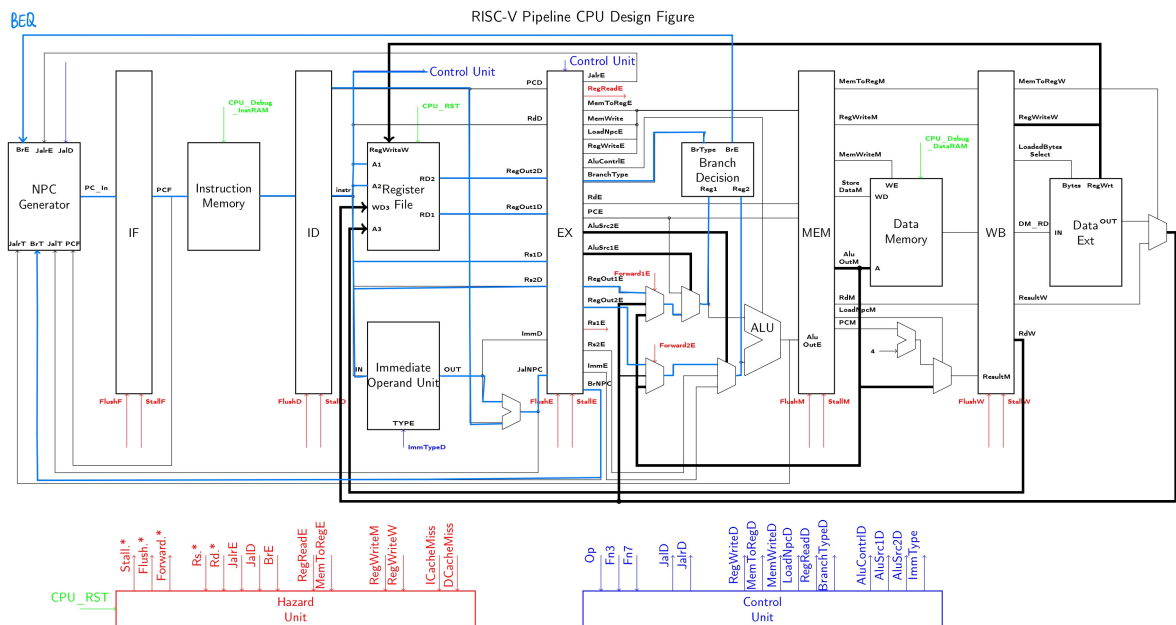
ID: 指令从寄存器ID中输出，op段输入control unit用于控制信号的产生（产生了RegWrite、AluContrl、AluSrc1D、AluSrc2D信号）并进入寄存器EX中，rs1、rs2段控制寄存器堆输出两个寄存器堆中存储的值并输入EX寄存器中，rs1、rs2、rd三个寄存器号同时也进入寄存器EX中；

EX: EX寄存器输出RegWriteE、RdE进入寄存器MEM用于下一级控制；AluContrl信号用于控制ALU模块做异或操作；Rs1E、Rs2E为两个寄存器号，用于Hazard Unit判断是否发生了数据相关从而产生Forward1E、Forward2E信号用于控制转发；RegOut1E、RegOut2E为两个寄存器的值，分别经过两次选择器选择，第一级选择器选择信号Forward1E、Forward2E控制转发，第二级选择器AluSrc1E、AluSrc2E为EX寄存器中的AluSrc1D、AluSrc2D信号的输出，控制Alu的输入，两个寄存器的值经过Alu进行异或运算后将值送入寄存器MEM中；

MEM: 由于并非跳转指令，LoadNpc信号控制结果为AluOut进入寄存器WB中为Result；Rd进入寄存器WB；RegWrite信号进入WB寄存器中用于下一步控制；

WB: 由于XOR指令未涉及到Memory的读取，因而MemToReg信号未生效，故Alu的计算结果随数据通路进入寄存器堆的数据写入端口，Rd的寄存器号传入寄存器堆的写入寄存器号的位置，寄存器的写使能MemWrite生效，将异或的运算结果写入寄存器堆中；

2、描述执行一条 BEQ 指令的过程（数据通路、控制信号等）。



2021.3.30

执行过程：

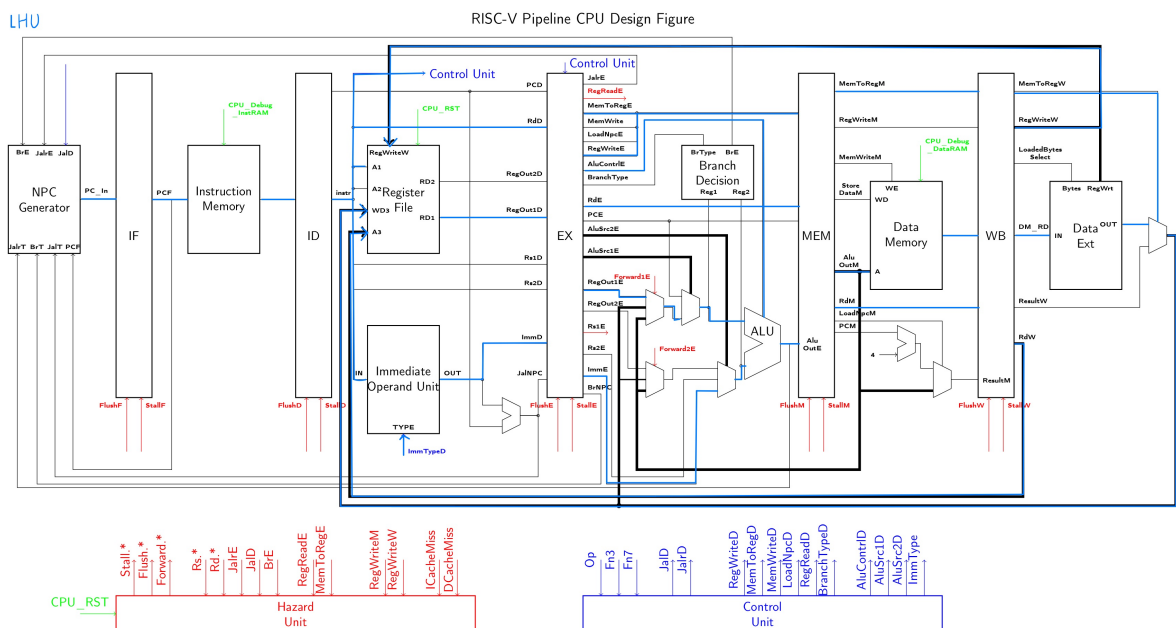
IF：产生新的PC值并经过指令存储读取指令，同时PC值进入下一级寄存器，用于计算跳转的地址；

ID：Op段输入控制模块，产生了BranchType信号进入下一级寄存器，产生ImmType信号控制立即数的扩展类型；Rs1、Rs2通过寄存器堆读取其存储的值，寄存器号同样通过流水线进入下一级用于判断有无数据相关；立即数段经过扩展和上一级存入PC值，通过一个加法器得到跳转的地址进入下一级寄存器中；

EX：两个寄存器号输入冒险模块判断是否存在数据相关，从而产生Forward信号用来控制转发；两个寄存器的值通过两级选择器，第一级选择器为转发的判断，第二级选择器选择的都为第一级选择器产生的值（BEQ指令不涉及该选择器选择信号的改变），两个寄存器的值最终进入Branch Decision模块，BranchType信号为Branch类指令的实际类型，对BEQ为判断两个寄存器的值是否相等，若相等，产生BrE信号，使NPC Generator更新PC值为BrT（BrT为ID段计算的跳转地址）；

BEQ指令的MEM和WB段实际不发生任何操作。

3、描述执行一条 LHU 指令的过程（数据通路、控制信号等）。



2021.3.30

ID：获取新的PC值并从指令存储器中读取新的指令；

IF: 指令的Op段进入控制模块, 产生了ImmType、MemToReg、RegWrite、AluContrl、AluSrc2信号; Rs段通过寄存器堆读取Rs寄存器中存储的值, 同时寄存器号本身进入下一级流水线用于判断是否存在数据相关; 立即数段通过立即数扩展模块进行扩展, 控制信号为ImmType, 产生LHU指令对应的立即数格式, 进入寄存器EX进入下一级; Rd寄存器号进入寄存器EX用于后续数据的写入

EX: Rs1E进入冒险模块, 判断是否存在数据相关从而产生对应的Forward1信号; RegOut1经过两级选择器, 第一级选择器由Forward1控制是否转发, 第二级不改变其值, 进入ALU进行相应运算; 立即数ImmE经过一级选择器, AluSrc2E信号选择ImmE输出, 进入ALU中与寄存器Rs的值计算出需要读取数据的地址;

MEM: MemToReg和RegWrite进入寄存器WB流入下一级; AluOut通过数据存储器读出相应地址中存取的值进入下一级流水线;

WB: MemToReg信号控制选择器选择经过扩展的存储器中存储的值并送入寄存器堆中, 寄存器Rd写入寄存器堆的A3端, RegWrite信号控制寄存器堆写入, 从而将存储器中的值写入寄存器Rd中。

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。

ID:

增加一个寄存器堆用于存储CSR寄存器的值, 其上有写使能, 写数据, 写地址, 读地址以及读数据的端口;

增加一个CSR寄存器值的扩展模块, 用于扩展CSR中存储的值, 并将该值送入下一级;

立即数扩展模块新增对CSR指令立即数的扩展;

控制模块产生AluSrc1、AluSrc2、CSRWrite、ImmType、AluContrl等信号;

CSR、Rs1、Rd寄存器号及值均传入下一级;

EX:

CSR、Rd寄存器号及值均传入下一级;

AluSrc1所控制的选择器增加一个连线到立即数, 用于选择Alu的一端为立即数/寄存器Rs1;

AluSrc2所控制的选择器增加两个连线, 一个为0, 一个为CSR寄存器的值, 用于ALU的另一个操作数;

ALU根据控制指令进行加/或/同或运算, 产生的结果送入下一级;

MEM:

CSR的值 (表示Rd寄存器的值), CSR寄存器号 (值为ALUOUT) 连入冒险模块用于数据相关, 控制信号由ID截断的控制模块产生并通过流水线转移;

WB:

CSR的值 (表示Rd寄存器的值), CSR寄存器号 (值为ALUOUT) 连入冒险模块用于数据相关, 控制信号由ID截断的控制模块产生并通过流水线转移;

CSR的寄存器号、ALUOUT的值连接CSR寄存器堆, 写使能有效, 修改CSR的值;

CSR寄存器的值, 写入寄存器Rd中。

5、Verilog 如何实现立即数的扩展？

立即数的原始值为扩展后的低位，高位根据立即数的符号位，若为0则全为0，若为1则全为1。

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

可分为两步来完成，以lw为例，若非字对齐，先后使用lwr指令和lwl指令读取两个部分，将小端置入寄存器的右部，大端置入左部，得到完整的数值。

7、ALU 模块中，默认 wire 变量是有符号数还是无符号数？

默认wire是无符号数

8、简述BranchE信号的作用。

BranchE信号由Branch Decision产生，用于控制NPC Generator模块更新的信号。Branch Decision通过条件判断是否跳转，若跳转，则Branch信号置1；当BranchE信号生效时，NPC Generator以BrT端口获取的PC值更新为新的PC值。

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

有优先级，存在两条连续的跳转指令，如BEQ、JAL，若BEQ指令跳转，则此时BEQ指令位于EX级，JAL位于ID级，BrE和JalE同时生效，显然应该BEQ指令跳转生效，因而存在优先级的差别。

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？

分支指令引起的PC冲突需要插入气泡，如JAL指令需要停顿一个周期，BEQ指令需要停顿两个周期等；

LW+R-type指令存在数据相关，需要stall一个周期；

LW+LW指令会存在数据相关，需要stall一个周期；

LW+SW指令会存在数据相关，需要stall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？

由于采用静态分支预测器，所以Branch指令执行过程中，后续的两条指令继续进入流水线，当预测成功时，无须改变任何信号，当预测失败时，需要使这两条指令失效，故控制EX和MEM寄存器的flush和stall信号使得其上全部信号失效，即stall两个周期。

12、0 号寄存器值始终为 0，是否会对 forward 的处理产生影响？

会产生影响；如果ADD指令的Rd寄存器为寄存器0，后一条指令用到寄存器0，则Harzard模块判断存在数据相关，将ADD指令的计算结果转发，而实际上寄存器0的值始终为0，从而导致了结果存在错误。