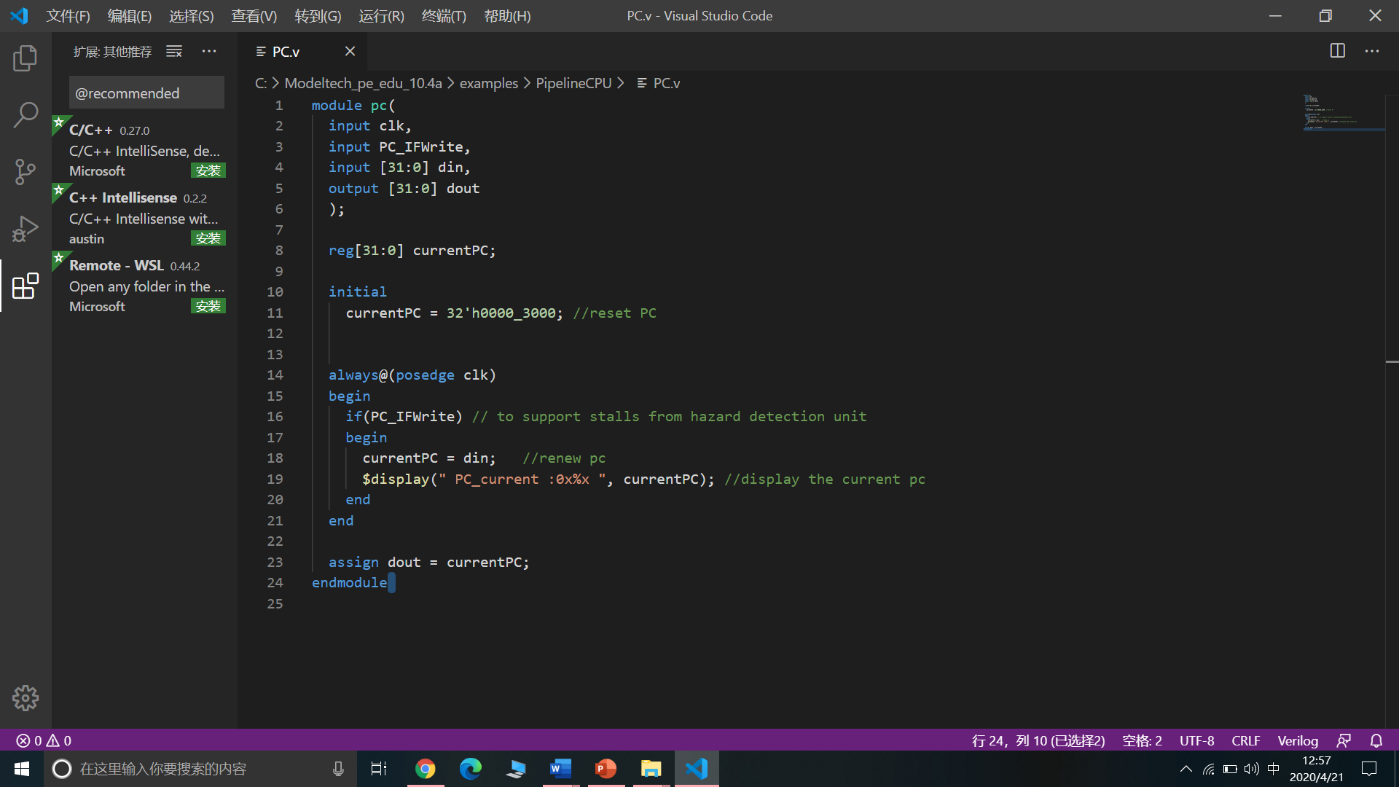
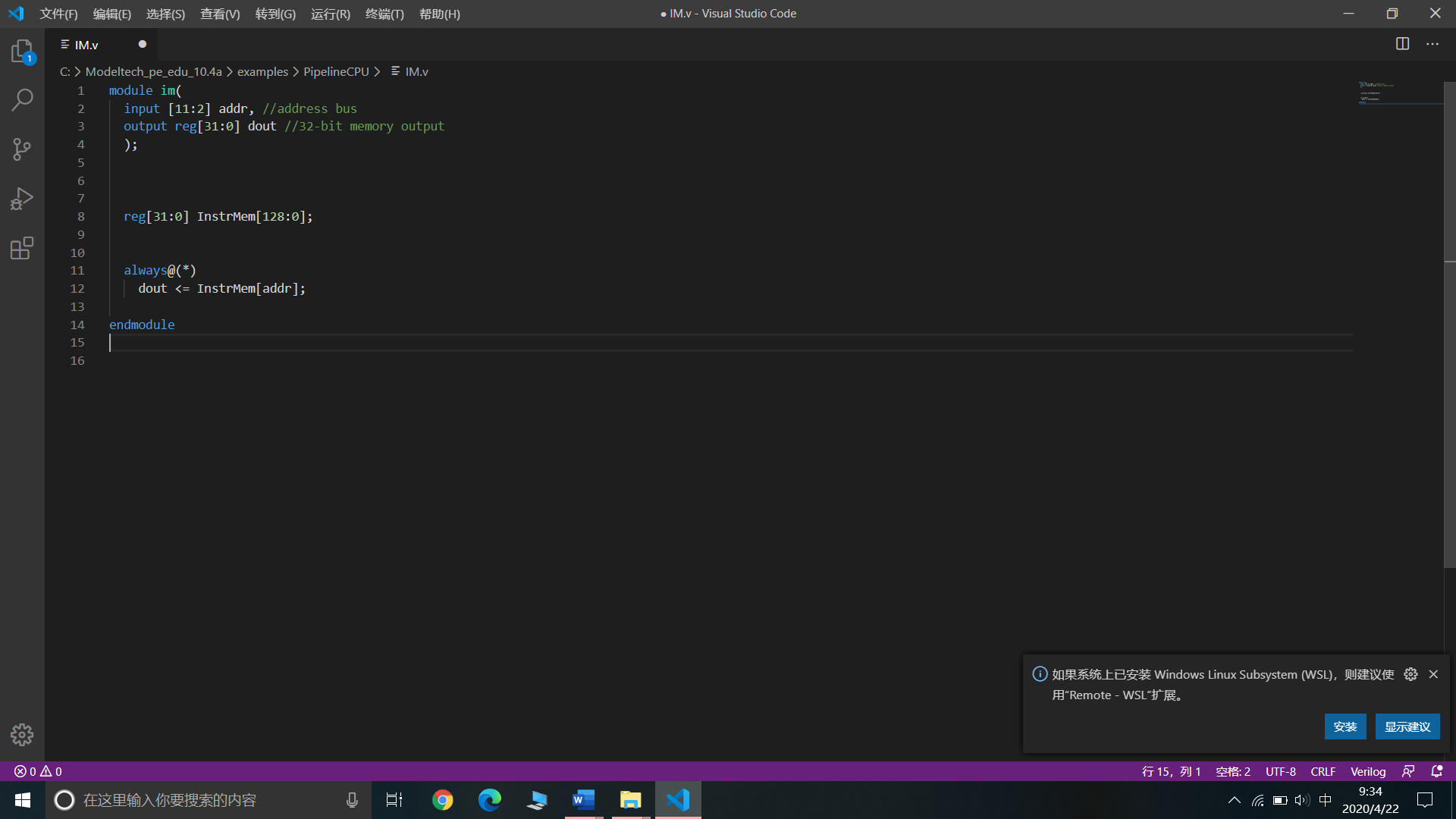
**4.3 功能部件详细设计--- 流水线**

4.3.1 **PC（程序计数器）**

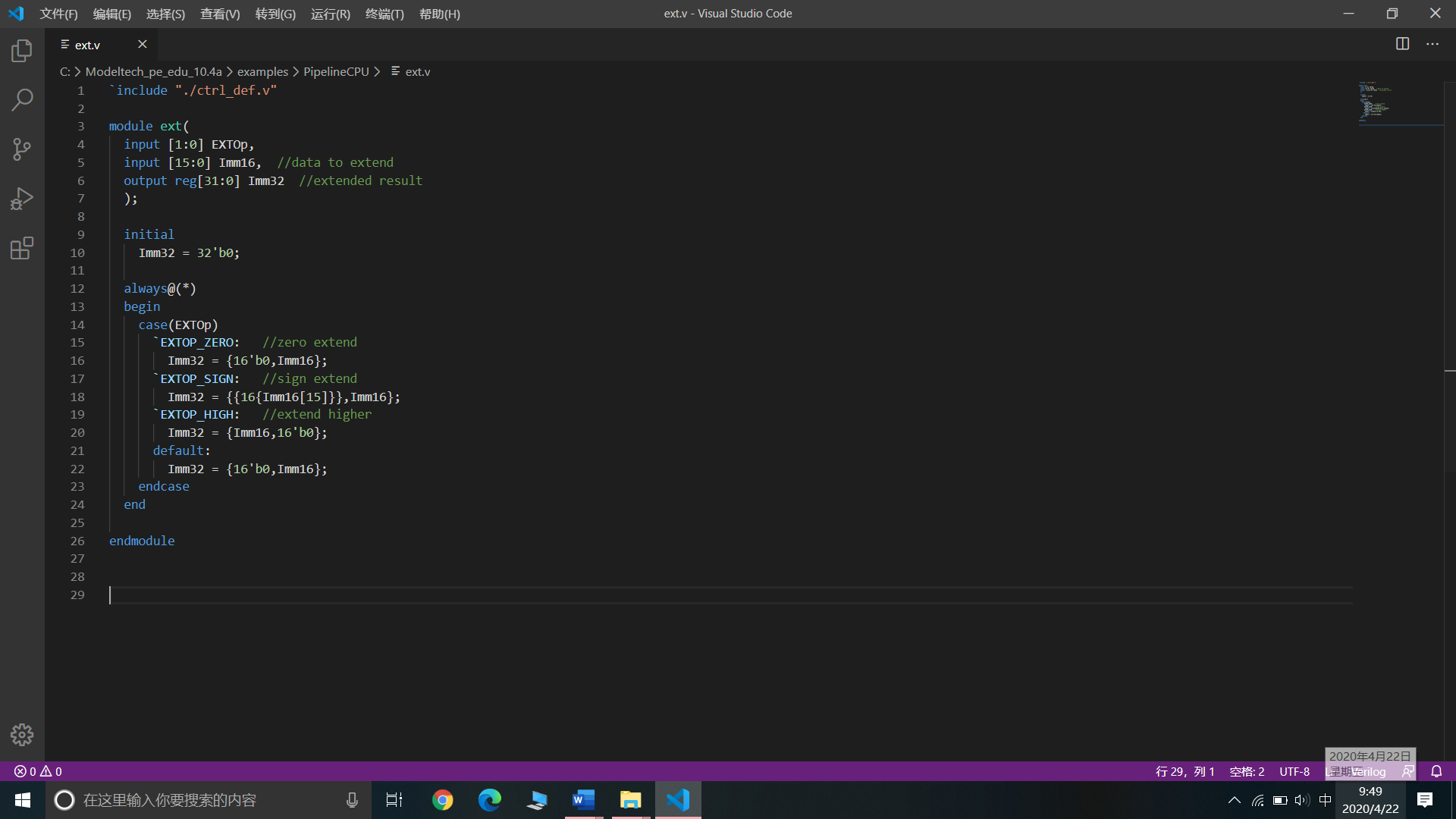
在clk上升沿时更新PC，初始时复位位32’h0000\_3000,为了与mars仿真时指令的起始地址相匹配。当冒险检测单元未检测到冒险时，产生PC\_IFWrite==1,更新PC。

4.3.2 IM （指令存储器）



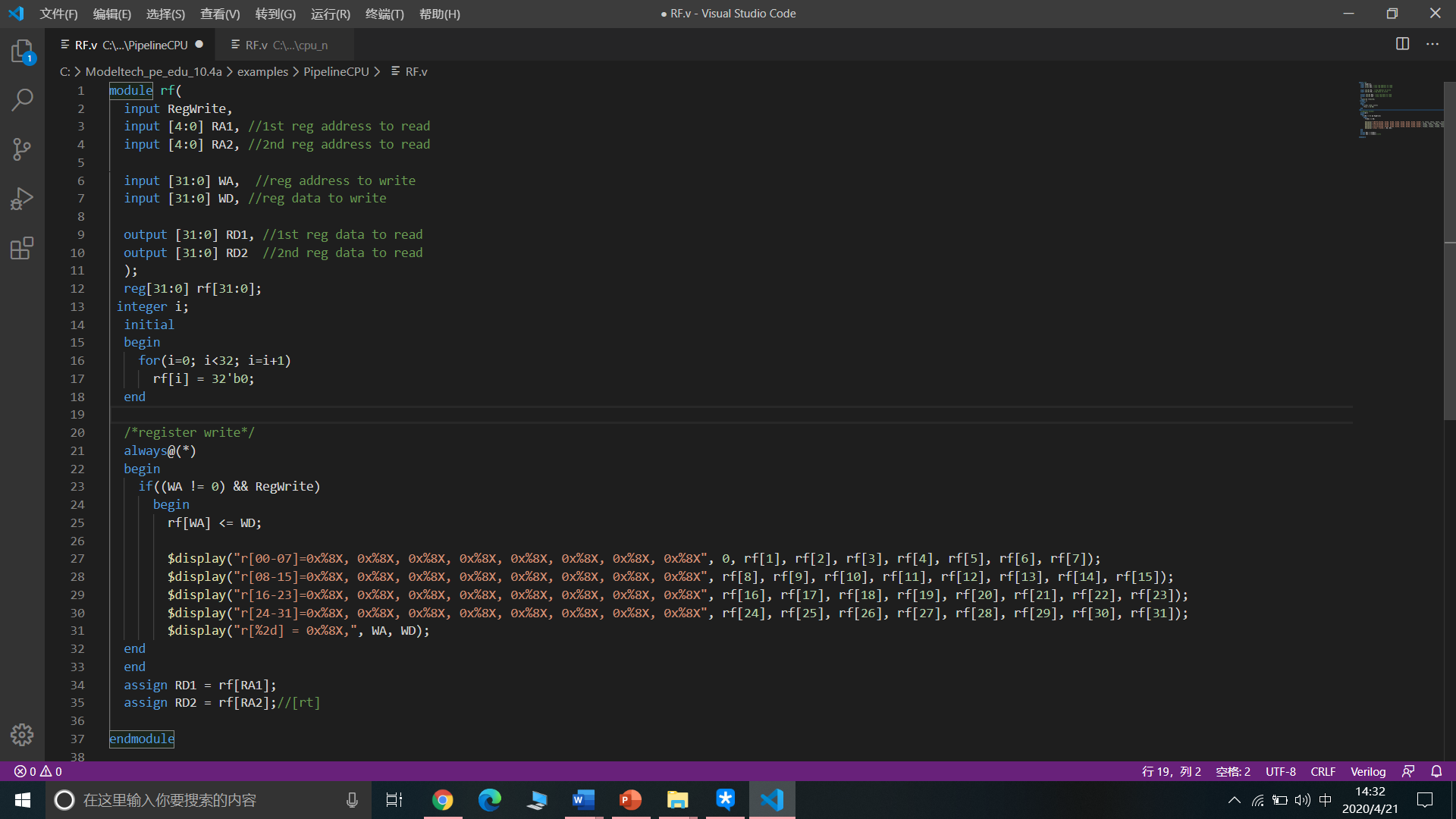
IM负责从指令文件中读取对应的指令。其中每个 PC 地址对应一条机器指令，每条指令长度为 32bit。在 PC变化时，立即读取指令。

4.3.2 EXT



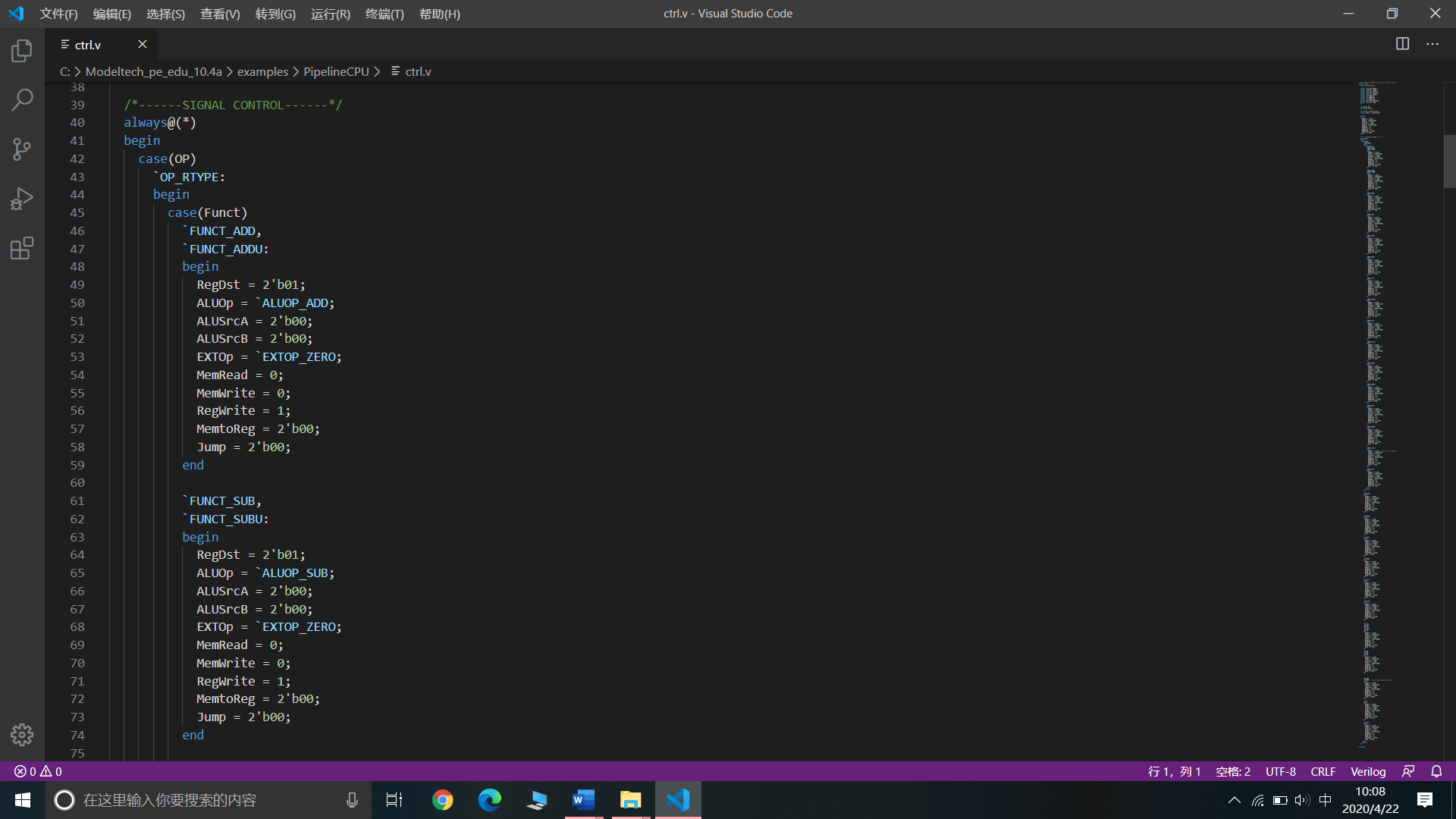
将16位立即数，按照操作码的不同，进行0扩展，符号扩展，或者高位扩展，以便后续进行相应的运算。

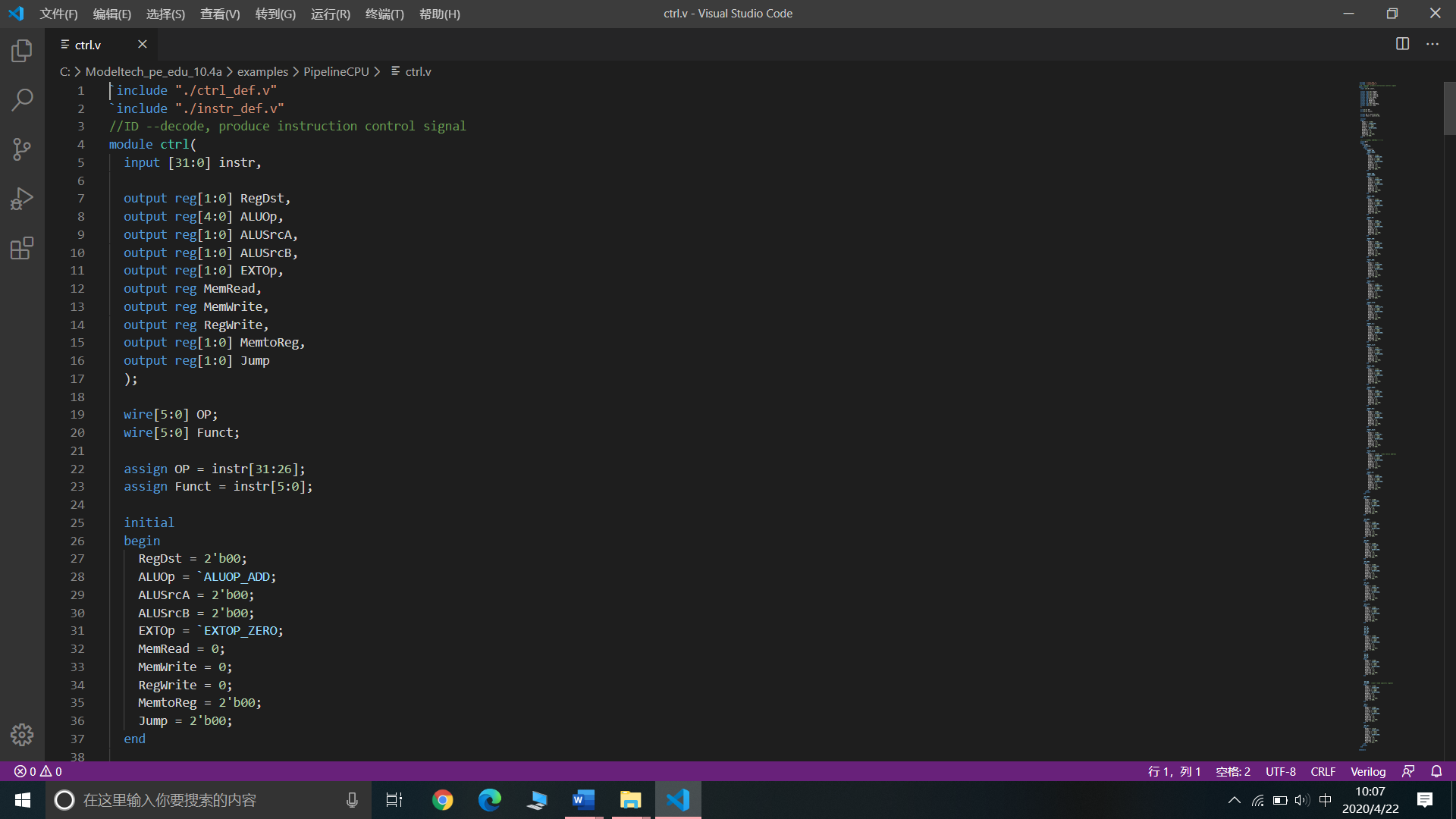
**4.3.4 RF （通用寄存器组）**

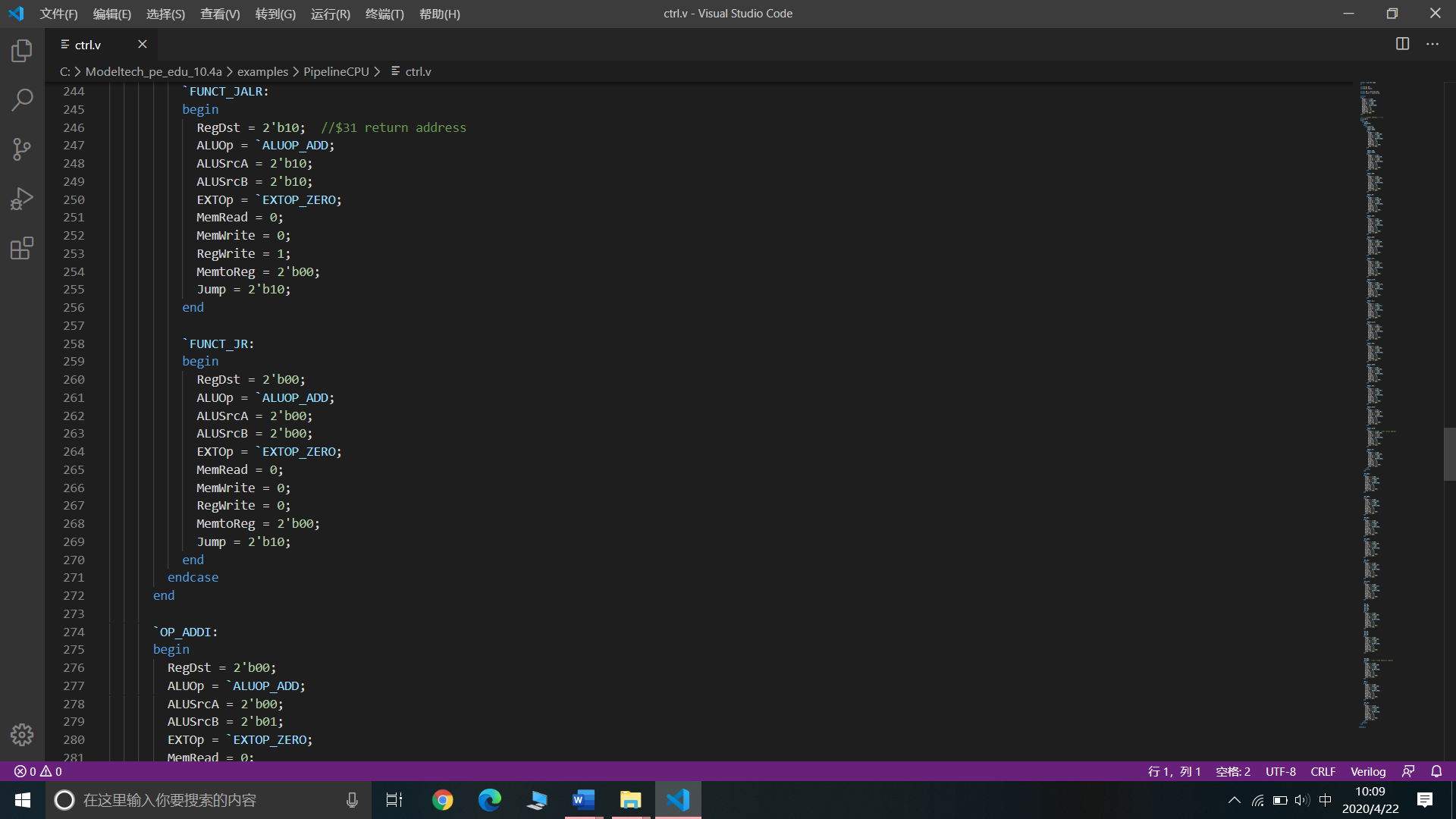
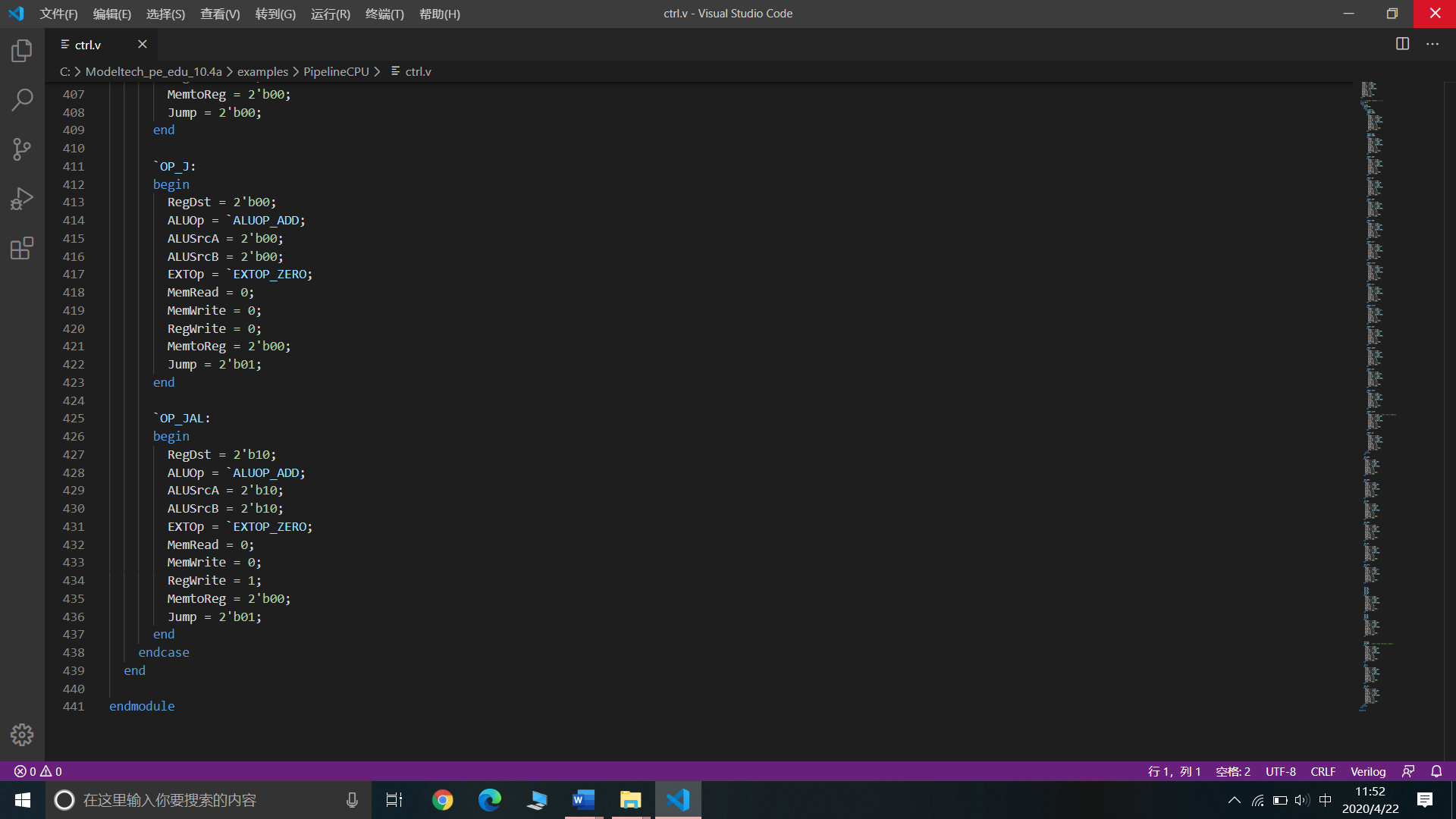
****

通用寄存器组在clk在上升沿时触发，判断写信号RFWr是否为1和写寄存器地址是否为0，是则将输入的数据WD保存到写入地址对应的寄存器，将读端口读到的数据输出到相应端口中。

4.3.5 Ctrl (信号控制单元)

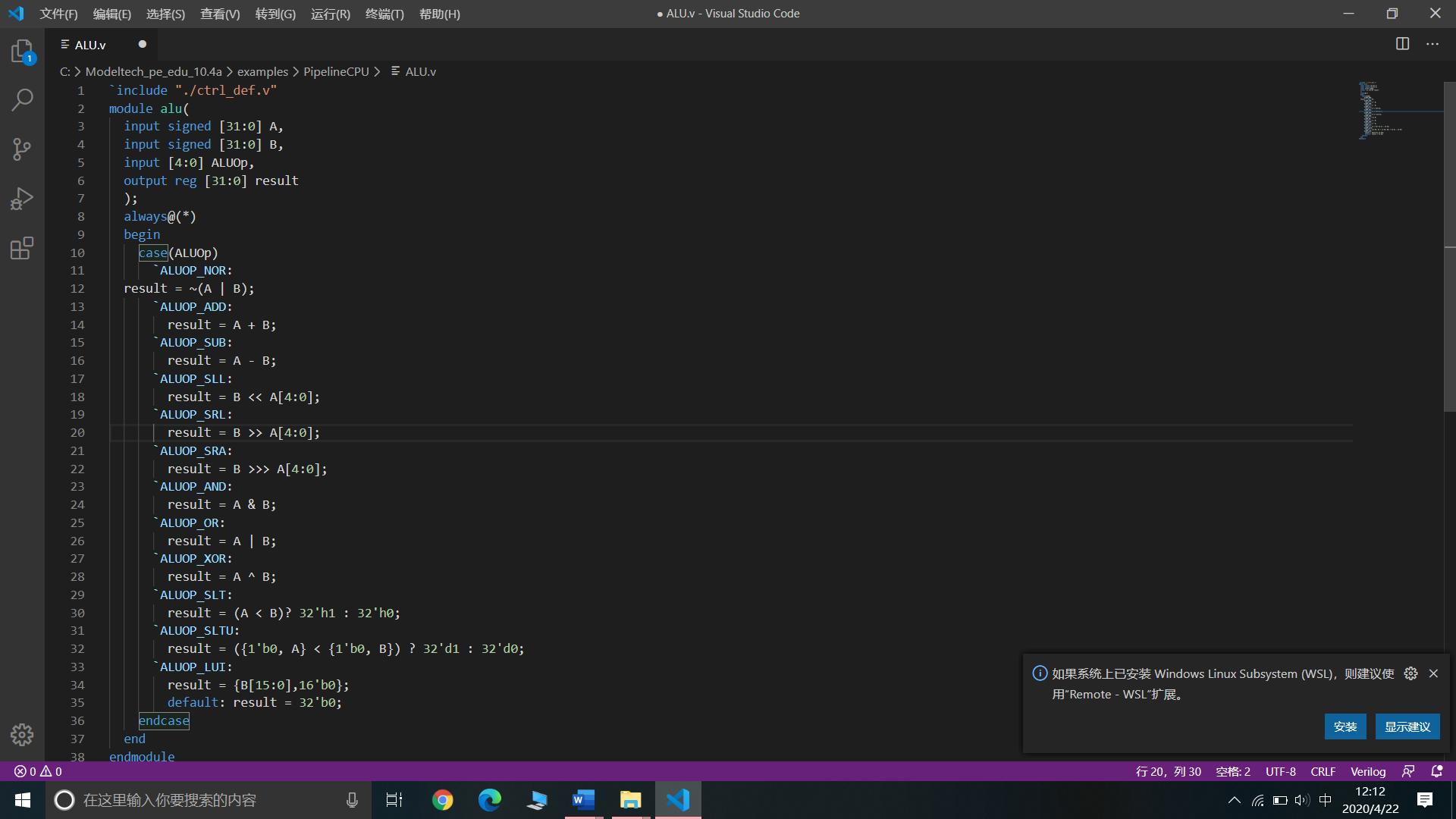
部分代码



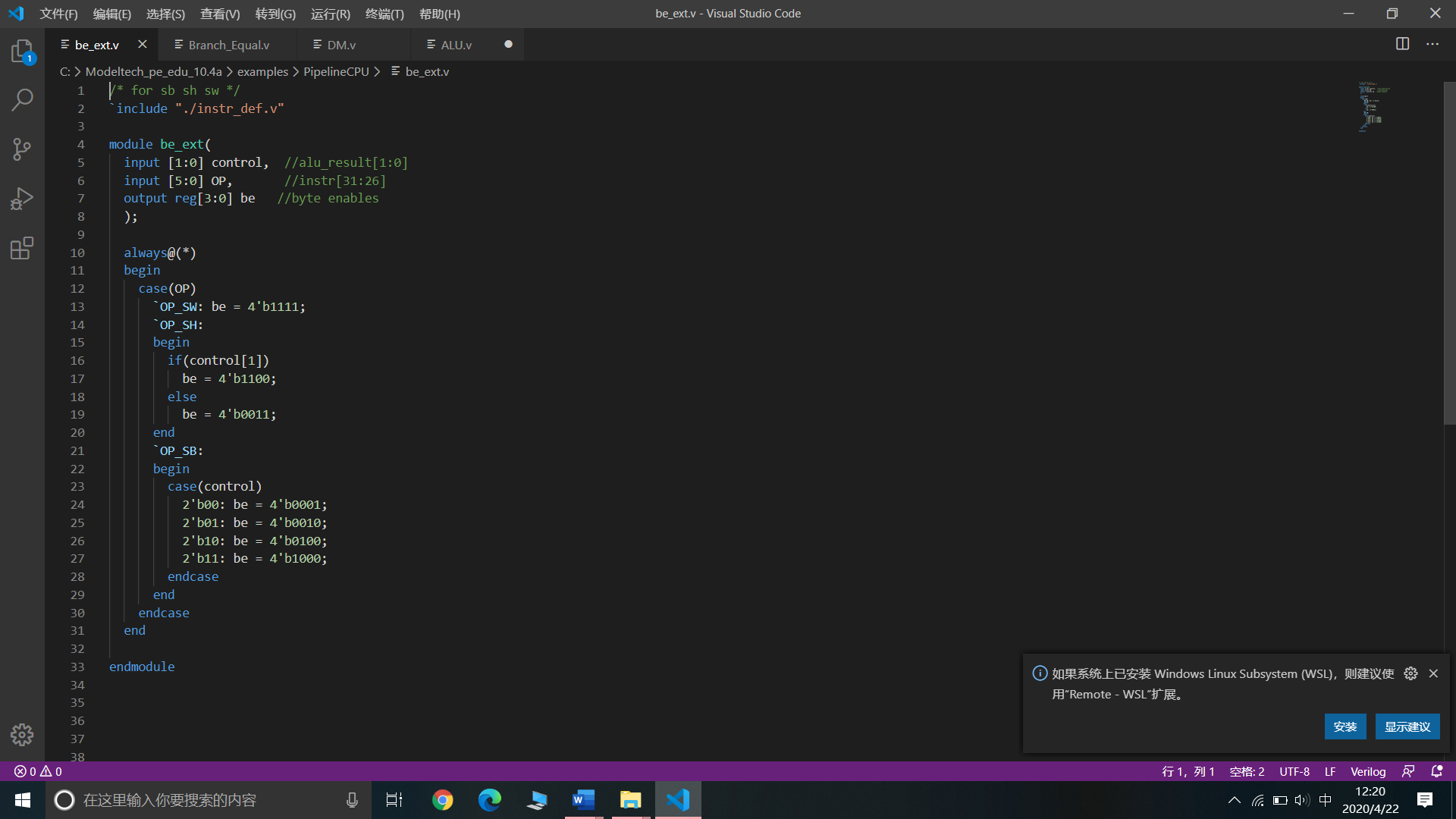


根据指令的Opcode 或funct字段，为相应指令的执行给出对应的信号。特别注意跳转指令是否要写寄存器，如jr指令和jalr指令，选择对应的寄存器写入地址的信号。

4.3.6 ALU （运算器）

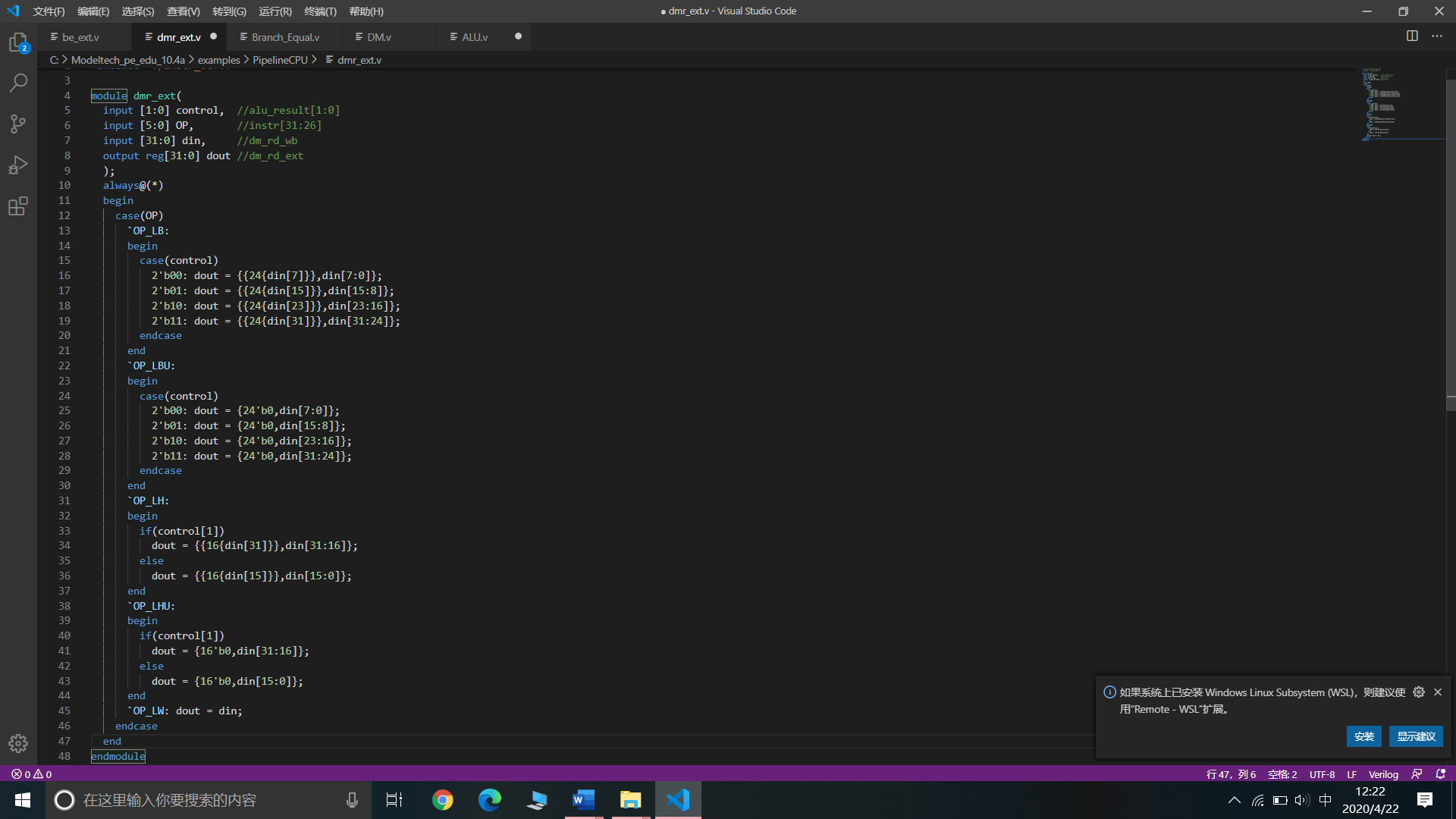


依据Ctrl单元译码判断出的指令类型，进行指令中操作数的运算。

4.2.7 Be\_ext (存数扩展控制单元)

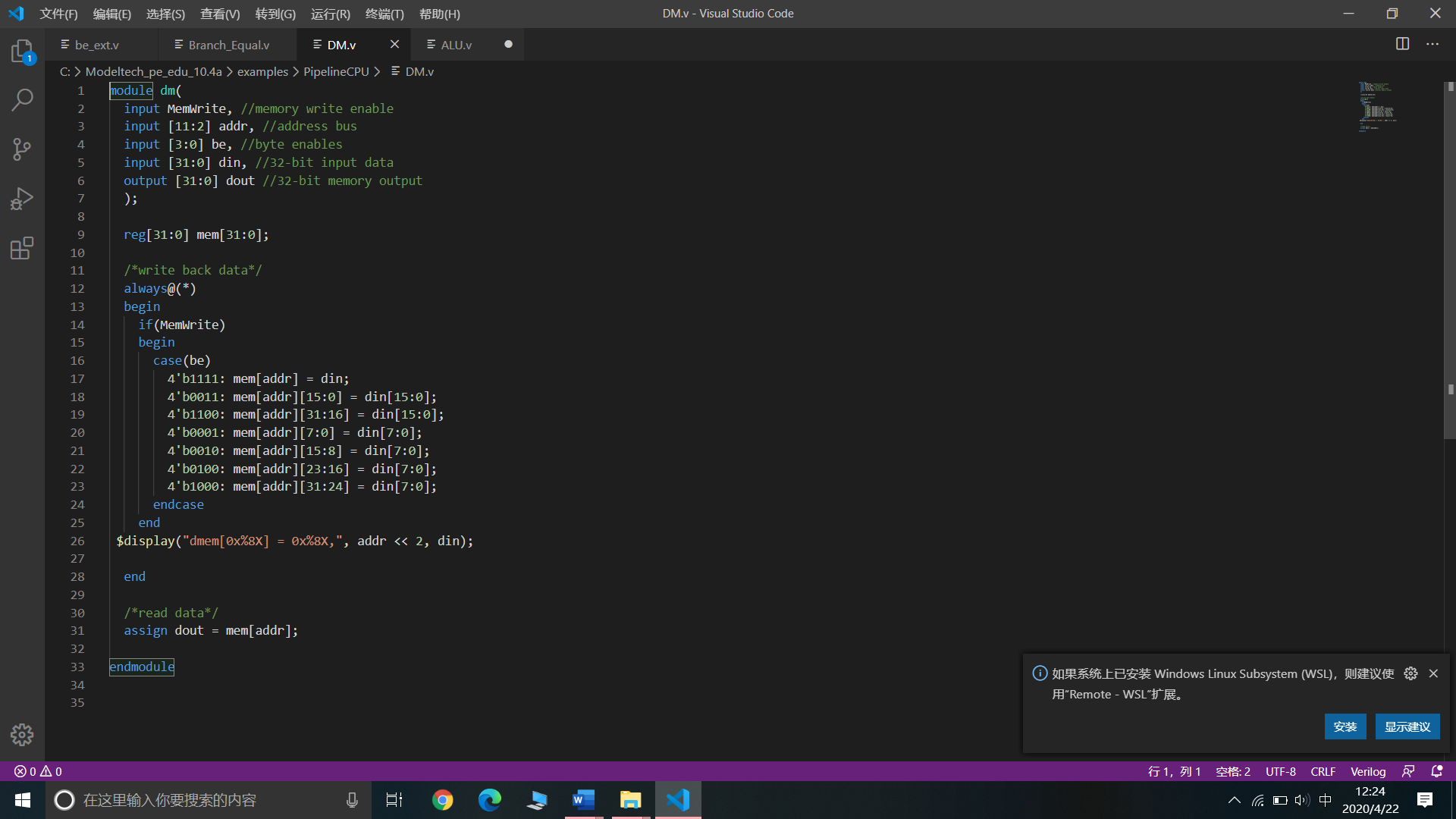
依据Alu运算结果的后两位和指令的Opcode,判断出是SW,SH,SB的哪一种，生成控制信号。

4.2.8 Dmr\_ext（取数扩展控制单元）



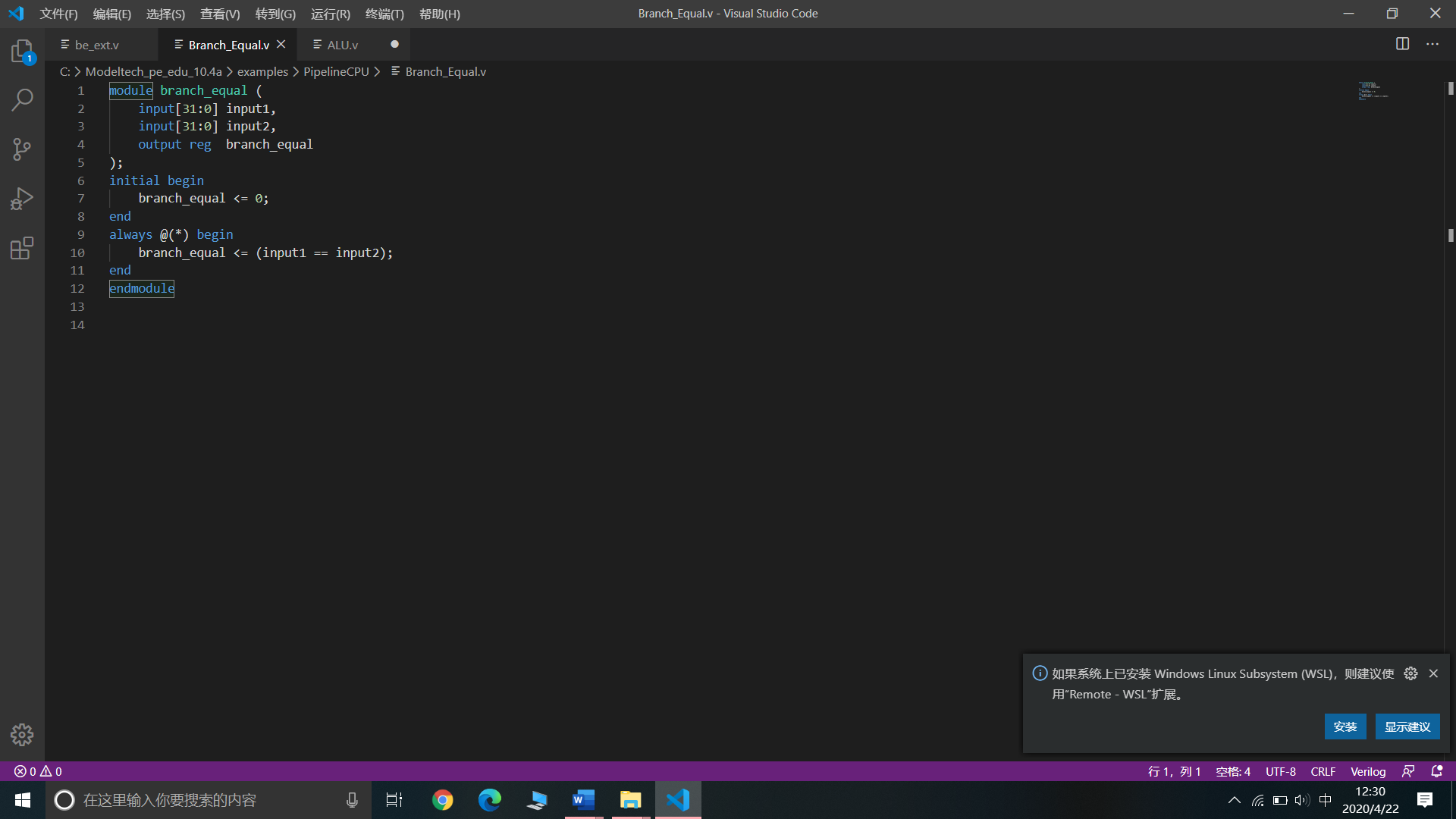
根据Alu运算结果的后两位和指令的Opcode,判断出是LW,LB,LBU,LH,LHU指令的哪一种，进行相应的数据扩展

4.2.9 DM（数据存储器）



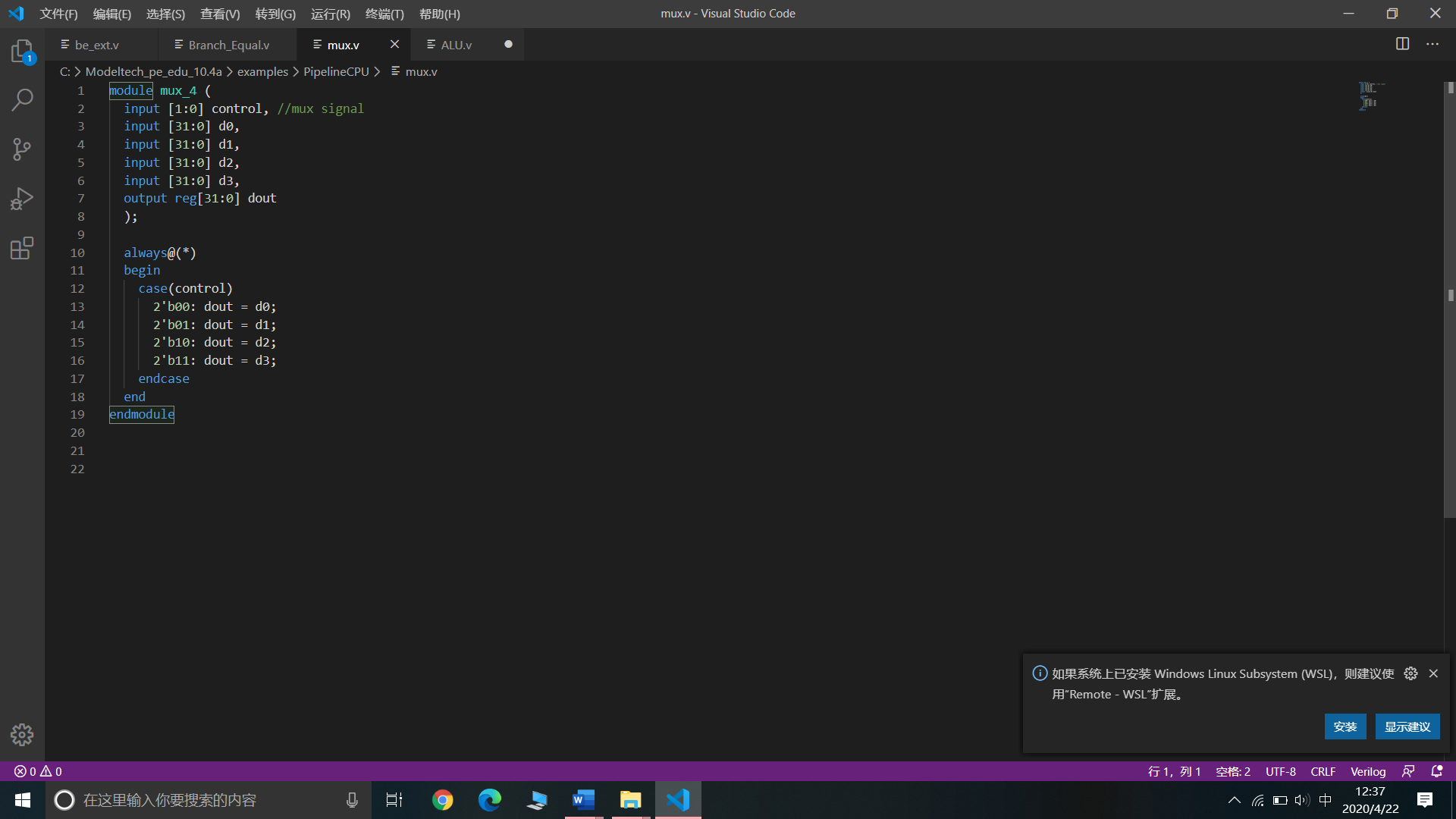
DM写入和读取的端口是共用的。只有在 clk 为 posedge 并且 MemWrite 为 1 时才允许写入。特别要关注sb,sh指令存放的位置。但是在任何时候都可以读出。

4.2.10 Branch\_Equal (判断分支执行条件单元)



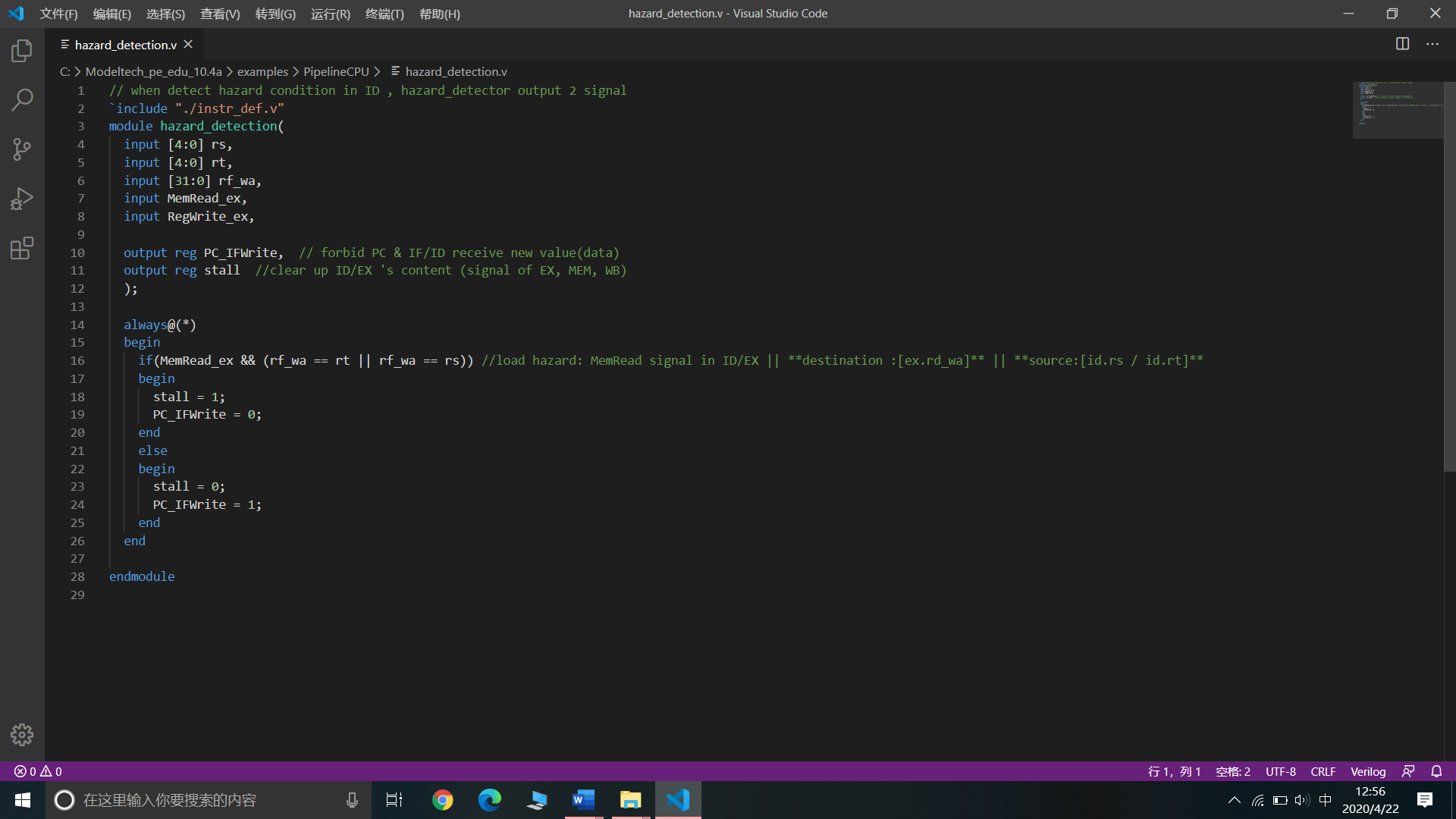
比较两个输入数据是否相等，如指令中存在数据相关，结合旁路（主要配合转发单元C/D使用），在ID阶段判断Beq,Bne跳转条件是否成立。

4.2.11 MUX（多路选择器）



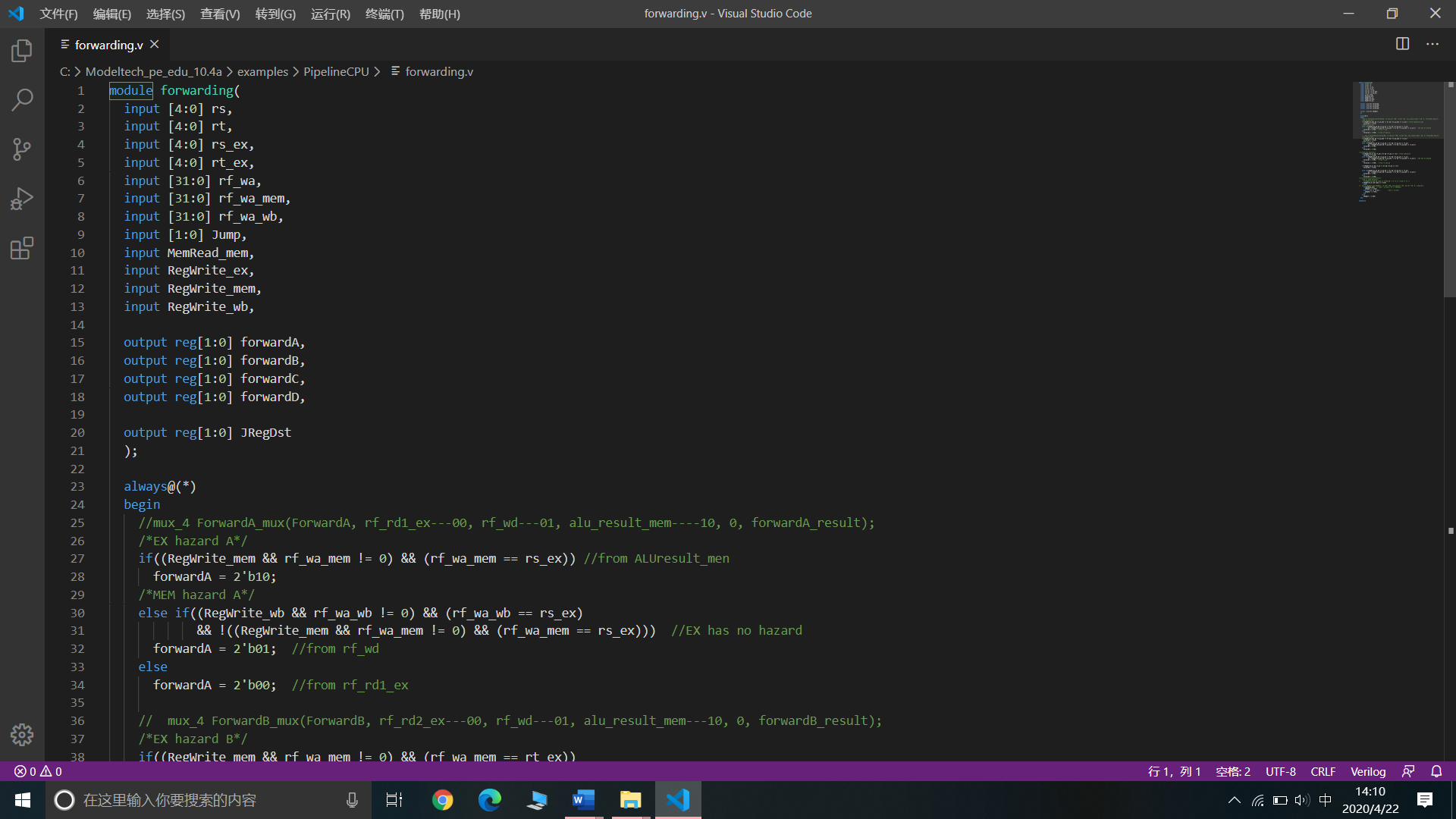
根据输入数据的位长和输入数据的个数，使用四选一多路选择器，当可选项不足四个时，直接用0代替，为了简洁，流水线CPU未设计二选一选择器，所以设计到选择的信号必须为2位。

4.2.12冒险判断单元（Hazard）



冒险判断单元会针对当前的跳转、分支控制信号，寄存器写入，在 IF 的 Rs 和 Rt,ID的Rt进行冒险判断。

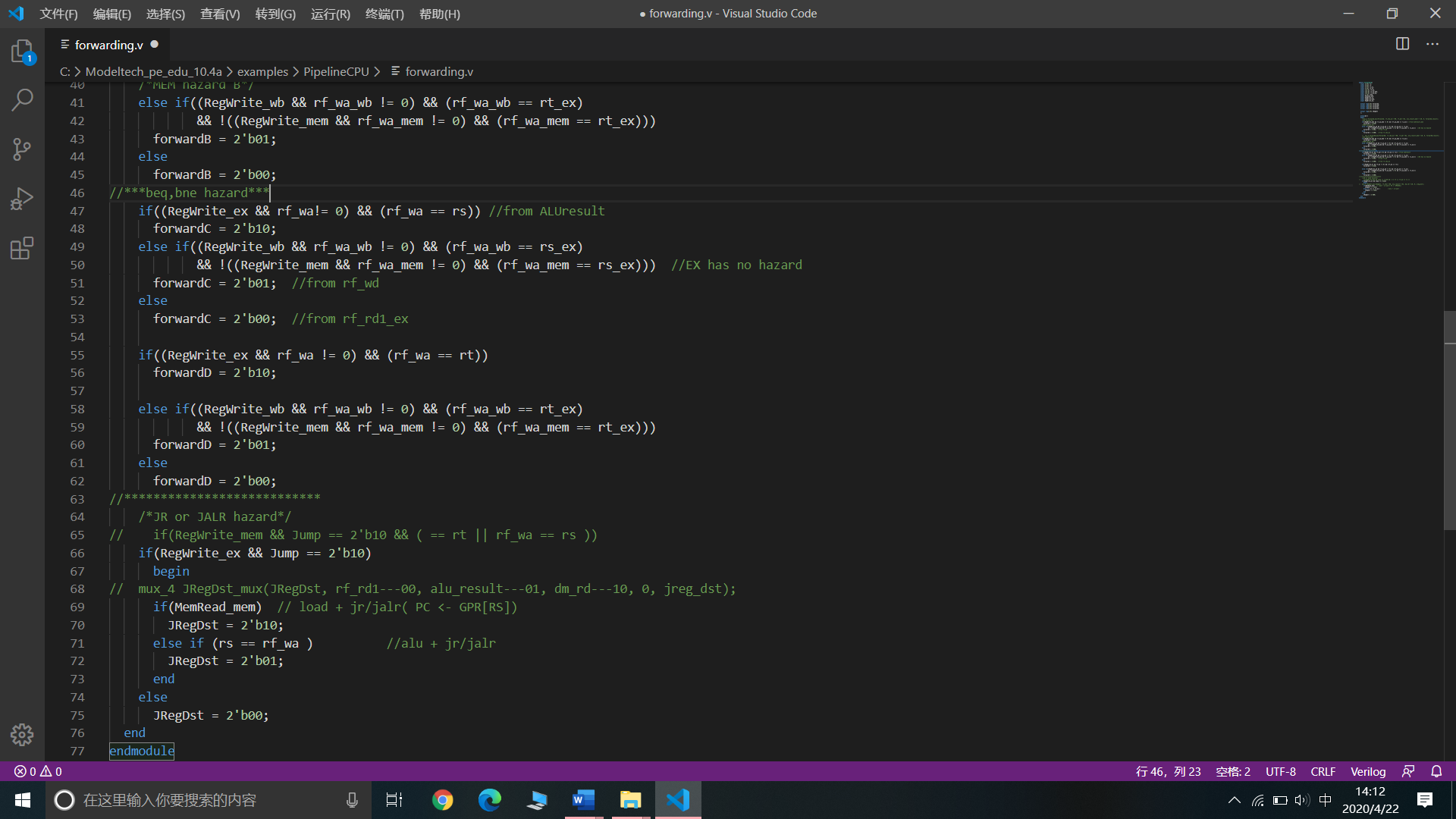
4.2.13 forwarding（旁路单元）



转发决定单元分别针对 ABCD 四个单元进行设置。负责在寄存器的对应数值 没有更新的时候正确取出需要的值。

ForwardA 的取值取决于后续使用的寄存器端口与ID 级 Rs是否存在冲突。

ForwardB 则取决于寄存器端口与ID 级 Rt 是否存在冲突。

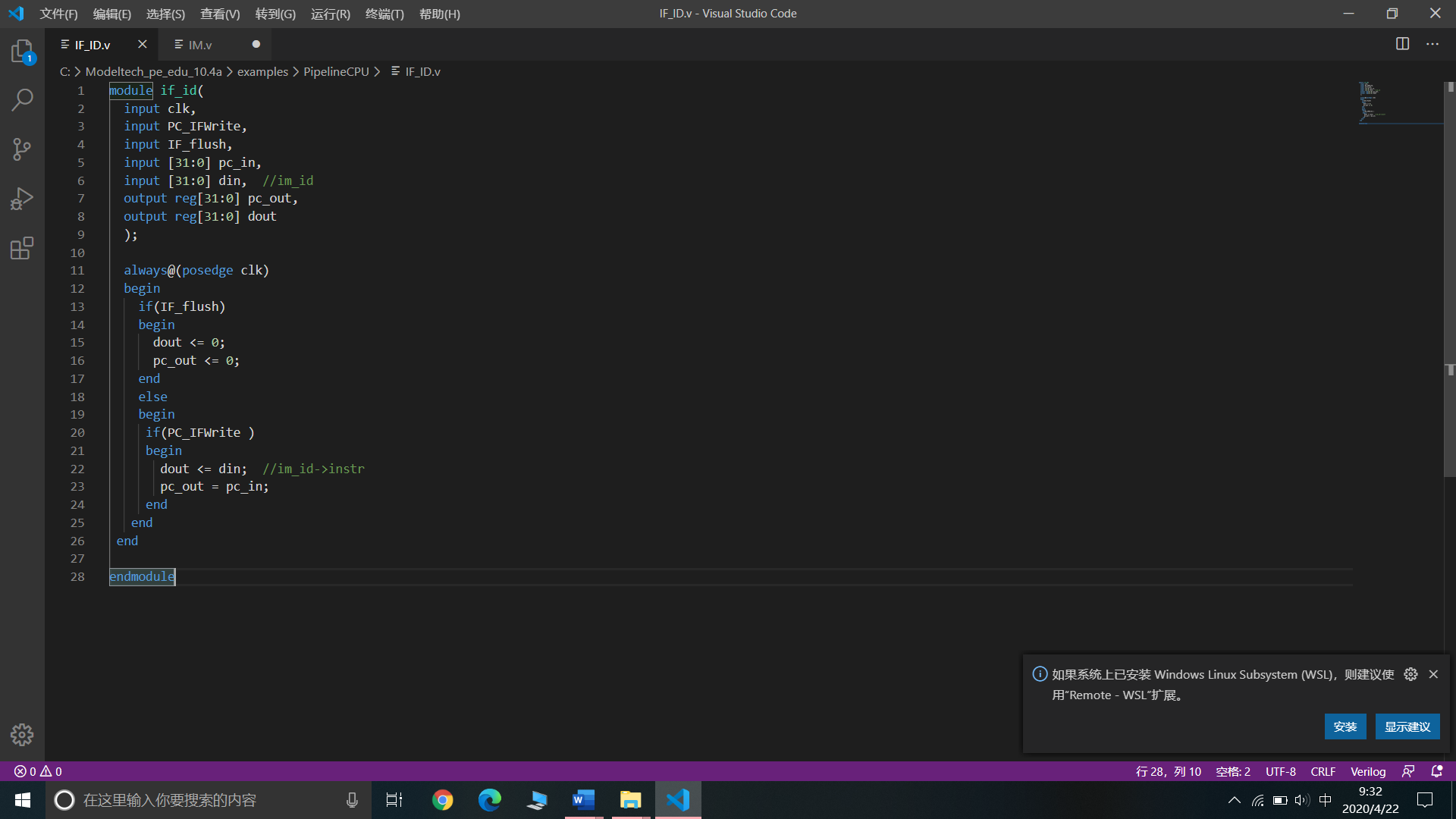


ForwardC 的取值取决于读取寄存器端口1是否和后续写入端口冲突，

ForwardD 则是读取寄存器端口2。

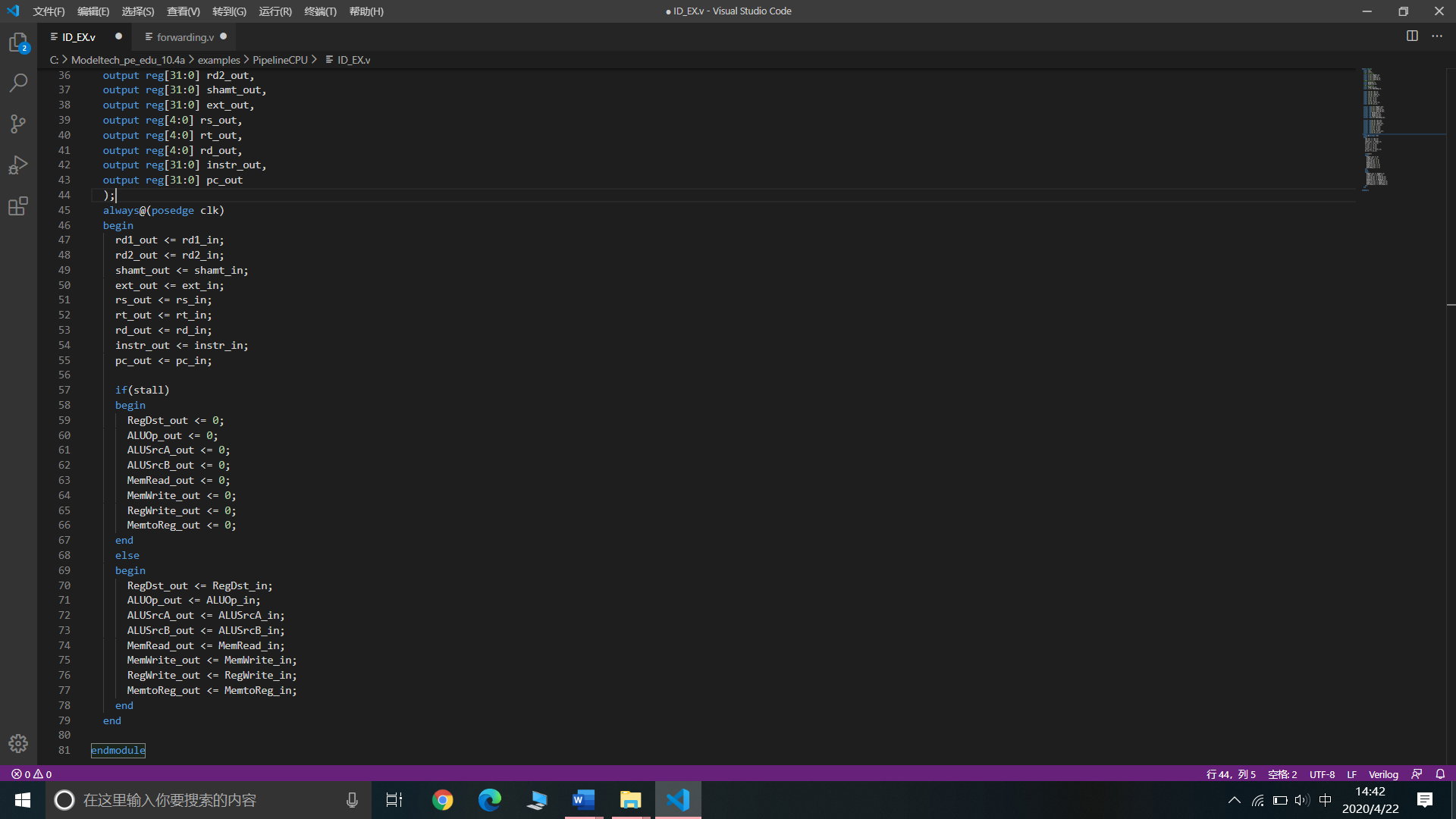
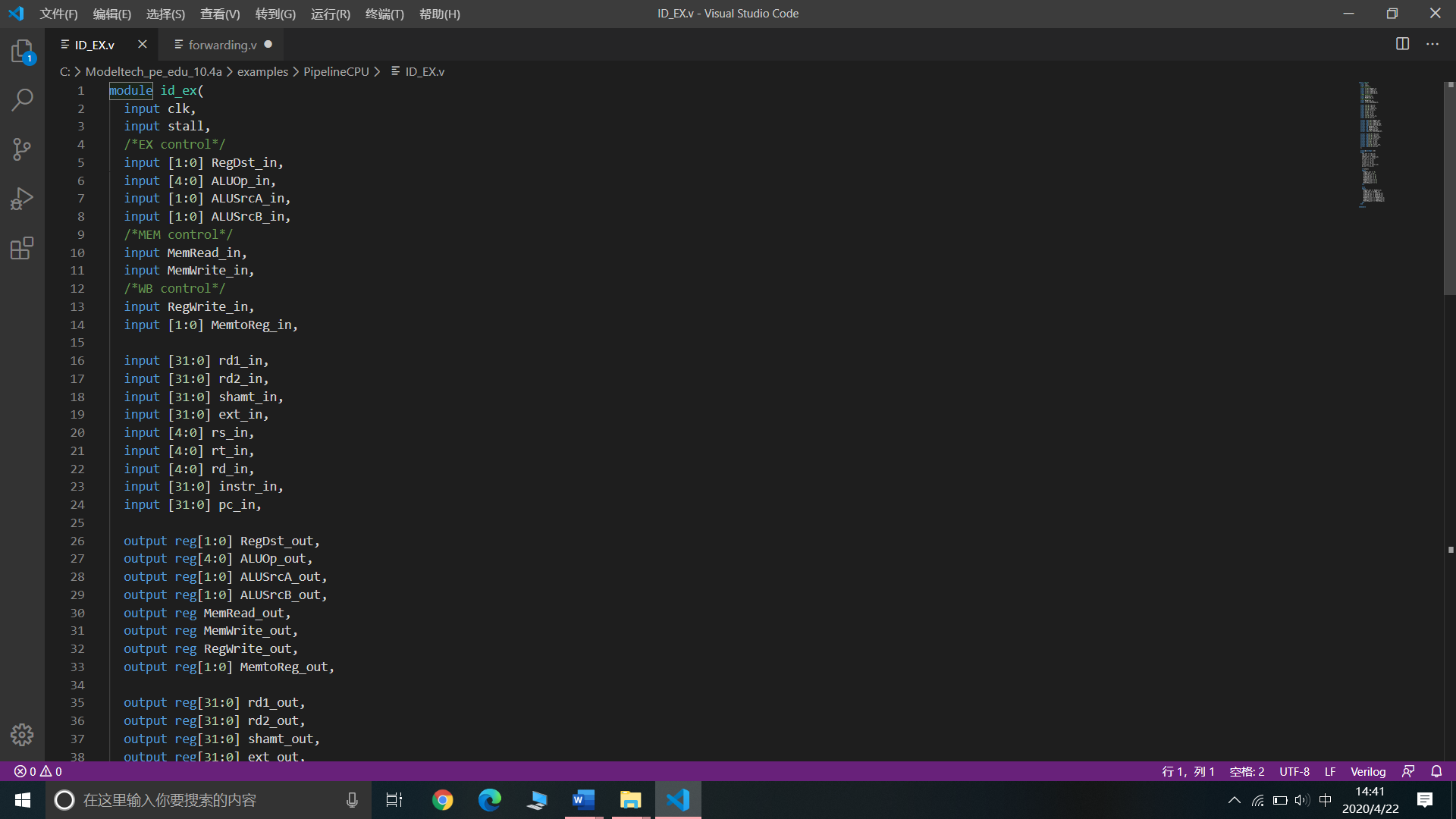
JR和JALR指令的数据相关设计到两个地方，一是这条跳转指令要读的寄存器是上一条运算指令的执行结果要存放的寄存器，而该结果还未写回寄存器；二是上一条指令是取数指令，还结果还未写回寄存器，编写代码时要结合波形图出现的值进行相应的检测条件设置。

4.3.14 IF\_ID（IF/ID级流水线寄存器）



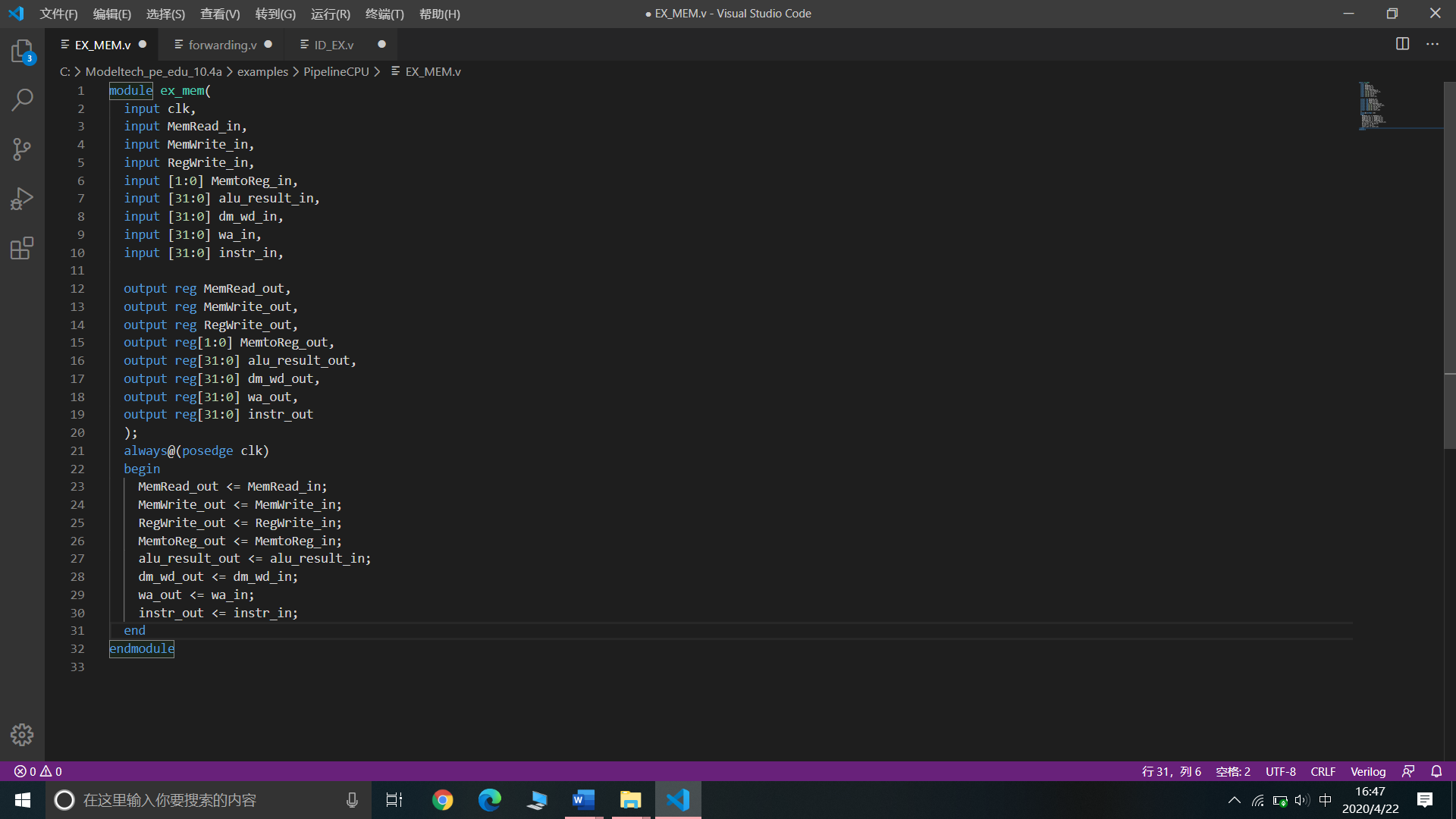
IF/ID 流水线寄存器负责将来自 IF 级的数据传递到 ID 级。需要 clk 才能进行传递。同时由于分支预测总发生，需要在预测错误时时清空此寄存器的内容以消除误读指令的影响。PC\_IFWrite为0时，禁止接受新的指令和PC，实现暂停。

4.3.15 ID\_EX（ID/EX级流水线寄存器）



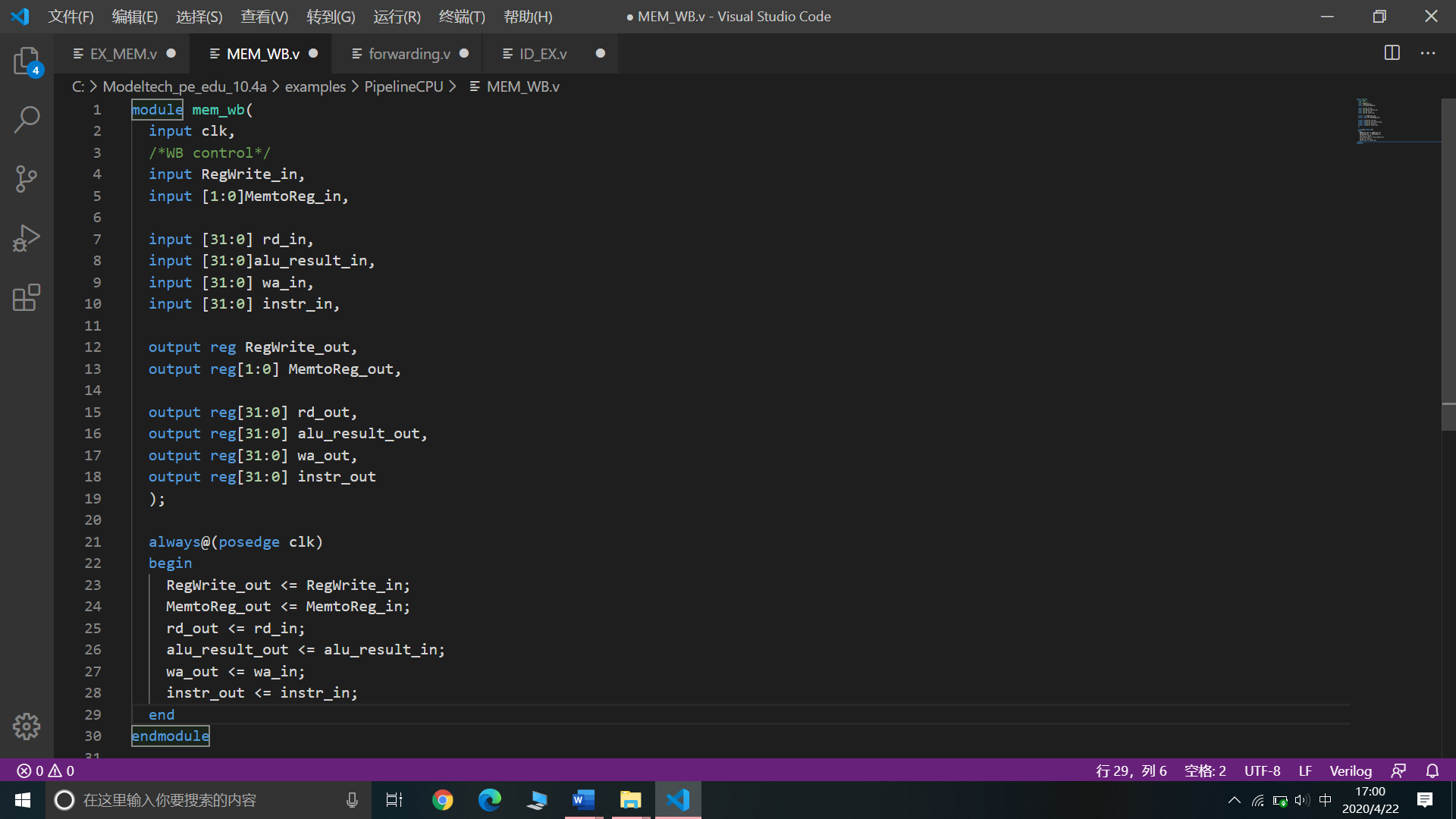
ID/EX 流水线寄存器，负责将来自 ID 级的数据传递到 EX 级，需要 clk 才能进行传递。如果stall信号值为1，则将ID级传递的数据清空，不再往下执行该指令。

4.3.16 EX\_MEM （EX/MEM级流水线寄存器）



EX/ME 流水线寄存器负责将来自 EX 级的数据传递到 ME 级。需要 clk 才能进行传递。

4.3.17 MEM\_WB （MEM/WB级流水线寄存器）



ME/WB 流水线寄存器负责将来自 ME 级的数据 传递到 WB 级。需要 clk 才能进行传递。