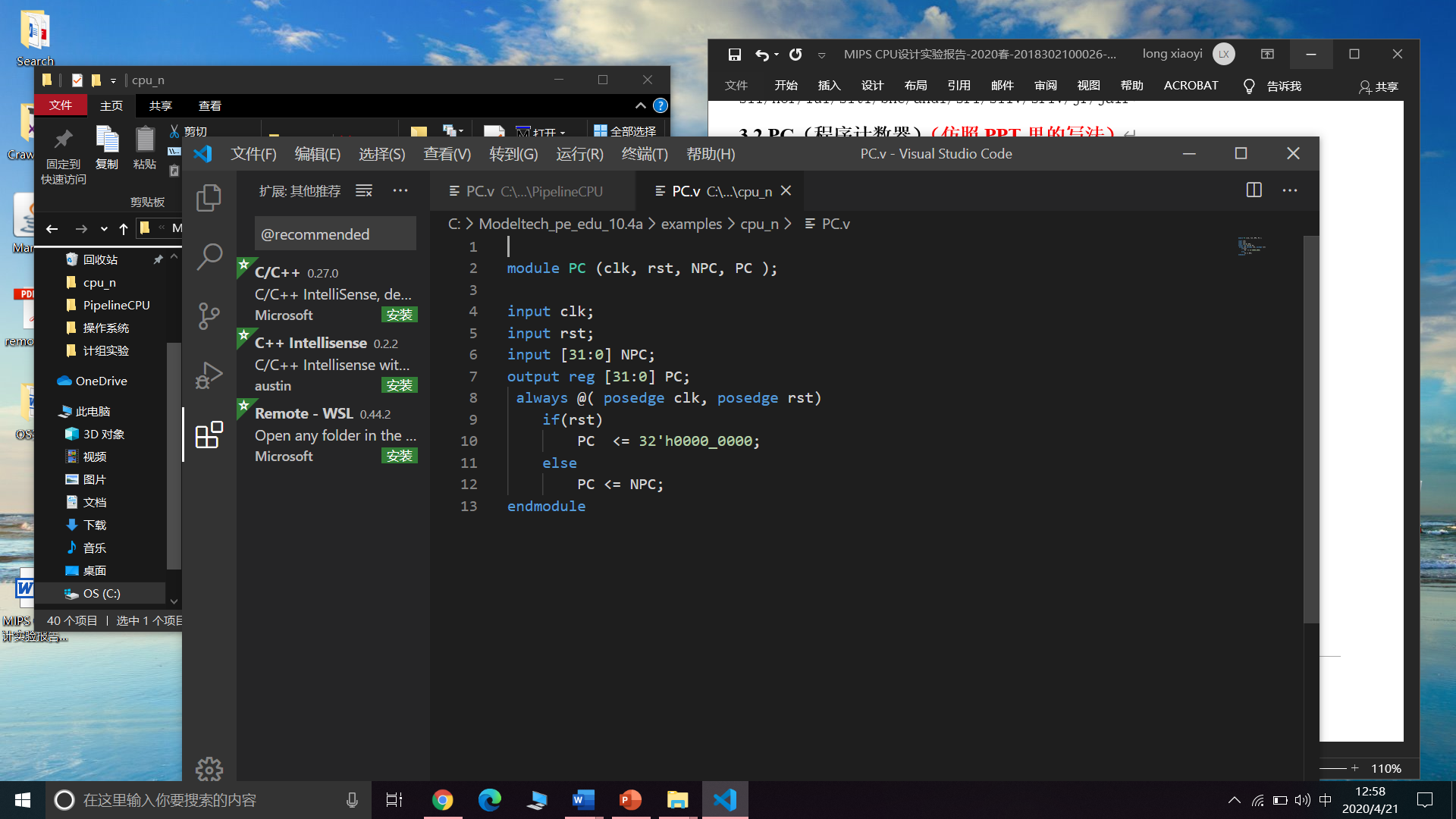
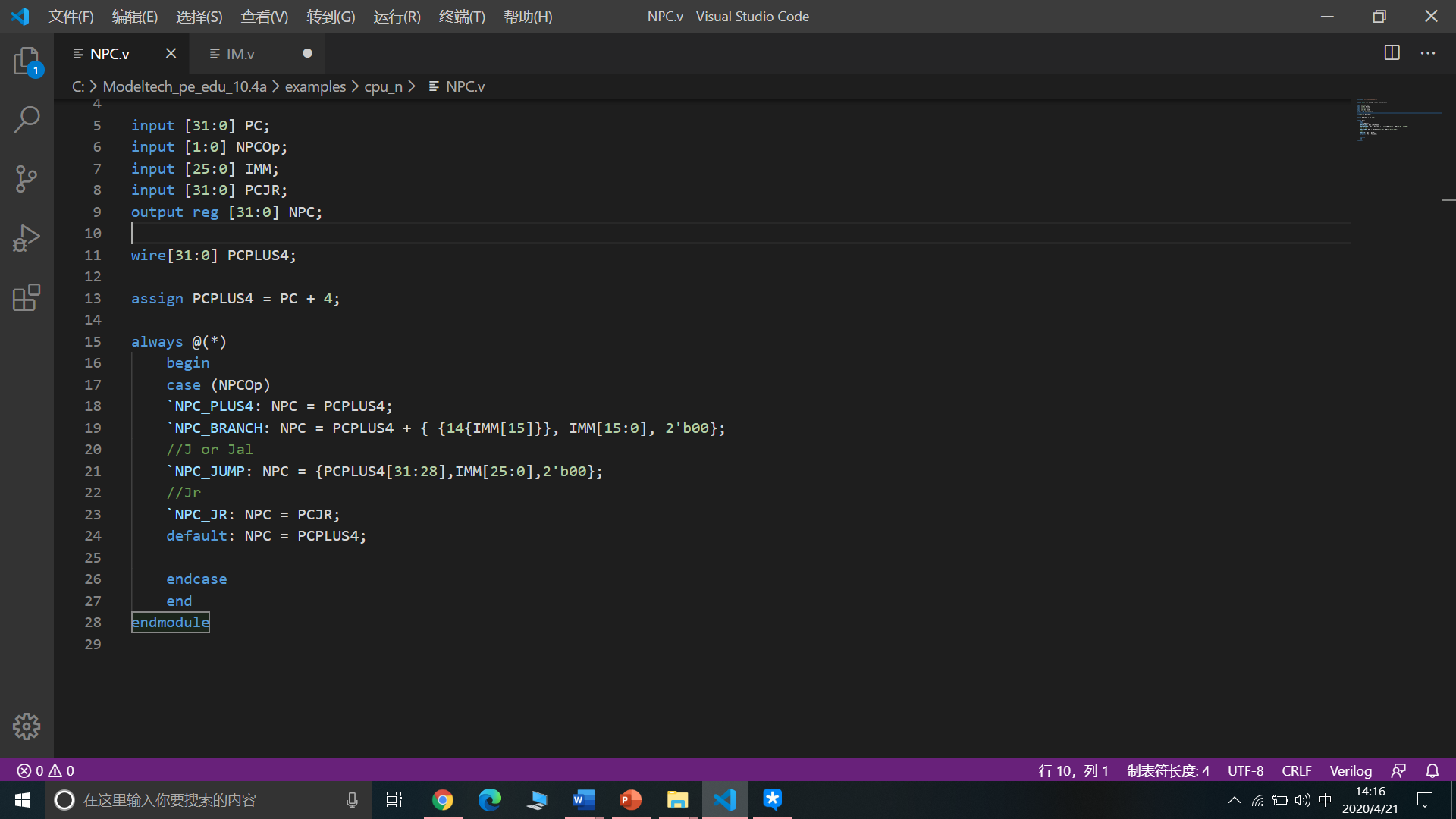
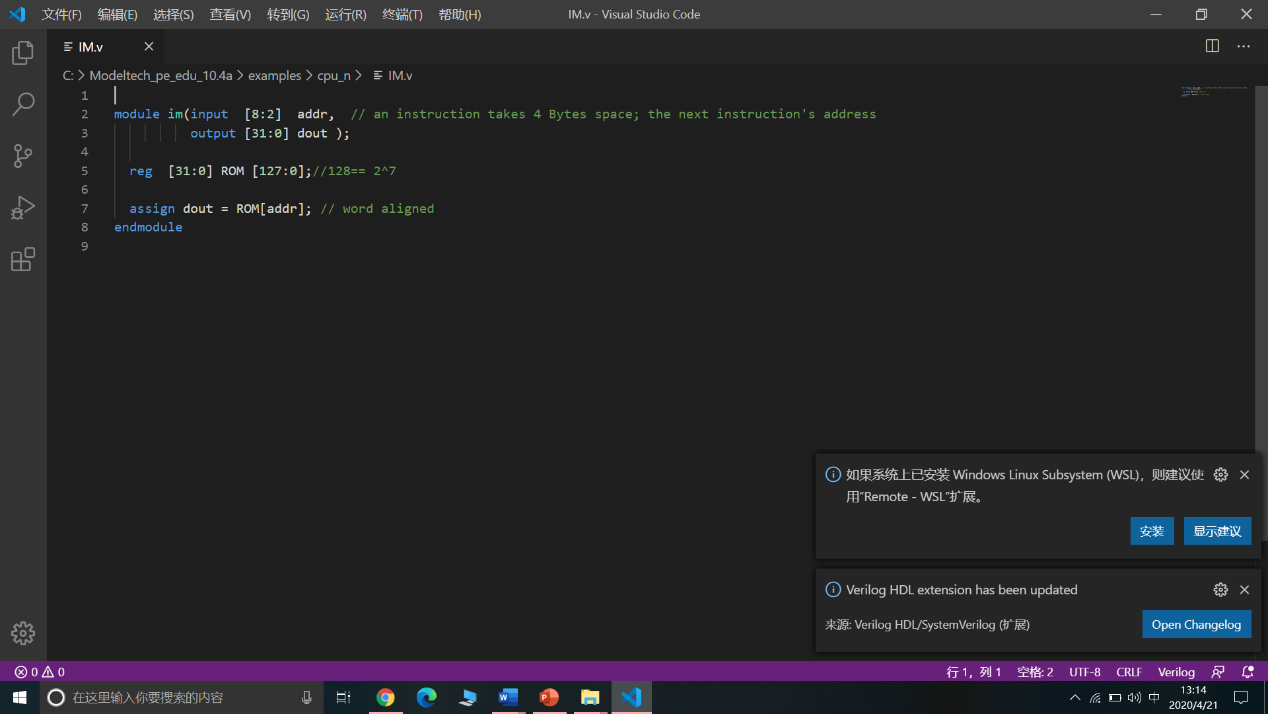
**4.2 功能部件详细设计---单周期（放代码和描述说明）**

**4.2.1 PC（程序计数器）**

当时钟上升沿到来时，若rst为0，写入下一条指令的PC；若rst值为1，则PC值清零。

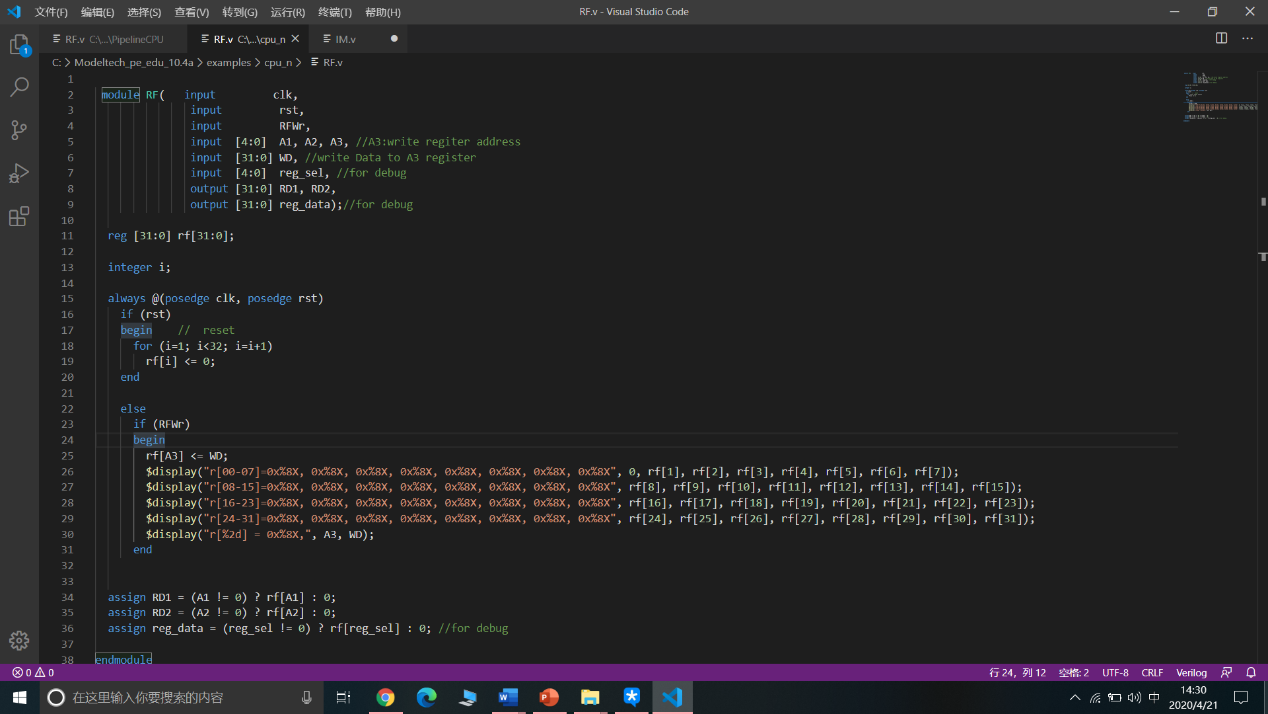
**4.2.2 NPC**

根据NPC的操作码Op，选择下一条指令地址的计算方式，有pc+4,分支指令的分支地址，跳转指令的跳转地址：j or jal指令的立即数字段，jr or jalr指令的寄存器rs的值作为跳转地址。

**4.2.3 IM（指令存储器）**

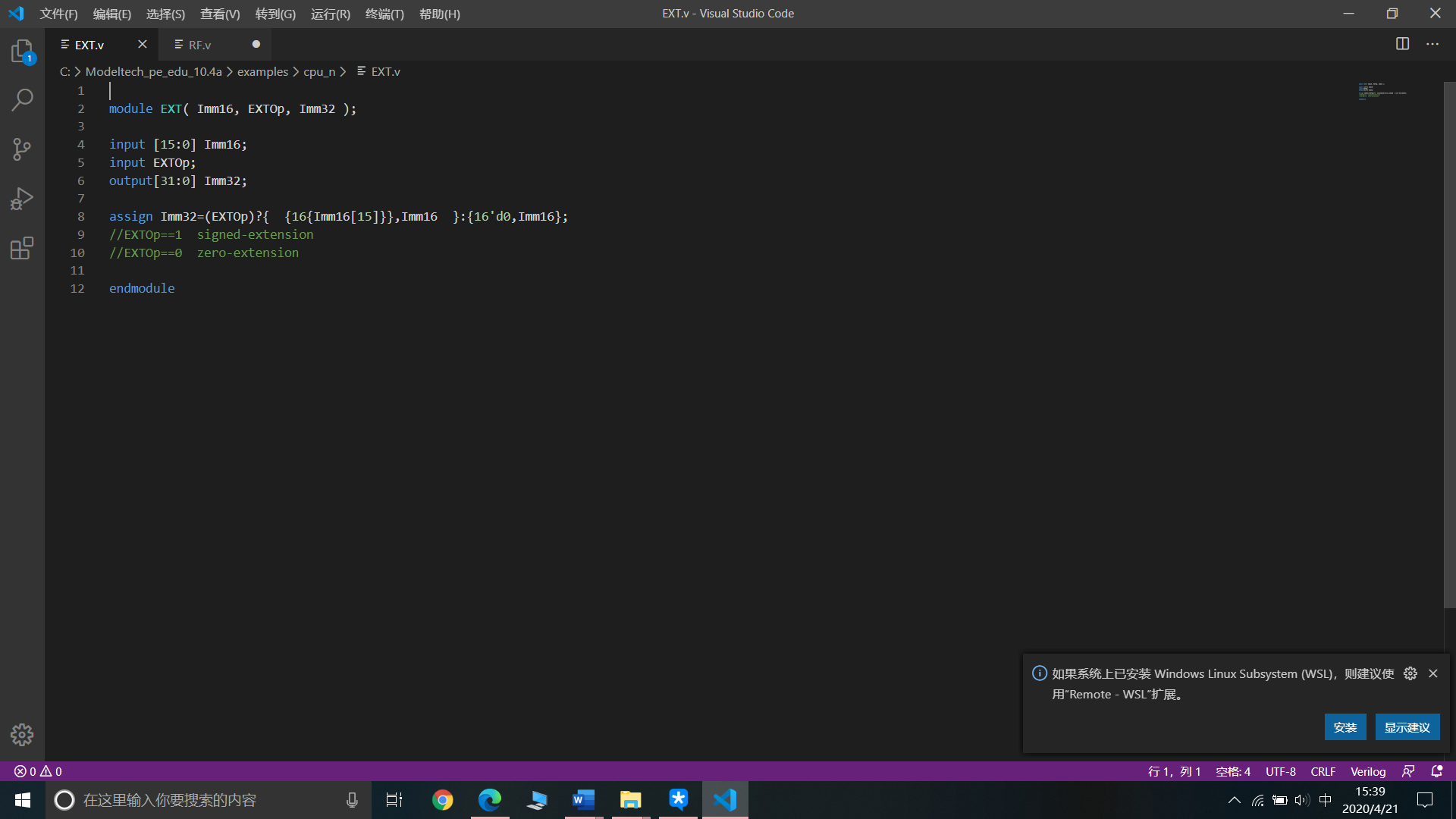
根据指令的地址，在指令存储器中取出相应指令。每个存储器单元为32位，共128个单元，即2^9字节的寻址空间。

**4.2.4 RF（寄存器堆）**



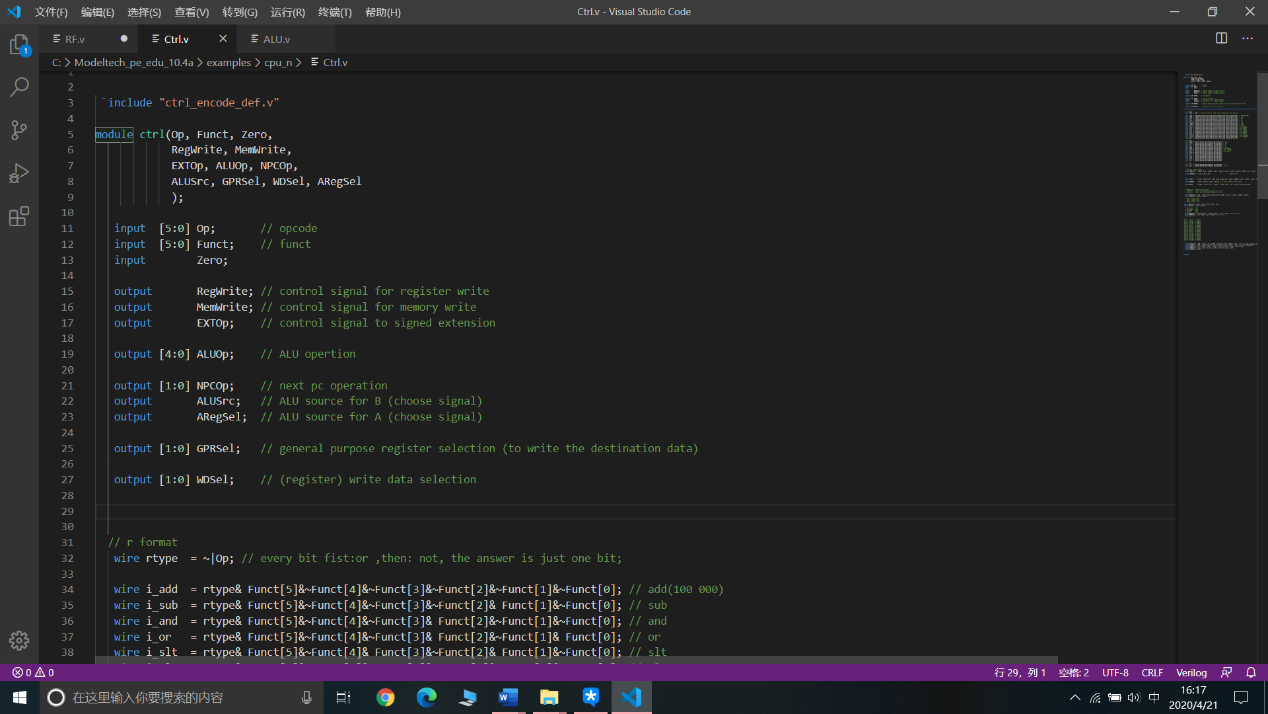
寄存器堆当clk在上升沿时触发，如果rst值为1，则寄存器值清零，如果rst值为0，再判断写信号RFWr是否为1，是则将输入的数据WD保存到写入地址对应的寄存器中。读端口如果读寄存器地址不为0，将读到的数据输出到相应端口中。

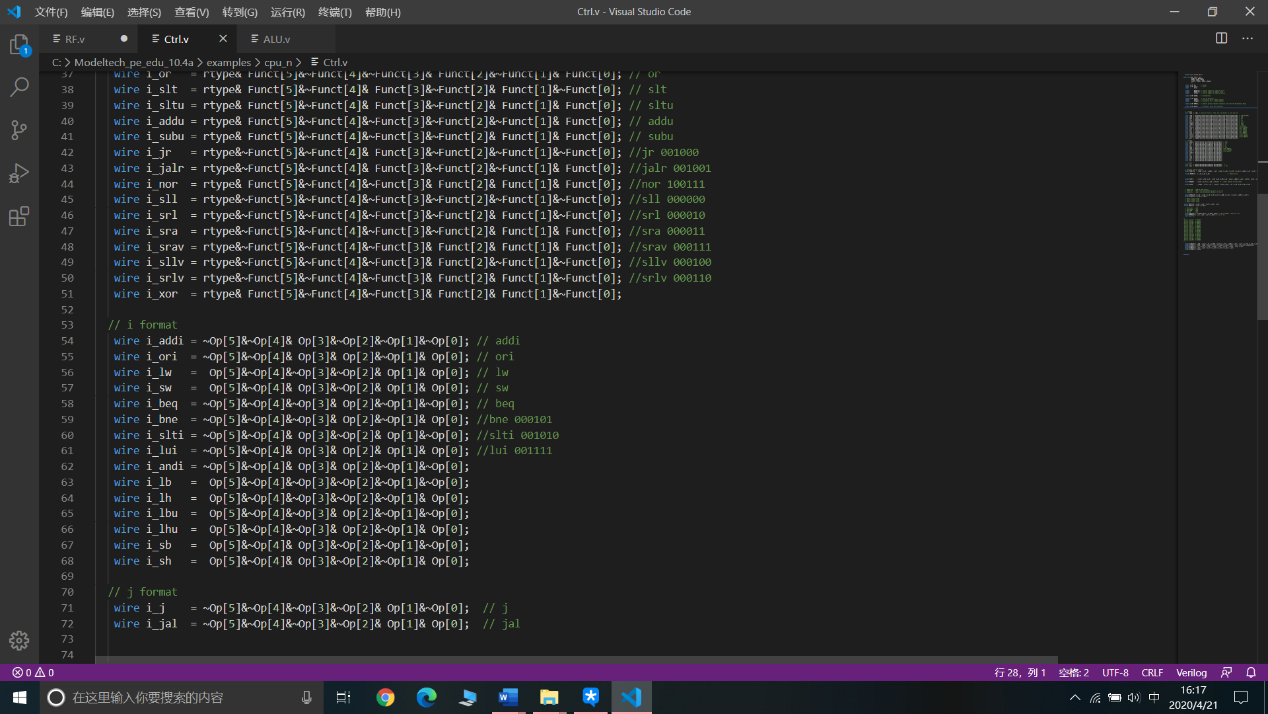
**4.2.5 EXT（符号扩展单元）**



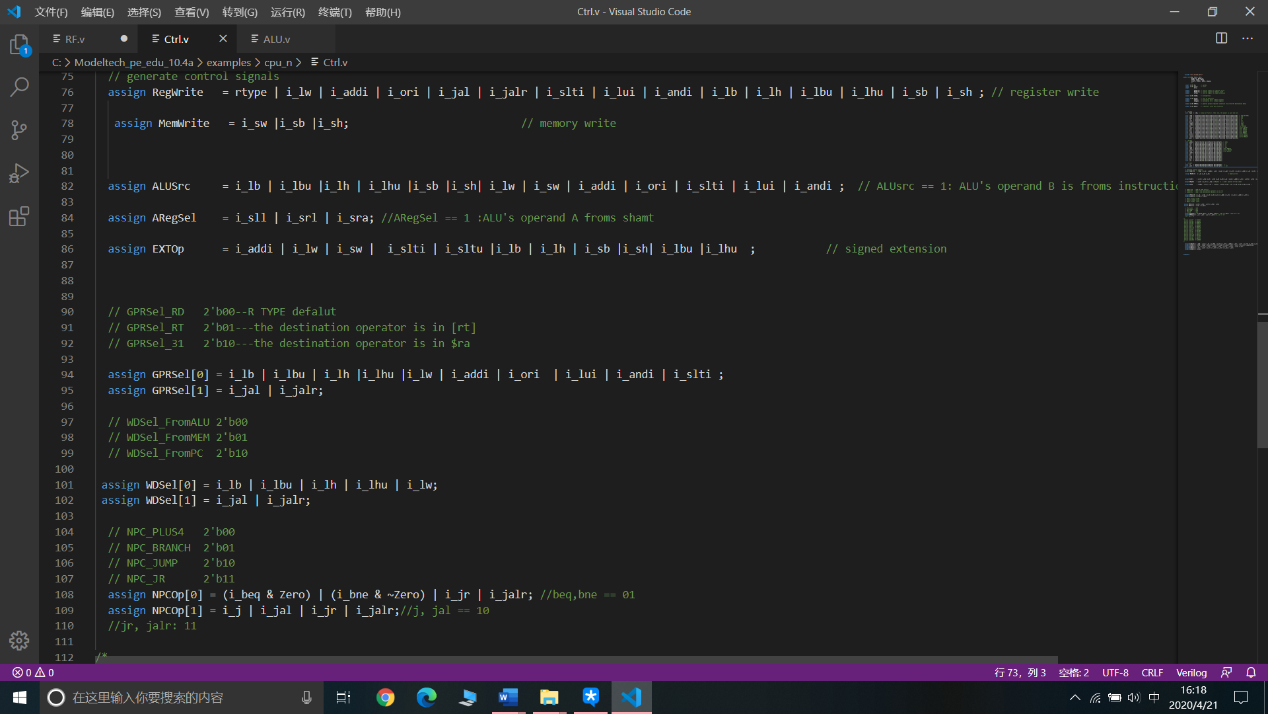
EXT中将指令字段里的16位立即数，EXTOp为1时进行符号扩展，为0进行0扩展，输出32为立即数。

**4.2.6 Ctrl（信号控制单元）**

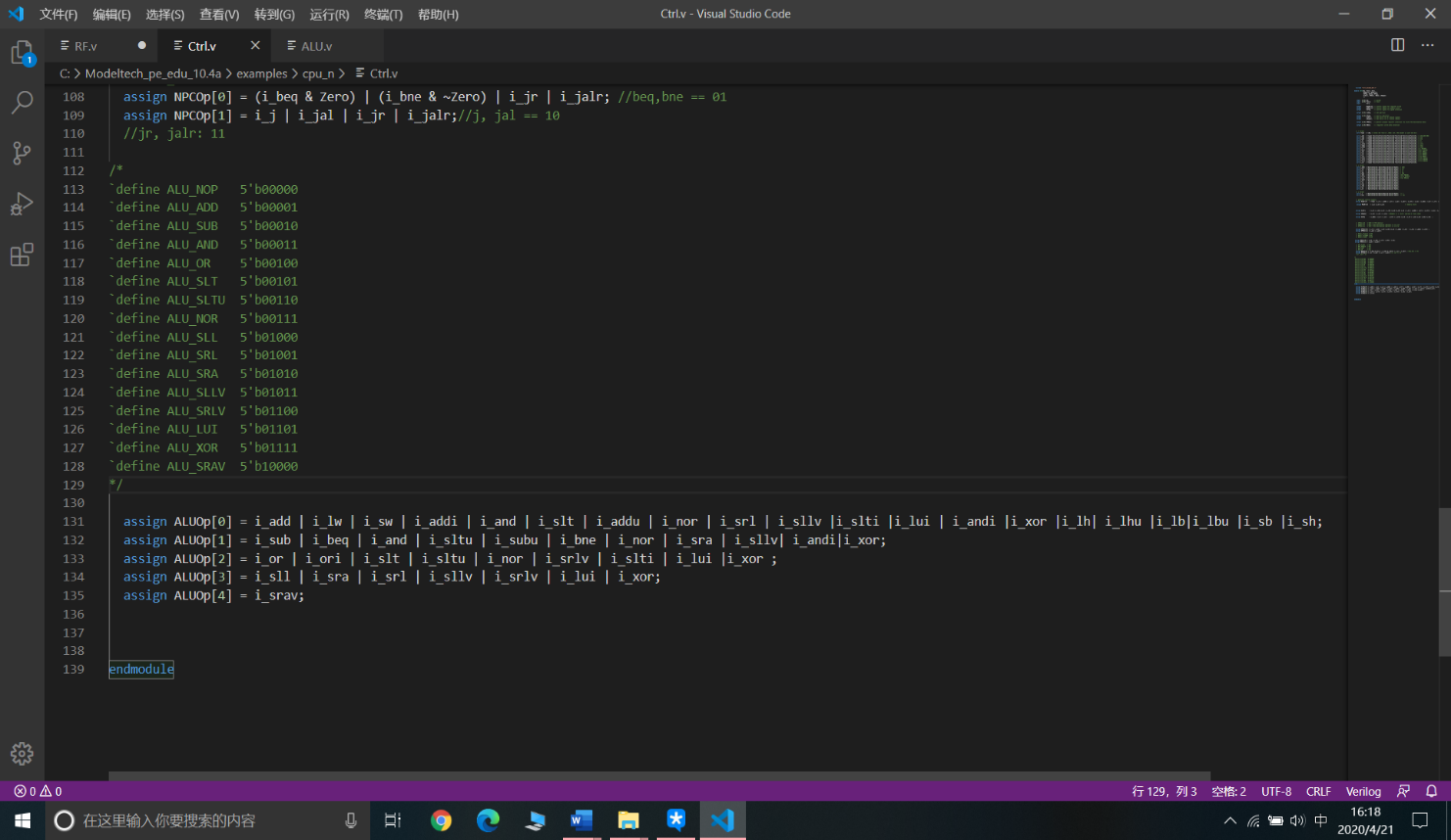




若输入指令符合哪一个op码的排列，则相应指令信号置1。

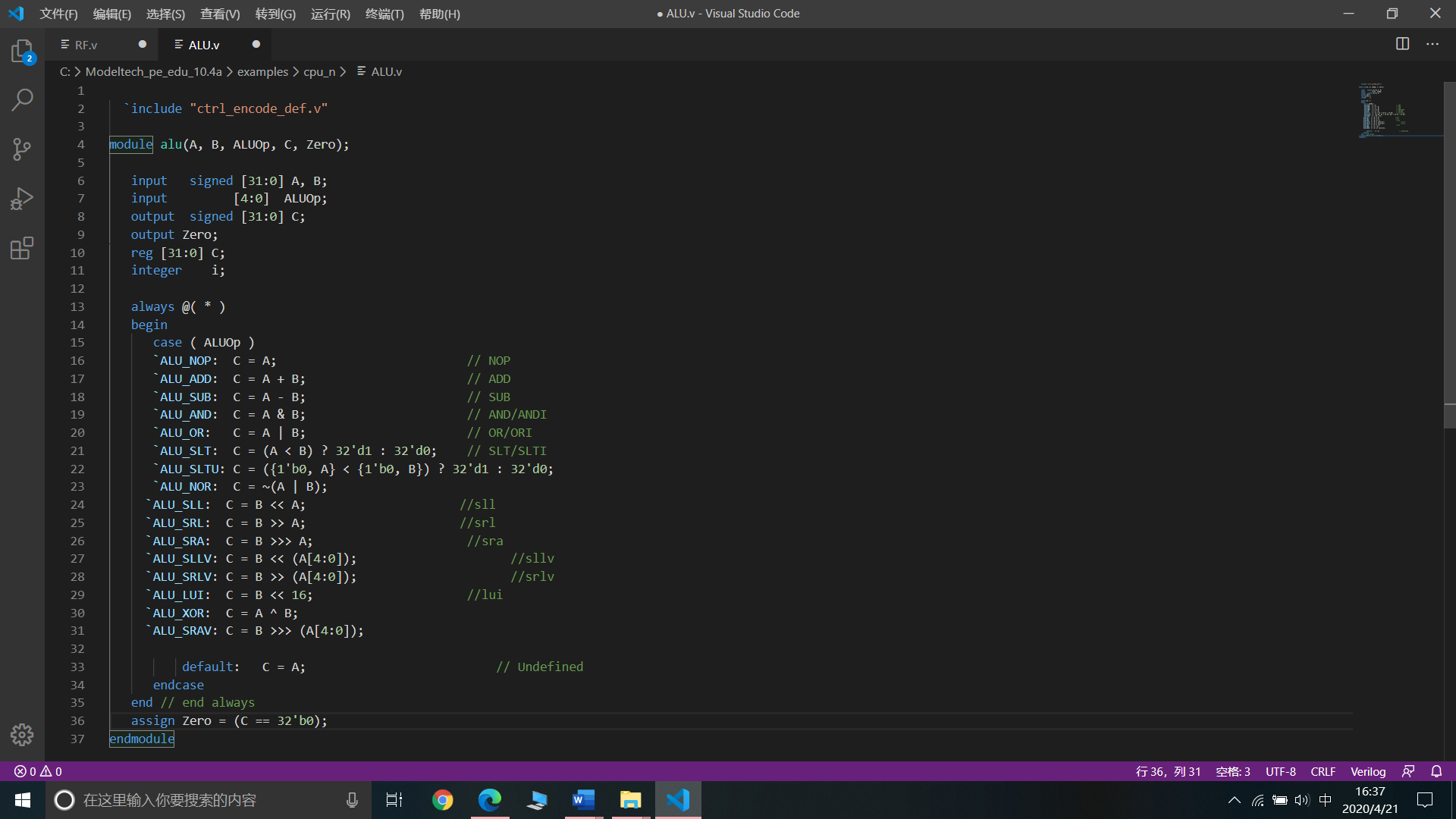


结和zero信号判断是否要跳转。

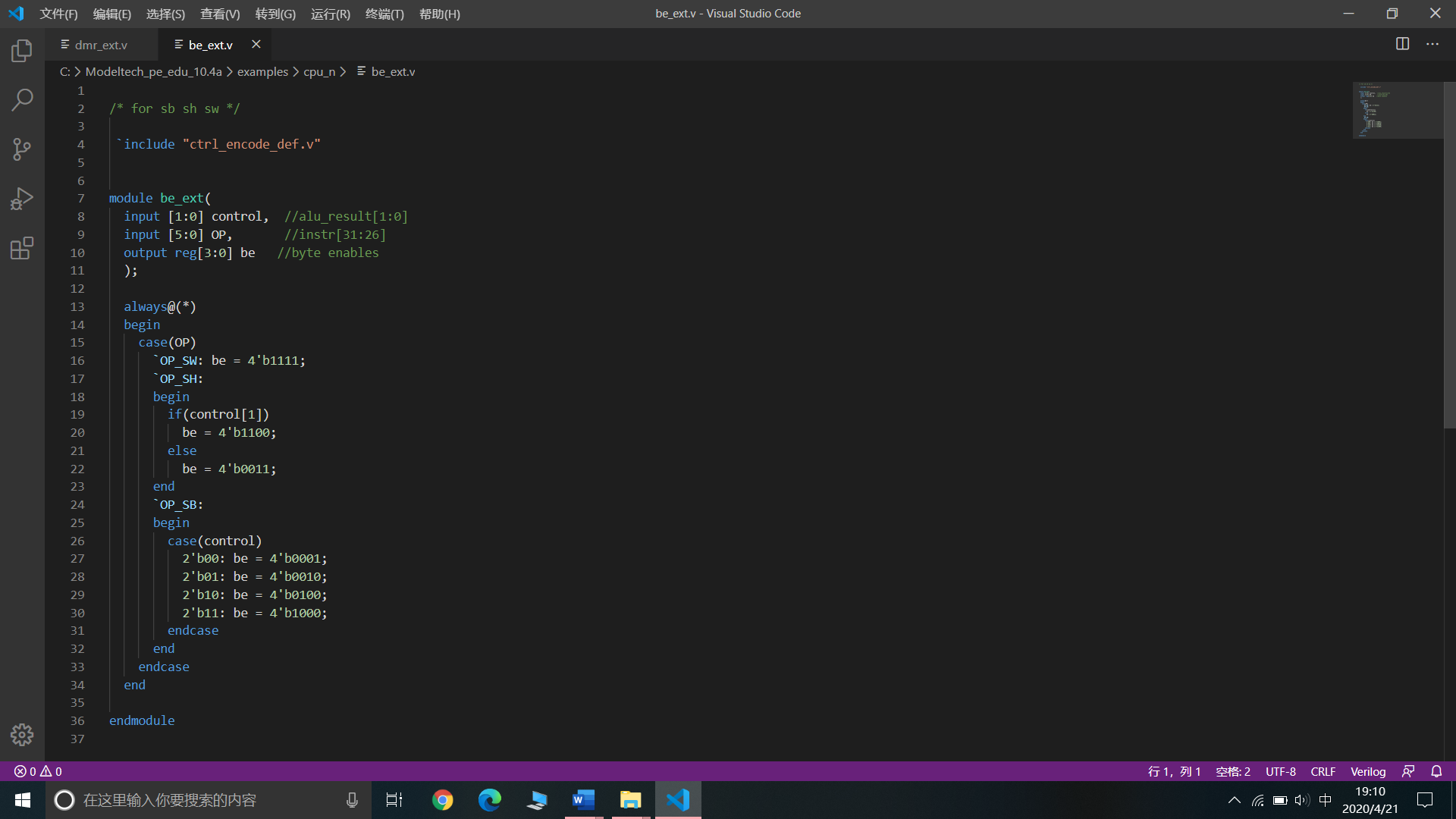


将指令译码，把指令中各个字段分解出来，并根据OP或者Funct字段判断出指令类型，产生ALU操作数来源信号，ALU的操作信号，NPC的选择信号，符号扩展型号，写寄存器信号和写寄存器的地址字段。

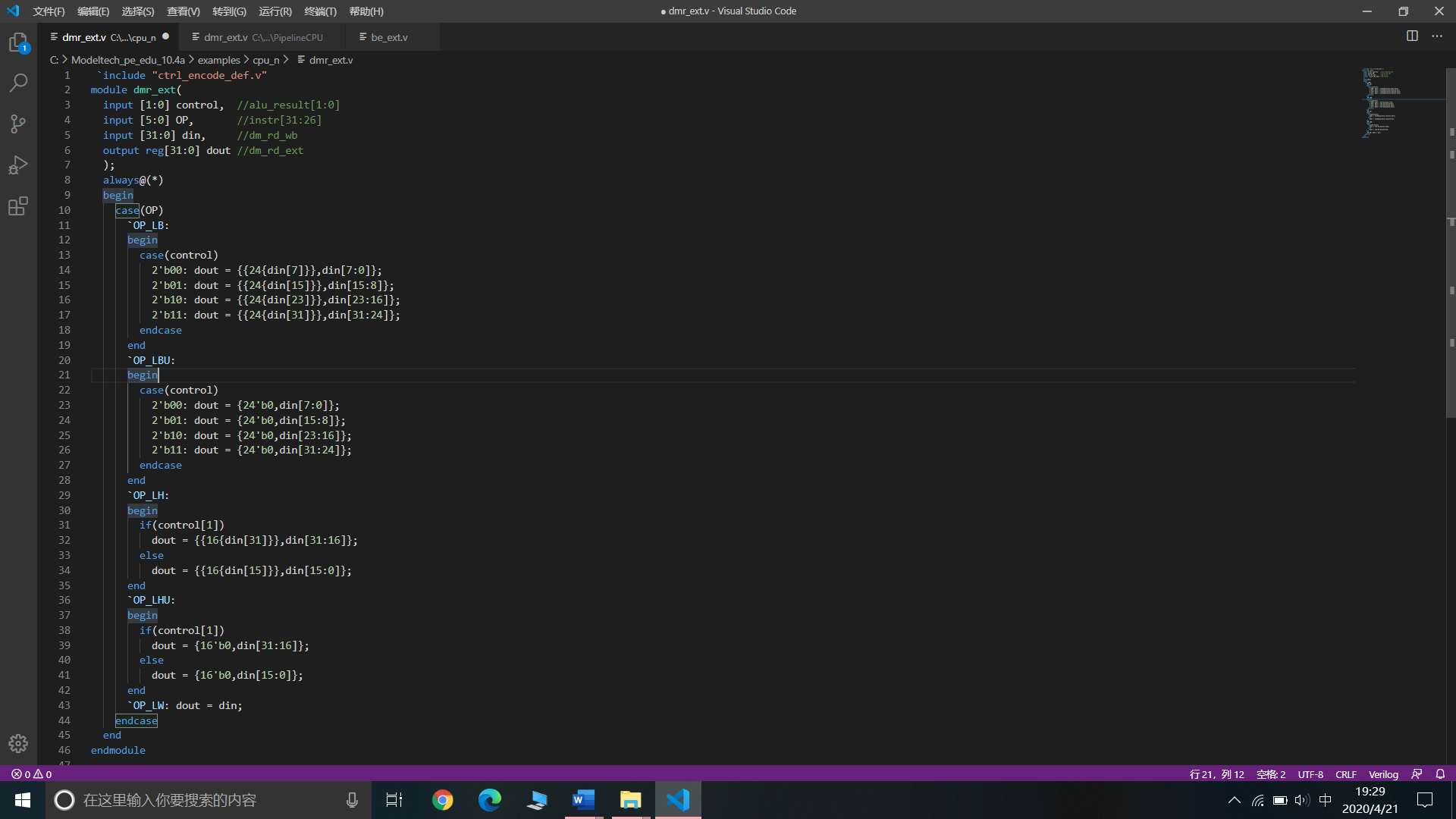
**4.2.7 ALU（运算器）**



ALU根据操作码，将输入的两个操作数进行相应的加，减，移位，与，或，非等运算，并将结果输出，当运算结果为0时，zero信号输出1。

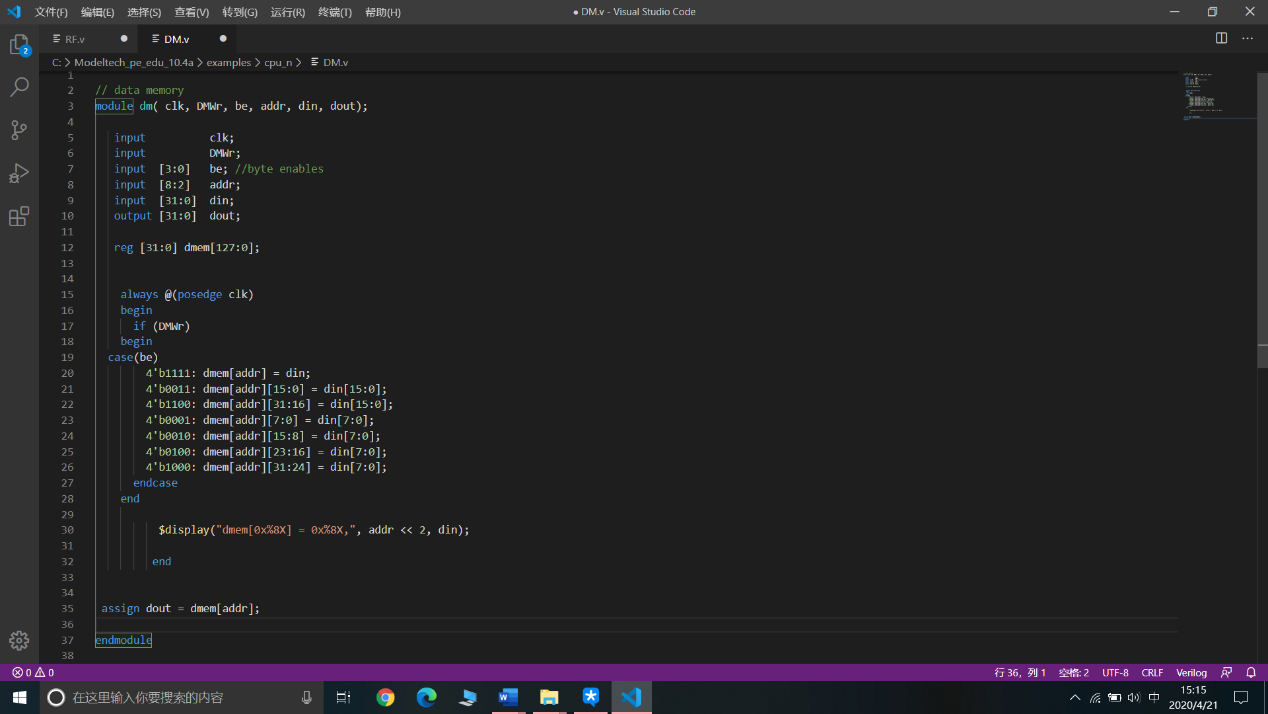
**4.2.8 Be\_ext（取数扩展控制单元）**

根据输入的指令Opcode和alu运算结果后两位判断是那一条存数指令，输出对应的控制信号。

**4.2.9 Dmr\_ext（存数扩展控制单元）**

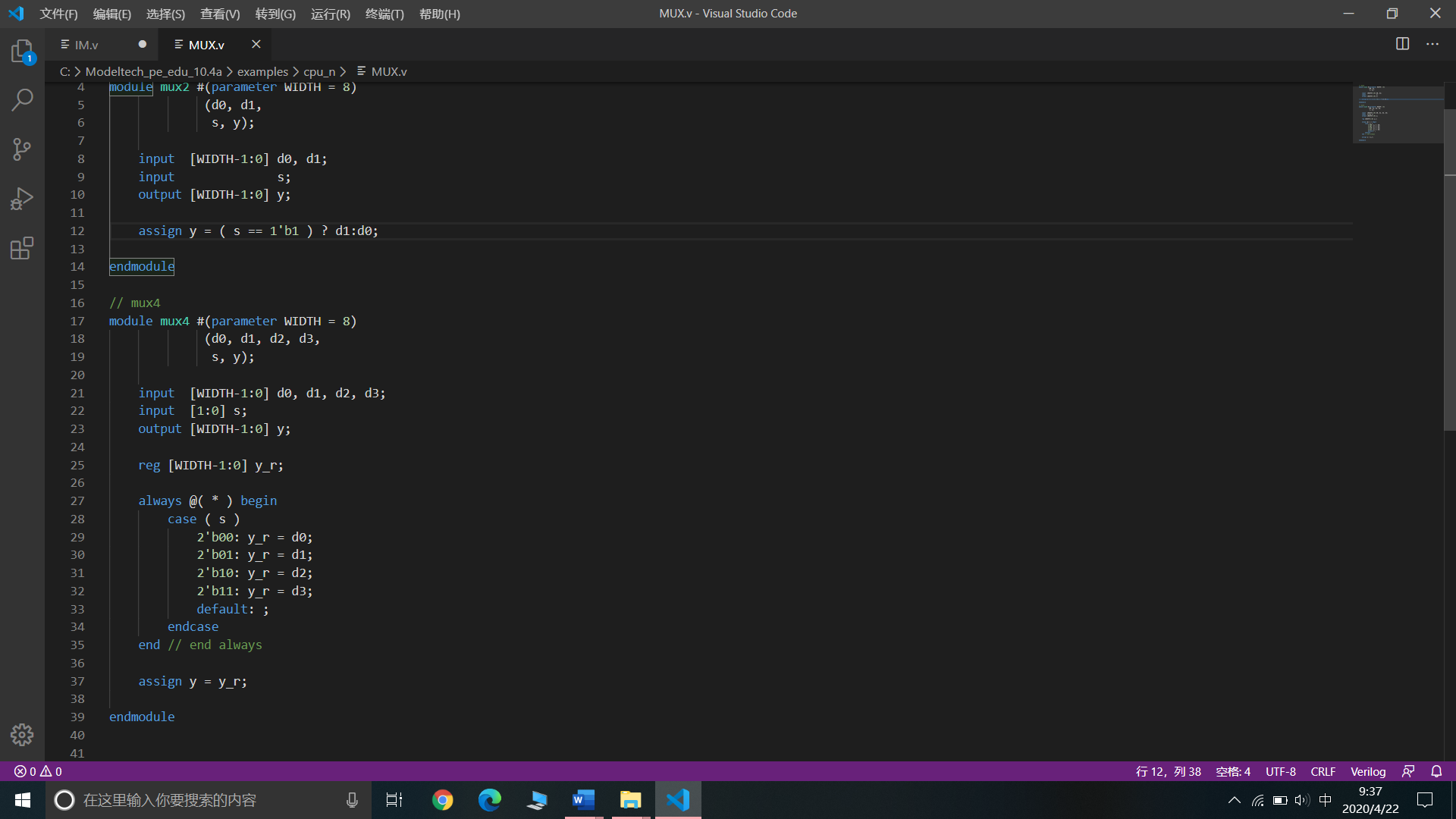
根据输入指令的Opcode和alu的运算结果判断是那一条取字指令，将取出的字节或半字，根据所在位置，进行相应的符号扩展至32位，lw指令直接输出，无需扩展。

4.2.10 DM（数据存储器）



从存数扩展控制单元来的be信号，根据指令要求存放的位置，存在相应的字节或半字，或字的位置中。

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，选择使用四选一多路选择器或者二选一多路选择器，数据位数长度在实例化时可以改变。