练习题 4.24　条件传送指令(简称 cmovXX)的指令代码为 IRRMOVQ。如图 4-28 所示，我们可以用执行阶段中产生的 Cnd 信号实现这些指令。修改 dstE 的 HCL 代码以实现这些指令。

4. 访存阶段

访存阶段的任务就是读或者写程序数据。如图 4-30 所示，两个控制块产生内存地址和内存输入数据(为写操作)的值。另外两个块产生表明应该执行读操作还是写操作的控制信号。当执行读操作时，数据内存产生值 valM。

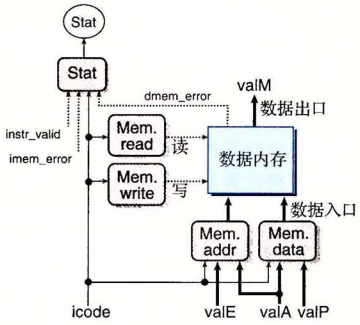

图 4-30　SEQ 访存阶段。数据内存既可以写，也可以读内存的值。从内存中读出的值就形成了信号 valM

图 4-18～图 4-21 的访存阶段给出了每个指令类型所需要的内存操作。可以看到内存读和写的地址总是 valE 或 valA。这个块用 HCL 描述就是：

```
word mem_addr = [
        icode in { IRMMOVQ, IPUSHQ, ICALL, IMRMOVQ } : valE;
        icode in { IPOPQ, IRET } : valA;
        # Other instructions don't need address
];
```

练习题 4.25　观察图 4-18～图 4-21 所示的不同指令的访存操作，我们可以看到内存写的数据总是 valA 或 valP。写出 SEQ 中信号 mem_data 的 HCL 代码。

我们希望只为从内存读数据的指令设置控制信号 mem_read，用 HCL 代码表示就是：

```
bool mem_read = icode in { IMRMOVQ, IPOPQ, IRET };
```

练习题 4.26　我们希望只为向内存写数据的指令设置控制信号 mem_write。写出 SEQ 中信号 mem_write 的 HCL 代码。

访存阶段最后的功能是根据取值阶段产生的 icode、imem_error、instr_valid 值以及数据内存产生的 dmem_error 信号，从指令执行的结果来计算状态码 Stat。

练习题 4.27　写出 Stat 的 HCL 代码，产生四个状态码 SAOK、SADR、SINS 和 SHLT(参见图 4-26)。

5. 更新 PC 阶段

SEQ 中最后一个阶段会产生程序计数器的新值(见图 4-31)。如图 4-18～图 4-21 中最后步骤所示，依据指令的类型和是否要选择分支，新的 PC 可能是 valC、valM 或 valP。用 HCL 来描述这个选择就是：

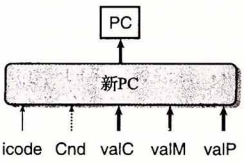

图 4-31　SEQ 更新 PC 阶段。根据指令代码和分支标志，从信号 valC、valM 和 valP 中选出下一个 PC 的值

```
word new_pc = [
        # Call.  Use instruction constant
        icode == ICALL : valC;
        # Taken branch.  Use instruction constant
        icode == IJXX && Cnd : valC;
        # Completion of RET instruction.  Use value from stack
```