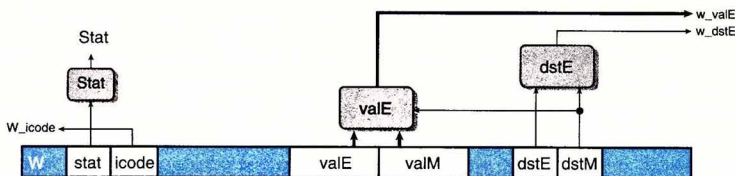


图 4-70 能够进行加载转发的执行和访存阶段。通过添加一条从内存输出到流水线寄存器 M 中 valA 的源的旁路通路，对于这种形式的加载/使用冒险，我们可以使用转发而不必暂停。这是家庭作业 4.57 的主旨

**\*\* 4.58** 我们的流水线化的设计有点不太现实，因为寄存器文件有两个写端口，然而只有 popq 指令需要对寄存器文件同时进行两个写操作。因此，其他指令只使用一个写端口，共享这个端口来写 valE 和 valM。下面这个图是一个对写回逻辑的修改版，其中，我们将写回寄存器 ID(W\_dstE 和 W\_dstM) 合并成一个信号 w\_dstE，同时也将写回值(W\_valE 和 W\_valM)合并成一个信号 w\_valE:



用 HCL 写执行这些合并的逻辑，如下所示：

```
## Set E port register ID
word w_dstE = [
    ## writing from valM
    W_dstM != RNONE : W_dstM;
    1: W_dstE;
];

## Set E port value
word w_valE = [
    W_dstM != RNONE : W_valM;
    1: W_valE;
];
```