

图 4-60 PIPE 的执行阶段逻辑。这一部分的设计与 SEQ 实现中的逻辑非常相似

练习题 4.35 d_valA 的 HCL 代码中的第二种情况使用了信号 e_dstE ，来判断是否要选择 ALU 的输出 e_valE 作为转发源。假设我们用 E_dstE ，也就是流水线寄存器 E 中的目的寄存器 ID，来作为这个选择。写出一个采用这个修改过的转发逻辑就会产生错误结果的 Y86-64 程序。

4. 访存阶段

图 4-61 是 PIPE 的访存阶段逻辑。将这个逻辑与 SEQ 的访存阶段(图 4-30)相比较，我们看到，正如前面提到的那样，PIPE 中没有 SEQ 中标号为“Data”的块。这个块是用来在数据源 $valP$ (对 $call$ 指令来说)和 $valA$ 中进行选择的，但是这个选择现在由译码阶段中标号为“Sel+Fwd A”的块来执行。这个阶段中的其他块都和 SEQ 中相应的部件相同，采用的信号做适当的重命名。在图中，你还可以看到许多流水线寄存器 M 和 W 中的值作为转发和流水线控制逻辑的一部分，提供给电路中其他部分。

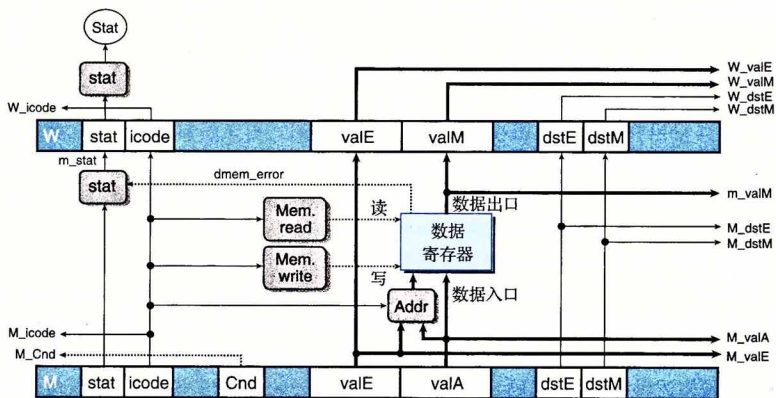


图 4-61 PIPE 的访存阶段逻辑。许多从流水线寄存器 M 和 W 来的信号被传递到较早的阶段，以提供写回的结果、指令地址以及转发的结果