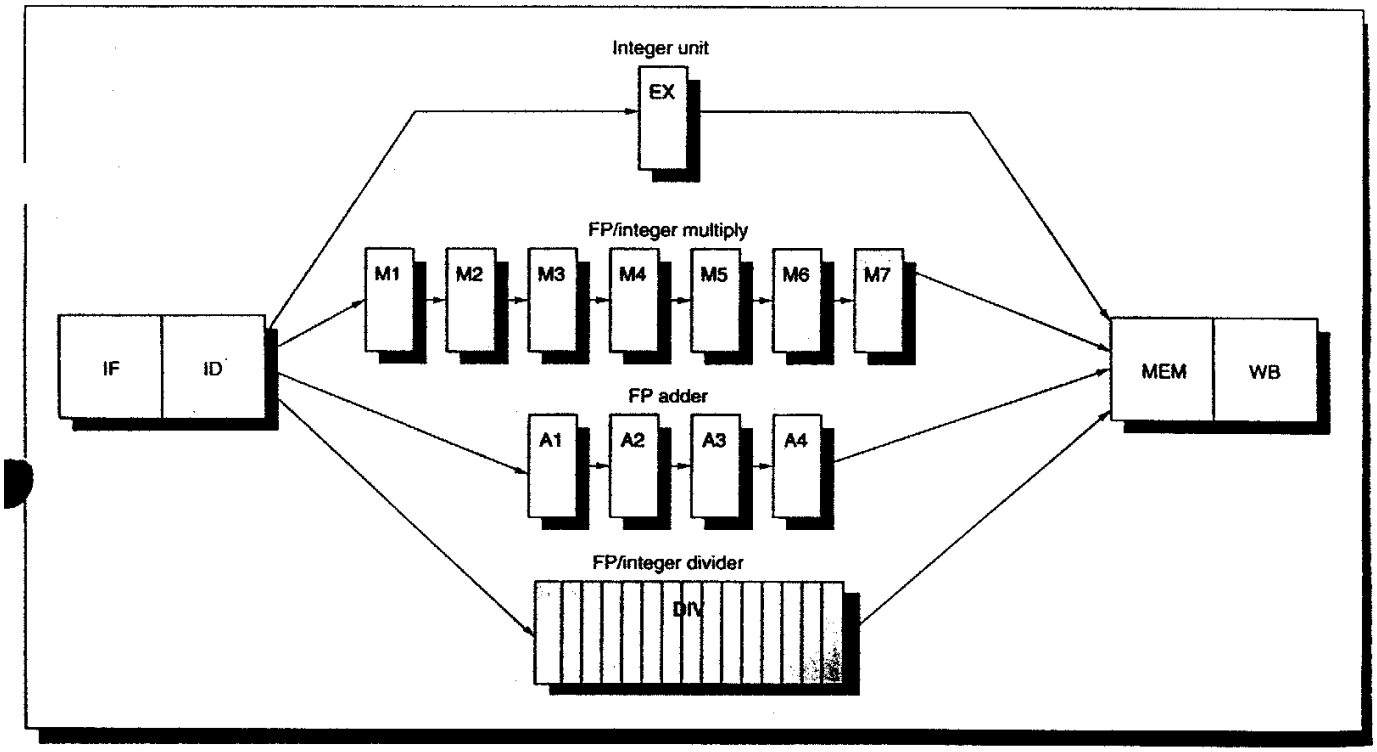

FIGURE 3.44 A pipeline that supports multiple outstanding FP operations. The FP multiplier and adder are fully pipelined and have a depth of seven and four stages, respectively. The FP divider is not pipelined, but requires 25 clock cycles to complete. The latency in instructions between the issue of an FP operation and the use of the result of that operation without incurring a RAW stall is determined by the number of cycles spent in the execution stages. For example, the fourth in ction after an FP add can use the result of the FP add. For integer ALU operations, the depth of the execution pipeline is always one and the next instruction can use the results. Both FP loads and integer loads complete during MEM, which means that the memory system must provide either 32 or 64 bits in a single clock.

1. The instructions below are run through the floating-point MIPS pipeline above. Show the timing of this instruction sequence with forwarding.

```
loop:   ld    $f0, 0($s2)
        ld    $f4, 0($s3)
        multd $f0, $f0, $f4
        addd  $f2, $f0, $f2
        sd    $f2, 0($s2)
        addi  $s2, $s2, 8
        addi  $s3, $s3, 8
        stl   $t0, $s3, 128
        bne   $t0, 0, loop
```

2. Find a code sequence for the above pipeline that will cause a structural hazard.

3. There are many stalls in the above solution. Unroll the loop to minimize stalls.