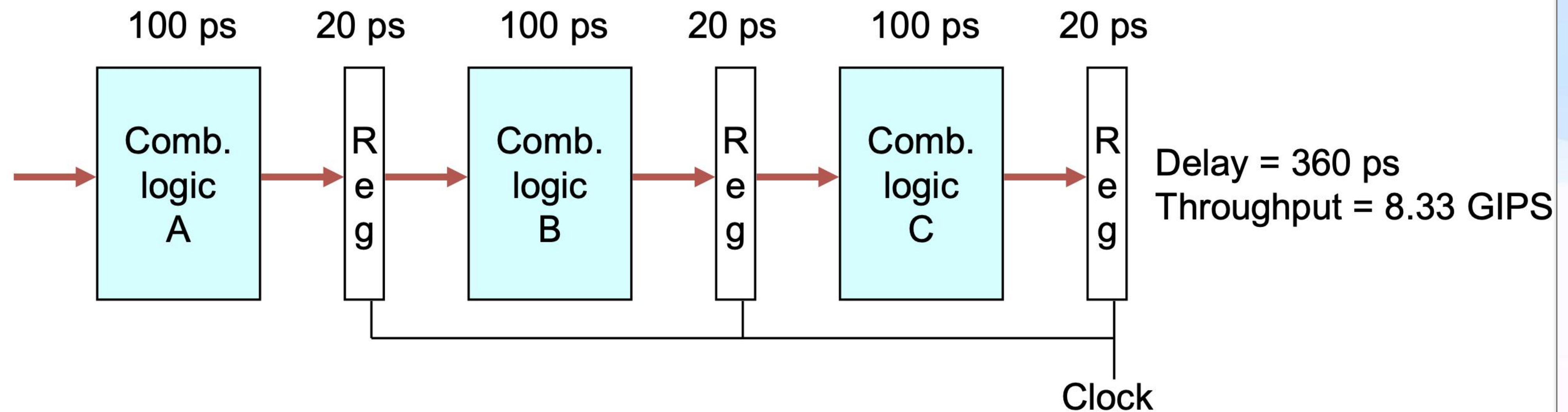


Pipelined Implementation

2024. 10. 23

流水线通用原理



性能分析

CPI: 执行一条指令所需要的平均时钟周期数 (单位: 时钟周期)

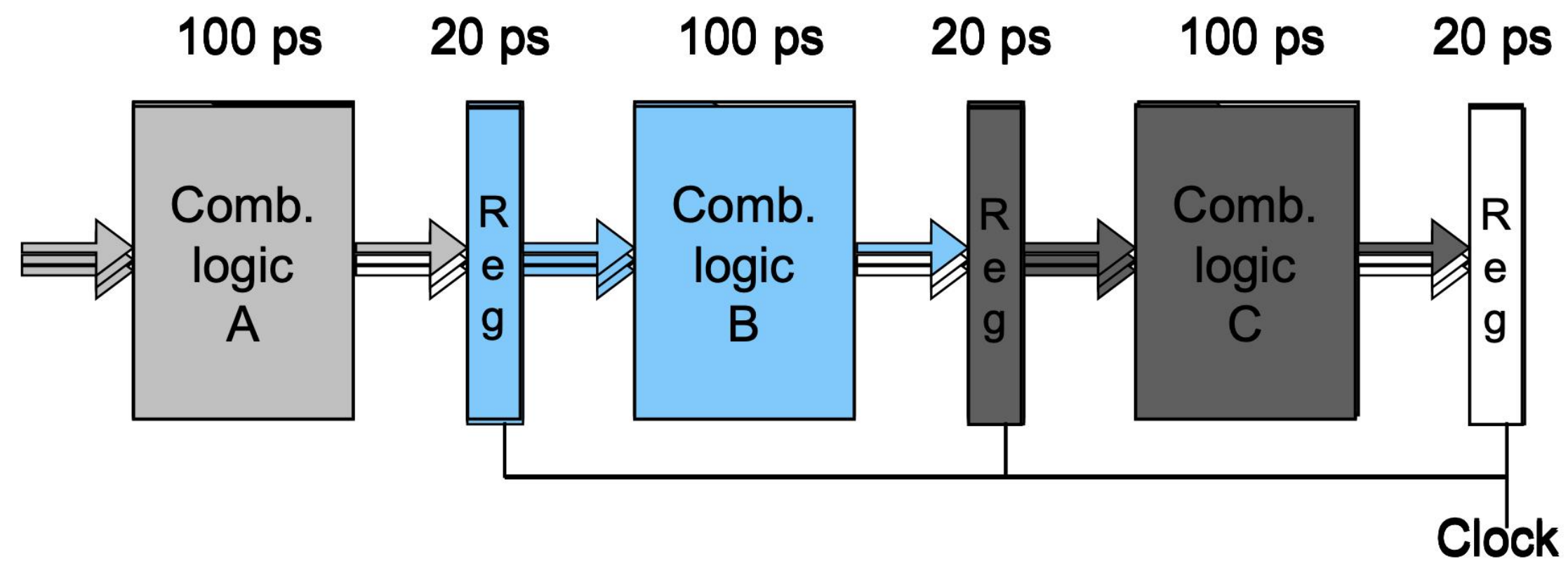
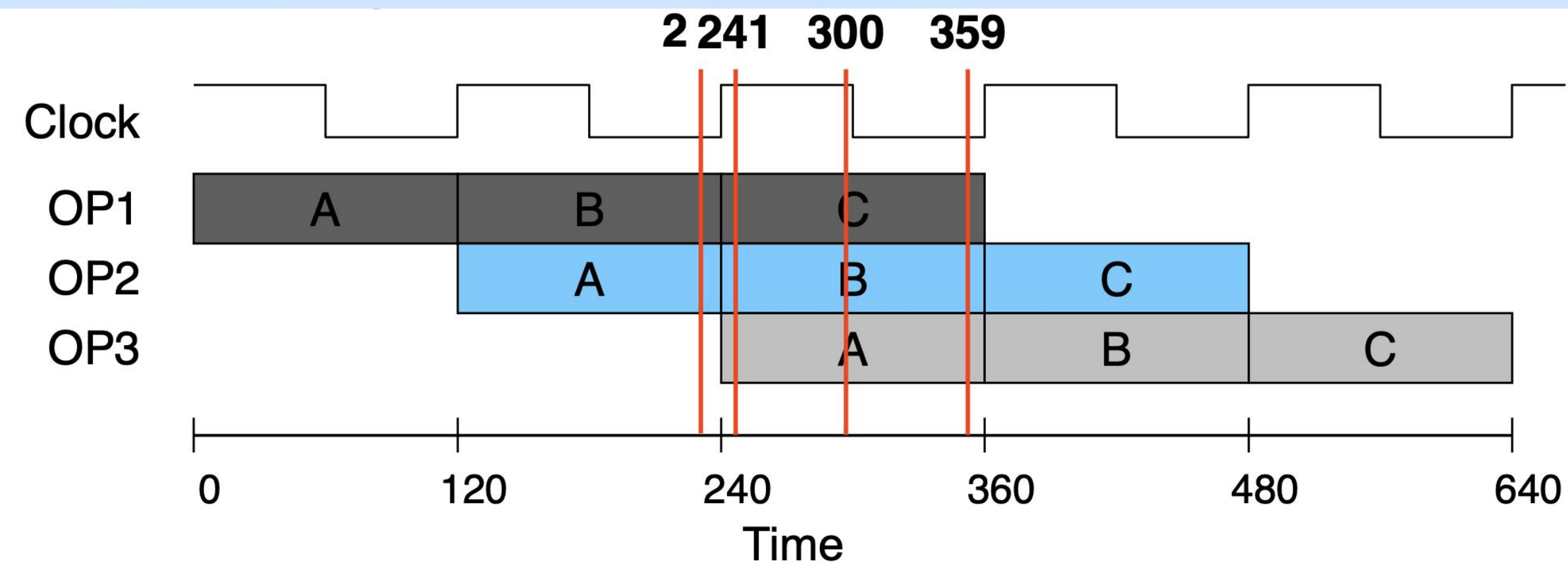
$$\text{CPI} = \frac{C_i + C_b}{C_i} = 1.0 + \frac{C_b}{C_i}$$
$$\text{CPI} = 1.0 + lp + mp + rp$$

- lp - 加载处罚, mp - 预测错误分支处罚, rp - 返回处罚

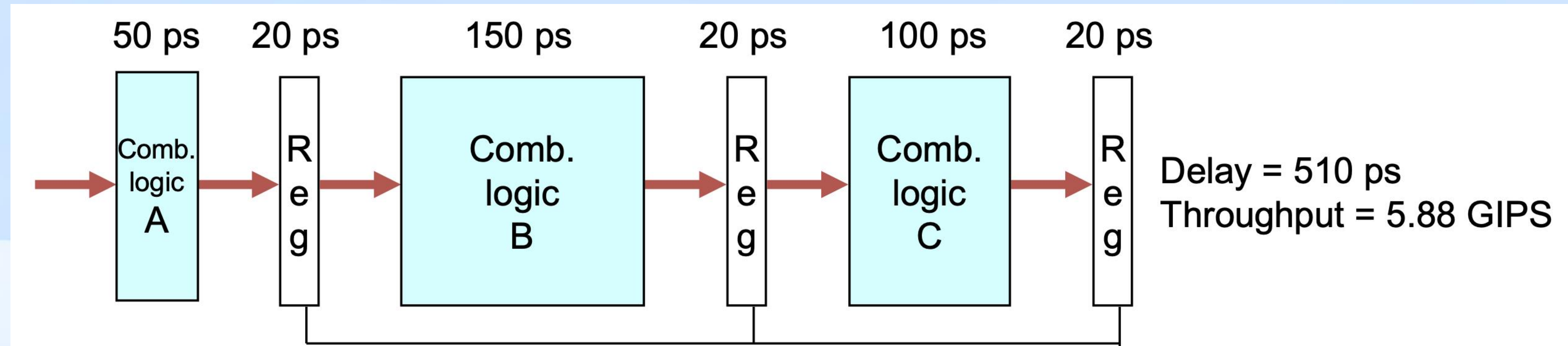
Cause	Name	Instruction frequency	Condition frequency	Bubbles	Product
Load/use	lp	0.25	0.20	1	0.05
Mispredict	mp	0.20	0.40	2	0.16
Return	rp	0.02	1.00	3	0.06
Total penalty					0.27

流水线的操作

图为 $t=359$ 时流水线状态

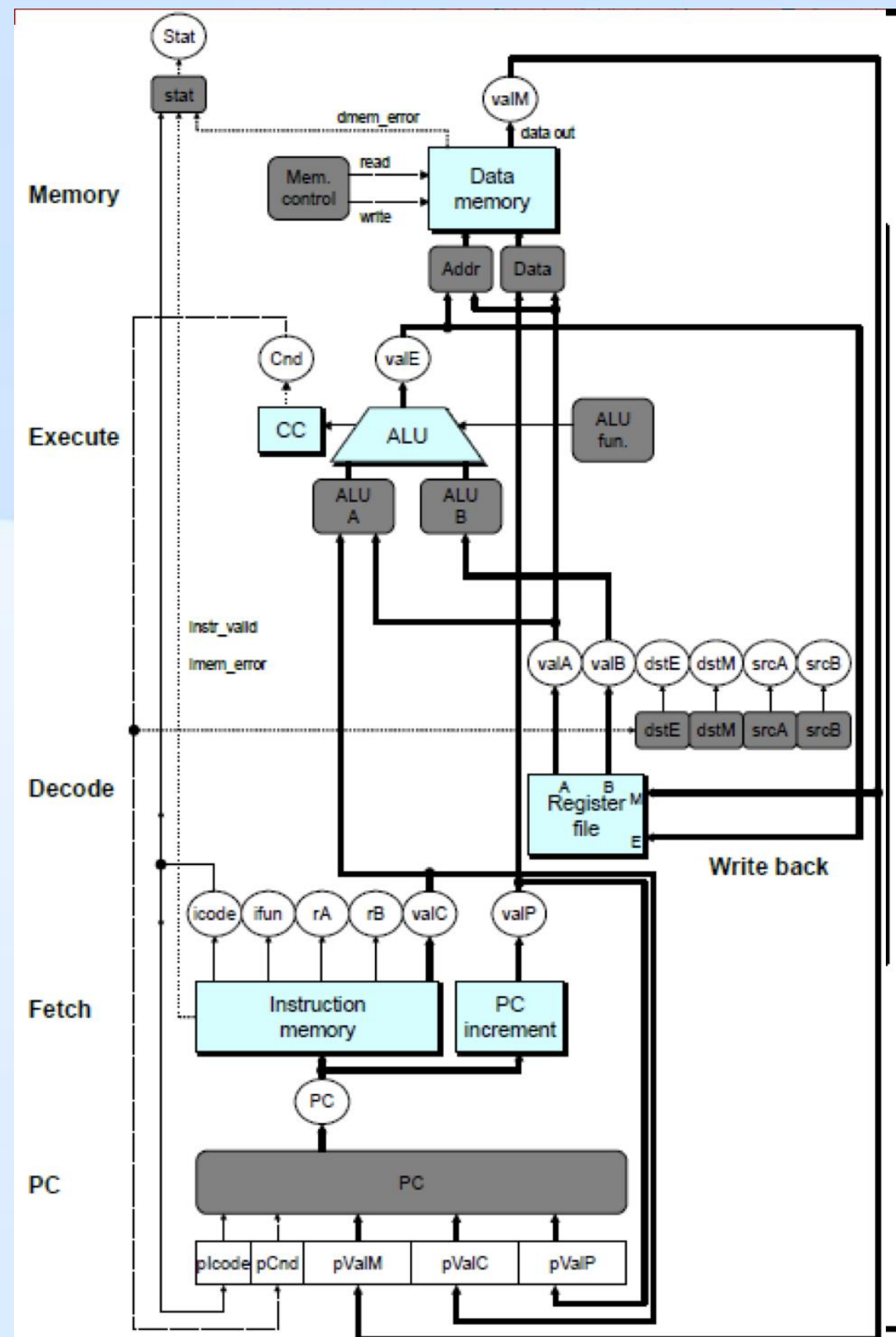


流水线的局限性



SEQ+: 重新安排计算阶段

对比：更新 PC 阶段在时钟周期开始执行（电路重定时）



流水线冒险

- 当相邻指令出现相关时会导致出现问题：数据相关 or 控制相关
- 数据冒险：相邻指令用到了同样的寄存器产生数据相关
- 控制冒险：由于跳转指令的存在，不知道下一步应该执行哪一条指令

数据冒险

SEQ+结构可能会遇到如下问题:

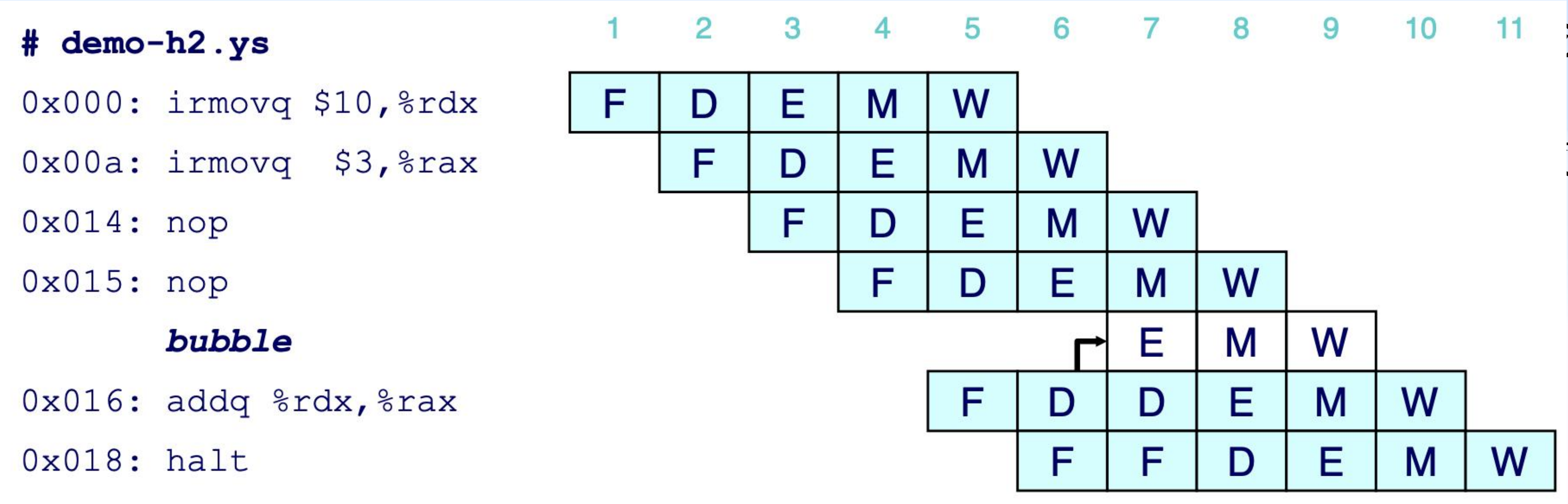
```
# demo-h0.ys
0x000: irmovq $10,%rdx
0x00a: irmovq $3,%rax
0x014: addq %rdx,%rax
0x016: halt
```

1	2	3	4	5	6	7	8
F	D	E	M	W			
	F	D	E	M	W		
		F	D	E	M	W	
			F	D	E	M	W

到写回阶段(W)，引发

数据冒险

暂停 (Stalling)

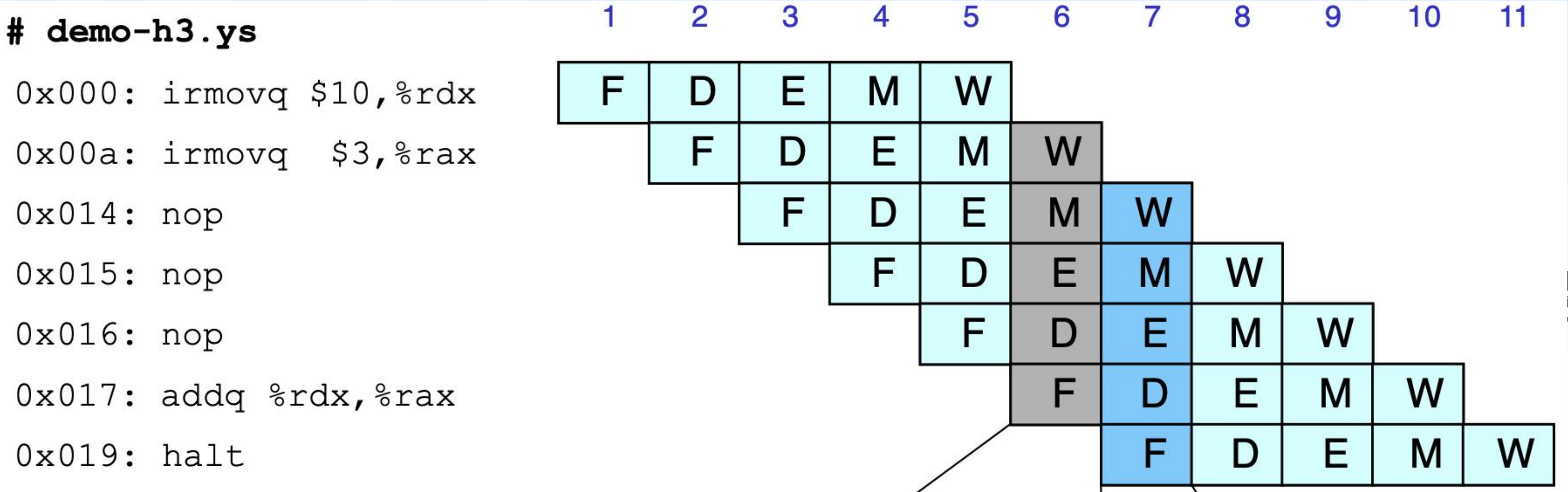


条件不满足

中插入一个气泡

数据冒险

解决方法：执行暂停指令，将 add 停留在译码阶段



写(W) 才可以执行！

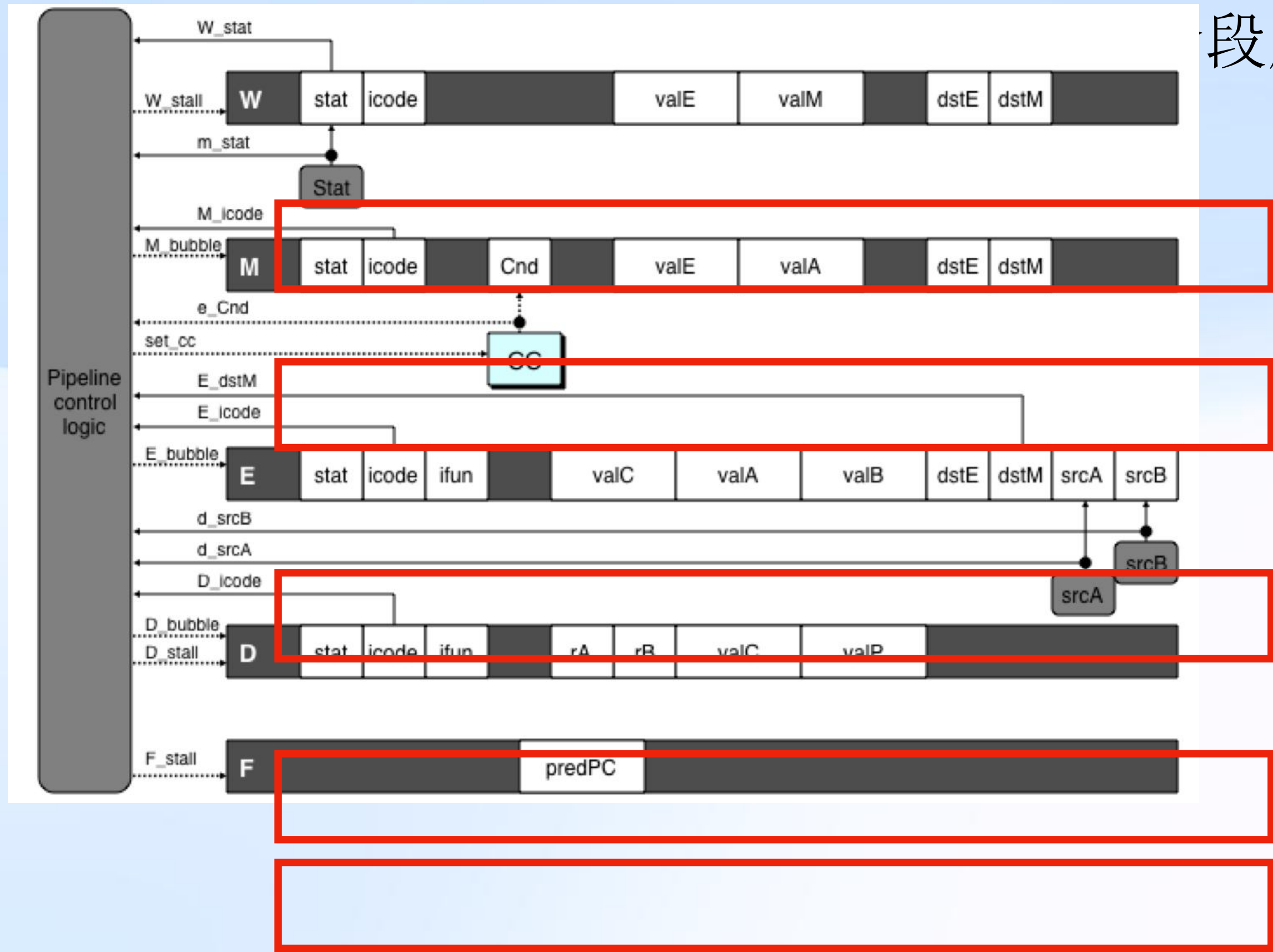
PIPE-

解决方案：插入流水寄存器！

- 目的：将每次操作阶段的结果保存在对应阶段的流水寄存器中，可以直接传输，使后续指令不需要等到写回指令(W)就可以正常执行，传输优先级 $E > M > W$
- F：保存程序计数器预测值
- D：保存最新取出的指令和信息，交给译码阶段处理
- E：保存最新译码的指令和信息和从寄存器文件读出的信息，交给执行阶段处理
- M：保存最新执行的指令和结果，以及用于处理条件转移的分支条件目标的信息
- W：将计算出来的值提供给寄存器文件写，完成 ret 时向 PC选择逻辑提供返回地址

数据冒险

转发（使用 PIPE-）



段所产生的值，并且将值直接传到较早阶段

加载/使用数据冒险

解决：在转发基础上加入一次暂停

```
# demo-luh.ys
```

```
0x000: irmovq $128,%rdx
```

```
0x00a: irmovq $3,%rcx
```

```
0x014: rmmovq %rcx, 0(%rdx)
```

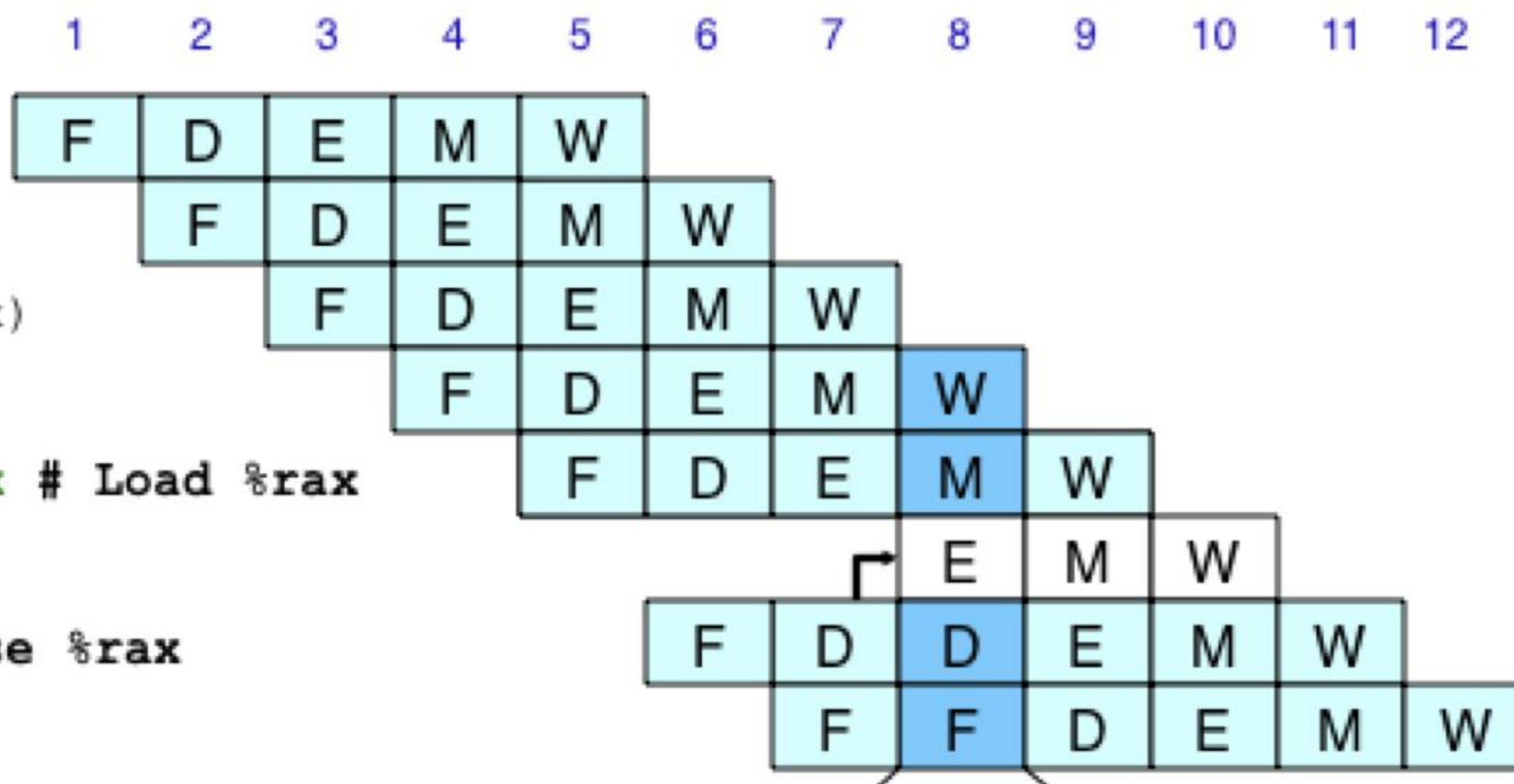
```
0x01e: irmovq $10,%rbx
```

```
0x028: mrmovq 0(%rdx),%rax # Load %rax
```

bubble

```
0x032: addq %rbx,%rax # Use %rax
```

```
0x034: halt
```



控制冒险

处理 ret 指令

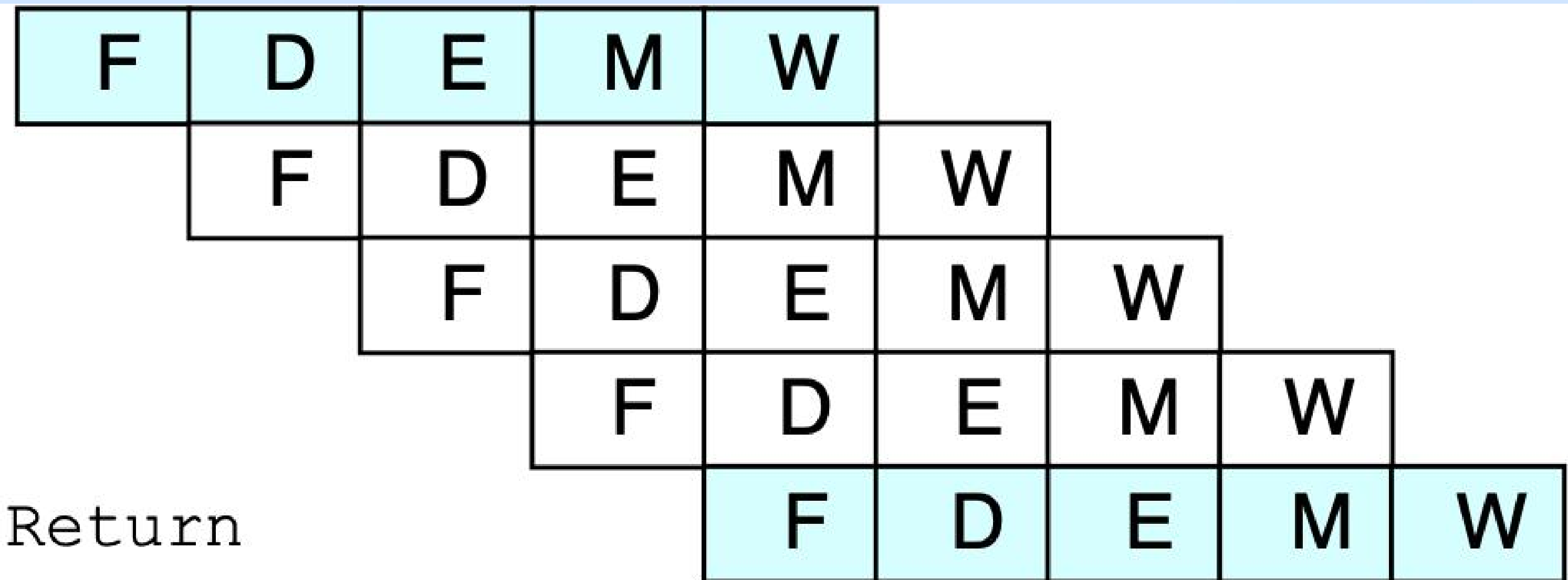
ret

bubble

bubble

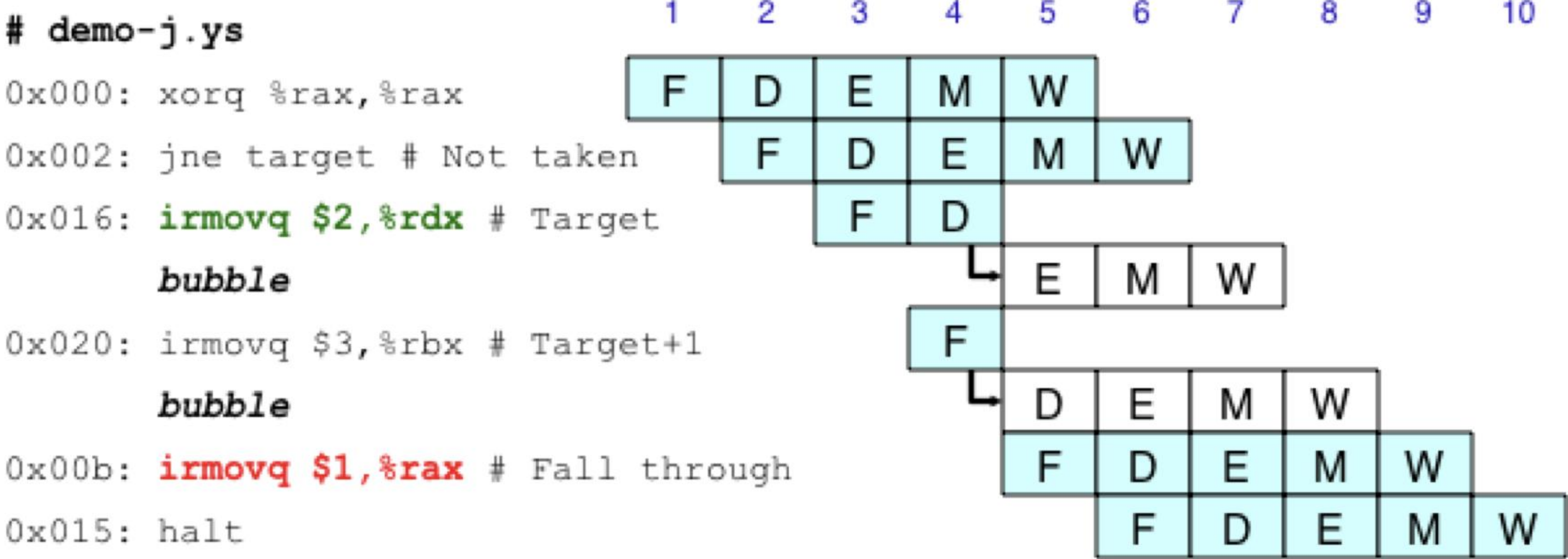
bubble

irmovq \$5,%rsi # Return



控制冒险

处理条件判断指令



若预测正确则正常
不正确指令

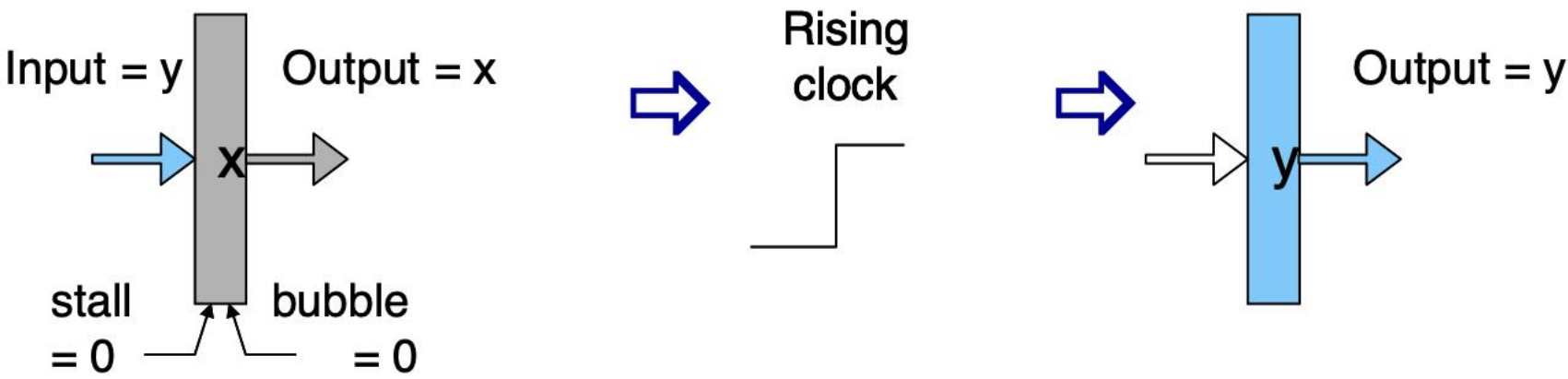
PIPE 各阶段的实现

PC 选择和取指阶段

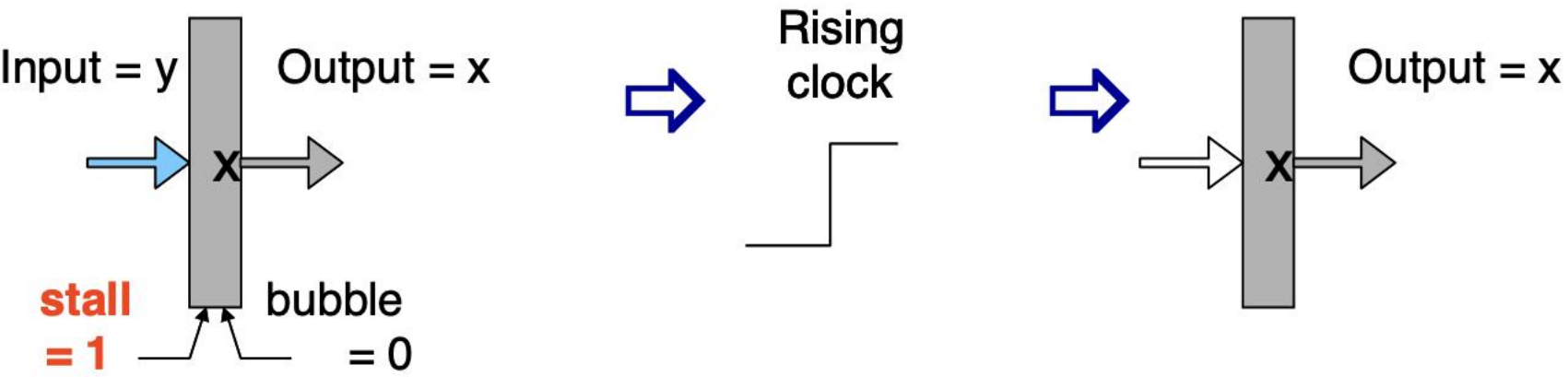
- 如何预测下一条指令的地址：
- 顺序执行：直接计算编码长度即可，值为 `valP`
- 跳转执行(`call`, `jxx`)：使用 `valC` 表示，若为 `call`, `jmp` 则值即为 `valC`
- `ret`：需要读取栈，直接使用 `valP`（无法预测）
- 会根据指令类型（`icode`）来判断下一条指令的地址为 `valP` or `valC`

暂停&气泡

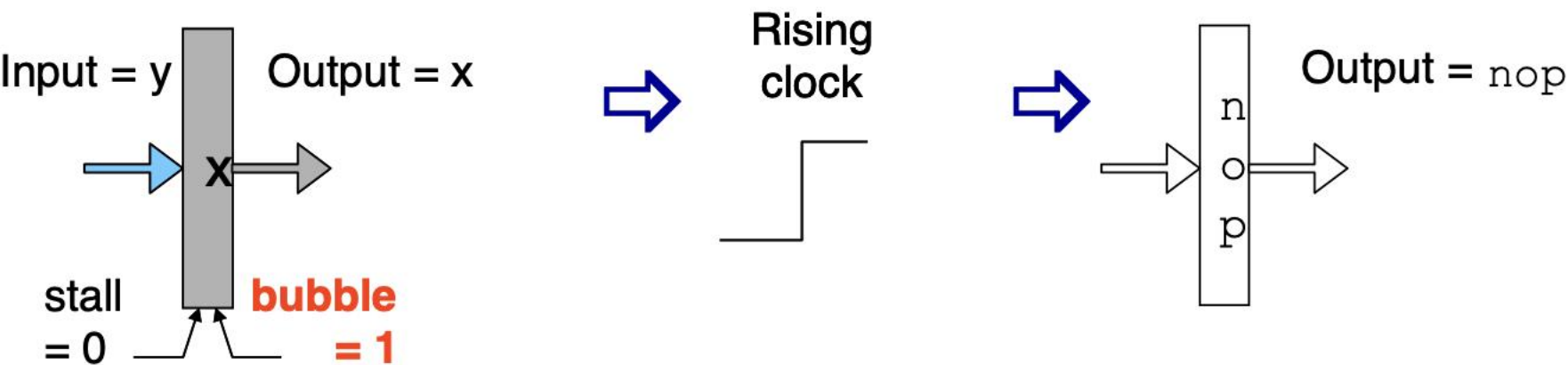
Normal



Stall

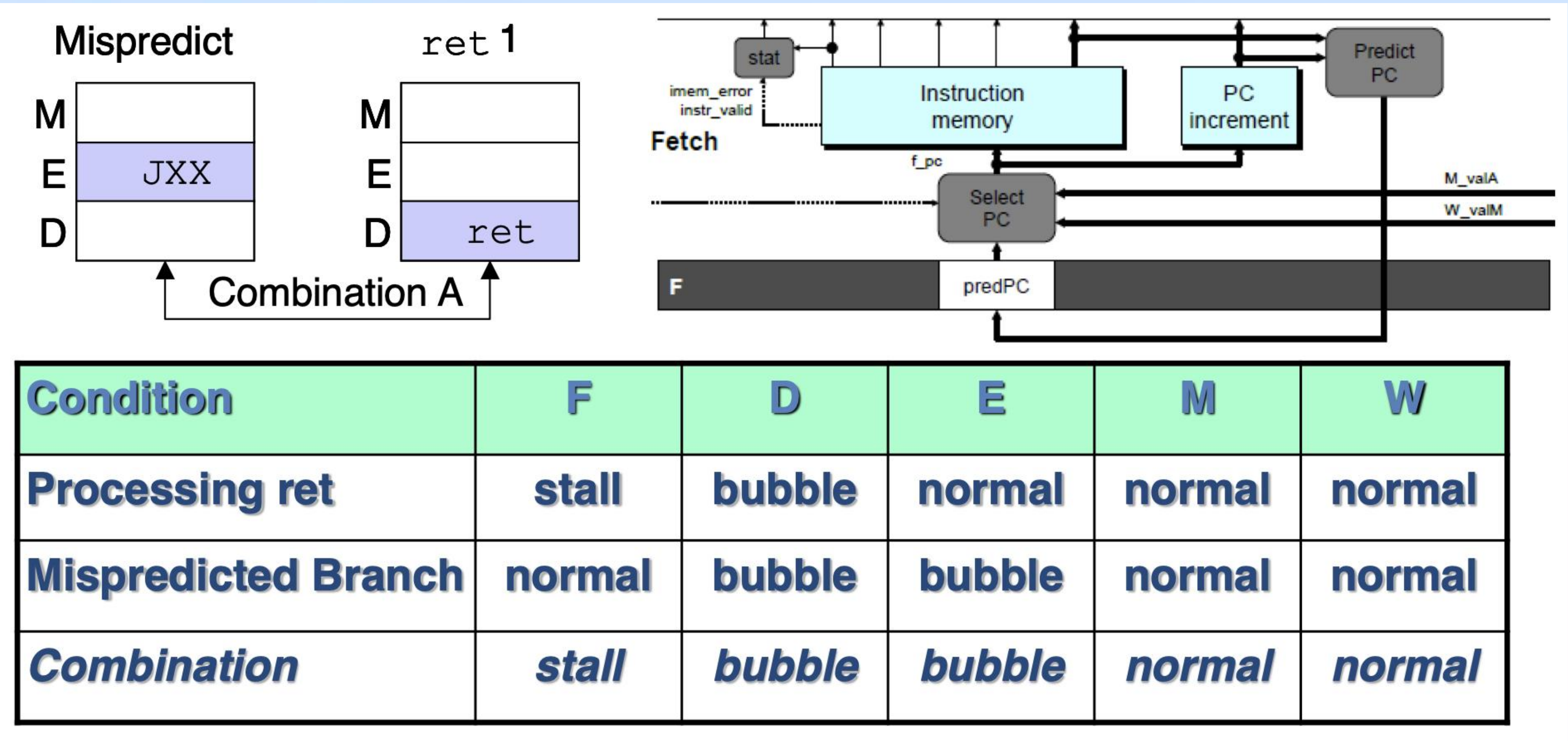


Bubble



控制条件的组合

组合 A (jxx+ret)



控制条件的组合

组合 B（加载/使用+ret）



Condition	F	D	E	M	W
Processing ret	stall	bubble	normal	normal	normal
Load/Use Hazard	stall	stall	bubble	normal	normal
Combination	stall	bubble + stall	bubble	normal	normal

✦ 谢谢！