

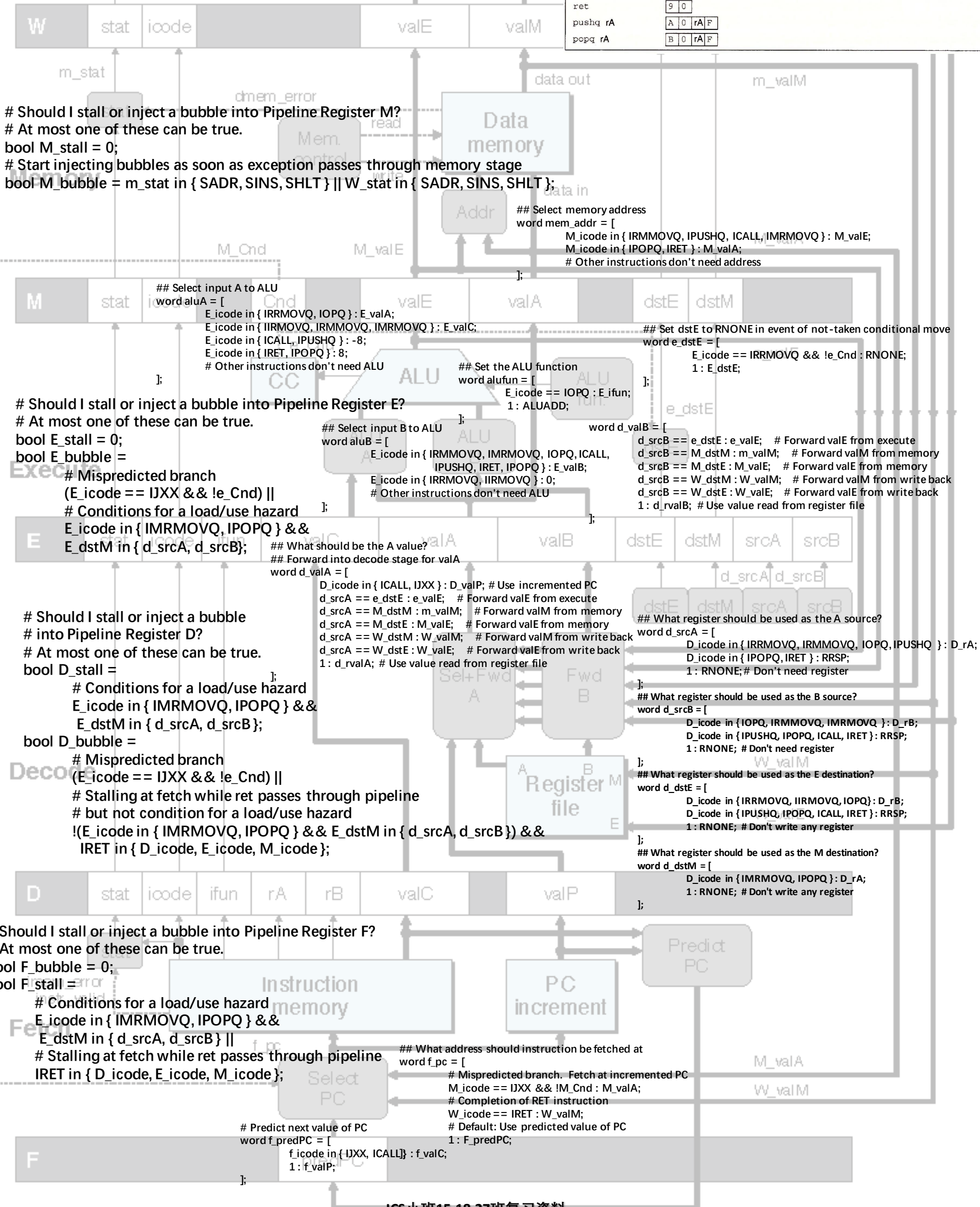
## 重点题型C: PIPE HCL语言补全

Write

# Should I stall or inject a bubble into Pipeline Register W?

bool W\_stall = W\_stat in { SADR, SINS, SHLT };

bool W\_bubble = 0;



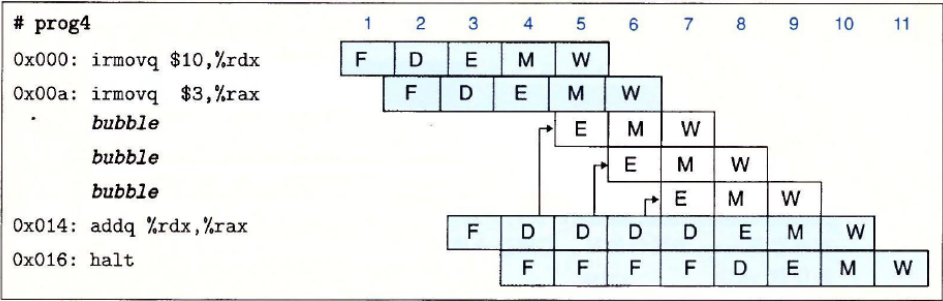
重点题型A: SEQ 跟踪指令的执行

Stage	OPq rA, rB	rrmovq rA, rB	irmovq V, rB	Stage	rmmovq rA, D(rB)	mrmovq D(rB), rA
Fetch	icode:ifun ← M <sub>1</sub> [PC] rA:rB ← M <sub>1</sub> [PC + 1]  valP ← PC + 2	icode:ifun ← M <sub>1</sub> [PC] rA:rB ← M <sub>1</sub> [PC + 1]  valP ← PC + 2	icode:ifun ← M <sub>1</sub> [PC] rA:rB ← M <sub>1</sub> [PC + 1] valC ← M <sub>8</sub> [PC + 2] valP ← PC + 10	Fetch	icode:ifun ← M <sub>1</sub> [PC] rA:rB ← M <sub>1</sub> [PC + 1] valC ← M <sub>8</sub> [PC + 2] valP ← PC + 10	icode:ifun ← M <sub>1</sub> [PC] rA:rB ← M <sub>1</sub> [PC + 1] valC ← M <sub>8</sub> [PC + 2] valP ← PC + 10
Decode	valA ← R[rA] valB ← R[rB]	valA ← R[rA]		Decode	valA ← R[rA] valB ← R[rB]	valB ← R[rB]
Execute	valE ← valB OP valA Set CC	valE ← 0 + valA	valE ← 0 + valC	Execute	valE ← valB + valC	valE ← valB + valC
Memory				Memory	M <sub>8</sub> [valE] ← valA	valM ← M <sub>8</sub> [valE]
Write back	R[rB] ← valE	R[rB] ← valE	R[rB] ← valE	Write back		R[rA] ← valM
PC update	PC ← valP	PC ← valP	PC ← valP	PC update	PC ← valP	PC ← valP

Stage	pushq rA	popq rA	Stage	jXX Dest	call Dest	ret
Fetch	icode:ifun ← M <sub>1</sub> [PC] rA:rB ← M <sub>1</sub> [PC + 1]  valP ← PC + 2	icode:ifun ← M <sub>1</sub> [PC] rA:rB ← M <sub>1</sub> [PC + 1]  valP ← PC + 2	Fetch	icode:ifun ← M <sub>1</sub> [PC]  valC ← M <sub>8</sub> [PC + 1] valP ← PC + 9	icode:ifun ← M <sub>1</sub> [PC]  valC ← M <sub>8</sub> [PC + 1] valP ← PC + 9	icode:ifun ← M <sub>1</sub> [PC]  valP ← PC + 1
Decode	valA ← R[rA] valB ← R[%rsp]	valA ← R[%rsp] valB ← R[%rsp]	Decode		valB ← R[%rsp]	valA ← R[%rsp] valB ← R[%rsp]
Execute	valE ← valB + (−8)	valE ← valB + 8	Execute	Cnd ← Cond(CC, ifun)	valE ← valB + (−8)	valE ← valB + 8
Memory	M <sub>8</sub> [valE] ← valA	valM ← M <sub>8</sub> [valA]	Memory		M <sub>8</sub> [valE] ← valP	valM ← M <sub>8</sub> [valA]
Write back	R[%rsp] ← valE	R[%rsp] ← valE R[rA] ← valM	Write back		R[%rsp] ← valE	R[%rsp] ← valE
PC update	PC ← valP	PC ← valP	PC update	PC ← Cnd ? valC : valP	PC ← valC	PC ← valM

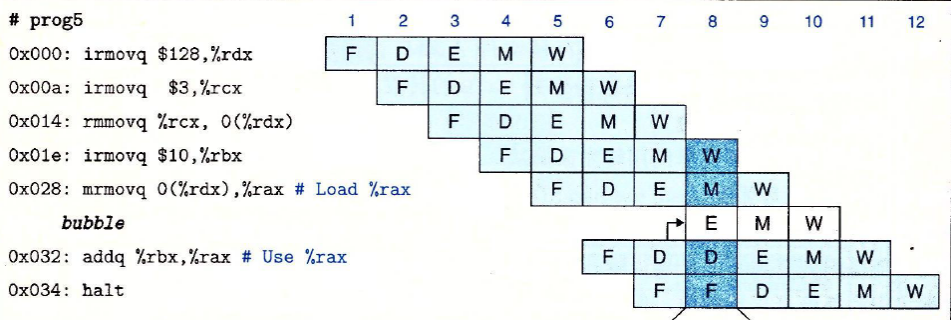
1.暂停避免数据冒险

解决方案：转发避免数据冒险，需要5个转发源(e\_valE, M\_valE, W\_valE, m\_valM, W\_valM)和2个转发目的(ValA, Val B)



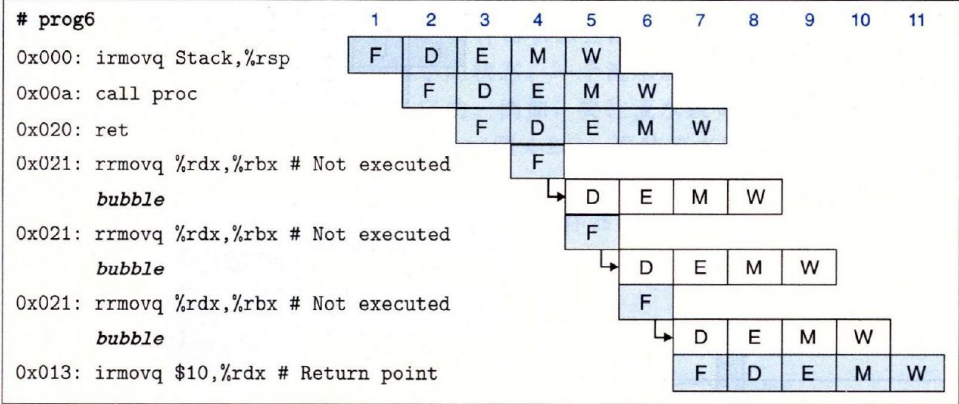
2.加载/使用冒险(属于数据冒险)

解决方案：加载互锁,1个气泡



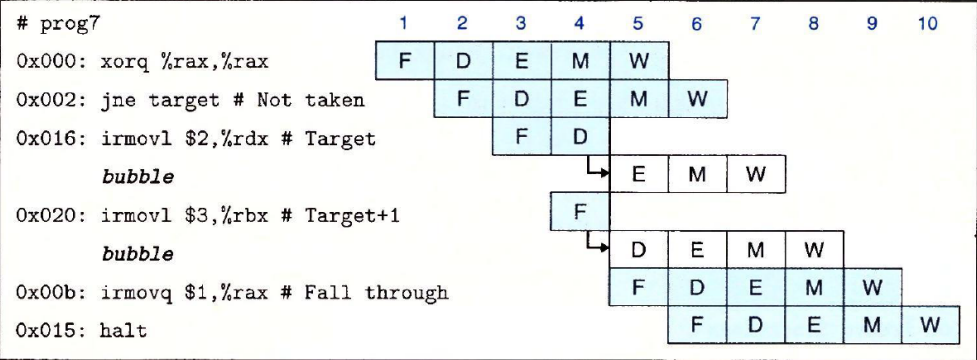
3.处理ret(属于控制冒险)

解决方案：3个气泡



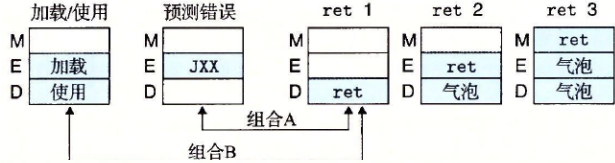
4.预测错误的分支(属于控制冒险)

解决方案：2个气泡



5.小结和多种冒险的组合

条件	流水线寄存器				
	F	D	E	M	W
处理 ret	暂停	气泡	正常	正常	正常
加载/使用冒险	暂停	暂停	气泡	正常	正常
预测错误的分支	正常	气泡	气泡	正常	正常



条件	流水线寄存器				
	F	D	E	M	W
处理 ret	暂停	气泡	正常	正常	正常
预测错误的分支	正常	气泡	气泡	正常	正常
组合	暂停	气泡	气泡	正常	正常

条件	流水线寄存器				
	F	D	E	M	W
处理 ret	暂停	气泡	正常	正常	正常
加载/使用冒险	暂停	暂停	气泡	正常	正常
组合	暂停	气泡+暂停	气泡	正常	正常
期望的情况	暂停	暂停	气泡	正常	正常