

Problem 1

Determine status code for fetched instruction

```
Int f_stat = [  
    imem_error : SADR;  
    !instr_valid : SINS;  
    f_icode == IHALT : SHLT;  
    1 : SAOK;  
];
```

Problem 2

```
bool E_bubble =  
    # Mispredicted branch  
    (E_icode == IJXX && !e_Cnd) ||  
    # Conditions for a load/use hazard  
    E_icode in { IMRMOVL, IPOPL } &&  
    E_dstM in { d_srcA, d_srcB };
```

Problem 3

If the two cases were reversed, then the write back from M_valE would take priority, causing the incremented stack pointer to be passed as the argument to the rrmovl instruction. This would not be consistent with the convention for handling popl %esp determined in section 4.1.6 in ICS book.

```
irmovl $1, %edx
addl %eax, %edx
irmovl $2, %ecx
subl %edx, %ecx
je loop
ret
```

1	2	3	4	5	6	7	8	9	10	11
F	D	E	M	W						
	F	D	E	M	W					
		F	D	E	M	W				
			F	D	E	M	W			
				F	D	E	M	W		
					F	D	E	M	W	
						F	D	E	M	W

E:*e_valE* = 1
e_dstE = \$edx

D:srcA=%eax
srcB=%edx
valA=0
valB=*e_valE*=1

W:w_dst=%edx
w_valE=1

M:m_dstE=%edx
M_valE=1

E:E_dstE=%ecx
e_valE=2

D:srcA=%edx
srcB=%ecx
valA=M_valE=1
valB=e_valE=2