Homework 14

Problem 1

```
#demo-retB.ys
0x000:        irmovl Stack,%esp # Intialize stack pointer
0x006:        call p # Procedure call
0x00b:        irmovl $5,%esi # Return point
0x011:        halt
0x020: .pos 0x20
0x020: p:     irmovl $-1,%edi # procedure
0x026:        ret
0x027:        irmovl $1,%eax # Should not be executed
0x02d:        irmovl $2,%ecx # Should not be executed
0x033:        irmovl $3,%edx # Should not be executed
0x039:        irmovl $4,%ebx # Should not be executed
0x100: .pos 0x100
0x100: Stack: # Stack: Stack pointer
```

In the above example, when **ret** at 0x026 is in its D, E, M stages, there would be **stall** at F stage and **bubble** at D stage. Explain why this happens and the difference between stall and bubble?

Problem 2

Suppose the order of the third and fourth cases (the two forwarding sources from the memory stage) in the HCL code for d_valA were reversed, as follows in table 1. Describe the resulting behavior of the rrmovl instruction(line 5) for the following program:

```
Int d_valA = [
      D_icode in { ICALL, IJXX } : D_valP;
      d_srcA == e_dstE : e_valE;
      d_srcA == M_dstM : m_valM;
      d_srcA == M_dstE : M_valE;
      d_srcA == W_dstM : W_valM;
      d_srcA == W_dstE : W_valE;
      1 : d_rvalA;
];
```

```
1       irmovl $5, %edx
2       irmovl $0x100, %esp
3       rmmovl %edx, 0(%esp)
4       popl %esp
5       rrmovl %esp, %eax
```

Problem 3

For each of the instruction that would cause exception, specify its cause (halt, bad address or invalid instruction) and the stage where exception can be detected, otherwise leave blank.

Instruction	Cause	Stage
halt		
.byte FE		
jmp $-100		
rmmovl %eax,0x10000(%eax) where %eax=$100		
xor %eax, %eax jne t ret t: .byte 0xFF		

Problem 4

Suppose we have the following statistics for a certain program:

1) Fraction of instructions that are loads: 15%
2) Fraction of load instructions requiring stall: 20%
3) Number of bubbles injected each time: 1
4) Fraction of instructions that are cond. Jumps: 35%
5) Fraction of cond. jumps mispredicted: 40%
6) Number of bubbles injected each time: 2
7) Fraction of instructions that are returns: 1%
8) Number of bubbles injected each time: 3

Calculate CPI.