Homework 3

Problem 1

Suppose the CPU is pipelined with Data Forwarding, there are five instructions:

```
irmovl $0,%eax
popl    %eax
popl    % ebx
addl    %eax, %ebx
pushl    %eax
```

Draw the pipelined diagram of these instructions, and point out forwarding in the diagram.
(1)Initially, all registers have value $0.
(2)Specify the src and dest of each forwarding value explicitly.
(3)Mem[%esp] = 1,Mem[%esp+4] = 2.

Problem 2

```
#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 will be **stall** at F stage and **bubble** at D stage. Explain why this happens and the difference between stall and bubble?

Problem 3

For each of the instruction that will cause exception, specify its cause(halt, bad address or invalid) and the stage where exception can be detected, otherwise leave blank. (%eax = $100)

Instruction	**Cause**	**Stage**
halt		
.byte FF		
jmp $-1		
rmmovl %eax,0x10000(%eax)		
Xor %eax, %eax jne t t: .byte 0xFF		

Problem 4

Suppose we have the following statistics for a certain program:
(1)Fraction of instructions that are loads: 10%
(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 mis-predicted: 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 the CPI.