Fine print:
Quiz is closed-book, no calculators; covers Interrupts and Real Time, Semaphores, and the Pipelined Beta -- up to L23 (Pipelined Beta II)/R21 (this recitation)

Practice, practice, practice:
Follow “Previous terms” link from http://6004.csail.mit.edu, pick a semester (the more recent, the better), click on the “Announcements” page for the semester, and find the PDF for Quiz 4 and 5 solutions. NOTE: We covered material in different order this year, skipped over some subjects, and focused more on others. Other Note: The Fall ’05 Quiz 5 is unlikely to be representative of this year’s Quiz 5; it would be good to go over, but it was an unusually hard Quiz 5.

Another perspective on the material – Margaret Chong’s Handbook:
Follow “Handouts” link from http://6004.csail.mit.edu, click on handbook link near the bottom of the page.

Handouts
The “Handouts” page also brings you to a copy of 5-stage Pipelined Beta. You should understand the significance of all additions made since the Unpipelined Beta.

Good topics to know:
Interrupts and Real Time
Interrupt latency: how much time is allowed to elapse between interrupt request and start of handler?
Priorities
Weak/non-preemptive: Nobody can interrupt the interrupt handler
Strong: Certain devices/events can interrupt lower-priority handlers
Semaphores
Operations:
wait(semaphore s)
    stall current process if s<=0, otherwise s=s-1
signal(semaphore s)
    s=s+1 (can let other processes proceed)
Deadlock
Each process is waiting for other processes to release resource(s)
Avoid (e.g., by ranking resources) or detect
**Pipelined Beta**

**Stages**

- **IF**
- **RF**
- **ALU**
- **MEM**
- **WB**

*Instruction Fetch stage:* Maintains PC, fetches one instruction per cycle and passes it to

*Register File stage:* Reads source operands from register file, passes them to

*ALU stage:* Performs indicated operation, passes result to

*Memory stage:* If it’s a LD, use ALU result as an address, pass mem data (or ALU result if not LD) to

*Write-Back stage:* Writes result back into register file.

**Hazards**

**Data Hazards**

- Problem: Instruction reading from a register with outdated value (new value is still in the pipeline)
- Solution: Bypass from stage of appropriate instruction to RF stage (note, may have to bypass PC+4 or data from memory)

*(Load Hazards)*

- Problem: “Appropriate instruction” (from which you need register data) is a LD that has not yet retrieved data from memory
- Solution: Stall instruction requesting the new register value until LD reaches WB stage (*i.e.*, until data is ready)

**Control Hazards**

- Problem: Instruction fetched, but should not be executed (instruction in RF stage branches, or interrupt received)
- Solution: Replace instruction with NOP (for a branch) or with BNE(R31,...,XP) (for interrupt)

**Faults** – can happen at any stage of the pipeline; need to replace all instructions following the fault with NOPs
Problem 1
(B) Which, if any, of the following factors contribute to the performance advantage of the pipelined Beta over the unpipelined version?

- Fewer clock period per instruction
- Higher clock frequency

(C) (5 points) The pipelined Beta is modified to include an additional MEM2 pipeline stage (for a total of 6 stages), to accommodate slower, pipelined memories. Relative to the 5-stage pipelined Beta, which of the following will be required as a result?

- More bypass paths: YES: ____; or NO: ____
- Additional branch delay slots after each BR: YES: ____; or NO: ____
- An additional multiplexor for instruction annulment: YES: ____; or NO: ____
- Additional pipeline stall cycles during execution of certain programs: YES: ____; or NO: ____

(for situations that can't be cured via bypasses)

Problem 3 (7 Points): Pipelined Beta

The following program fragment is executed on our standard, 5-stage pipelined Beta whose diagram is attached to this quiz:

```
ADDC(R31, 4, R0)
SUBC(R0, 3, R2)
MUL(R0, R1, R2)
BR(XXX, LP)
XOR(R2, R1, R1)
XXX: ST(LP, 0x100, R0)
```

(A) (1 point) What memory location is written by the ST instruction at XXX?

Address of ST instruction (HEX): 0x_______

The ADDC instruction is fetched during clock cycle i. The following questions deal with control signals during subsequent clock cycles, namely clock cycles i+1 through i+6. You may find the scratch pipeline diagrams attached at the end of the quiz useful in answering these questions.

(B) (2 points) During clock cycle i+3, which (if any) bypass path is selected for the A and B operands?

Enter one of RF (for no bypass), ALU, MEM, WB, PC^ALU, or PC^MEM or don't care (if it doesn't matter).

Bypass source for A operand (RF for none):

Bypass source for B operand (RF for none):
Quiz #5 Spring 2003

Process A          Process B

int Y;
wait(S);
A1:   Y=X*2;
A2:   X=Y;
signal(S)

int Z;
wait(S);
B1:   Z=X+1;
B2:   X=Z;
signal(S)

S is set to 1 and X is set to 5 prior to execution; the processes are run in parallel.
List all possible final values for X