CS/ECE 552: Introduction to Computer Architecture
Department of Computer Sciences
University of Wisconsin-Madison

Prof. Karthikeyan Sankaralingam

Mid-term Examination
In-Class
March 13, 2008
Approximate Weight: 25%

1 hour 15 minutes

CLOSED BOOK. You can bring one-cheat sheet (two-sides 8.5 x 11 page).

Exam is **one-sided, total of 7 numbered pages**. Plan your time carefully.
One blank page included in the end for rough work. Also use left-side blank pages for rough work.

NAME: _____

Email: _____

Problem number	Maximum points	Actual points
1	8	
2	8	
3	15	
4	30	
5	25	
Total	**86**	

Problem 1 (8 points)

Program A runs for 1,000,000 instructions and executes a branch every 11 instructions (i.e., averages 10 sequential instructions and one branch every 11 instructions). Program B runs for 3,000,000 instructions and branches every 5 instructions. Consider a workload where A and B are run equally often. Write an expression for the average number instructions per branch.

Problem 2 (8 points)

Consider a workload with 15% stores, 25% loads, 20% branches, and 40% other integer instructions.
Consider two machines BASE and NEW.
On BASE, let stores take *1 cycle*, loads *2 cycles*, branches 3 cycles, and others 1 cycle.
On NEW, let stores take *2 cycles*, loads *1 cycle*, branches 3 cycles, and others 1 cycle.
In this case, the speedup of NEW with respect to BASE is S. **Write an expression for S.**

Problem 3 (15 points)

Part A:

Indicate the true dependences in the following MIPS code

```
sw    $3, 0($2)
add   $3, $3, $1
lw    $1, 0($3)
add   $4, $3, $1
lw    $4, 0($2)
```

Part B:

What are structural hazards, data hazards, and control hazards?

Part C:

Explain three techniques to solve structural hazards

Problem 4: (30 points)

Short answers: Any assumptions you need to make if the question is ambiguous please state in your work. Instructor will not answer clarifications for this problem!

1) How does pipelining speedup execution of a program? How is the throughput of a pipeline related to latency of execution of tasks in the pipeline? (5 points)

2) Assume we make an enhancement to a computer so that some mode of execution is improved by a factor of 5. (10 points)

a) Under what circumstances will overall program speedup be 3?

b) If the enhanced mode contributes to 50% of the time (and this 50% is percentage of the total execution time when the enhanced mode is used), what is the overall speedup? What percentage of the original execution time is converted to the faster mode?

3) Consider a server computer using a 1 GHz Intel PentiumIII processor versus another server computer using a 667 MHz Compaq Alpha 21264 processor. Assume that benchmark results show that the Alpha-based system is faster than the PentiumIII-based system running SPEC95 integer benchmarks. In the space below (you may not use any additional space!), propose at least two different explanations why the system with slower clock speed might be faster. (5 points)

4) Pipelining improves performance by decreasing latency of execution of each instruction. TRUE or FALSE? (2.5 points)

5) A barrel shifter to shift a 32-bit input number left by up to 31 bit positions has 8 stages of 2-1 multiplexors. TRUE or FALSE? (2.5 points)

6) One of the main reasons to have general purpose registers in a CPU is to save main memory bandwidth. TRUE or FALSE? (2.5 points)

7) Consider two computers A and B, running a program P. It is possible for computer A to have a lower execution time than computer B, if computer A executes MORE instructions than B. TRUE or FALSE? (2.5 points)

Problem 5: (25 points)

High performance datapaths use bypass paths (also known as data forwarding logic) to reduce pipeline stalls. However, bypass paths are relatively expensive, especially in some wire constrained technologies. To reduce the cost (and potential cycle time impact), some architects have explored omitting some of the possible bypass paths. Consider the datapath illustrated above (note that the PC update logic and all control logic is intentionally omitted). This pipelined datapath is similar to the one in the book, *but only has bypass paths on one side of the ALU.* Assume that the register file internally bypasses the value, so that if register $i is read and written in the same cycle, then the read returns the new value. Assume that the control logic bypasses the data as soon as possible using the given forwarding data paths, and stalls in decode otherwise. You may NOT add additional data paths.

In this problem, you will look at how a program snippet performs on this pipeline. Recall that R-format instructions have the form:

```
opcode rd, rs, rt
```

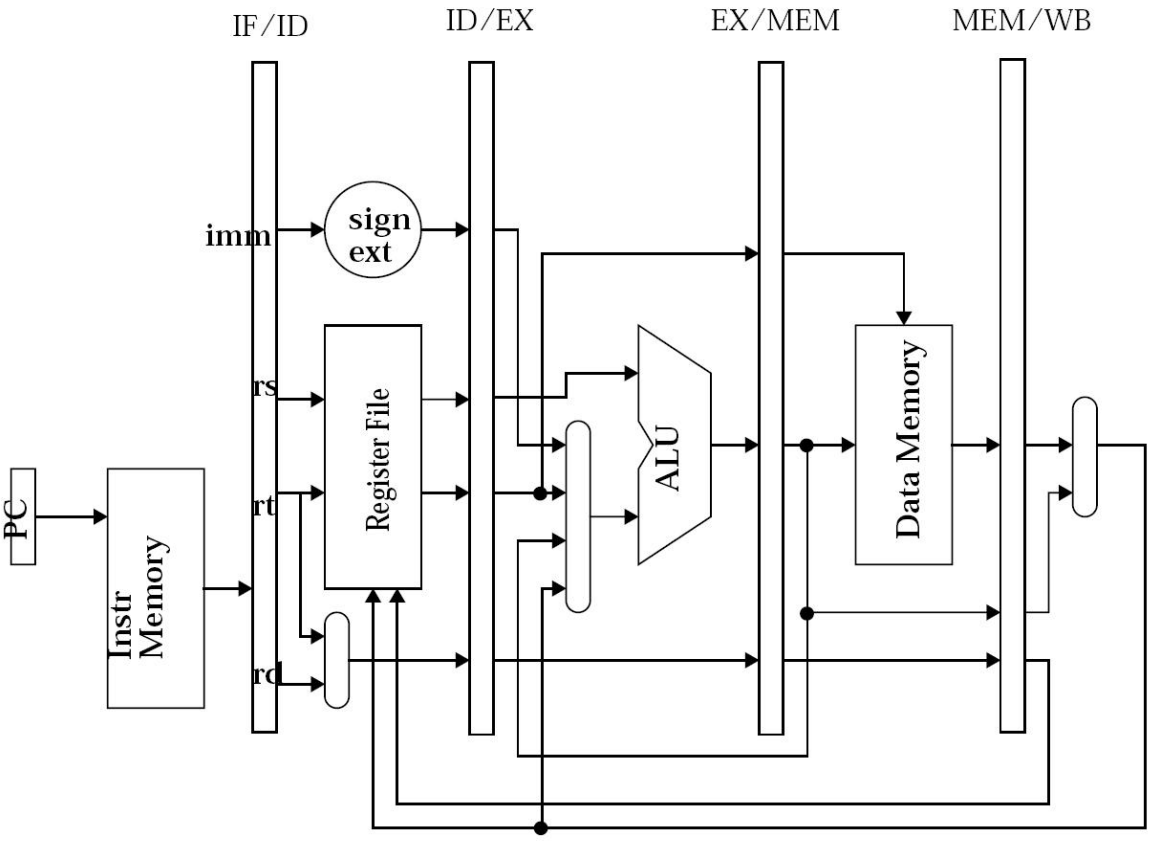
and I-format instructions have the form

```
opcode rt, imm(rs)
```

or

```
opcode rt, rs, imm
```

Use the table on the next page to show how the given instruction sequence flows through the pipeline and where stalls are necessary to resolve hazards.



	Cycle																			
Instruction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
add $1, $2, $3	F	D	X	M	W															
sub $4, $1, $5		F	D																	
or $6, $1, $4																				
and $7, $4, $8																				
lw $9, 4($7)																				
add $1, $9, $2																				
sw $1, 4($7)																				

Consider the code and pipeline above. Show the execution of this code on the pipeline above.
Use the letters, F, D, X, M, and W.

For each cycle where a stall occurs explain why.