

Department of Electrical and Computer Engineering  
The University of Texas at Austin

EE 382N, Spring 2008  
Y. N. Patt, Instructor  
Rustam Miftakhutdinov and Aater Suleman, TAs  
Exam 1, March 24, 2008

Name : \_\_\_\_\_

Problem 1 (12 points): \_\_\_\_\_

Problem 2 (12 points): \_\_\_\_\_

Problem 3 (12 points): \_\_\_\_\_

Problem 4 (12 points): \_\_\_\_\_

Problem 5 (12 points): \_\_\_\_\_

Problem 6 (12 points): \_\_\_\_\_

Problem 7 (12 points): \_\_\_\_\_

Problem 8 (12 points): \_\_\_\_\_

Problem 9 (12 points): \_\_\_\_\_

Problem 10(12 points): \_\_\_\_\_

Bonus for legibility on

all answers (4 points): \_\_\_\_\_

Total (100 points): \_\_\_\_\_

Directions: The first problem of this exam is a required problem. You may answer any 7 of the last 9 problems. Place an "X" in the 2 lines above for the 2 problems that you choose not to answer.

Note: Please be sure that your answers to all questions (and all supporting work that is required) are contained in the space provided.

Note: Please be sure your name is recorded on each sheet of the exam.

**GOOD LUCK!**

Name: \_\_\_\_\_

**Problem 1 - Required (12 points):** A Texas A&M graduate student implemented the register dependency check logic for a machine which can fetch/decode/issue up to two instructions each cycle. He elected to use a scoreboard which contains one bit for each architectural register. He adopted the convention: set the bit in the third stage of the pipeline (register dependency check and operand access) when an instruction uses a register as a destination register, and clear the bit when the register is written with the result produced by that instruction. The register dependency check logic should stall the pipeline when necessary to protect against incorrect execution. Unfortunately, he made a mistake.

**Your job:** Fix his verilog code. Please make the change inside the box. You can assume that the instruction in pipeline0 is always older (in terms of program order) than the instruction in pipeline1. The code the Aggie wrote is reproduced below:

```
module dep_check(out_stall_0,      //output -- stall pipeline0
                out_stall_1,      //output -- stall pipeline1
                instr0_src1[2:0], // input -- source operand for pipe0 instruction
                instr0_src2[2:0], // input -- source operand for pipe0 instruction
                instr1_src1[2:0], // input -- source operand for pipe1 instruction
                instr1_src2[2:0], // input -- source operand for pipe1 instruction
                instr0_dest[2:0], // input -- dest operand for pipe0 instruction
                instr1_dest[2:0], // input -- dest operand for pipe1 instruction
                SB_out[7:0],      // input -- bits from scoreboard, one for each register
                clk
                );
    ... //port declarations

    wire instr0_src1_status, instr1_src1_status;
    wire instr0_src2_status, instr1_src2_status;
    wire instr0_dest_status, instr1_dest_status;

    mux8$ i0s1mux(instr0_src1_status, SB_out, instr0_src1);
    mux8$ i0s2mux(instr0_src2_status, SB_out, instr0_src2);
    mux8$ i0dmux(instr0_dest_status, SB_out, instr0_dest);

    or3$ or3gate0(out_stall_0, instr0_src1_status, instr0_src2_status, instr0_dest_status);

    mux8$ i1s1mux(instr1_src1_status, SB_out, instr1_src1);
    mux8$ i1s2mux(instr1_src2_status, SB_out, instr1_src2);
    mux8$ i1dmux(instr1_dest_status, SB_out, instr1_dest);

    or3$ or3gate1(out_stall_1, instr1_src1_status, instr1_src2_status, instr1_dest_status);

endmodule

//mux8$ selects one of the 8 data lines based on the sel signal
module mux8$(out, // output
            data[7:0], //the 8 signals from which to select
            sel[2:0]
            );
    ...
```

Name: \_\_\_\_\_

**Problem 2 (12 points):** The performance equation has three factors: length, CPI, and cycle time. When comparing the AMD Barcelona product to the Intel Pentium IV product, we can simplify the performance equation. Explain.

Name: \_\_\_\_\_

**Problem 3 (12 points):** An IA 64 instruction bundle consists of three 41-bit instructions, packaged in a 128-bit unit. What are the extra five bits used for? What value do they provide over previous designs (VLIW) by the same architects?

Name: \_\_\_\_\_

**Problem 4 (12 points):** Performance is enhanced by removing bottlenecks from the critical path.

**Part a:** “Loads” are always on the critical path. Explain.

**Part b:** In what situations are “stores” also on the critical path. Explain.

Name: \_\_\_\_\_

**Problem 5 (12 points):** The Block-Structured ISA is fundamentally different from the Superblock. How so?

Name: \_\_\_\_\_

**Problem 6 (12 points): Part a:** What benefit does “inactive issue” provide to a microarchitecture that has a trace cache?

**Part b:** What property of trace cache must be implemented in order for inactive issue to even be possible? Explain.

Name: \_\_\_\_\_

**Problem 7 (12 points):** A customer at your neighborhood microarchitecture store needs advice in choosing the parts to build a processor. **Your job:** Choose the parts which provide the highest performance on his software, keeping in mind that money should not be spent unnecessarily.

You have the following parts available in your store:

Pipelines (All run at 2GHz and include an L1 cache)	in-order (\$2) out-of-order with 128-entry ROB (\$8) out-of-order with 256-entry ROB (\$16)
Branch Predictors	2-bit counter (Free) Gshare (\$1) Perceptron (\$2)

Note: memory latency is 400 processor cycles.

**Part a:** The customer wants to run a program which computes a dot product of two vectors A and B.

```
dot-product
  sum = 0
  for I = 1 to 1000000
    sum = sum + A[I]*B[I]
```

Specify which components you will pick and why.

**Part b:** Another customer wants to run a program which counts the nodes in a singly-linked list.

```
count
  ptr = HeadPtr
  count = 0
  while(ptr != NULL)
    count = count + 1
    ptr = ptr->NextPtr
```

Specify which components you will pick for this customer and why.



Name: \_\_\_\_\_

**Problem 8 (12 points):** Sometimes a major improvement in microarchitecture is simply due to good fortune. What do we mean by that? Give an example of a major improvement that is simply due to good fortune.

Name: \_\_\_\_\_

**Problem 9 (12 points):** In class, almost every day, we have provided references to additional material on the topic of the day. Identify one reference that you have looked up after class and give one interesting thing you got from the paper above and beyond what we covered in class.

Name: \_\_\_\_\_

**Problem 10 (12 points):** The Pentium microarchitecture introduced a split I-Cache, split in the sense that although the line size was 16 bytes, and each I-Cache fetch produced 16 bytes of information, one could obtain the last eight bytes of one line and the first eight bytes of the next line, as well the entire 16 bytes of a single line, depending on the address one uses to access the I-Cache. Why do you think the designers introduced this complication, and do you think it was a good idea?