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

EE 382N, Spring 2012
Y. N. Patt, Instructor
Veynu Narasiman and Carlos Villavieja, TAs
Exam 1, March 28, 2012

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 exam contains ten problems. You are required to do the first two and 6 of the remaining 8. Note this cover page provides space for recording the grades for all ten problems. Please put an "X" in the space provided for the two problems you do not want me to grade. All problems are equally weighted.

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): You are part of a team designing the memory system of an in-order processor. Specifically, you are responsible for the module that checks if a memory instruction in the read stage needs to be stalled because there is an address dependency with a memory instruction in the later write back stage.

Implement the following module in structural Verilog:

```
module  MemoryDependency(
         RDstall,         // output -- the read instruction needs to be stalled
         RDaddress[31:0], // input  -- Read address
         RDsize[1:0],     // input  -- Read memory size, 00 = 1 byte, 01 = 2 bytes,
                                        10 = 4 bytes, 11 = 8 bytes
         WRaddress[31:0], // input  -- Write address
         WRsize[1:0],     // input  -- Write memory size, 00 = 1 byte, 01 = 2 bytes,
                                        10 = 4 bytes, 11 = 8 bytes
         );
```

You can use the following elements from the team library:

```
mux4_3 (out,in0,in1,in2,in3,sel0,sel1);  // 3 bit Multiplexer 4 to 1
add32 (out,in0,in1);                     // 32 bit adder
Any basic logic gate (inverter, or, ...) you need.
```

Name:________________________________

Name:________________________________

Problem 2 (12 points):
Assemble the following x86 instructions into hexadecimal format (e.g: 80 73 AB ...) and store the code in memory locations starting at byte-addressable location hex 1000. You may not need all the locations provided. If you think an instruction can be assembled more than one way, choose the way that uses fewer bytes.

In the pseudo assembly code below, the first operand is the destination. The [ ] indicate a memory location. Immediates and displacements begin with the $ symbol.

```
OR [(ES<<16) + EDX + (ECX*4) + $0x12345678], $0x87654321
MOV AX, $0xAB87
ADD AL, $0x49
DAA
```

1000	
1001	
1002	
1003	
1004	
1005	
1006	
1007	
1008	
1009	

100A	
100B	
100C	
100D	
100E	
100F	
1010	
1011	
1012	
1013	

1014	
1015	
1016	
1017	
1018	
1019	
101A	
101B	
101C	
101D	

What is the final value of AL? □

Name:________________________________

Problem 3 (12 points): In class, I have given you more than 50 references to look up additional material with respect to specific topics. Listed below are four of them, with a question or two regarding each. Select one of the four, and answer the question(s) asked in the space provided.

(1) Pan, So, Rahmeh, "Improving the accuracy of dynamic branch prediction using branch correlation," ASPLOS, 1992.

Question: What does the author mean by "correlation"?
Question: In class, I talked about nine different two-level branch predictors: GAg, GAs, GAp, SAg, SAs, SAp, PAg, PAs, PAp. Which of the nine was introduced by Pan et.al.? Explain what the three letters in Pan's predictor signify.

(2) Kim, Mutlu, Stark, Patt, "Wish Branches: Combining Conditional Branching and Predication for Adaptive Predicated Execution," Micro 2005.

Question: The paper defines three types of wish branches. What are they and how are they used?

(3) Chappell et.al., "Simultaneous Subordinate Microthreading (SSMT), ISCA 1999.

Question: The paper describes three obvious uses for SSMT. What are they?

(4) Burger, et. al., "Scaling to the End of Silicon with EDGE Architectures," IEEE Computer, 2004.

Question: This paper describes the work done in the Computer Science Department on the TRIPS project. The ISA defines instructions. What is the most important characteristic of an instruction in this ISA? Explain.
Question: Does the paper suggest anything about the amount of work performed by a single instruction? What mechanism is used to obtain the desired amount of work in a single instruction?

ANSWER: Name of paper:________________________________
Commentary:

Name:________________________________

Problem 4 (12 points): In our discussion of the Block Structured ISA, I remarked that in an expanded block, ”instructions” contained two types of branches: faulting branches and trapping branches.
Their most important difference is that faulting branches

,

while trapping branches

.

Show a diagram (like the one in class) of a program fragment made up of blocks that are not expanded, and what that same fragment looks like after the compiler produces expanded blocks. Identify a faulting branch and a trapping branch.

Name:________________________________

Problem 5 (12 points): The enhanced trace cache derived some performance benefit from Branch Promotion. Explain what this term means, and why you think it provided performance benefit.

Problem 6 (12 points): The HEP computer designed by Burton Smith allowed multiple threads to execute concurrently. What, if anything, did it use for a branch predictor? Explain.

Name:________________________________

Problem 7 (12 points): Competition among manufacturers of products with different ISAs in the early 1990s classified their products as either brainiacs or speed demons. This classification related to whether the product was designed to optimize on

in the case of speed demons, or on

in the case of brainiacs.

In the case of the trace cache, one could argue that Jim Smith and I approached the design issue from different perspectives, one a brainiac, one a speed demon. Which of us was which? Explain.

Name:________________________________

Problem 8 (12 points): In order to exploit performance we need to bring lots of instructions into the execution core, we need to make the required data available as quickly as possible, and we need an execution core that does not wait unnecessarily to execute ready instructions.

To obtain the required level of instruction supply, three things are needed. What are they, and how do we go about satisfying each need?

Name:________________________________

Problem 9 (12 points): I suggested three important design principles that every microarchitect needs to pay attention to:

Critical Path Design
Bread and Butter Design
Balanced Design.

Please select two of the three, describe the design principle, and explain why it is important to pay attention to.

Name:________________________________

Problem 10 (12 points): A load instruction misses in a write back cache. The replacement policy selects a dirty block to replace to free up space for the line required by the load instruction. Describe the order of activities you would cause to happen to satisfy the cache miss. Your design point is high performance. Please be complete but concise.

Step 1:

Step 2:

Step 3: