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

EE 382N.19, Spring 2018
Y. N. Patt, Instructor
Siavash Zangeneh and Ali Fakhrzadehgan, TAs
Written midterm
March 7, 2018

There are 12 problems on the exam. You are being asked to solve problems 1 and 2, and to select 6 of the remaining 10 problems. That is, you are asked to solve a total of 8 of the 12 problems.

Name :_____

Problem (1):_____

Problem (2):_____

Problem ( ):_____

Problem ( ):_____

Problem ( ):_____

Problem ( ):_____

Problem ( ):_____

Problem ( ):_____

Bonus points for legibility of all answers (4 points):_____

Total (100 points):_____

Please make your handwriting clear, legible, and use a dark enough pen or pencil that it is easily readable. If I can not easily read your handwriting, the answer will be marked wrong. I apologize if you take this as offense, but I think it is unreasonable to scribble on the exam and expect me to squint and struggle to decipher your handwriting.

Note: Be sure to sign your name in the space below to acknowlege that you've read the instructions and will not cheat on the exam. Please be sure your name is recorded on each sheet of the exam.

I WILL NOT CHEAT ON THIS EXAM

Signature:_____

GOOD LUCK!

Name:_____

Problem 1 - Required (12 points): We want to prioritize memory reads over memory writes in a pipelined processor with a single-ported D-cache. We achieve this by inserting memory write requests in a write buffer instead of writing to the D-cache directly. The write requests are only issued to D-cache when D-cache is not busy serving a read request.

The write buffer is an 8-entry FIFO. Write requests are inserted to the entry pointed by the *tail* of the FIFO. The write request pointed by the *head* pointer is issued to D-cache if D-cache is not busy. A 4-bit register keeps track of the current number of entries in the write buffer.

Complete the module "write_buffer_controller" which controls the internal state of the write buffer and its interface to D-cache. The interface to the module is given below

```
module write_buffer_controller(
    input [3:0] curr_size,  // number of valid entries in the write buffer
    input writeback_WE,     // 1 if the write back stage wants to write data to the write buffer
    input dcache_port_busy, // 1 if the D-cache is currently serving a read request
    input dcache_hit,       // 1 if the write request hits in the D-cache
    output inc_head,        // 1 if the head pointer should be incremented
    output inc_tail,        // 1 if the tail pointer should be incremented
    output dcache_WE,       // 1 if the write buffer should issue a write to D-cache
    output buffer_full      // 1 if the write buffer is full
);
```

The only modules you are allowed to use are 4-input OR gates, 4-input AND gates, and inverters, as specified below:

```
module  or4$(out, in0, in1, in2, in3);
module  and4$(out, in0, in1, in2, in3);
module  inv1$ (out, in);
```

Problem 2 - Required (12 points):

Decode the x86 instruction stream shown below in hex into its component instructions.

```
  83 c6 03 52 65 66 0f b1 4c 4b 26 9a 24 13 24 13 24 13
```

'83' is byte 0, and '13' is byte 17. Show the exact address calculation for any memory operand. Use M_32[ ] to indicate a 32 bit memory location. Immediates, displacements, and literals should begin with the $ symbol.

Show below is some example assembly. Please use this as a reference for notation.

```
OR M_32[(ES<<16) + EDX + (ECX*4) + $0x12345678], $0x87654321
MOV AX, $0xAB87
DAA
```

Name:_____

Problem 3 (12 points): HPS built on the out-of-order processor of Tomasulo by adding (1) wide issue, fetching more than one instruction each cycle, (2) branch prediction and as a result speculative execution, (3) in-order retirement, and (4) doing all this for more than just the floating point unit. Which ONE of these additions do you believe is responsible for HPS being adopted by almost all manufacturers, whereas there was no follow-on product of the Tomasulo's idea, even at IBM.

Problem 4 (12 points): Daniel Jimenez' PhD dissertation made a strong case for why branch prediction is a natural application for the machine learning paradigm. What is it about branch prediction and machine learning that makes this so?

Problem 5 (12 points): I have been advocating for decades that professionals operating at each layer of my transformation hierarchy (problems in natural language, algorithms, programs in mechanical languages, ISA, microarchitecture, circuits, electrons) need to be able to talk to professionals at other layers if we are to continue to obtain increased performance. Actually, there are examples of this already being the case in some instances. Give an example of a mechanism that we have discussed in class where the mechanism happened because professionals at more than one layer of the transformation hierarchy were able to communicated with each other.

Problem 6 (12 points): Since a trace cache segment contains up to three basic blocks, and each block ends in a conditional branch, predicting a full trace segment accurately will depend on three conditional branches being predicted correctly. If the probability of predicting correctly is 0.9, one would expect a correct prediction of all three to be 0.9 times 0.9 times 0.9 = 0.729. Is this expectation realistic? Explain.

Problem 7 (12 points): Trace-driven simulation has been used to evaluate the performance benefits of various characteristics of a cache. A good idea or a bad idea? Explain.

Problem 8 (12 points): What is the purpose of Runahead Execution? Hint: none of the results produced while the machine is in Runahead mode are saved for retiring after the microarchitecture comes out of Runahead mode.

Problem 9 (12 points): What is a victim cache? Why is it a good idea? Should such a cache be direct mapped, fully associative, or something in between? Explain.

Problem 10 (12 points): Intel's Pentium chip and DEC's Alpha 21064 were released at approximately the same time. Both were two-wide issue machines. The Pentium had a branch target buffer (BTB), the Alpha 21064 did not. What benefit did the BTB provide. Explain.

Problem 11 (12 points): In class we discussed several fundamental design principles, one of which was "Critical path design." Explain what it means, in the context of specifying the cycle time of your design.

Problem 12 (12 points): In class we showed that we could evaluate a fourth degree polynomial in five units of time if we had three functional units available to us. We could accomplish this by executing three operations in each of the first three units of time, and then one operation in each of the last two units of time. That is, 11 operations, but with three processing elements, only 5 units of time. With one processing unit, no parallelism is possible, so it would take us 11 units of time. Therefore, the speed-up is 11/5 = 2.2. Is it correct, that the speedup is 2.2? Explain.