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

EE 382N, Spring 2014
Y. N. Patt, Instructor
Faruk Guvenilir and Milad Hashemi, TAs
Exam 1, March 5, 2014

Name :______________________________

Problem 1 (10 points):________

Problem 2 (10 points):________

Problem 3 (10 points):________

Problem 4 (10 points):________

Problem 5 (10 points):________

Problem 6 (10 points):________

Problem 7 (10 points):________

Problem 8 (10 points):________

Problem 9 (10 points):________

Problem 10 (10 points):________

Problem 11 (10 points):________

Problem 12 (10 points):________

Problem 13 (10 points):________

Total (100 points):____________

Directions: The exam contains 10 problems. You are required to do #1, #2, and 8 of the remaining 11. Please circle on the cover sheet the 8 questions you are answering.

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: 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 (10 points): You are part of a team designing the pipeline of an in-order processor. Specifically, you are responsible for a 4-entry, 16 bit register file, with one read port and one write port (e.g. the register file can be read from and written to at the same time).

The register file supports six different accesses:

1. A 16 bit read.
2. A 16 bit write.
3. Reading the high byte (bits 15:8). The output data must be produced on the low byte (bits 7:0) of the register file output port.
4. Writing the high byte (bits 15:8). The input data is available on the low byte (bits 7:0) of the register file input port.
5. Reading the low byte (bits 7:0).
6. Writing the low byte (bits 7:0).

Implement the following module in structural Verilog and draw a block diagram of your design:

```
module  RegFile(
RData0 [15:0], // output -- Read Port Data
RAddress0 [1:0], // input -- Register Read Address
RSize0 [1:0], // input -- Register Size 0, 00 = 16 bit, 01 = 8 bit low, 10 = 8 bit high, 11 unused
WData[15:0], // input -- Data to be written
WAddress[1:0], // input -- Register Write Address
WSize [1:0], // input -- Register Write Size 0, 00 = 16 bit, 01 = 8 bit low, 10 = 8 bit high, 11 unused
clk // input -- Clock
);
```

You can use the following elements from the team library. Feel free to use 8-bit and 16-bit muxes as necessary (e.g. mux4_8 or mux4_16).

```
reg8(out[7:0], in[7:0], we, clk) //8-bit register, active high
mux4_1 (out, in0, in1, in2, in3, sel[1:0]);  // 1-bit Multiplexer 4 to 1
mux2_1 (out, in0, in1, sel);  // 1-bit Multiplexer 2 to 1
Any basic logic gate (inverter, or, and, xor) you need.
```

Name:______________________________

Name:________________________________

Problem 2 - Required (10 points):

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

```
04 0F 26 66 0F B1 9C FC 0F 42 87 27 F3 A7 F4
```

For each instruction, include any prefixes. Show the exact address calculation for any memory operand. Use M_32[ ] to indicate a 32 bit memory location. Immediates and displacements should begin with the $ symbol. We have provided room for 7 instructions. There are 7 or fewer instructions in the above code.

1

2

3

4

5

6

7

The operations performed by one of the above instructions can be performed by another instruction in the x86 ISA – that is, a different opcode, and a different format for the instruction.

Which instruction?

What is the other instruction, and how would it be encoded to require the identical operations be performed. Show in hexadecimal.

Name:________________________________

Problem 3 (10 points): Many references have been suggested to you in class for further edification. You are not expected to read all, but I do expect you to look at some. I have listed below four papers. **Pick one**, and answer the accompanying question(s).

(1) M.D. Smith, M Johnson, M.A. Horowitz. Limits on multiple instruction issue. ASPLOS-3, 1989.

Question: The authors claimed that it made no sense to have an issue width greater than how many instructions per cycle width?
Question: On what basis did they make that assertion?

(2) H. Esmaeilzadeh, E, Blem, R St. Amant, K. Sankaralingam, D. Burger, A. Seznec, P. Michaud. Dark Silicon and the End of Multicore Scaling. ISCA, 2011.

Every generation introduces a new term into our computer architecture lexicon. One of the more recent ones is "dark silicon."

Question: What does this term refer to? Do the authors suggest this is a bug or a feature? Why? Does Dr. Patt suggest it is a bug or a feature? Why?

(3) Guriandar S. Sohi et al. Multiscalar Processors. ISCA, 1995.

Question: Describe briefly the fetch mechanism of the Multiscalar Processor. What serious problem is introduced by the instruction ADD R1,R2,R3 followed by ADD R4,R5,R1 in a subsequent fetch? What mechanism does Multiscalar use to guarantee that this sequence executes correctly?

(4) Onur Mutlu, Jared Stark, Chris Wilkerson, and Yale N. Patt. Runahead Execution: An Alternative to Very Large Instruction Windows for Out-of-order Processors. HPCA-9, 2003.

Question: Why does the processor enter runahead mode? When does the processor enter runahead mode?

ANSWER: Name of paper:________________________________
Commentary:

Name:________________________________

Problem 4 (10 points): Pipelines can create bubbles in the presence of conditional branches. Before branch prediction became the answer most processors adopted, several suggestions were put forward, among them the Delayed Branch.

How does the Delayed Branch work? Is it a good idea, or a bad idea? When would it work great? When would it not work great?

Name:________________________________

Problem 5 (10 points): The wish branch provides an option for either predicating or predicting a conditional branch. The wish branch requires activity at compile time and at run time.

What does the compiler do at compile time?

What does the hardware do at run-time? Be complete but be concise.

Name:______________________________

Problem 6 (10 points): The Pentium Pro provided for simultaneous decode of three x86 instructions in a single cycle provided the three instructions obeyed a certain constraint. What was that constraint?

Suppose the compiler could not find three successive instructions obeying that constraint, what would the compiler produce for the static object file?

Name:______________________________

Problem 7 (10 points): The HEP had an 8-stage pipeline that was kept running due to multiple threads with no bubbles.

How is this possible since LD instructons took far more than 8 cycles to complete the load?

How is this possible since Conditional Branch instructions needed the predicate generated by the previous instruction.

Name:______________________________

Problem 8 (10 points): Single thread parallelism started with a simple SISD machine with pipelining. Then came SIMD, VLIW, DAE, HPS, and pure data flow. Complete the table below by explaining what each paradigm added to the previous paradigm.

SIMD	
VLIW	
DAE	
HPS	
Data Flow	

Name:______________________________

Problem 9 (10 points):
The Superblock was invented to solve what problem?

It terminated on a branch. What did Scott Mahlke invent to extend the size of the superblock? Explain how it extended the size of the Superblock. Be complete, but at the same time concise.

Name:______________________________

Problem 10 (10 points):
A microprocessor that implements the GAg two-level branch predictor records the last 10 branches in the branch history register thus: 10011101010001. The next branch that gets processed will be predicted taken or not taken based on what? Please be precise and concise.

Name:________________________________

Problem 11 (10 points):
In response to noting Flynn's bottleneck, some architects designed a front end capable of fetching six instructions each cycle, and decoding six instructions each cycle.

What can you tell me about the format of the instructions in the ISA? How do you know?

They designed the execution core to consist of two ALUs, a LD/ST unit, and a branch resolution unit. Does this design make sense? Explain.

Name:______________________________

Problem 12 (10 points):
I have said many times that computer architecture and microarchitecture and system architecture is all about tradeoffs. Whether or not to allow unaligned accesses to the memory system is an example of tradeoff. Explain.

Name:________________________________

Problem 13 (10 points):
Speculation can be a good thing and speculation can be a bad thing. Explain.

If one wants a 4-way set associative cache, one can either design it, or one can approximate it with way prediction. How does way prediction work? What are the tradeoffs?