Mid-Term Exam — ECEN 350 (SPRING 2017)

March 2, 2017

Name:

UIN:

Sign the following statement:

On my honor, as an Aggie, I have neither given nor received unauthorized aid on this academic work

- This exam booklet should have **11** printed pages (including this one).
- This test has **4** questions.
- This is a *open-book* exam. You are allowed to use any books and notes that you wish.
- You are not allowed to use other people's homework assignments, labs, or exams during this test.
- Note: the exam proctor is not allowed to answer any questions. If you are unclear about a problem, make your best guess and write down any assumptions you make.
- Calculators are permitted, but no communication devices of any kind are allowed.
- Each question is marked with its number of points, use your time wisely.
- Show your answers in the space provided for them. Write neatly and be well organized.
- Show your work if you want to get partial credit.
- The test is due exactly at the end of the class period.
- Write all numbers in either hex or binary, points will be deducted for decimal notation.

Good luck!

Problem	Maximum	Score
1	30	
2	35	
3	15	
4	20	
Total	100	

I. Performance Metrics and Measurement (30 Points)

In this problem you will quantitatively evaluate different processor configurations under typical performance metrics.

For parts A and B of this problem, consider the performance of two x86 processors executing the same program to completion.

- Processor A, produced by AMD, has a clock rate of 2.5 GHz, and has an average CPI for the given program of 2.
- Processor B, produced by Intel, has a clock rate of 3 GHz, and has an average CPI for the given program of 1.5.

A. (5 points)

What is the speedup of processor A versus B.

B. (5 points)

If both machines cost the same amount and you are primarily interested in running this program, which processor should you buy?

For the next four parts to this problem consider a program with the following instruction makeup and CPI on a given machine:

Instruction Type	**Instruction Frequency**	**CPI**
Simple ALU (ADD, SUB, ORR, AND, etc.)	25%	1
Complex ALU (MUL, DIV, FP)	15%	4
Memory operations (LDUR, STUR, etc.)	30%	3
Control flow (CBZ, CBNZ, B, BL, BR, etc.)	30%	4

C. (5 points)

What is the overall average CPI of the program?

D. (5 points)

If the program is 1,000,000 instructions long (dynamic instruction count), and the clock frequency is 2GHz, how long will it take to execute?

E. (5 points)

Imagine you were to design a new processor to execute the given program. You have the option of either spending effort to improve the memory operation performance by 1 cycle, or to improve the performance of control flow instructions by 3 cycles. Which would be the better option? Use speedup to justify your answer.

F. (5 points)

Imagine you could chose to spend effort improving one of the instruction times above (ie. Simple ALU, Complex ALU, etc.). Which instruction class would you choose to improve? What is the maximum performance gain (speedup) to be had (imagine they took 0 time)?

II. COMPILATION AND ASSEMBLY (35 POINTS)

For this problem, consider following C code:

```
int y;
int main() {
  int z = 5
  ...
  y = my_funct(12);
  z = z+y;
  ...
}
int my_funct(unsigned int x) {
  if (x == 0) {
    return 0;
  }
  return (x + my_funct(x>>2));
}
```

A. (5 points)

What value is left in z at the end of `main()`?

B. (10 points)

Compile the function `my_funct()` into ARMv8 assembly (do not compile main()). Be sure to use the conventions discussed in lecture and used in lab and only those instructions discussed in class. Note, converting the function into a simple, nonrecursive loop will yield 0 points.

C. (5 points)

Draw a small diagram of the application's memory map.

- Mark it with 0xFFFFFFFFFFFFFFFF at the top to 0x0000000000000000 at the bottom.
- Mark out the approximate locations of the different regions discussed in class (stack, heap/dynamic, text, global/static).
- On this diagram mark (approximately) where the following data or instructions would be located:
  1) The application's instructions
  2) The variable "int y"
  3) The variable "int z"
  4) Where the return address back to the `main()` function after the `my_funct()` has been called is stored

D. (5 points)

Assuming your function is initially called by main with an argument of `x=7` (ie. `my_funct(7)`). How many words are stored on the stack at its deepest point?

Draw a diagram showing the full contents of the stack at this point. You should assume that the BL to get to my_funct() from main() is at address 0x430100 and that my_funct() starts at address 0x440000 in memory. Also you may assume that the SP is set to 0x7F0FFF3000 when BL is called the first time.

For the next two parts, consider the following assembly:

```
 .section .data
X: .quad 0x123456789ABCDEF0
Y: .quad 0x0
 .section .text
 main:
ADR X19, X
ADR X20, Y
LDURB X19, [X19, 2]
STURB X20, [X20, 6]
LDUR X9, [X20, 0]
```

Note: .quad specifies a constant long long int, similar to .word specifying an int in the assembler.

E. (5 points)

Assuming ARMv8 is Big Endian, what value would be left in X9 after the code starting with $main$ is exectued?

F. (5 points)

If ARMv8 were Little Endian, what value would be left in X9 after the code starting with $main$ is exectued?

III. Processor Implementation (15 Points)

For the following problem assume that the Verilog bus `Inst[31:0]` contains the instruction after being fetched from the Instruction memory.

A. (5 points)

For R-Type instructions, what bits of the `Inst[31:0]` bus should be wired into each of the following ports in the register file:

Port	**Bits**
Read Addr 1	
Read Addr 2	
Write Addr	

B. (5 points)

For the $LDUR$ instruction, what bits should go to the sign extender?

C. (5 points)

For the CBZ instruction, what bits should go to the sign extender?

IV. Qualitative questions (20 points)

Answer the following questions with short answers.

A. (5 points)

List the instructions which require an entry in the relocation table, and which must be relocated upon final linking. Under what conditions do they need to be in the relocation table?

B. (5 points)

Describe the difference between a RISC ISA and a CISC ISA. Give examples of each.

C. (10 points)

Why does the ARMv8 ISA provide the distinction between caller-save and callee-save registers? What benefit can this convention provide?