

National Exams May 2017

98-Comp-A3, Computer Architecture

3 hours duration

NOTES:

1. If doubt exists as to the interpretation of any question, the candidate is urged to submit with the answer paper, a clear statement of any assumptions made.
2. This is an OPEN BOOK EXAM.
One of two calculators is permitted; any Casio or Sharp approved model.
3. FIVE (5) questions constitute a complete exam paper.
The first five questions as they appear in the answer book will be marked.
4. Each question is of equal value.
5. Most questions require an answer in essay format. Clarity and organization of the answer are important.

Marking Scheme

1. (a) 5 marks (b) 5 marks (c) 5 marks (d) 5 marks
2. (a) 10 marks (b) 10 marks
3. (a) 10 marks (b) 10 marks
4. (a) 5 marks (b) 5 marks (c) 5 marks (d) 5 marks
5. (a) 5 marks (b) 7 marks (c) 8 marks
6. (a) 10 marks (b) 10 marks

1. (a) Two 32-bit registers A and B contain respectively the values 0xFFFF FFFE (shown in hexadecimal) and 0x0000 0003. What is the value of $A + B$, if (i) A and B are to be interpreted as unsigned integers, and (ii) A and B are to be interpreted as signed integers in 2's complement. Explain your answer.

(b) The IEEE standard for single precision floating-point representation of real numbers uses 32-bits in total comprising an 8-bit exponent E, a sign bit S, and a 23 bit mantissa M. The number encoded is:

$$(-1)^S \times 2^{(E - 128)} \times 1.M$$

Given two floating point numbers A and B, can " $A \times B$ " be represented precisely in this format? If not, how many more bits would be needed for the mantissa, exponent, and sign?

(c) Assume a 4GB byte-addressable address space: How many bits will the addresses need to be when accessing memory? What if the address space is word-addressable and each word is 32bits long?

(d) For the address space of part (c) explain how a unidimensional array A of 16-bit values having 128 elements will be stored in memory. Which memory address $A[i]$ would be at if the array is stored starting from memory address 0x1000?

2. A processor has 32 32-bit general-purpose registers and a 4GB byte-addressable address space. All instructions are encoded using 32-bits. The processor uses the following two instruction formats:

Format R

4	5	5	5	13
Opcode	Destination Register	Source Register 1	Source Register 2	unused

Format I

4	5	5	18
Opcode	Destination Register	Source Register	Immediate

The top row shows the width, in bits, of each field.

Opcode 0x0 through 0x9 use Format R, while opcodes 0xA through 0xE use Format I. Here are two examples of instructions that use the aforementioned formats:

Format R: ADD R1, R2, R3

Format I: ADDI R1, R2, 0x1000

Where R1-R3 are registers and 0x1000 is an immediate.

- (a) We would like to introduce additional instructions. Can we? How many can we introduce? If this is possible, show an example where we introduce at least one instruction that has one destination register and two source registers. How many more instructions of this type can we introduce? Otherwise, explain why it is not possible to do.
- (b) What if instead we wanted to introduce additional registers? If this is possible, show an example where an instruction uses 2 source registers, 1 destination register and where there are 64 registers in total. Otherwise, explain why it is not possible to do so. How many registers in total can we introduce for instructions that use 2 source registers and 1 destination register, and where any combination of registers is possible?
3. (a) A processor encodes each instruction using 4 bytes. Its address space is 4GB and is byte-addressable. A program executes 1000 instructions, 10% of which are memory reads. Each memory read, reads 4 bytes from memory. What is the minimum number of bytes that a processor will have to read from memory to execute this program? There is only one memory in the system.
- (b) Discuss the pros and cons of using polling to communicate with peripheral devices. Do the same for interrupts.
4. (a) A processor has a 4GB byte-addressable address space. How will a 32KB, 2-way set-associative cache with 32-byte blocks be indexed. Explain your answer.
- (b) A block cached in set 0x1 (hexadecimal) of the cache described in part (a) is tagged with 0x20 (hexadecimal). What is the range of addresses it contains in hexadecimal?
- (c) Why do modern processors use caches? What would be the pros and cons of having more registers instead?

(d) Typical caches use cache blocks or frame of 32 bytes or more. Typical processor instructions that access memory only access up to 8 bytes. Why do caches use larger blocks/frames?

5. (a) A memory chip has the following interface: A0-A15 are 16 single bit input address lines specifying which row is accessed, a single bit input signal R/W! specifies whether the access is a read (1) or a write (0), E is a single bit input signal that must be 1 to access the chip. The data values that are read or written appear on D3-D0. Each of D3-D0 is a single bit output/input pin. When E is 0, D3-D0 pins are in high-Z. What is the total capacity of this memory chip in bytes?

(b) Using as many as necessary of the chips described in part (a), synthesize the equivalent of an 8-bit wide 128KB memory. You can use a few additional logic gates as needed.

(c) A memory interface has the following signals: L0-L31 are 32 single-bit output address lines specifying which memory address is accessed, a single bit output ME signal is 1 when an access is taking place, a single bit R/W! output signal is 1 or 0 for reads and writes respectively, D0-D7 are eight single-bit bi-directional data signals that either provide the value to be written or accept the value on reads. Connect two 1KB chips with the interface described in part (b) (appropriately configured) to this memory interface. The chip should be activated for accesses in the 2KB range starting at address 0xF000 0000. Addresses should be byte-interleaved across the two chips, so that address 0xF000 0000 maps to the first, address 0xF000 0001 to the second, address 0xF000 0002 to the first, and so on.

6. (a) A multi-cycle implementation of a processor requires 5 cycles for the LI instruction, 6 cycles for all ARITHMETIC and BRANCH instructions, 7 cycles for the MEMORY READ instruction, and 5 cycles for the MEMORY STORE instruction. No other instructions exist. The implementation can be modified so that the clock frequency is increased by 5% but at the expense of needing 6 cycles for the LI instruction.

On the average, programs execute instructions with the following frequency:

LI	15%
ARITHMETIC	45%
MEMORY READ	20%
MEMORY STORE	10%
BRANCH	10%

Which implementation (original or modified) will execute programs faster and by how much assuming the aforementioned mix of instructions? Explain your answer.

(b) The x86 processor family from Intel has been around for decades. The first few generation of this family did not use caches. Caches appeared around the mid-80s and for the 80286 processor. Explain what caches are used for. Why didn't the first few generations of 86 processors did not use caches? Why did the later generation processors use them?