UNIVERSITY OF SWAZILAND

Faculty of Science

Department of Computer Science

MAIN EXAMINATION December 2016

Title of Paper: **COMPUTER ORGANISATION II**

Course Number: CS341

Time Allowed: 3 hours Total Marks: **100**

Instructions to candidates:

*This question paper consists of **FIVE (5)** questions.*

*ANSWER **ONE** QUESTION FROM **SECTION A***
*ANSWER **THREE** QUESTIONS FROM **SECTION B***

All questions carry equal marks.

SPECIAL REQUIREMENTS:

NO CALCULATORS ALLOWED

THIS EXAMINATION PAPER SHOULD NOT BE OPENED UNTIL PERMISSION HAS BEEN GRANTED BY THE INVIGILATOR

SECTION A

QUESTION 1 (COMPULSORY)

A) Describe what is meant by Instruction Set Architecture (ISA). In addition briefly describe any 3 sections you would expect to see in an ISA specification document. [5]

B) An instruction set has 4 bits for *opcode* and 32 bits for addresses. What percentage change in instructions and memory resolution results if the *opcode* is increased by 2 bits without altering the instruction length (by taking bits from address portion). [4]

C) Explain one advantage and one disadvantage of the ASCII system compared with Unicode. [2]

D) Discuss the advantages and disadvantages of minimizing the length of an instruction format. [3]

E) With the aid of suitable diagrams compare the following:
i) Cache miss and page fault
ii) Conditional and unconditional branching [8]

F) One of your friends has just come bursting into your room at 3 A.M., out of breath, to tell you about his brilliant new idea: an instruction with two opcodes. Should you send your friend off to the patent office or back to the drawing board? [3]

SECTION B (ANSWER ANY *THREE* QUESTIONS FROM THIS SECTION)

QUESTION 2

A) What is the main purpose of the control store? Communication with memory can be done in 2 different ways. State and briefly describe the 2 ways in which memory access can be achieved. [7]

B) With the aid of appropriate diagrams, describe the following allocation algorithms:

i) First fit
ii) Worst fit [6]

C) Design an expanding opcode to allow all the following to be encoded in a 36-bit instruction:

- 7 instructions with two 15-bit addresses and one 3-bit register number
- 500 instructions with one 15-bit address and one 3-bit register number
- 40 instructions with no addresses or registers. [8]

D) Is it possible to design an expanding opcode to allow the following to be encoded in a 12-bit instruction? A register is 3 bits.

- 4 instructions with three registers
- 255 instructions with one register
- 16 instructions with zero registers [4]

QUESTION 3

A) Define any 6 addressing modes. [6]

B) What is the difference between an instruction and a pseudoinstruction? [2]

C) A machine has a 32-bit byte-addressable virtual address space. The page size is 4 KB. How many pages of virtual address space exist? [3]

D) Briefly explain how the concept of speculative execution is useful in improving performance. [4]

E) Evaluate the following reverse polish expression, where each number is a (decimal) digit.

$$(A-B)\times(((C-D\times E)/F)/G)\times H$$ [5]

F) Compare internal fragmentation to external fragmentation. What can be done to alleviate each? [5]

QUESTION 4

A) Using correct terminology and illustrations, in short paragraphs, describe the following terms:

i) Segmentation. [5]
ii) Paging. [5]
iii) Paged segmentation. [3]

B) Explain, using a suitable illustration, how semaphores work. [4]

C) Outline the main steps carried out by a two-pass assembler? [8]

QUESTION 5

A) Using Amdahl's law on a given program which has 50% sequential code and 50% parallel:

i. What is the speed up anticipated with two processors? [5]
ii. What about 4 processors? [4]
iii. How many processors would result in a 4-fold speedup? [4]

B) For a certain program, 2% of the code accounts for 50% of the execution time. Compare the following three strategies with respect to programming time and execution time. Assume that it would take 100 man-months to write it in C, and that assembly code is 10 times slower to write and four times more efficient.

i. Entire program in C. [4]
ii. Entire program in assembler. [4]
iii. First all in C, then the key 2% rewritten in assembler. [3]

<< End of Question Paper >>