82

UNIVERSITY OF SWAZILAND

Faculty of Science

Department of Computer Science

Main Examination, November 2010

Title of paper: COMPUTER ORGANISATION - II

Course numbers: **CS341**

Time allowed: 3 hours

Instructions: Answer any 5 out of the 6 questions. Each question carries 20 marks.

Question 1

(a) Write microinstruction code words for Mic-1's implementation of the IJVM IOR instruction:

```
iorl      MAR = SP = SP - 1; rd
ior2      H = TOS
ior3      MDR = TOS = MDR OR H; wr; goto Mainl
```

Assume that the control store addresses of the following microinstructions are:

- 0x01: Main1
- 0xF1: ior2
- 0x08: ior3

[5]

(b) Mic-1 has 4 ways of determining the next microinstruction's address. Describe each of them in detail. Your answer must include at least the following:

- Expected content of the JAM and NEXT_ADDRESS field.
- Method by which the MPC register is assigned.
- Example of a microinstruction that uses the method.

[15]

Question 2

(a) Contrast between instructions, micro-instructions and micro-steps. [3]

(b) Draw a labelled diagram of the main components of Mic-2's Instruction Fetch Unit. [4]

(c) Mic-3's 3 buses are each equipped with a latch. Explain the reasons for dividing the data path into 3 sections in this manner. [3]

(d) (i) Describe the main components of Mic-4's Decoding and Queueing Units. Illustrate your answer with diagrams. [5]

(ii) Describe the different ways in which Mic-4's Decoding and Queueing Units can interact with each other. [5]

Question 3

(a) Explain how each of the following can improve the performance of a microarchitecture:
(i) Cache memory
(ii) Out-of-Order execution and Register Renaming [6]

(b) Describe the main kinds of inter-instruction dependency, and explain why dependencies are, in general, detrimental to high-performance in pipelined architectures. [8]

(c) Calculate the *average access time* of a two-level cache that has the following characteristics:

- Level 1 *miss* rate: 10%
- Level 2 *hit* rate: 60%
- Level 1 access time: 5ns
- Level 2 access time (after level 1 miss): 15ns
- Main-memory access time (after level 2 miss): 60ns [6]

85

Question 4

(a) Write the integer 255 in binary notation and in binary coded decimal (BCD). How does this example demonstrate the inefficiency of BCD? [2]

(b) Discuss the advantages and disadvantages of *minimizing* the length of an instruction format. [4]

(c) Design a *minimal* instruction format for encoding the following instruction set, assuming that operands are encoded in 2 bits:
- 25 zero-operand instructions
- 3 two-operand instructions [6]

(d) Design an instruction format for encoding the following instruction set into a 20-bit instruction word:
- 16 zero-operand instructions
- 16 single-register operand instructions
- 32 two-operand instructions, with 1 resister and 1 address operand

Assume that the architecture has 4 registers and 2000 bytes of addressable memory. [8]

Question 5

(a) Explain how FOR and IF-THEN-ELSE control structures (provided by high level languages such as Pascal) can be written in IJVM assembly language. Give code examples in IJVM assembly language to illustrate your answer. [7]

(b) (i) Outline the main steps carried out by a *two-pass* assembler. [7]

(ii) Describe the structure of the symbol table and explain how it is used for computing branch offsets. [6]

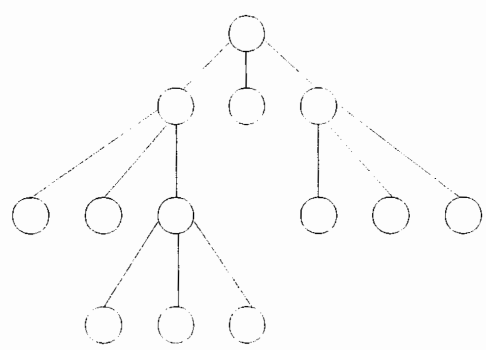
Question 6

(a) Contrast between each of the following pairs:
(i) Shared- and distributed-memory systems.
(ii) Loosely- and tightly-coupled systems. [4]

(b) (i) Define *degree* and *diameter* of interconnection topologies. [2]

(ii) For high performance, is it more advantageous for diameter of a topology to be high or low? Explain. [2]

(iii) A *ternary tree* interconnection topology is a kind of tree where each node (other than leaf nodes) has 3 children. The diagram below shows a ternary tree of size 13. Give the degree and diameter of this tree: [2]



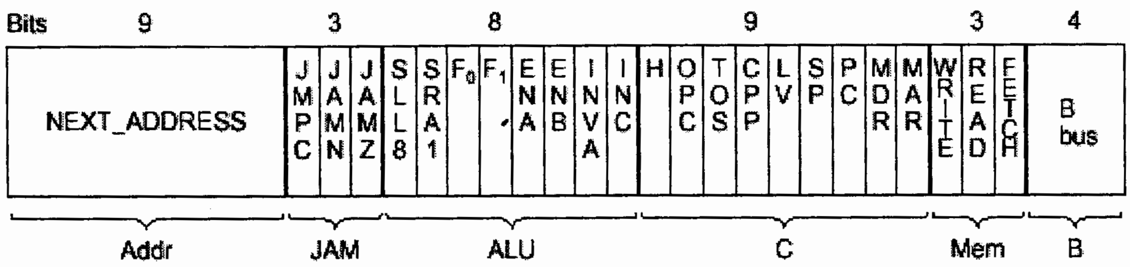
(iv) Explain the advantages and disadvantages of the ternary tree compared with the *ring* topology. [3]

(c) Define *speedup* in parallel computing. In addition, explain the concept of linear speedup and give reasons why linear speedup is rarely attained in practice. [7]

*** END OF QUESTION PAPER ***

Appendix – Microarchitecture & Instruction Set Datasheet

(Source: Andrew S. Tanenbaum, Structured Computer Organization, 5th ed., Prentice-Hall,)



B bus registers

0 = MDR	5 = LV
1 = PC	6 = CPP
2 = MBR	7 = TOS
3 = MBRU	8 = OPC
4 = SP	9-15 none

F_0	F_1	ENA	ENB	INVA	INC	Function
0	1	1	0	0	0	A
0	1	0	1	0	0	B
0	1	1	0	1	0	$\bar{A}$
1	0	1	1	0	0	$\bar{B}$
1	1	1	1	0	0	A + B
1	1	1	1	0	1	A + B + 1
1	1	1	0	0	1	A + 1
1	1	0	1	0	1	B + 1
1	1	1	1	1	1	B – A
1	1	0	1	1	1	B – 1
1	1	1	0	1	1	–A
0	0	1	1	0	0	A AND B
0	1	1	1	0	0	A OR B
0	1	0	0	0	0	0
0	1	0	0	0	1	1
0	1	0	0	1	0	–1

Hex	Mnemonic
0x10	BIPUSH *byte*
0x59	DUP
0xA7	GOTO *offset*
0x60	IADD
0x7E	IAND
0x99	IFEQ *offset*
0x9B	IFLT *offset*
0x9F	IF_ICMPEQ *offset*
0x84	IINC *varnum const*
0x15	ILOAD *varnum*
0xB6	INVOKEVIRTUAL *disp*
0x80	IOR
0xAC	IRETURN
0x36	ISTORE *varnum*
0x64	ISUB
0x13	LDC_W *index*
0x00	NOP
0x57	POP
0x5F	SWAP
0xC4	WIDE