

University of Swaziland

**Faculty of Science
Department of Computer Science**

Supplementary Examination, 2006/07

Title of Paper: Computer Organisation I

Course Number: CS241

Time Allowed: Three (3) hours

Instruction: Answer all questions. Questions carry equal marks.

You are reminded that in assessing your work, account will be taken of the accuracy of the material, of the language used and the general quality of expression, together with the layout and presentation of your answer. Remember full answers will usually *define, explain and exemplify*.

Special Requirements:

Calculators are prohibited.

This examination paper should not be opened until permission has been granted by the invigilator.

Question 1.

Referring to figure 1,

a) Give the micro-code that is executed when the MIR of Tanenbaum's Mic1 subset of JVM contains:

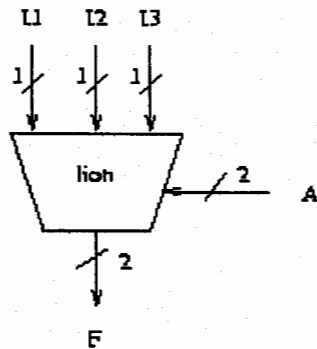
001015201021₈

b) Translate the following micro-code into the bits of the MIR and write the answer in hexadecimal.

H=TOS; wr; goto 23₁₆

Question 2.

The following defines an arithmetic and logical unit:



A ₁	A ₂	F
0	0	I ₁ - (I ₂ OR I ₃)
0	1	I ₂ OR I ₃ OR I ₂
1	0	I ₃ - I ₂ - I ₁
1	1	I ₁ + I ₂ + I ₃
		+ and - are arithmetic

a) What is the output when:

	I ₁	I ₂	I ₃	A ₁	A ₂
i)	1	0	1	1	0
ii)	1	0	1	0	0
iii)	0	1	1	0	1

b) When:

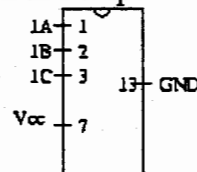
I₁ = I₂ = I₃ = A₁ = A₂ = 0
what happens?

Question 3.

a) Differentiate between the terms *bus* and *bus protocol*.

b) What is a wrist strap?

c) What are the functions of A, B and C on this chip?



d) What is anonymous FTP?

e) How do manufacturers ensure you put their CPU into the motherboard correctly?

Question 4.

- a) Express -10_{10} in binary.
- b) Convert 2649_{10} into hexadecimal.
- c) Convert 139_{16} to octal.
- d) Why would you choose JVM and not Java in which to write a program?

Question 5.

- a) Show how the OR logic function can be implemented using transistors.
- b) How do you implement a two input AND gate using NOR gates only?

Addr(9)	J (3)			ALU (8)							C (9)					M (3)			B (4)				
	JMP	JMN	JMZ	shifter		ALU proper					H	OPC	TOS	CPP	LV	SP	PC	MDR	MAR	write	read	fetch	one only of:
address in the control store of next micro-instruction to be obeyed				SLL8	SRA1	F ₀	F ₁	enable H reg	enable B bus	invert H reg	increment output												0 = MDR
																							1 = PC
																							2 = MBR
																							3 = MBRU
																							4 = SP
																							5 = LV
																							6 = CPP
																							7 = TOS
																							8 = OPC
																							9 - 15 undefined

what appears on o/p of ALU	F ₀	F ₁	enable VP from H reg	enable VP from B bus	invert VP from H reg	force carry to LSB of O/P
Hreg	0	1	1	0	0	0
Bbus	0	1	0	1	0	0
not(H)	0	1	1	0	1	0
not(Bbus)	1	0	1	1	0	0
H + Bbus	1	1	1	1	0	0
H + Bbus + 1	1	1	1	1	0	1
H + 1	1	1	1	0	0	1
Bbus + 1	1	1	0	1	0	1
Bbus - H	1	1	1	1	1	1
Bbus - 1	1	1	0	1	1	0
-H	1	1	1	0	1	1
H and Bbus	0	0	1	1	0	0
H or Bbus	0	1	1	1	0	0
0	0	1	0	0	0	0
1	0	1	0	0	0	1
-1	0	1	0	0	1	0

F ₀	F ₁	ALU function
0	0	Hreg and Bbus
0	1	Hreg or Bbus
1	0	not(Bbus)
1	1	Hreg + Bbus

Figure 1: A reminder of the tables introduced during the course, defining the MIC1.

End of examination paper.