University of Swaziland

Faculty of Science
Department of Computer Science

Supplementary Examination, 2009

Title of Paper: Computer Organisation I

Course Number: CS241

Time Allowed: Three (3) hours

Instruction: Answer five 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 *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. [20]

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

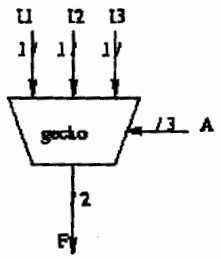
0X80C0A11

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

MAR = SP = SP -1; read

Question 2. [20]

The following defines the *gecko* arithmetic and logical unit:



A_1	A_2	A_3	F
1	0	1	I_1 EOR I_2
0	0	0	I_1 IOR I_3
0	1	1	$I_3 - I_1 + I_2$
1	1	1	$I_3 + I_2 + I_1$
1	0	0	$I_2 - (I_1 * I_3)$
1	1	0	I_1 shifted logically left I_2 places
gecko ALU			+, * and - are arithmetic

What is the output when:

	I_1	I_2	I_3	A_1	A_2	A_3
i)	1	0	0	1	0	1
ii)	0	0	1	0	0	1
iii)	1	1	1	0	0	0
iv)	1	0	0	0	1	1
v)	1	1	0	1	1	1
vi)	1	1	1	1	0	0
vii)	0	1	1	1	1	0

Question 3. [20]

a) Draw the symbol of the PNP transistor showing *base*, *collector* and *emitter*. Develop the circuit diagram for an OR gate using this transistor only.

b) Write JVM code to increment CPP by 600.

c) What is, and what is the puropse of, the *notch* on a chip?

d) What is a (computer) *protocol*?

e) Is it possible to insert a 25 D-way male incorrectly into a 25 D-way female? If not, why not?

Question 4. [20]
Express:
a) 43 (decimal) in binary.

b) 0x43 in binary.

c) 43_8 in binary.

d) 32768_{10} in binary.

e) $7fff_{16}$ in decimal.

Question 5. [20]
a) Show how the AND logic function can be implemented using transistors.

b) How do you implement a two input AND gate using NOR gates only?

Question 6. [20]
a) Using the MIC1, you have to program: *TOS - MDR* and put the result in the *H register*. Do you use either of these code snippets, and if so, why? If you use neither, how do you do it?

i)
H = TOS
H = H - MDR

ii)
H = TOS
H = MDR - H
H = -H

b) What Java statement would be implemented using the IJVM *invokevirtual* statement?

c) You have been authorised to upload a file to the indlu file server (IP address 192.168.4.155). Explain the commands you issue to upload this file called *pass.jpg*.

d) Describe the differences between 1s and 2s complement.

e) Code this Java snippet into the IJVM of MIC1, giving any assumptions you make.

Z = (3*I) + 4

Addr(9)	J (3)			ALU (8)							C (9)									M (3)			B (4)
				shifter		ALU proper																	
address in	JMPC	JAMN	JAMZ	SLL8	SRA1	F_0	F_1	enable	enable	invert	increment	H	OPC	TOS	CPP	LV	SP	PC	MDR	MAR	write	read	fetch	one only of:
the control								H reg	B bus	H reg	output													0 = MDR
store of next																								1 = PC
micro-																								2 = MBR
instruction to																								3 = MBRU
be obeyed																								4 = SP
																								5 = LV
																								6 = CPP
																								7 = TOS
																								8 = OPC
																								9 – 15 undefined

F_0	F_1	ALU function
0	0	Hreg and Bbus
0	1	Hreg or Bbus
1	0	not(Bbus)
1	1	Hreg + Bbus

what appears on o/p of ALU	F_0	F_1	enable I/P from H reg	enable I/P from B bus	invert I/P from H reg	force carry to LSbit 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

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

hex	mnemonic	meaning
10	BIPUSH byte	push byte onto stack
59	DUP	copy top word on stack and push onto stack
A7	GOTO offset	unconditional branch
60	IADD	pop two words from stack; push their sum
7E	IAND	pop two words from stack; push Boolean AND
99	IFEQ offset	pop word from stack; branch if it is zero
9B	IFLT offset	pop word from stack; branch if it is less than zero
9F	IF_ICMPEQ offset	pop two words from stack; branch if equal
84	IINC varnum const	add a constant to a local variable
15	ILOAD varnum	push local variable onto stack
B6	INVOKEVIRTUAL disp	invoke a method
80	IOR	pop two words from stack; push Boolean OR
AC	IRETURN	return from method with integer value
36	ISTORE varnum	pop word from stack; store in local variable
64	ISUB	pop two words from stack; push their difference
13	LDC_W index	push constant from constant pool onto stack
00	NOP	do nothing
57	POP	delete word on top of stack
5F	SWAP	swap the top two words on the stack
C4	WIDE	prefix instruction; next instruction has 16-bit index

Figure 2: Table of IJVM instructions

End of examination paper.