University of Swaziland

Faculty of Science
Department of Computer Science

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

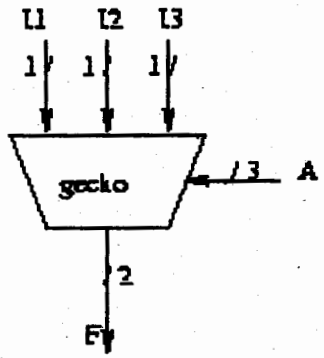
0X203D9025

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

MAR = SP = SP +1; read; goto 44_8;

Question 2. [20]

The following, from a badly printed document, defines the *gecko* arithmetic and logical unit:



A_1	A_2	A_3	F
1	1	1	$I_3 + I_2 + I_1$
0	0	0	I_1 IOR I_3
1	0	0	$I_2 - (I_1 * I_3)$
1	0	1	I_1 OR I_2
0	1	1	$I_2 - I_1 - I_3$
1	1	0	I_1 shifted logically right 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	1	0	1	0	1
ii)	0	1	1	0	0	1
iii)	1	0	1	0	0	0
iv)	1	0	1	0	1	1
v)	1	1	1	1	1	1
vi)	1	1	1	1	0	0
vii)	1	1	0	1	1	0

Question 3. [20]

a) Draw the symbol of the NPN transistor showing *base*, *collector* and *emitter*. How does it function as an inverter?

b) What is folding?

c) You have found a chip in the laboratory. How do you know what it does?

d) What is anonymous FTP?

e) Is it possible to insert the BIOS chip onto a motherboard of a PC incorrectly? If not, why not?

Question 4. [20]
a) Express -25_{10} in binary.

b) Convert 1234_{10} into hexadecimal.

c) Convert 888_{16} to octal.

d) Why would you choose Java in which to write a program?

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) In the microprogram for the MIC1 (the relevant extract is reproduced here), at label *if_icmpeq3*, MDR is copied to H. A few lines later it is subtracted from TOS to check for equality. Surely it is better to have one statement here:
if_compeq3 Z = MDR - TOS; rd

Why is this not done?

---

if_icmpeq1	MAR = SP = SP - 1; rd	Read in next-to-top word of stack
if_icmpeq2	MAR = SP = SP - 1;	Set MAR to read in new top-of-stack
if_icmpeq3	H = MDR; rd	Copy second stack word to H
if_icmpeq4	OPC = TOS	Save TOS in OPC temporarily
if_icmpeq5	TOS = MDR	Put new top of stack in TOS
if_icmpeq6	Z = OPC - H; if (Z) goto T; else goto F	If top 2 words are equal, goto T, else goto F

---

b) Give the Java statement that produced the following IJVM code, where i, j and k refer to both the operand of the ISA code and the Java variable, as in the course text:

```
ILOAD k
ILOAD j
ISUB
BIPUSH 6
ISUB
DUP
IADD
ISTORE i
```

c) Outline how to obtain a file in the /cs241 directory on the indlu file server (IP address 192.168.4.155).

d) In an old Intel CPU, the word length was 4 bits. What would have happened when 0111_2 was added to 1_2?

e) Write IJVM code to test if bit 5 is set in LV5, and if it is not, goto label XYZ.

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