

**University of Swaziland**

**Faculty of Science  
Department of Computer Science**

**Supplementary Examination, 2005**

Title of Paper: Computer Organisation I  
Course Number: CS241  
Time Allowed: Three (3) hours  
Instruction: Answer all questions  
Special Requirement: Table of IJVM instructions (appended)

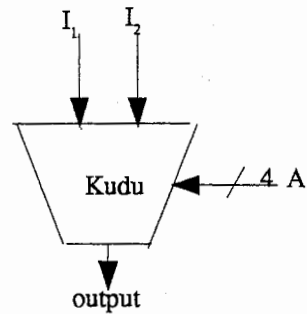
This examination paper should not be opened until permission has been granted by the invigilator.

1. [11]

An extract from a provisional data sheet of an experimental ALU is given here.

Data sheet for the Kudu ALU

A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	I <sub>1</sub>	I <sub>2</sub>	output
0	0	1			+
1	0	0			-
1	0	1			OR
0	1	0			XOR
0	0	0			AND



i) What function of I1 and I2 appears at the output for the following settings of A and I?

	A <sub>1</sub>	A <sub>2</sub>	A <sub>3</sub>	I <sub>1</sub>	I <sub>2</sub>
a)	1	0	1	1	0
b)	1	1	1	1	0
c)	0	0	0	0	1
d)	0	0	0	1	1
e)	0	1	0	1	1

ii) Do you have any comments on the content of the Kudu data sheet?

2. [10]

(i) Write the following Pascal statement

`m := m + 1;`

in:

- a) Java
- b) Java assembly language – using only four instructions
- c) Java JVM machine code

ii) Why is the code produced in (b) bad code?

3. [5]

Why is it safe to run a Java applet on your computer?

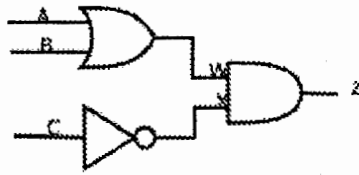
4. [10]

Explain fully, with a diagram, each of the following terms:

- a) pin 15 is held at +5V
- b) pin 5 is asserted
- c) pin 6 is read control

5. [11]

a) Write the truth table for this circuit:



b) Does this circuit implement a single NAND, AND, OR or NOR gate?

Table of JVM instructions

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

End of examination paper