University of Cape Town
Department of Computer Science

Computer Science CSC116S

Test 3 - 5 October 2005

- Answer all questions.
- All questions that refer to elements of programming make reference to the Java programming language as studied in class.
- Good luck !

Marks: 40

Time: 40 minutes

- Approximate marks per question are shown in brackets
- The use of calculators is permitted

Surname Initials

NAME:

STUDENT NO: **COURSE CODE:** CSC

This paper consists of 6 questions and 6 pages (including this cover page).

Mark Allocation							
Quest	Marks	Internal	External	Quest	Marks	Internal	External
1	[15]			4	[3]		
2	[5]			5	[2]		
3	[1]			6	[14]		
	Total				**Total**		
					Grand Total		
					Final Mark		
Internal Examiner:				**External Examiner:**			

Section 1. Number Systems, Boolean Algebra and Logic

Question 1. [15 marks]

Show all calculations for the following questions.

a) Convert 117.375_{10} to radix 2.

```
2 117
2 58 r 1
2 29 r 0
2 14 r 1
2 7  r 0
2 3  r 1
2 1  r 1
  0  r 1 [1]
.375 * 2 = 0.750 intpart =0
.750 * 2 = 1.5   intpart =1
.5   * 2 = 1.0   intpart = 1 [1]
answer: 1110101.011_2
```

[2]

b) Convert 345_8 to hexadecimal.

```
345_8 = (011)(100)(101)_2  [1]
      =(0)(1110)(0101)_2 = E5_16 [1]
```

[2]

c) Use 4-bit 2's complement binary addition to calculate $6_{10} - 2_{10}$.

```
6_10  2_10
= 0110 + 2comp(0010)
= 0110 + 1110  [1]
= 0100 carry 1  discard  [1]
= 4_10   [1]
```

[3]

d) What is the decimal value of `0 10000010 11000000000000000000000` in IEEE 754 format?

```
sign = 0 so positive
biased exponent = 10000010 = 130_10
actual exponent = 130  127 = 3 [1]
significand = .11 = (.5 + .25) = .75 [1]
value: (-1)^0 * (1 + .75) * 2^3 = 1.75 * 8 = 14 [1]
```

[3]

e) In IEEE 754 format, what is the difference between exponent overflow and exponent underflow? What values can be used as approximations in each case?

overflow is when the exponent is too large (positive) to be represented [1] while underflow is when the exponent is too small (negative) to be represented [1]. overflow is approximated with infinity and underflow with zero. [1]

[3]

f) Using an example, show how the alignment of two floating point numbers, for addition, can result in a loss of precision.

If we use 4 bit significands to add $1.0001x2^0$ and $1.0001x2^{-1}$, after alignment the second value is truncated to $0.1000x2^0$, which is different from the actual value of $0.10001x2^0$. [2]

[2]

Question 2. **[5 marks]**

a) If $A = 0$, $B = 1$ and $C = 0$, what is the value of $F = A + (\overline{A} \cdot B) + C$?

F = 0 + (1.1) + 0 = 1 [1]

b) Using a truth table, prove De Morgan's Law : $\overline{A} \cdot \overline{B} = \overline{A + B}$

A	B	~A	~B	~A.~B	A+B	~(A+B)
0	0	1	1	1	0	1
0	1	1	0	0	1	0
1	0	0	1	0	1	0
1	1	0	0	0	1	0

One mark per line. Need only show LHS vs RHS

[4]

Section 2. MIPS

Refer the the attached MIPS instruction set specification when answering these questions.

Question 3. [1 marks]

What is the size, in bits, of a register in the MIPS machine?

32 bits [1] [1]

Question 4. [3 marks]

Explain the purpose for which the following registers in the MIPS machine are used

a) Instruction Register

Instruction Register — Holds a copy of the current instruction being obeyed. [1]

b) Program Counter

Program counter — Holds the address in Main memory of the current instruction being obeyed [1]

c) Register $0

Register $0 — holds the value zero and this value can never be changed. It is useful because the value 0 is frequently used in the CPU [1]

Question 5. [2 marks]

Give the 4 steps that the Control Unit of a computer does.

```
Load Instruction
Increment the Program counter
Execute the Instruction
Go to step 1

2 marks if correct.
```

[2]

Question 6. [14 marks]

Write a MIPS assembler program that does the same as the following Java program. Note that the program may not make a lot of sense, but that it is logically correct.

```
Public static void main(String args[ ]) {
   int a=10, b=15, c=20;
   int ans;
   if (a>b) { ans = a + b + c; }
   else {
      ans = b  a;
      System.out.println(ans);
   }
}
```

Solution :

```
.data [1]
ans: .word 0
a: .word 10
b: .word 15
c:.word 20 [1]

.text
.globl main [1]

Main:  lw $1, a $1 = a
lw $2, b $2 = b [1]

bgt $1, $2, plus #jump to plus if a>b [1]

sub  $3, $2, $1 #$3 = b-a [1]
sw $3, ans # store in memory [1]

li $2, 1 # print an int $2=1 [1]
add $4, $3, $0 #put ans in $4 [1]
syscall # print [1]
j end # jump to end [1]

plus: add $3, $1, $2 # $3 = a+b [1]
lw $4, c # $4=c [1]
add $3, $3, $4 # $3= a+b + c [1]
sw $3, ans

end: jr $ra # exit from program [1]

max marks 14
```

[14]