THE UNIVERSITY OF AUCKLAND

FIRST SEMESTER, 2007
Campus: Tamaki

COMPUTER SCIENCE

TEST

Computer Systems 1
(Time allowed: 50 Minutes)

NOTE: Attempt **ALL** questions.
Write your answers in the space provided.
There is space at the back for answers that overflow the allotted space.
No Calculators are permitted.

Surname	
Forenames	
Student ID	
Login (UPI)	

Question	Marks	Out of
Question 1 (Number Representation)		**12**
Question 2 (Bit Operations)		**6**
Question 3 (Unicode)		**7**
Question 4 (Encoding)		**5**
Question 5 (Branches)		**4**
Question 6 (Instruction Formats)		**8**
Question 7 (Loads & Stores)		**4**
Question 8 (Miscellaneous)		**4**
Total		**50**

Question 1: Number Representation [12 marks]

A. Convert the following unsigned binary numbers to decimal. (2 marks)
   a. 10101
   b. 110110

10101
Answer = 21
1 mark

110110
Answer = 54
1 mark

B. Convert the following unsigned decimal numbers to octal and thence to binary. (2 marks)
   a. 47
   b. 321

47
Octal = 57
Binary = 101 111

321
Octal=501
Binary 101 000 001

C. Perform the following multiplication of unsigned
   a. Binary numbers: 10 * 11 (2 marks)
   b. Octal numbers: 14 * 23 (2 marks)

Multiplication of unsigned Binary numbers: 10 * 11

```
  10
* 11
----
 100
  10
----
 110
```

(1 mark for answer, 1 mark for steps)

Multiplication of unsigned Octal numbers: 14 * 23

```
  14
* 23
----
 300
  44
----
 344
```

(1 mark for answer, 1 mark for steps)

D. Perform the following binary subtractions of signed 2-s complement binary numbers by adding the 2's complement of the subtrahend. You MUST indicate whether or not overflow occurs.
   a. 00010101 - 0010 0100 (2 marks)
   b. 10101010 - 0100 0100 (2 marks)

00010101 - 0010 0100

Two's complement: 1101 1100 (0.5 mark)

```
 00010101
+11011100
 00111000 carries
-----------------
 11110001
```
(1 mark)

Carry into = carry out => Valid (0.5 mark)

10101010 - 0100 0100

Two's complement: 1011 1100 (0.5 mark)

```
 10101010
+10111100
101110000 carries
-----------------
101100110
```
(1 mark)

Carry into != carry out => Invalid (overflow) (0.5 mark)

Question 2: Bit Operations [6 marks]

What is the output of the following code fragment?

```
public class BitOperatos {
      public static void main(String[] args) {
            int x = 47;
            int y = 21;
            int a = 0x15;
            int b = 015;
            System.out.println(a);
            System.out.println(b);
            System.out.println( x & y );
            System.out.println( x ^ y );
            System.out.println( x >> 2 );
            System.out.println( x << 1 );
      }
}
```

```
21 (16 + 5)
13 (8 + 5)
5  (101111 & 10101 = 000101)
58 (101111 ^ 10101 = 111010)
11 (101111 >> 2) = 001011
94 (101111 << 1) = 1011110
```

1 mark each
0.5 mark each if answer is in Binary

Question 3: Unicode [7 marks]

A. Given the following Unicode table:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
03A	…															
03B	ΰ	α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
03C	π	ρ	ς	σ	τ	υ	φ	χ	ψ	ω	…					

What are the Unicode of the following characters? (2 marks)

Character: α Unicode: __03B1__

Character: β Unicode: __03B2_____

(Example : Character : ΰ Unicode: 03B0)

B. Complete the following paint method to display the above two characters on the frame. (2 marks)

```
public void paint(Graphics g) {
    Font font =  new Font("Arial Unicode MS", Font.BOLD, 18);
    g.setFont(font);
```

```
String s = "\u03B1\u03B2";

0.5 for "\u", 0.5 for the code x 2
```

```
    g.drawString(s, 50, 50);

 }
```

C. Convert the Unicode character (α) to UTF8 form. Give your answer in hexadecimal. (3 marks)

Character: α

Unicode (Ucs-2): _03B1

UTF-8 :

Steps :
03B1 (hex)
0000 0011 1011 0001 (binary) <- 1 mark

Case 2 : 5 – 8 leading zeros
110 01110
10 110001 <- 1 mark

`= CE B1 <- 1 mark`

Question 4: Encoding [5 marks]

A. The Alpha instruction set consists of approximately 515 unique instructions. The Alpha opcode is only 6 bits. How can an instruction specify 515 instruction with only a 6-bit opcode? (3 marks)

> There can be only be $2^6=64$ unique opcodes, but some opcodes indicate a class of instructions, e.g., arithmetic, and the "operate" format allows an additional 11 bits to specify various instructions within the class (several versions each of addition, subtraction, multiplication, and compare).

B. How many address bits are required to specify a memory of at least 4 billion bytes? (2 marks)

> $2^{21} <$ 4 billion $< 2^{22}$ so the minimum is 22 address bits. (Any number close to 22 was accepted.

Question 5: Branches [4 marks]

A. The branch instruction format allocates 21 bits for "displacement". Obviously this restricts the range of instructions that can be the target of the branch. What is that range? (2 marks)

> Branch instructions can branch forward 2^{20} or backward 2^{20}-1 longwords (instructions). That is, they can adjust the PC (after it has been incremented) by adding or subtracting a constant X in the range $-2^{22} < X \leq 2^{22}$.

B. Is it possible to transfer control outside of this range? How? (2 marks)

> Yes. There are other ways to transfer control, e.g., using the JMP instruction, which provides the full PC address in a register.

Question 6: Instruction Formats [8 marks]

Below are the four formats used for all Alpha instructions.

31 – 26	25 – 21	20 – 16	15 – 5	4 – 0	
Opcode	Number				PALcode Format
Opcode	RA	Disp			Branch Format
Opcode	RA	RB	Disp		Memory Format
Opcode	RA	RB	Function	RC	Operate Format

A. Give a specific example of an Alpha instruction that uses the Operate Format, and explain what each of the fields in the instruction specifies. (2 marks)

The ADDQ instruction (one of many) uses the operate format. The opcode indicates the instruction is an arithmetic operate instruction. The Function field indicates the operation (including whether the second operand is a literal or taken from the RB field). RA specifies the first source register, and RB specifies the second register (or is part of the literal). RC indicates where the result is placed.

B. Give a specific example of an Alpha instruction that uses the Memory Format, and explain what each of the fields in the instruction specifies. (2 marks)

The LDQ instruction (one of many) uses the memory format. The opcode specifies the LDQ operation; the RA field indicates the destination of the quadword read from memory. The RB field specifies the register whose content is added to the signed-integer constant taken from the Displacement field to derive the address from which the quadword is read.

C. Give a specific example of an Alpha instruction that uses the Branch Format, and explain what each of the fields in the instruction specifies. (2 marks)

The BNE instruction (one of many) uses the branch format. The opcode specifies that it is a BNE instruction; the RA field specifies the register to be compared against zero. The displacement field provides the signed integer constant specifying the (longword) offset to adjust the PC.

D. Give a specific example of an Alpha instruction that uses the the PALcode Format, and explain what each of the fields in the instruction specifies. (2 marks)

The PAL_CODE instruction uses the PALcode format. The opcode indicates it is a PAL_CODE instruction and the Number field indicates the service being requested.

Question 7: Loads & Stores [4 marks]

A. Load and store instructions specify the effective address in two parts, a dynamic part and a static part. Why is the dynamic part necessary? (2 marks)

> The static part is stored in the instruction and can't be modified at runtime. In order to access different addresses using the same instruction, part of the address is provided dynamically by a register, which can be modified by other instructions.

B. Which part requires more bits to specify? How many bits are used to specify each part? (2 marks)

> The static part (the displacement) requires 16 bits of the instruction. The RA field, which specifies the base register, requires 5 bits.

Question 8: Miscellaneous [4 marks]

A. Registers make up only a tiny memory and are expensive. They respond only slightly faster than main memory. What purpose do they serve? That is, why are they needed? (2 marks)

> Registers can be specified in 5 bits. Memory takes much more. Without registers, a 32-bit instruction can't even specify a single complete address, let alone 3! (No one received full credit for this question.)

B. Explain the difference between the shift right arithmetic (SRA) instruction and the shift right logical (SRL) instruction. (2 marks)

> The SRA instruction performs the mathematical operation of integer division by a power of 2. (In practical terms, this means that the bits are shifted right, with signbit fill.) The SLA instruction performs the logical operation of shifting the bits right, zero-filling from the left.