THE UNIVERSITY OF AUCKLAND

FIRST SEMESTER, 2005
Campus: Tamaki

COMPUTER SCIENCE

COMPUTER SYSTEMS

(Time allowed: TWO hours)

NOTE: **DO NOT START, DO NOT OPEN SCRIPT!**
UNTIL INSTRUCTED TO DO SO
Please write your family name, given name and student ID at the top of every page.
Answer all questions on the exam paper in the spaces provided.
The exam is worth 70% of your final grade.
No calculators are allowed!
There are three parts to the test. Part A (worth 10%) is on Data Representation, Part B (worth 30%) is on Unix, Part C (worth 60%) is on Assembly.

Section:	**A**	**B**	**C**	Total
Possible marks:	10	30	60	100
Awarded marks:				

SURNAME:

FORENAME(S):

STUDENT ID:

Surname: _____

Forename(s): _____

Section A: Data Representation

1. **Multiple choice (up to 6 marks)**: **Put a tick or cross in the box on the left of the correct answer (or answers).**
   Important: In some questions you need to possibly mark more than one box in a given question to get full marks for that question. Incorrect answers on the MCQ are penalised (one mark off for each incorrect answer).

   (a) The number 78_{10} is equal to the following:

   ☐ 66_{12} ☐ 116_8 ☐ 01001100_2 ☐ $4D_{16}$

   (b) The number $1B6_{16}$ is equal to the following:

   ☐ None of the others ☐ 1101101010_2 ☐ 666_8 ☐ 436_{10}

   (c) Express the unsigned binary 00111100.11010_2 as a decimal, assuming the format bbbbb.bbbbb$_2$:

   ☐ None of the others ☐ 31.56250 ☐ -30.56250 ☐ 30.6250

   (d) The integer binary expression $10111001_2 << 3 - 101_2 >> 2$ is equivalent to:

   ☐ none of the others
   ☐ $10111_2 - 10110_2$
   ☐ 00011_2
   ☐ 10110_2

   (e) Appendix A gives a table for 7-bit ASCII. Using this table, give the hexadecimal value corresponding to the encoding of the ascii string "1234" (Assume each 7-bit code occupies the space of an 8-bit byte with the MSB=0):

   ☐ 64636261_8
   ☐ 31323334_{16}
   ☐ 34333231_{16}
   ☐ 61626364_8

Surname: _____

Forename(s): _____

(f) Can the irrational number π be expressed exactly in IEEE single precision floating point format?:

- ☐ definitely not
- ☐ only with 64-bits VAX data representation
- ☐ yes
- ☐ possibly

---

2. Compute the 20-bit binary fraction representation of $\frac{1}{63}$. Show your working. [4 marks]

Surname: _____

Forename(s): _____

Section B: Unix

3. Assuming this is a content of the file called Info.txt give the output produced by executing each of the following commands.

```
With 5 different campuses and
  30,000 students, it can be
somewhat daunting, especially
as a new student, to find your
feet.  The Schools Partnership
   Office (contact (09)3737969)
```

(a) grep '[0-9]' Info.txt [2 marks]

(b) grep -n '^[A-Z]' Info.txt [2 marks]

(c) grep -n '^[^A-Z]' Info.txt [2 marks]

Surname: _____

Forename(s): _____

(d) cat Info.txt | wc -w [2 marks]

(e) cat Info.txt | grep '[0-9]' |tr -d"("|tr -d")" [2 marks]

4. (a) Give a shell pipeline command to count the number of lines that contain the text "**UNIX**" in a file named **Compsci210.txt**. [2 marks]

(b) Write a command to clear the group's read permission flag for a file "**readme.txt**" in the current directory. [2 marks]

Surname: _____

Forename(s): _____

(c) Give a shell command to list all files that end with ".s", in your **current directory** and any of it's **sub-directories**. [2 marks]

(d) Give a one line command which will save in alphabetically **reverse** order, a list of all files in the current working directory, to a file called **reverse.txt**. [2 marks]

(e) Explain the difference between wildcards and regular expressions. Give examples of commands you will use them in. Give one symbol that looks the same but have different meaning and one example of symbol that has same meaning, but look different in wildcards and regular expression.
[2 marks]

Surname: _____

Forename(s): _____

5. Write a Bash script that is called ”Calculator.bash”, this script will accept 3 parameters, 1st parameter is a first number, 2nd parameter is either + or - and indicates an operator and 3rd parameter is the second number. [5 marks]
   Shell script should return result of either subtraction or addition of two number depending on the parameter.

   For example:
   Calculator.bash 2 - 1
   1
   Calculater.bash 2 + 5
   7

6. Write a shell script called ”**Conv.bash**” that accepts a line of space delimited binary numbers, coverts those to hexadecimal and prints them out. [5 marks]
   For example:
   Conv.bash 1000 0100 0010 0001
   0x8421
   Conv.bash 0101 0100
   0x54

Surname: _____

Forename(s): _____

Section C: Assembly

7. (a) Provide 2 different ways to subtract the content of register $**T1** from register $**T2**, $**T2** holding the result. [2 marks]

(b) What is the main difference between the instructions **ldwu** and **ldw**? [2 marks]

(c) Show content of the registers $**T0** and $**T1** after following commands have been executed: [2 marks]
ldl $T0, $0x9178$;
ldlu $T1, $0x9178$;

(d) What is the purpose of the value held in the $\$RA$? [2 marks]

(e) What is the purpose of the following instruction **stq $ra, savRet($sp);** [2 marks]

Surname: _____

Forename(s): _____

8. For each of the following questions answer either True or False. You will be awarded 2 marks per right answer.

   (a) On Alpha there are total of 64 floating and integer registers. They all are 64 bits long.

   (b) Command **ldiq** $**T0, a;** loads char "**a**" into register $**T0**.

   (c) Command **stb** $**T0, 1(**$**T2);** retrieves byte stored in memory at the location of **1(**$**T2)** and stores it into $**T0**.

   (d) The instruction **sra** $**T0,** $**T1;** is a genuine alpha instruction and performs arithmetic shift right by $**T1** bits, of the content $**T0**.

   (e) Command **br while;** means break while loop.

Surname: _____

Forename(s): _____

9. (a) Assume the following section program is to be executed:

```
data {
align quad;
one: byte 0x29;
align quad;
two: quad 0x123456789abcdefg;
align quad;
message: asciiz ”0xabc948”;
align quad;
three: quad;
align quad;
last: word 0xbeee;
}
```

Further assume that the memory address corresponding to label one is **0x100000** and the memory has been reset beforehand. Indicate the contents of each byte of memory (and the correct addresses labels are referring to) after the program section has executed. Display all values in hexadecimal. Use little Endian format. [15 marks]

label	memory address	contents	label	memory address	contents

Surname: _____

Forename(s): _____

(b) Assuming initially the memory content from the previous question, compute the value of registers modified after executing each instruction of the assembly code provided below. [10 marks]

ldiq $t0, two;	$t0=	
ldwu$t1, 4($t0);	$t1=	
ldiq $t3, last;		
ldb $t4, 0($t3);		
cmpeq $t4, $t3, $t5;		
cmple $t4, $t1;		
mov 24, $t6;		
ldiq $t0, 2;		
addq $t0, 10;		
negq $t0;		

Surname: _____

Forename(s): _____

10. The opcode and function code (if any) of some Alpha instructions are shown in the table below. All values are in hexadecimal. The formats of integer operate, memory and branch instructions are displayed below.

instruction	opcode	function code	instruction	opcode	function code
addq	10	20	mulq	13	20
subq	10	29	sll	12	39
stq	2d		sra	12	3c
cmple	10	6d	lda	8	
cmpeq	10	2d	ldq	29	
bne	3d		stl	2c	

Operate instruction format

31 26	25 21	20 16	15 13	12	11 5	4 0
opcode	Ra	Rb	000	LF	function	Rc

31 26	25 21	20 13	12	11 5	4 0
opcode	Ra	value	LF	function	Rc

Branching instruction format

31 26	25 21	20 0
opcode	Ra	displacement

Memory instruction format

31 26	25 21	20 16	15 0
opcode	Ra	Rb	displacement

Surname: _____

Forename(s): _____

(a) Disassemble the hexadecimal value **0x40410523** as an Alpha instruction.
Show your working. [5 marks]

(b) Produce the hexadecimal value associated to the following instruction:
`comple $T5, 0x2, $T1;`
Show your working. [5 marks]

Surname: _____

Forename(s): _____

(c) Produce the hexadecimal value associated to the following instruction: `lda $T1, 1($T0);`
Show your working. [5 marks]

Surname: _____

Forename(s): _____

Overflow page 1

Surname: _____

Forename(s): _____

Overflow page 2

Surname: _____

Forename(s): _____

Appendix A

b7 b6 b5 / BITS b4 b3 b2 b1	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
	CONTROL		SYMBOLS NUMBERS		UPPER CASE		LOWER CASE	
0 0 0 0	0 NUL 0	16 DLE 20	32 SP 40	48 0 60	64 @ 100	80 P 120	96 ‘ 140	112 p 160
0 0 0 1	1 SOH 1	17 DC1 21	33 ! 41	49 1 61	65 A 101	81 Q 121	97 a 141	113 q 161
0 0 1 0	2 STX 2	18 DC2 22	34 ” 42	50 2 62	66 B 102	82 R 122	98 b 142	114 r 162
0 0 1 1	3 ETX 3	19 DC3 23	35 # 43	51 3 63	67 C 103	83 S 123	99 c 143	115 s 163
0 1 0 0	4 EOT 4	20 DC4 24	36 $ 44	52 4 64	68 D 104	84 T 124	100 d 144	116 t 164
0 1 0 1	5 ENQ 5	21 NAK 25	37 % 45	53 5 65	69 E 105	85 U 125	101 e 145	117 u 165
0 1 1 0	6 ACK 6	22 SYN 26	38 & 46	54 6 66	70 F 106	86 V 126	102 f 146	118 v 166
0 1 1 1	7 BEL 7	23 ETB 27	39 ’ 47	55 7 67	71 G 107	87 W 127	103 g 147	119 w 167
1 0 0 0	8 BS 10	24 CAN 30	40 (50	56 8 70	72 H 110	88 X 130	104 h 150	120 x 170
1 0 0 1	9 HT 11	25 EM 31	41) 51	57 9 71	73 I 111	89 Y 131	105 i 151	121 y 171
1 0 1 0	10 LF 12	26 SUB 32	42 * 52	58 : 72	74 J 112	90 Z 132	106 j 152	122 z 172
1 0 1 1	11 VT 13	27 ESC 33	43 + 53	59 ; 73	75 K 113	91 [133	107 k 153	123 { 173
1 1 0 0	12 FF 14	28 FS 34	44 , 54	60 < 74	76 L 114	92 \ 134	108 l 154	124 | 174
1 1 0 1	13 CR 15	29 GS 35	45 – 55	61 = 75	77 M 115	93] 135	109 m 155	125 } 175
1 1 1 0	14 SO 16	30 RS 36	46 . 56	62 > 76	78 N 116	94 ^ 136	110 n 156	126 ~ 176
1 1 1 1	15 SI 17	31 US 37	47 / 57	63 ? 77	79 O 117	95 _ 137	111 o 157	127 DEL 177

LEGEND: dec / hex / oct / CHAR

Figure 1: American Standard Code for Information Interchange (ASCII)

Surname: _____

Forename(s): _____

Appendix B-Unix

Useful commands

cd
chmod
cp
ls
mkdir
mv
rm
rmdir echo
cat
head
tail
uniq
sort
cut
paste
find
grep