# TEST
# COMPSCI.210.S1.T
# Computer Systems

13th April 2006, 14:35 - 15:25pm

(TIME ALLOWED: 50 MINUTES)

## 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 test paper in the spaces provided. The test is worth 15% of your final grade.**

**No calculators are allowed!**

**There are two parts to the test. Part A (worth 50%) is on Data Representation, Part B (worth 50%) is on Assembly.**

Print name clearly: ____________________

## PART A-Data Representation: multiple choice (worth 50%)

**Put a tick or cross in the box on the left of the correct answer.**
**Signed 8-bits two's representation assumed (unless otherwise stated).**
**incorrect answers will be penalized.**

---

1. The number $233_{10}$ is equal to the following:

   ☐ $307_6$ ☐ $351_8$ ☐ $10110001_2$ ☐ $11101001_2$

---

2. The number $233_5$ is equal to the following:

   ☐ $572_{16}$ ☐ $11111010_2$ ☐ $154_{16}$ ☐ $510_8$

---

3. What is the decimal value of the 10-bit two's representation binary $1011100011_2$ :

   ☐ $285_{10}$ ☐ $739_{10}$ ☐ $-738_{10}$ ☐ $-285_{10}$

---

4. What is the 12's complement of 3 :

   ☐ (9's complement of 3) +3
   ☐ 5
   ☐ $-8$
   ☐ 4

---

5. Express the signed twos representation 10 bits binary $11111.11010_2$ as a signed decimal, assuming the format $\text{bbbbb.bbbbb}_2$ :

   ☐ $-3.06250$ ☐ $3.56250$ ☐ $-31.81250$ ☐ $-1.81250$

---

6. Compare the Unsigned Hexadecimal number $fff00_{16}$ with other unsigned numbers:

   ☐ $= 3333330000_4$ ☐ $< 67532_{10}$ ☐ $< ffe00_{16}$ ☐ $> 100000000000000_{10}$

---

Print name clearly: ______________________________

7. **Assume unsigned representation**
The sum $110101_2 + 0.101_2$ is equivalent to:

- [ ] none of the others
- [ ] $53.624_{10}$
- [ ] $= 65_8 + 0.4_8$
- [ ] $110101.101_2$

8. $ff7g$ may represent :

- [ ] a number in base 16
- [ ] a number in base 10
- [ ] a number in base 15
- [ ] a number in any base above base 16

9. **Assume 6-bit twos representation.** The binary product $110110_2 \times 101_2$ is equivalent to:

- [ ] none of the others
- [ ] $110111_2 + 11011100_2$
- [ ] $110111_2 << 2 + 1101010_2$
- [ ] $-50_{10}$

10. Appendix A gives a table for 7-bit ASCII. Using this table, give the hexadecimal value corresponding to the encoding of the ascii string "CS210" (Assume each 7-bit code occupies the space of an 8-bit byte with the MSB=0):

- [ ] $3031324565_{16}$
- [ ] $4353323130_{16}$
- [ ] $3031323334_{16}$
- [ ] $101102102101_{16}$

11. What decimal value has to be added to the ASCII for the upper case letter "I‘" to obtain the ASCII for the lower case letter "j":

- [ ] $21_{16}$
- [ ] $20_{16}$
- [ ] 20
- [ ] $21_8$

Print name clearly: ______________________________

12. From Appendix A, the octal ASCII code for the letter "g" is:

☐ $67_{16}$ ☐ $67_{10}$ ☐ $103_8$ ☐ $147_8$

13. The unsigned binary number $1101.111101_2$ is equivalent to:

☐ $23.5625_{10}$
☐ $11011111.01 \times 2^3$
☐ $0.110101101 \times 2^-5$
☐ $1101111101 \times 2^6$

14. Let suppose a decimal value $V$ in XS-$K$ (*excess* $K$) representation. Its unsigned decimal value assuming base 10 would be:

☐ $V \div K$ ☐ $V - K$ ☐ $V + K$ ☐ $V \times K$

15. Given the binary $110111_2$, assuming XS-32 (*excess 32*) representation, its associated decimal value (assuming XS-32) is:

☐ none of the others ☐ negative ☐ $> 64$ ☐ 23

16. A longword comprises how many bytes:

☐ $< 2$ ☐ 24 ☐ 32 ☐ 4

17. Can the rational number 2/3 be expressed exactly in 10 bits binary fraction representation?:

☐ definitely not
☐ only under certain circumstances
☐ yes
☐ possibly

Print name clearly: ______________________________

18. Can the fraction number 0.0005 be expressed exactly with 10-bits binary fraction representation?:

- [ ] none of the others
- [ ] only on a forever computer
- [ ] yes it is equal to 0.00010111111
- [ ] definitely not

19. How many bits (in the binary fraction) are needed to represent $1/2+1/2^{10}$ with 1% accuracy:

- [ ] 11
- [ ] it is impossible
- [ ] 1
- [ ] 10

20. The integer binary expression $10111001_2 << 3$ is equivalent to:

- [ ] $sll\ \ \$T0, 3;$ (with $\$T0$ equals to $b9_{16}$)
- [ ] $10110_2$
- [ ] $00011_2$
- [ ] $10110_2$

21. What is the 8-bit binary fraction representation of $\frac{1}{13}$.

- [ ] $0.00010011_2$
- [ ] you cannot do it
- [ ] $0.10101000_2$
- [ ] $1.00010011_2$

Print name clearly: ____________________

# Part B Alpha Assembly Programming (worth 50%)

**C. 1**

*For each of the following, answer with a simple* Yes *or* No. *[2 marks per correct answer]*

1. `bne $a0, end;` *implies, if a0 is equal to 1, branch to the label* `end`.

2. *The instructions* `addq $T1, $T0` *and* `addq $T0, $T1, $T0` *do the same thing.*

3. *On the alpha, the 32 integer registers respect little-endian representation.*

4. *Both* `ldiq` *and* `ldq` *instructions access the memory to load a quadword.*

5. *If I do answer this question correctly, I'll get 3 marks.*

Print name clearly: ______________________________

**C. 2**

*Suppose we have the following assembly data section*

```
data {
align quad;
a:  quad 0x123456789abcdef0;
align quad;
b:  byte 0x9988;
align quad;
c:  asciiz "CS210";
d:  byte 0x23;
e:  word 0x4321;
}
```

*Fill out the table below to show, in hexadecimal, the given contents of memory assuming that label a refers to address 0x10000000 (assume memory is byte addressable):* *[15 marks]*

| *memory address* | *contents* | *memory address* | *contents* | *memory address* | *contents* |
|---|---|---|---|---|---|
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

*Assuming the initial memory content from the previous question, fill out the table below to show, in hexadecimal, the amended contents of memory, as well as the register contents, after the execution of the instructions listed below:* *[25 marks]*

```
ldiq   $T0, 0x10000002;
ldwu   $T1, 0xfffe($T0);
ldb    $T2, 6($T0);
ornot  $T1, $T2, $T3;
sll    $T3, 4, $T4;
addq   $T3, $T4, $T5;
cmpeq  $T4, $T5, $T6;
cmoveq $T6, 31, $T7;
stb    $T7, +5, $T0;
stq    $T5, +8, $T0;
```

*Print name clearly:* ____________________________________________

| *memory address* | *contents* | *memory address* | *contents* | *register* | *register contents* |
|---|---|---|---|---|---|
| | | | | *T0* | |
| | | | | *T1* | |
| | | | | *T2* | |
| | | | | *T3* | |
| | | | | *T4* | |
| | | | | *T5* | |
| | | | | *T6* | |
| | | | | *T7* | |

**C. 3**

```
block main uses register {
data {
a:  quad 0x1234567890abcdef;
}
code {
public enter:
{
ldiq $t0, a;
ldq $t1, ($t0);
sll $t1, 62, $t2;
srl $t2, 63, $t3;
stq $t3, ($t0);
}
}
}
```

*Briefly describe the purpose of this assembly program and its outcome.* *[10 marks]*

Print name clearly: ______________________________

Rough working area (will not be marked).

# Appendix A

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

LEGEND: dec CHAR hex oct

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