UNIVERSITY OF DUBLIN
TRINITY COLLEGE

Faculty of Engineering, Mathematics and Science

School of Computer Science and Statistics

JF BA (Mod) Computer Science **Trinity Term 2010**

CS1021 & CS1022 – Introduction to Computing I & II

Wednesday, 12 May 2010 **GOLDSMITH HALL** **14:00 – 17:00**

Dr Jonathan Dukes

You **MUST** Answer
Question 1 from **Section A** and
THREE out of **FOUR Questions** from **Section B**

Non-programmable calculators are permitted for this examination

Please indicate the make and model of your calculator
on each answer book used

To be accompanied by an **ARM Instruction Set and Addressing Mode Summary** booklet

Where you are asked to write an assembly language program, you must provide an adequate explanation of your program, for example, in the form of pseudo-code comments.

SECTION A (25 marks)

You MUST answer Question 1 from this section

Suggested time allocation: 35-40 minutes

1. (a) How many binary digits (bits) are required to store the binary equivalent of the following decimal values?

   (i) 320

   (ii) 1045

(1 marks)

(b) Convert the following signed decimal values to their binary equivalents using an 8-bit 2's Complement representation. (**<u>Note:</u>** marks will be awarded for showing **how** you have performed the conversion. Answers consisting of just the final result will receive zero marks.)

   (i) +15

   (ii) -100

(2 marks)

(c) Write an ARM Assembly Language program to compute the following mathematical expression, assuming x is in R1 and y is in R2. Your program should store the result in R0.

$$3x^2+9xy+y^2$$

(3 marks)

1. ... continued ...

(d) Consider each of the following independent ARM Assembly Language instruction sequences. In each case, calculate the final value in R0 and state whether each of the **N** (negative), **Z** (zero), **C** (Carry) and **V** (oVerflow) flags would be set or clear after the execution of the final instruction in each sequence. Assume the flags are all clear before the execution of the first instruction in each sequence.

(i)
```
LDR     R1, =0x6E0074F2
LDR     R2, =0x211D6000
ADDS    R0, R1, R2
```

(3 marks)

(ii)
```
LDR     R1, =0xBF2FDC1E
LDR     R2, =0x40D032E2
ADDS    R0, R1, R2
```

(3 marks)

(e) Consider each of the following independent ARM Assembly Language LDR instructions. State precisely what effect executing each of the LDR instructions would have and, in particular, the memory address accessed. Assume that R1=0xA1001000 and R2=0x00000020.

(i) `LDR     R0, [R1, R2, LSL #2]`

(ii) `LDR     R0, [R1], #16`

(iii) `LDR     R0, [R1, R2]!`

(3 marks)

(f) Provide ARM Assembly Language instructions to perform the following stack operations. Assume a full-descending stack using R13 as the stack pointer.

(i) Push the word-size value in R2 on to the top of the stack

(1 mark)

(ii) Pop off and discard (do not load) the top three words off the stack

(1 mark)

(g) Translate each of the following pseudo-code extracts into ARM Assembly Language. Assume all values are unsigned word-size values, a, b and c are stored in R0, R1 and R2 respectively.

(i)

```
if (a >= 60 && a < 70) // && is Logical AND
{
    b = b + 1;
}
```

(2 marks)

(ii)

```
if ((a & 0xFFFFFFDF) == 'n') // & is bitwise AND
{
    b = 0;
}
```

(2 marks)

(iii)

```
if (a > 0)
{
    while (b >= a)
    {
        b = b - a;
        c = c + 1;
    }
}
```

(4 marks)

SECTION B (75 marks)

You MUST answer THREE out of FOUR questions from this Section
All questions are worth 25 marks

(Suggested time allocation: 40-45 minutes per question)

2 (a) Show how you would perform each of the following operations using ARM Assembly Language:

(i) clear the least significant 5 bits of the word in R1

(1 mark)

(ii) invert the most significant byte of the word in R2

(1 mark)

(iii) change the sign of the word in R3, assuming a signed 2's complement number system

(1 mark)

(iv) multiply the value in R4 by 16 using only the MOV instruction

(1 mark)

(b) Design and write an ARM Assembly Language program that will generate an ASCII string representation, in decimal form, of a signed (2's complement) word-size integer value stored in register R1. The string must be prefixed with leading space characters so that the overall length of the string, including the sign and magnitude (digits), is nine characters and the signed value appears to the right of the string. The string generated by your program must be NULL-terminated. Store your generated string in memory starting at the address contained in R2.

For example, if register R1 contains the word-size value 0xFFFFFF88, your program should generate a string containing the following sequence of ASCII characters:

' ', ' ', ' ', ' ', ' ', '-', '1', '2', '0', 0 (NULL)

Your answer must include:

(i) an explanation of your approach to solving the problem

(6 marks)

(ii) an ARM Assembly Language listing for your program with adequate comments

(15 marks)

3. Write ARM Assembly Language **subroutines** to perform the string operations described below. Assume all strings are composed of byte-size ASCII characters and are NULL (zero) terminated. Provide adequate comments to explain your programs and document the interface to each of your subroutines.

(a) Determine the number of ASCII characters in a string, excluding the NULL character.

(5 marks)

(b) Convert each alphabetic character in a string to uppercase. Assume that the string may contain non-alphabetic characters.

(7½ marks)

(c) Determine whether one string is a substring of a second string. Your comparison should be case sensitive.

(12½ marks)

4. Design and write an ARM Assembly Language **subroutine** that will check the correctness of a "Sudoku" solution. Assume that the solution is stored in memory as a 9 × 9 2-dimensional array of word-size values. The following is an example of a correct solution:

3	6	2	8	4	9	1	5	7
4	1	9	5	6	7	8	3	2
8	7	5	3	1	2	9	4	6
6	9	8	1	3	4	2	7	5
5	2	7	6	9	8	3	1	4
1	3	4	2	7	5	6	9	8
7	5	6	9	8	3	4	2	1
2	8	3	4	5	1	7	6	9
9	4	1	7	2	6	5	8	3

Your program must check the following:

- Each of the digits from 1 to 9 appears exactly once in each row
- Each of the digits from 1 to 9 appears exactly once in each column
- Each of the digits from 1 to 9 appears exactly once in each 3 × 3 sub-array

Your answer must include:

(i) an explanation of your approach to solving the problem

(6 marks)

(ii) a description of your interface to the subroutine

(2 marks)

(iii) an ARM Assembly Language listing for your subroutine with adequate comments

(15 marks)

(iv) an ARM Assembly Language listing for a program showing how your subroutine would be invoked, using the interface that you have described

(2 marks)

5 The ARM7TDMI-S microcontroller raises an Undefined Instruction exception if an attempt is made to decode an undefined instruction. Show how this mechanism can be used to extend the instruction set with a **DIV** instruction that divides one integer (the dividend) by another (the divisor).

Use the following undefined instruction template, where Rd is the destination register, Rm is the dividend and Rn is the divisor.

DIV instruction template

cond	**0111**	**1111**	**0100**	**Rn**	**Rd**	**1111**	**Rm**

Your answer must include:

(i) a new exception handler for Undefined Instruction exceptions that tests for your new **DIV** instruction and, when it is detected, decodes the instruction and implements the division operation

(15 marks)

(ii) a program to install the new exception handler in the exception vector table and test the new **DIV** instruction

(5 marks)

(iii) a discussion of the steps that take place when the "undefined" DIV instruction is fetched and the Undefined Instruction exception is raised

(5 marks)