UNIVERSITY OF DUBLIN
TRINITY COLLEGE

Faculty of Engineering, Mathematics and Science

School of Computer Science and Statistics

BA (Mod) Computer Science
Junior Freshman Examination

Trinity Term 2011

CS1021 & CS1022 – Introduction to Computing I & II

Wednesday 18th May 2011 **Lower Luce 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: 40-45 minutes

1. (a) Write an ARM Assembly Language program to evaluate the following function:

$$f(x) = ax^2 + bx + c$$

where a, b and c are values stored in **R1**, **R2** and **R3** respectively and x is a word-size value stored in memory at address **0xA1001000**. Store the result in **R0**.

(4 marks)

(b) Provide an ARM Assembly Language instruction, or a sequence of instructions, to multiply a value in **R1** by the following constants, **without using the MUL instruction**. Store the result in each case in **R0**.

(i) 16 (1 mark)

(ii) 21 (1 mark)

(c) Consider the following ARM Assembly Language listing, which also shows the address at which each assembled instruction will be stored in memory. Calculate the branch target offset that would be encoded in each of the two highlighted branch instructions and, in each case, show how you computed your result.

```
address  label   instruction
00000000 start   MOV     R0, #0
00000004         MOV     R1, #1
00000008 wh      CMP     R1, #15
0000000C         BHI     endwh
00000010         ADD     R0, R0, R1
00000014         ADD     R1, R1, #1
00000018         B       wh
         endwh
0000001C stop    B       stop
```

(3 marks)

(d) Convert the following pseudo-code program into ARM Assembly Language. You must provide suitable comments to explain your answers. Assume that **i**, **j** and **N** are unsigned values stored in **R1**, **R2** and **R3** respectively and that **A** is an array of word-size values beginning at the address stored in **R4**.

```
if (i * i <= N)
{
    j = 2;
    while (i * j <= N)
    {
        A[i*j] = 0;
        j = j + 1;
    }
}
```

(7 marks)

(e) Provide pairs of values which, if stored in registers **R1** and **R2**, would cause the following condition code flag states after the execution of the instruction **ADDS R0, R1, R2**. Explain your answer in each case.

(i) N=0; Z=1; C=1; V=0 (2 marks)

(ii) N=0; Z=0; C=1; V=1 (2 marks)

Question 1 continues on Page 4

(f) Consider the following ARM Assembly Language subroutine, which computes the ith Fibonacci number, F_i. Describe fully, with the use of a detailed diagram, the contents of the system stack (SP) during the execution of the most deeply nested recursive invocation of the **fibonacci** subroutine when computing F_4.

```
; fibonacci subroutine
; Compute F_i, the (i+1)th Fibonacci number,
; Parameters:     r0: result F_i
;                 r1: i (0, 1, 2, ...)
fibonacci
        STMFD   sp!, {r4-r5,lr}  ; save registers
        MOV     r4, r1
        CMP     r4, #0           ; if (i == 0)
        BNE     else1
        MOV     r0, #0
        B       endif
else1   CMP     r4, #1           ; else if (i == 1)
        BNE     else2
        MOV     r0, #1
        B       endif
else2   SUB     r1, r4, #1       ; else (i >= 2)
        BL      fibonacci
        MOV     r5, r0
        SUB     r1, r4, #2
        BL      fibonacci
        ADD     r0, r0, r5
endif   LDMFD   sp!, {r4-r5,pc}  ; restore registers
```

(5 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. Design and write an ARM Assembly Language subroutine to count the number of pairs of identical adjacent words in a text string.

You may assume that strings are composed of byte-size ASCII characters and are NULL (zero) terminated. You may also assume that the words in the string contain only lower-case alphabetic characters (i.e. 'a' ... 'z') and that adjacent words are separated by a single space character. For example, given the following string:

"one two two three three three"

your subroutine should return the value 3, since "two two", the first "three three" and the second "three three" are pairs of adjacent identical words.

Your answer must include:

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

(6 marks)

(ii) A description of the interface to your subroutine

(1 mark)

(iii) An ARM Assembly Language listing for your subroutine. **You must provide adequate comments to explain your subroutine.**

(18 marks)

3. (a) Design and write an ARM Assembly Language subroutine that will read a sequence of ASCII characters typed by a user in a "console" window and convert the sequence of characters to the value they represent. The only valid characters that may be entered are those representing the decimal digits '0' … '9'. For example, if a user enters the characters '2', '4', '7', '5' and '3', your subroutine should return the value 24753. The sequence of characters ends when the user presses the **RETURN** key, which you may assume corresponds to ASCII character code **0x0D**.

A single ASCII character can be read from a "console" window by invoking the **getkey** subroutine, which will return the ASCII character code in **R0**. Since the key pressed is not automatically also displayed in the window, your subroutine should also invoke the **sendchar** subroutine to display each character as it is read. The character to be displayed should be passed to **sendchar** in **R0**.

Your answer must include:

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

(3 marks)

(ii) A description of the interface to your subroutine

(1 mark)

(iii) An ARM Assembly Language listing for your subroutine. **You must provide adequate comments to explain your subroutine.**

(8 marks)

(b) Design and write an ARM Assembly Language subroutine that will output a sequence of ASCII characters to a "console" window representing an unsigned word value in decimal form.

Your subroutine should output the value one ASCII character at a time by invoking the **sendchar** subroutine, as described in part (a) above.

You may assume the existence of a **divide** subroutine that divides the contents of **R2** by the contents of **R3**, storing the quotient in **R0** and the remainder in **R1**.

Your answer must include:

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

(3 marks)

(ii) A description of the interface to your subroutine

(1 mark)

(iii) An ARM Assembly Language listing for your subroutine. **You must provide adequate comments to explain your subroutine.**

(9 marks)

4 (a) A Magic Square is a two-dimensional array, with dimensions $N \times N$, that contains each of the integers $1 \ldots N^2$ arranged such that the sum of each row, each column and each diagonal is equal to:

$$\frac{N(N^2+1)}{2}$$

Design and write an ARM Assembly Language subroutine that will determine if a two-dimensional array stored in memory is a Magic Square.

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.

(1 marks)

(iii) An ARM Assembly Language listing for your subroutine. **You must provide adequate comments to explain your subroutine.**

(17 marks)

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

(1 marks)

5 (a) Explain in detail how an external interrupt is handled by the NXP LPC2468 microcontroller that you have been using. Your answer should describe the sequence of steps taken by the microcontroller beginning from when the hardware interrupt occurs to the point where the original program resumes execution. You should make reference to terms such as "vectored interrupt controller", "exception vector table", "exception handler" and "processor mode".

(10 marks)

(b) Suppose you have been asked to write the software for a computer system to count the number of cars entering a car park. The system is to be developed using an NXP LPC2468 microcontroller.

There is a sensor embedded in the surface of the road and connected to the **EINT0** external interrupt pin on the LPC2468 microcontroller. The sensor causes a low-to-high transition on the **EINT0** line when a car drives onto it and a high-to-low transition when the car drives off it.

You may assume that the physical design of the sensor is such that each car causes exactly one low-high-low transition on the **EINT0** line, regardless of the number of axels on each vehicle, the speed of the vehicles and the distance between them.

Design and write an appropriate ARM Assembly Language program that will maintain a count of the number of vehicles entering the car park in register **R0**. Your program should include appropriate code to initialise the system and to handle the interrupts raised by the road sensor.

(15 marks)