Programme Code : TU857
Module Code : CMPU2013
CRN: 22510

TECHNOLOGICAL UNIVERSITY DUBLIN

City Campus - Grangegorman

TU857 Computer Science (Infrastructure)

Year 2

SEMESTER 1 EXAMINATIONS 2023/24

Microprocessors

External Examiner
Caroline McEnroy

Internal Examiners
Frank Duignan
Dr. Paul Doyle

Exam Duration: 2 hours

Instructions: **Answer 3 out of the following 4 questions**

Special Instructions/Handouts:
Numbers that are prefixed with '**0x**' are in hexadecimal

Question 1 (33 marks)

(a) Explain the operation of the function **setBit** shown in Listing Q1a below:

```
void setBit(volatile uint32_t *locn, uint32_t bitnumber)
{

  uint32_t value;
  uint32_t mask;
  value = *locn;
  mask = (1 << bitnumber);
  value = value | mask;
  *locn = value;
}
```

Listing Q1a

[6 marks]

(b) What is the unsigned numeric range of the following C types:

(i) `int8_t`

(ii) `uint32_t`

[4 marks]

(c) What value will the variable X have after the following code is executed?

```
int8_t X=127;
X = X + 3;
```

[5 marks]

(d) A little-endian computer system stores the C-string "1234" in memory starting at address 100. It also stores the 16 bit integer value 0x1234 starting at address 200. What values are stored in memory addresses 100 to 104 and memory addresses 200 to 201? The ASCII code for '1' is 49.

[4 marks]

(e) The Arm Cortex M0 executes most instructions in a single clock cycle. If the clock speed is 48MHz, how many microseconds will it take to execute a function with 6000 instructions?

[2 marks]

(f) An STM32F031 has a 12 bit analogue to digital converter with an input voltage range of 0 to 3.3V. What number will it output if the input voltage is 0.8V?

[4 marks]

(g) What role do each of the following play in a computer system?

(i) Address bus

(ii) Data bus

[4 marks]

(h) Using suitable examples, outline the main differences between serial and parallel data communications.

[4 marks]

Question 2 (33 marks)

Figure Q2a shows the pinout for an STM32F031. Pin numbers are shown inside the device outline. Pin functions (e.g. PA9) are shown outside the device outline.

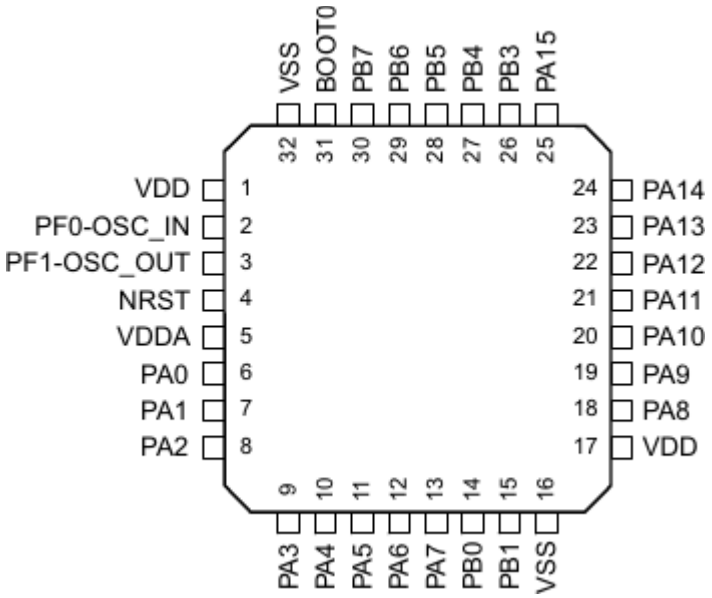

Figure Q2a

An LED and a Button are connected to the device as follows:

Red LED	pin 18
Button	pin 27

A '1' in a port bit turns on an LED. When the button is pressed, the pin to which it is connected is pulled down to '0' otherwise it defaults to '1'.

(a) Write C code that will turn on the Red LED without affecting other outputs.

[4 marks]

(b) Write C code that will wait for the button to be pressed.

[4 marks]

(c) The MODE register for a GPIO port is shown below in Figure Q2b. If it contains the hexadecimal value 0x00005040 which port bits are configured as outputs?

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:0 **MODER[15:0][1:0]:** Port x configuration I/O pin y (y = 15 to 0)
These bits are written by software to configure the I/O mode.
00: Input mode (reset state)
01: General purpose output mode
10: Alternate function mode
11: Analog mode

Figure Q2b

[6 marks]

(d)

```
void pinMode(GPIO_TypeDef *Port,uint32_t BitNumber,uint32_t Mode)
{
      uint32_t mode_value = Port->MODER;
      Mode = Mode << (2 * BitNumber);
      mode_value = mode_value & ~(3u << (BitNumber * 2));
      mode_value = mode_value | Mode;
      Port->MODER = mode_value;
}
```

Listing Q2a

The function pinMode is listed above in Listing Q2a. It is used to configure GPIO pins for an input/output port. Explain the operation of the function by showing which bits are changed when it is called as follows:

pinMode(GPIOA,2,1);

[9 marks]

(e) The SysTick timer in Arm Cortex-M microcontrollers can be used to generate a periodic interrupt.

(i) Outline how this timer operates

(ii) What is meant by the phrase 'interrupt' in this context?

(iii) A Systick timer has an input clock frequency of 48MHz. What value should be placed in its Reload Value Register (RVR) to cause it to generate an interrupt every millisecond?

[10 marks]

Question 3 (33 marks)

```
    .text
    .syntax unified language
    .cpu cortex-m0
    .global  main
    .equ N,10

main:
    LDR R1,=SourceString
    MOVS R0,#0
strlen_loop:
    LDRB R2,[R1]
    CMP R2,#0
    BEQ strlen_exit
    ADDS R0,R0,#1
    ADDS R1,R1,#1
    B strlen_loop

strlen_exit:

    B .     // loop here forever
SourceString: .asciz "abcde"
    .data
DestinationString: .space N

    .end
```

Listing Q3a

(a) Listing Q3a contains ARM Cortex M0 assembly language code which can determine the length of a null terminated ASCII string.

(i) Explain the operation of the assembly language code (starting at "main" and finishing at "B .")

[6 marks]

(ii) List and explain the effect of two assembler directives in Listing Q3a

[4 marks]

(iii) Modify the code so that it copies the contents of the string whose start address is identified by "SourceString" to a destination string whose start address is identified by "DestinationString".

[8 marks]

(b) The MOVS instruction provides an efficient way of loading a limited range of values into a Cortex M0 register.

(i) What range of values is allowed?

[2 marks]

(ii) How would you load a constant 32 bit value into an ARM Cortex M0 register using assembly language?

[4 marks]

(c) Using suitable examples illustrate the operation of the following addressing modes:

(i) Immediate addressing

[2 marks]

(ii) Register indirect with offset addressing

[4 marks]

(iii) Relative addressing

[3 marks]

Question 4 (33 marks) CPU architecture

(a) What are the functions of the following Arm Cortex M0 registers?

(i) PC

(ii) SP

[2 marks]

(b) The Arm Cortex M0 has 4 ALU flags. What are they and what calculations causes each of them to be set to 1?

[4 marks]

(c) Listing Q4a shows how a simple "leaf" function call can be implemented in ARM Cortex M0 assembly language. Explain exactly what the following lines do:

(i) bl sqr

(ii) bx lr

[4 marks]

(d) How is the return address managed for non-leaf functions?

[4 marks]

(e) Local variables are allocated on the stack. With reference to ARM Cortex M0 registers, explain how they are allocated, used and deallocated.

[9 marks]

(f) Modern microprocessors typically include additional hardware to accelerate their performance. Describe what is meant by each of the following and how they increase performance.

(i) Instruction Pipeline

(ii) CPU Cache

[10 marks]

```
main:
    movs R0,#5
    bl sqr

exit_loop:
    b .

sqr:
    muls r0,r0,r0
    bx lr
```

Listing Q4a