UNIVERSITY OF CAPE TOWN
DEPARTMENT OF ELECTRICAL ENGINEERING

EEE2039W (Module D), EEE2026S (Module D), EEE3070S
Introduction to Microcontrollers

FINAL EXAM

TIME: 90 minutes

MARKED OUT OF: 89
AVAILABLE MARKS: 92

INSTRUCTIONS:

1. Attempt all questions.
2. This is a closed book examination.

Examiner: Mr James Gowans
External Examiner: Dr Marc Welz (SKA-SA)

KEEP YOUR ANSWERS SHORT AND TO THE POINT.
Questions apply to the STM32F051 on the UCT development board.

Question 1: (14)

a) Our micro uses a load/store architecture. What is meant by the terms *loading* and *storing*? (2)

b) Compare what happens to the PC when an branch instruction is executed and the branch is taken compared to when an instruction other than a branch is executed. (2)

c) Compared to a normal branch instruction, what else does a `BL` instruction do and why is it useful? (3)

d) Consider the attached extract from the ARM Architecture v6-M reference manual. We want to place the machine code for the following instruction at effective address 0x08000ABC. Which addresses should be set to which values? (3)

```
ADDS R2, R3, #0x04
```

e) Besides the PC changing, what other two things happen when the following instruction is executed? Specify them in the order which they occur. (2)

```
PUSH {R4}
```

f) At a hardware level, how are breakpoints implemented? (2)

Question 2: (15)

a) Assume we want to get the number 0xAA into a CPU register. Which instruction is preferable, a `MOV` instruction or a `LDR` instruction? Explain why. (marks for explanation) (3)

b) What does the assembler directive `.align` do? (2)

c) Write a block of assembly code which will move the single byte at address 0x2000 0123 to address 0x2000 0042. You must use labels for literal loading. (3)

d) Write a single line of C code which does the same as the above. (2)

e) In assembly, when one labels a line of code, a symbol is created. What is the value of the symbol equivalent to? (1)

f) What are the full names of the 4 bits in the APSR? (4 x 0.5 = 2)

g) Will the following branch be taken? Why or why not? (marks for reason) (2)

```
MOVS R0, #0
SUBS R1, R0, #1
ANDS R0, R0, R1
BEQ foo
```

Question 3: (16)

a) Before the registers of a peripheral like a timer or a GPIO port can be interacted with, we have to set bits in the AHBENR or APBENR. What does setting these bits do and why is our micro designed such that these bits must be explicitly set? (2)

b) In the GPIO MODER, how many bits control each pin? Explain why. (2)

c) Describe the algorithm we use to detect when a button is pressed (pressed, not held). (3)

d) Assume a perfect 6-bit ADC with a Vref of 3 V. If the digital output of the ADC is 0x1C, what is the corresponding input voltage range? (3)

e) What does ADC calibration attempt to achieve? (2)

f) Discuss the tradeoffs which should be considered when selecting an ADC resolution. (2)

g) Assuming you're using TIM6 and it's running off of a 8 MHz clock. What is the longest update event period (time between update events) that we're able to get? (2)

Question 4: (11)

a) What is meant by 'dereference a pointer'? (1)

b) Consider the following block. Implement a suitable `find_maximum` function which after being called will have caused `max_value_ptr` to point to the largest value in the array. Be very careful of the data types you specify in your function header. (5)

```
int32_t myArray[] = {-42, 69, 146, 88, -2, -642, 18, ... , -16, 81};
uint16_t myArray_length = sizeof(myArray)/sizeof(myArray[0]);
int32_t *max_value_ptr = NULL;
find_maximum(myArray, myArray_length, &max_value_ptr);
```

c) Explain how the line in the above block which ascertains the number of elements in `myArray` works. (2)

d) What is the value (not the actual number but rather what the number represents) of the first argument passed to the `find_maximum` function? (1)

e) What is the value of `foo` after the following? Express in hex. (2)

```
uint16_t foo = 7 << 14;
```

Question 5: (16)

a) What is the main advantage of using the data types provided `stdint.h` rather than the usual data types such as `int` or `short`? (1)

b) What is the difference between automatically allocated variables and statically allocated variables in terms of how they are defined, their lifespan and their location in memory. (5)

c) C is said to be "pass by value." What does this mean and what's the implication for the case where we want a function that we're calling to be able to modify a variable that is local to where we're calling from? (3)

d) Describe a case where we may want the VMA and LMA for a section to be different. (3)

e) Write down the general format for a makefile rule. Be sure to indicate where the critical but ‘invisible’ character goes. (2)

f) What is the purpose of the linker script? (2)

Question 6: (17)

a) What is the smallest (most negative) and largest (most positive) value which a 16-bit 2’s compliment signed number can take on. Express in hex. (2)

b) In the context of exception, what is a vector? (2)

c) What is the purpose of the NVIC? (2)

d) Discuss what a stack frame is, making reference to when it is used and what it’s used for for. (3)

e) What is the advantage of having an interrupt driven system? (2)

f) When the C preprocessor encounters a line like the the following in a source file, what does it do? (2)

```
#include “myLibrary.h”
```

g) You load some code onto your micro and 'continue' it. When you stop execution you find it's stuck in a hard fault handler. Drat. Discuss a debugging procedure you'd use to find the problem. (4)

Bonus: (3)

Describe the details of the bus and protocol used to interface with the EEPROM chip.