CMPE 364 (L52)
Midterm Exam 1
Spring 2017
March 30th, 2017

Instructions: Read carefully through the entire exam first, and plan your time accordingly. Note the relative weights of each segment. Make sure that you have all pages of the exam. If you are missing any pages, inform the instructor immediately.

During this exam you may use the book *Introduction to Microprocessor Based Systems Using the Arm Processor* by Kris Schindler as well as any notes handwritten by you. The use of any other reference material will result in the confiscation of your exam and you receiving a score of 0. No electronic devices are allowed.

Write your answers on this exam. **Do not write on the back of any pages.** If you need additional space, use the extra pages at the end of the exam. Anything written on the backside of a page will not be graded.

When answering questions that require calculations, be sure to show your work. Answers without appropriate work will not receive credit. When answering questions that request an explanation, keep your explanation short and correct. Explanations containing incorrect or irrelevant information will be marked wrong, even if correct information is also included.

When you are done, present your completed exam to the instructor at the head table. If leaving before the exam period is concluded, please leave as quietly as possible as a courtesy to your neighbors.

TL;DR:

- **You can use the book and handwritten notes, but nothing else.**
- **Don't write on the back of any pages, use the extra pages at the end.**
- **Show your work or you won't get points.**
- **Keep justifications concise and correct.**

Name:

Student ID Number:

Signature:

1. Short answer

   Answer the follow short questions. For answers requiring an explanation, use 1-2 sentences. For answers requiring calculations, show your work.

   (a) (5 points) What is the value of register r3 after the following instruction has executed? Assume that r1 and r2 contain `0x51287c4e` and `0xa0294977` respectively.

   ```
   ADD r3, r2, r1
   ```

   Solution: 0xF151C5C5

   (b) (5 points) The ARM architecture is described as a load-store architecture. What does that mean?

   Solution: It means that data processing operations can only be performed on registers, so if you want to operate on data in memory you must first load it into a register, then do the operation, then store it back to memory.

2. Memory

(a) (5 points) Consider the following map of memory. What are the contents of registers r2 and r4 after the execution of the following instruction? Assume that r2 and r4 initially contain `0x01596273` and `0x80084020` respectively. Also assume that processor is configured in little-endian mode.

```
LDR r2, [r4], #4
```

Address	Contents
0x80084024	0x2f
0x80084023	0xd4
0x80084022	0x1d
0x80084021	0x6a
0x80084020	0xe8
0x8008401F	0xcd
0x8008401E	0x41
0x8008401D	0xeb

Solution:

r4 = 0x80084024
r2 = 0xd41d6ae8

(b) (5 points) Consider the following map of memory. Determine the contents of registers r2, r3, r4, and `sp` after the execution of the following instruction, given an initial value of `0x80084018` for `sp`.

```
LDMEA sp!, {r4-r2}
```

Address	Contents
0x80084024	0x00111100
0x80084020	0x44555544
0x8008401C	0x88999988
0x80084018	0xCCDDDDCC
0x80084014	0x11333311
0x80084010	0x22444422
0x8008400C	0xAACCCCAA

Solution:

r4 = 0x11333311
r3 = 0x22444422
r2 = 0xAACCCCAA
sp = 0x8008400C

3. Simple Coding

Consider the following short coding questions.

(a) (5 points) Rewrite the following sequence of instructions to be more efficient (i.e., use less instructions):

```
start:  add r1, r1, #1
        cmp r1, #5
        bge end
        b start
end:    swi 0x11
```

Solution:

```
start:  add r1, r1, #1
        cmp r1, #5
        blt start
        swi 0x11
```

(b) (10 points) Write a short program which implements the following line of high level language pseudocode. Assume all source registers may be used as scratch registers.

```
r0 := (r3 * 255) / 64 + (r4 + 65) / 2 - r5
```

Solution:

```
mov r5, #255
mul r3, r5, r3
mov r3, r3, asr #6
add r4, r4, #65
mov r4, r4, asr #1
add r0, r3, r4
sub r0,r0, r5
```

4. (15 points) Arrays

Write subroutine called `find_primes` that counts the number of elements from an integer array that are prime. Your subroutine should follow the conventions specified by the AAPCS. Your answer should include only the subroutine.

You may assume that a subroutine called `is_prime` exists and that you can use it. `is_prime` takes an integer as a parameter and returns 1 if the integer is prime and 0 otherwise. (Reminder: You are *not* writing `is_prime`, you just assume it already exists.)

Parameters for `find_primes`:

- Address of an array of integers.
- Number of elements in the array.

Return value of `find_primes`:

- The number of elements in the array that are prime.

Solution:

```
; r4 address of the array
; r0 number of elements in the array
find_primes:
    stmfd sp!, {r4-r7, lr}

    ; r5 is the # of elements
    mov r5, r0
    ; r6 is the loop counter
    mov r6, #0
    ; r7 is the number of prime numbers
    mov r7, #0

    loop:
        cmp r6, r5
        bge end
        add r6, r6, #1
        ldr r0, [r4], #4
        bl is_prime
        cmp r0, #1
        bne loop
        add r7, r7, #1
        b loop
    end:
        mov r0, r7
        ldmfd sp!, {r4-r7, lr}
        bx lr
```

End of Exam.

Extra Work Page 1.
If you need more space to write your answer to a problem, then write it here and make a note on the problem's main page referring to this space.

Extra Work Page 2.
If you need more space to write your answer to a problem, then write it here and make a note on the problem's main page referring to this space.