The University of Alabama in Huntsville
Electrical and Computer Engineering Department
CPE 221 01
Test 1
October 8, 2019

This test is closed book, closed notes. You may not use a calculator. You should have the 6 page ARM Instruction Reference. *You must show your work to receive full credit.*

Name: _____

1. (2 points) The range of numbers that can be represented in n-bit signed 2's complement is _____ through _____.
2. (1 point) A _____ logic circuit is one whose outputs depend only on its current inputs.
3. (1 point) _____ (True or False) x OR x' = 0.
4. (1 point) DCW defines a constant that represents _____ bytes in memory.
5. (10 points) Convert decimal +243 and -205 to binary, using signed-2's complement representation and enough digits to accommodate the numbers.

6. (3 points) What is the decimal equivalent of 11010_3 (assume positional notation and unsigned integer formats)?

7. (12 points) If `r1 = 0x000F 0FFF` and `r2` = 20, what is the value of `r0` after each of the following instructions has been executed? Assume that each instruction uses the same data.

(a) `ADD r0, r1, r1, LSL #7`

(b) `ADD r0, r1, r1, ROR #8`

(c) `ADD r0, r1, r1, LSR r2`

8. (10 points) For each of the following operations on 6 bit signed numbers, calculate the values of the C, Z, V, and N flags

(a) 111010 - 001001

(b) 011011 + 001010

9. (15 points) For each of the following cases,
   1. Explain the effect of each of the following instructions using register transfer notation.
   2. Give the value in `r2` after each instruction executes.
   3. Give the value of the effective address.

   Assume that `r2` contains the initial value `0x0F00 ED10` and that `r0` contains `0xFF9F 5400`. Use these initial values for each instruction individually.

```
(a) LDR   r1, [r2]
```

Register Transfer _____

r2 _____

Effective Address _____

```
(b) STR   r1, [r2, #2_1101]
```

Register Transfer _____

r2 _____

Effective Address _____

```
(c) LDR   r1, [r2, #0xEE]!
```

Register Transfer _____

r2 _____

Effective Address _____

```
(d) STR  r1, [r2], #4
```

Register Transfer _____

r2 _____

Effective Address _____

```
(e) LDR  r1, [r2, r0, ASR #4]
```

Register Transfer _____

r2 _____

Effective Address _____

10. (25 points) Consider the following ARM program. Trace the values of the registers shown as they change during program execution. Also, trace the writes to memory by any `STR` instructions. There may be unused columns or rows in the tables. If you need to add columns or rows, you may do so. `DCD 1` reserves one word of storage and sets it equal to 1. `SPACE 3` reserves 3 bytes of memory but does not give those bytes a value.

```
      AREA   PROB_10, CODE, READONLY
      ENTRY
      ADR   r10, x                  ; (0)
      MOV   r9, #0                  ; (4)
      ADR   r5, p                   ; (8)
      LDR   r0, i                   ; (12)
      LDR   r1, size                ; (16)
loop  CMP   r0, r1                  ; (20)
      BGE   done                    ; (24)
      SUB   r2, r1, r0              ; (28)
      SUB   r2, r2, #1              ; (32)
      LDR   r3, [r10, r0, LSL #2]   ; (36)
      LDR   r4, [r10, r2, LSL #2]   ; (40)
      CMP   r3, r4                  ; (44)
      STRNE r9, p                ; (48)
      ADD   r0, r0, #1              ; (52)
      B     loop                    ; (56)
done  B     done                    ; (60)
size  DCD   5                       ; (64)
x     DCD   13000, 298, -4730, 698, 698 ; x has addresses (68-87)
p     DCD   1                       ; (88)
i     DCD   0                       ; (92)
      END
```

r0									
r1									
r2									
r3									
r4									
r9									
r10									

Results of any `STR` instructions.

Memory Address	Contents

11. (20 points) Complete the ARM assembly language program below so that it implements the following C++ statements.

```
const int size = 10;
int x[size] = {8, 2, 9, 6, 7, 0, 1, 3, 5, 4};
int y[size] = {399, -87, 12, 0, 42, -367, 57, 92, -1000, 25};
for i = 0; i < 10; i++)
  z[i] = y[x[i]] + 20;
```

```
      AREA   PROB_11, CODE, READONLY
      ENTRY
      ADR    r10, x
      LDR    r0, i
      LDR    r1, size
      ADR    r11, y
      ADR    r12, z
```

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

```
x       DCD     8, 2, 9, 6, 7, 0, 1, 3, 5, 4
y       DCD     399, -87, 12, 0, 42, -367, 57, 92, -1000, 25
size    DCD     10
z       SPACE   40
i       DCD     0
        END
```