The University of Alabama in Huntsville
Electrical and Computer Engineering Department
CPE 221 01
Test 1
February 21, 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. (1 point) An ARM processor has _____ registers.
2. (1 point) Each register holds ____ bits.
3. (1 point) In the ARM statement, `ADD r0, r1, r2` _____ is the destination register.
4. (1 point) _____ is a notation to define operations.
5. (1 point) The _____ is a register that holds the address of the instruction currently being executed.

6. (10 points) Convert decimal +508 and +-193 to binary, using signed-2's complement representation and enough digits to accommodate the numbers.

7. (3 points) What is the decimal equivalent of 010100_4 (assume positional notation and unsigned integer formats)?

8. (12 points) If `r1 = 0x000F 0FFF` and `r2` = 4, 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 #3`

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

9. (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) 110000 - 000001

(b) 011111 + 011111

10. (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 `0xFF00 1110` and that `r0` contains `0x0F9F 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, #0x2C]!
```

Register Transfer _____

r2 _____

Effective Address _____

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

Register Transfer _____

r2 _____

Effective Address _____

```
(e) LDR  r1, [r2, r0, ASR #3]
```

Register Transfer _____

r2 _____

Effective Address _____

11. (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_11, CODE, READONLY
        ENTRY
        ADR    r10, x                    ; 0
        ADR    r11, y                    ; 4
        LDR    r0, size                  ; 8
        MOV    r1, #0                    ; 12
        MOV    r2, #5                    ; 16
Loop    CMP    r1, r0                    ; 20
        BGE    done                      ; 24
        SUB    r3, r0, r1                ; 28
        STR    r3, [r10, r1, LSL #2]     ; 32
        SUB    r3, r2, r1                ; 36
        STR    r1, [r11, r3, LSL #2]     ; 40
        ADD    r1, r1, #1                ; 44
        B      loop                      ; 48
done    B      done                      ; 52
size    DCD    6                         ; 56
x       SPACE  24                        ; 60
y       SPACE  24                        ; 84
        END
```

r0																	
r1																	
r2																	
r3																	
r10																	
r11																	

Results of any `STR` instructions.

Memory Address	Contents

12. (20 points) Complete the ARM assembly language program below so that it implements the following C++ statements.

```
const int size = 10;
int x[size];
int flip = 1;
for i = 0; i < size; i++)
  if (x[i] != x[size – i – 1])
    flip = 0;
```

```
      AREA   PROB_12, CODE, READONLY
      ENTRY
      ADR    r10, x
      LDR    r0, i
      LDR    r1, size
      MOV    r11, #1
      STR    r11, flip
      MOV    r9, #0
```

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

```
x       SPACE  40
size    DCD    10
flip    SPACE   4
        END
```