## The University of Alabama in Huntsville Electrical and Computer Engineering Department CPE 221 01 Sample Test 1

## 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.

- (1 point) \_\_\_\_\_\_ are used to hold data when faster access than main memory is needed.
- 3. (1 point) \_\_\_\_\_\_ instructions can alter the normal flow of control from executing the next instruction in sequence.
- 4. (1 point) In RTL, the symbol \_\_\_\_\_ is used to indicate a data transfer.
- 5. (1 point) \_\_\_\_\_\_ is an example of an addressing mode found in processors.
- 6. (10 points) Represent 193 and -259 as signed (2s complement) 16-bit numbers

7. (10 points) If  $r1 = 0 \times 00$  FF 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 #2
- (b) ADD r0, r1, r1, ROR #17
- (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) 001011 + 001101 (b) 111111 + 000001

9. (15 points) Assume that r2 contains the initial value 0xFF001000. Explain the effect of each of the following instructions, and give the value in r2 after each instruction executes. Use register transfer notation.

(a) LDR r1, [r2] (b) STR r1, [r2, #2\_10010] (c) STR r1, [r2, #0x24]! (d) STR r1, [r2], #8 (e) STR r1, [r2, r0, ASR #8] 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 the STR instruction. 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.

loop	AREA ADR ADR LDR LDR SUBS BPL LDR LDR ADD STR ADD ADD ADD	PROB_10, CODE, READONLY r0, x r1, y r2, z r3, size r4, i r5, r4, r3 done r5, [r0] r6, [r1] r5, r5, r6 r5, [r2] r0, r0, #4 r1, r1, #4 r2, r2, #4 r4, r4, #1
	B	
done	B	done
size	DCD	6
i	DCD	0
Х	DCD	100, 3, -1, 2, 4, 4
У	DCD	-53, 247, 95, -7, 481, 91
Z	SPACE END	24

r0							
r1							
r2							
r3							
r4							
r5							
r6							

## Results of the ${\tt STR}$ instruction.

Memory	Contents
Address	

11. (25 points) Complete the ARM assembly language program below so that it implements the following C++ statements.

```
const int size = 10;
int x[size] = {100, 3, -1, 2, 4, 4, 2, -1, 3, 100};
int y[size] = {-53, 247, 95, -7, 481, 91, -33, 1500, 29, -83};
int z[size];
int i;
for (i = 0; i < size; i++)
 if (x[i] > y[i])
    z[i] = x[i];
  else
    z[i] = y[i];
;
        This program examines two arrays, element by element and copies the
;
        largest number of each pair into a third array.
;
;
                PROB 11, CODE, READONLY
        AREA
```

x DCD 100, 3, -1, 2, 4, 4, 2, -1, 3, 100 y DCD -53, 247, 95, -7, 481, 91, -33, 1500, 29, -83 z SPACE 40 i DCD 0 size DCD 10