The University of Alabama in Huntsville
Electrical and Computer Engineering Department
CPE 221 01
Final Exam
December 4, 2018

Name: _____

This test is closed book, closed notes. You may use a calculator. You should have the ARM reference packet. You must show your work to receive full credit. Before you begin, please make sure that you have all eight pages of the exam.

1. (1 point) _____ memories retain stored data only as long as they receive electric power.
2. (1 point) _____ employs a semiconductor technology that stores data as an electrostatic charge in a capacitor.
3. (1 point) Address _____ deals with the way in which address components are mapped onto a processor's physical address space.
4. (1 point) _____ RTL is implementation independent.
5. (1 point) (True or False) _____ A two-way set associative cache has two sets.
6. (4 points) In an ARM computer, r2 contains a value of -925 in decimal. What is the binary value of r1 after this instruction is executed?

```
LSR r1, r2, #7
```

7. (4 points) In an ARM computer, r2 contains a value of 497 in decimal. What is the binary value of r2 after this instruction is executed?

```
MOVT   r2, #497
```

8. (2 points) In an ARM computer, r2 contains a value of -925 in decimal while r3 contains a value of 497 in decimal. What is the binary value of r1 after this instruction is executed?

```
ORN   r1, r2, r3
```

9. (13 points) (a) (8 points) What are the values of the following registers when the program executes "`48 B loop`" for the sixth time? Answer in decimal.

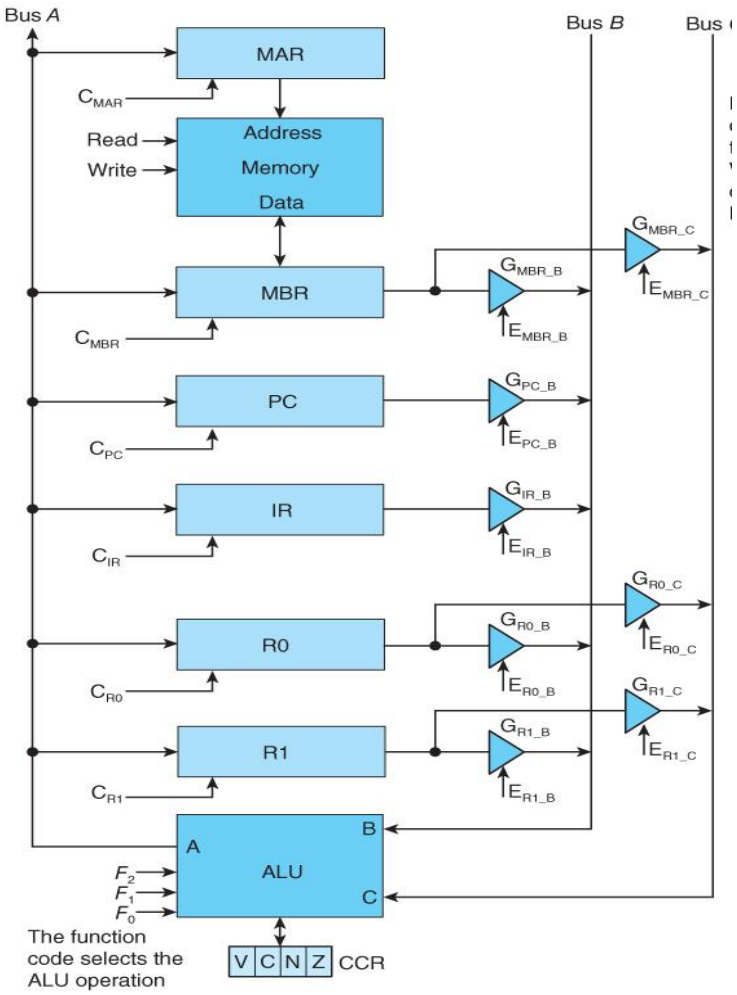
r2: _____ r4_____ r10: _____ r1: _____

(b) (5 points) What value is written by the `48 STR r2, max` instruction? Answer in decimal.

`48 STR r2, max` _____

```
          AREA PROB_9, CODE, READONLY
          ENTRY
0         ADR   r10, array
4         LDR   r1, size
8        LDR   r2, [r10]
12        MOV   r4, #1
16  loop  SUBS  r5, r4, r1
20        BPL   done
24        LDR   r7, [r10, #4]!
28        SUBS  r8, r7, r2
32        BMI   next
36        MOV   r2, r7
40  next  ADD   r4, r4, #1
44        B     loop
48  done  STR   r2, max
52  stop  B     stop
56  size  DCD   10
60  max   SPACE 4
64  array DCD   5, 3, -1, 2, 4, 37, -100, 13, -5, 0
          END
```

10. (15 points) For the architecture shown, write the concrete RTL and the sequence of signals and control actions necessary to execute the instruction `STRM P, R1, R0`, that stores R1 – R0 in the memory location P. Assume that the address P is in the instruction register, IR.
Abstract RTL: M[P] ← R1 – R0



F_1	F_0	Operation
0	0	A = B'
0	1	A = B
1	0	A = B + C
1	1	A = B + 1

Cycle	Concrete RTL	Signals
1		
2		
3		
4		
5		
6		
7		
8		

11. (10 points) A certain memory system has a 32 GB main memory and a 128 MB cache. Blocks are 4 words and each word is 32 bits. Show the fields in a memory address if the cache is 2-way set associative. This memory system is byte addressable.

12. (6 points) You are tasked with building a memory with 35 bits of address so that there are a total of 2^{35} data words. Each data word consists of 64 bits. The only parts you have available to you are static RAM chips that have 18 address bits, so that there are a total of 2^{18} entries. Each chip outputs 8 bits of data. (a) (2 points) How many rows are required? (b) (2 points) How many columns are required? (c) (2 points) How many chips in all?

13. (6 points) A RISC processor executes the following code. There are data dependencies. A source operand cannot be used until it has been written.

```
LDR   r2, [r4]
MOV   r3, r2
STR   r6, [r2]
```

Assuming a five-stage pipeline (fetch (IF), operand fetch (OF), execute (E), memory access (M), and register write (W)), how many extra cycles are required to ensure that the correct value of r2 is available for the `MOV` instruction?

	1	2	3	4	5	6	7	8	9	10	11
`LDR r2, [r4]`											
`MOV r3, r2`											
`STR r6, [r2]`											

14. (20 points) Complete the ARM assembly language program below so that it implements the following C++ statements.

```
;
;       This program adds two arrays, element by element into a third
;       array. It also writes a 0 into an array called sign if the sum
;       sum is negative and a 1 into the array called sign if the sum
;       is positive.
;
;       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 sign[size];
;       int i;
;       for (i = 0; i < size; i++)
;       {
;         z[i] = x[i] + y[i];
;         if (z[i] < 0)
;           sign[i] = 0;
;         else
;           sign[i] = 1;
;       }
```

```
        AREA    PROB_11, CODE, READEXECUTE
        ENTRY
        ADR     r0, x
        ADR     r1, y
        ADR     r2, z
        LDR     r3, size
        LDR     r4, i
        ADR     r5, sign
```

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

```
done    B       done
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
sign    SPACE    40
i       DCD      0
size    DCD      10
        END
```

15. (15 points) Consider the following ARM program. Trace the stack activity, including all changes to the stack pointer and to the contents of the stack. Clearly indicate the value of the sp.

```
0         MOV    sp, #0x00000000
4         B      main
8   swap  SUB    sp, sp, #4
12        LDR    r1, [sp, #8]
16        LDR    r2, [r1]
20        STR    r2, [sp]
24        LDR    r0, [sp, #4]
28        LDR    r3, [r0]
32        STR    r3, [r1]
36        LDR    r3, [sp]
40        STR    r3, [r0]
44        ADD    sp, sp, #4
48        MOV    pc, lr
52  main  SUB    sp, sp, #8
56        ADR    r6, x
60        STR    r6, [sp, #4]
64        ADR    r6, y
68        STR    r6, [sp]
72        BL     swap
76        ADD    sp, sp, #8
80  stop  B      stop
84  x     DCD    2
88  y     DCD    3
```

Address	Value
FFFF FFF0	
FFFF FFF4	
FFFF FFF8	
FFFF FFFC	

Instruction: ______________________

Address	Value
FFFF FFF0	
FFFF FFF4	
FFFF FFF8	
FFFF FFFC	

Instruction: ______________________

Address	Value
FFFF FFF0	
FFFF FFF4	
FFFF FFF8	
FFFF FFFC	

Instruction: ______________________

Address	Value
FFFF FFF0	
FFFF FFF4	
FFFF FFF8	
FFFF FFFC	

Instruction: ______________________

Address	Value
FFFF FFF0	
FFFF FFF4	
FFFF FFF8	
FFFF FFFC	

Instruction: ______________________

Address	Value
FFFF FFF0	
FFFF FFF4	
FFFF FFF8	
FFFF FFFC	

Instruction: ______________________

Address	Value
FFFF FFF0	
FFFF FFF4	
FFFF FFF8	
FFFF FFFC	

Instruction: ______________________

Address	Value
FFFF FFF0	
FFFF FFF4	
FFFF FFF8	
FFFF FFFC	

Instruction: ______________________

Address	Value
FFFF FFF0	
FFFF FFF4	
FFFF FFF8	
FFFF FFFC	

Instruction: ______________________