NAME: _____ GRADE: _____

EE442 / EE592 Real-Time Digital Signal Processing
Quiz #1
Allowed: Textbook, Notes, and Calculator

The following table may be useful for Problems 1 - 4

%0000	$0	0	2^8	256.0	2^-8	0.00390625000000
%0001	$1	1	2^7	128.0	2^-9	0.00195312500000
%0010	$2	2	2^6	64.0	2^-10	0.00097656250000
%0011	$3	3	2^5	32.0	2^-11	0.00048828125000
%0100	$4	4	2^4	16.0	2^-12	0.00024414062500
%0101	$5	5	2^3	8.0	2^-13	0.00012207031250
%0110	$6	6	2^2	4.0	2^-14	0.00006103515625
%0111	$7	7	2^1	2.0	2^-15	0.00003051757812
%1000	$8	8	2^0	1.0	2^-16	0.00001525878906
%1001	$9	9	2^-1	0.50000000000000	2^-17	0.00000762939453
%1010	$A	10	2^-2	0.25000000000000	2^-18	0.00000381469727
%1011	$B	11	2^-3	0.12500000000000	2^-19	0.00000190734863
%1100	$C	12	2^-4	0.06250000000000	2^-20	0.00000095367432
%1101	$D	13	2^-5	0.03125000000000	2^-21	0.00000047683716
%1110	$E	14	2^-6	0.01562500000000	2^-22	0.00000023841858
%1111	$F	15	2^-7	0.00781250000000	2^-23	0.00000011920929

1. Write the hexadecimal values for the following 24-bit fixed-point decimal fractions. The first one is done for you.

+0.5000000	$400 000
-0.7187500	$

2. Write the 24-bit fixed-point, decimal fractions (at least 7 digits of precision) for the following hexadecimal values. The first one is done for you.

+0.2500000	$200 000
	$C20 000

3. Write the hexadecimal values for the following 56-bit fixed-point accumulator (decimal integer/fraction). The first one is done for you.

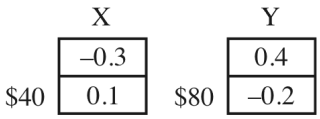
+128.0000000	$40 000 000 000 000
+ 2.1875000	$

4. For the following hexadecimal values in accumulator a, determine the resulting 6-digit hexadecimal value in register x0, when the following instruction is executed:

```
move a1,x0
```

a	x0
$00 101010 800000	$
$FC 100000 000000	$

5. Consider the following data in X- memory (starting at address $40) and Y- memory (starting at address $80)



Write a short code that uses address registers r2 and r6 in order to read the data into x0 and y0 data registers, computes (0.1)(0.4)+(–0.3)(–0.2), and stores the result in accumulator a.

6. Using the supplied code and comments, fill in the *missing* code and comments (denoted with ?) in order to compute the following

$$(0.7)(0.35)+(0.6)(0.3).$$

```
move  #0.7,x0          ;?
move  ?,?              ;y1 = 0.35
mpy   ?,?,?            ;b = (0.7)(0.35)
?                      ;?
?                      ;y0 = 0.3
mac   ?,x1,?           ;b = (0.7)(0.35)+(0.6)(0.3)
```

For the following questions, determine the register, accumulator, and memory states after execution of the instruction. Put 6 hexadecimal digits in each box (or 2 in the case of a2 and b2). If you leave the box blank, we assume the state does not change. If the instruction cannot be executed, write "CANNOT BE EXECUTED."

7. `move x0,x:(r2)+n2`

Before Execution

x1	x0
020202	707070

y1	y0
030203	908090

a2	a1	a0
12	345678	9ABCDE

b2	b1	b0
35	13579B	68ACE0

	r2	n2	m2
	000202	000002	00FFFF

	r3	n3	m3
	000201	000002	000002

	r4	n4	m4
	000403	000001	00FFFF

	r5	n5	m5
	000402	000002	000003

	X		Y
$203	010000	$403	000200
	000300		060000
	000005		000080
$200	007000	$400	00A000

After Execution

x1	x0

y1	y0

a2	a1	a0

b2	b1	b0

	r2	n2	m2

	r3	n3	m3

	r4	n4	m4

	r5	n5	m5

	X		Y
$203		$403	
$200		$400	

8. `move x1,x:(r3+n3)`

Before Execution

x1	x0
020202	707070

y1	y0
030203	908090

a2	a1	a0
12	345678	9ABCDE

b2	b1	b0
35	13579B	68ACE0

	r2	n2	m2
	000202	000002	00FFFF

	r3	n3	m3
	000201	000002	000002

	r4	n4	m4
	000403	000001	00FFFF

	r5	n5	m5
	000402	000002	000003

	X		Y
$203	010000	$403	000200
	000300		060000
	000005		000080
$200	007000	$400	00A000

After Execution

x1	x0

y1	y0

a2	a1	a0

b2	b1	b0

	r2	n2	m2

	r3	n3	m3

	r4	n4	m4

	r5	n5	m5

	X		Y
$203		$403	
$200		$400	

```
9.    move a,y:-(r4)
```

Before Execution

x1	x0
020202	707070

y1	y0
030203	908090

a2	a1	a0
12	345678	9ABCDE

b2	b1	b0
35	13579B	68ACE0

r2	n2	m2
000202	000002	00FFFF

r3	n3	m3
000201	000002	000002

r4	n4	m4
000403	000001	00FFFF

r5	n5	m5
000402	000002	000003

	X		Y
$203	010000	$403	000200
	000300		060000
	000005		000080
$200	007000	$400	00A000

After Execution

x1	x0

y1	y0

a2	a1	a0

b2	b1	b0

r2	n2	m2

r3	n3	m3

r4	n4	m4

r5	n5	m5

	X		Y
$203		$403	
$200		$400	

```
10.   move a2,y:(r5)-
```

Before Execution

x1	x0
020202	707070

y1	y0
030203	908090

a2	a1	a0
12	345678	9ABCDE

b2	b1	b0
35	13579B	68ACE0

r2	n2	m2
000202	000002	00FFFF

r3	n3	m3
000201	000002	000002

r4	n4	m4
000403	000001	00FFFF

r5	n5	m5
000402	000002	000003

	X		Y
$203	010000	$403	000200
	000300		060000
	000005		000080
$200	007000	$400	00A000

After Execution

x1	x0

y1	y0

a2	a1	a0

b2	b1	b0

r2	n2	m2

r3	n3	m3

r4	n4	m4

r5	n5	m5

	X		Y
$203		$403	
$200		$400	

This page EE592 only

EE442 students: if time permits, you may wish to try this problem, however, no additional points will be earned.

11. (+20 points) For the following code, determine the register, accumulator, and memory states after assembly and execution of the code. You may use decimal or hexadecimal ($) values in your answers. If you leave a box blank, we assume the state does not change.

```
N     equ   3

      org x:$101
fifo  dsm   N
      org x:fifo
      dc    0.1, -0.2, 0.3

      org y:$102
data  dc    0.9, 0.45, 0.3

      org p:$100
      move  #fifo,r0
      move  #N-1,m0
      move  #data,r5
      clr   a           x:(r0)+,y0        y:(r5)+,x0
      mpy   x0,y0,a     x:(r0)+,y0        y:(r5)+,x0
      mac   x0,y0,a     x:(r0)+,y0        y:(r5)+,x0
      macr  x0,y0,a
```

After Execution