

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
+0.0312500	\$
-0.6250000	\$

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
	\$248 000

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
- 4.6250000	\$

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 234567 89ABCD	\$
\$FF 000000 800000	\$

5. 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)

```

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

	X		Y
\$41	-0.3	\$81	0.4
\$40	0.1	\$80	-0.2

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.

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

Before Execution						After Execution					
x1	x0	r2	n2	m2	x1	x0	r2	n2	m2		
020202	707070	000202	000002	00FFFF							
y1	y0	r3	n3	m3	y1	y0	r3	n3	m3		
030203	908090	000201	000002	000002							
a2	a1	a0	r4	n4	m4	a2	a1	a0	r4	n4	m4
12	345678	9ABCDE	000403	000001	00FFFF						
b2	b1	b0	r5	n5	m5	b2	b1	b0	r5	n5	m5
35	13579B	68ACE0	000402	000002	000003						
	X		Y			X		Y			
\$203	010000	\$403	000200		\$203		\$403				
	000300		060000								
	000005		000080								
\$200	007000	\$400	00A000		\$200		\$400				

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

Before Execution						After Execution					
x1	x0	r2	n2	m2	x1	x0	r2	n2	m2		
020202	707070	000202	000002	00FFFF							
y1	y0	r3	n3	m3	y1	y0	r3	n3	m3		
030203	908090	000201	000002	000002							
a2	a1	a0	r4	n4	m4	a2	a1	a0	r4	n4	m4
12	345678	9ABCDE	000403	000001	00FFFF						
b2	b1	b0	r5	n5	m5	b2	b1	b0	r5	n5	m5
35	13579B	68ACE0	000402	000002	000003						
	X		Y			X		Y			
\$203	010000	\$403	000200		\$203		\$403				
	000300		060000								
	000005		000080								
\$200	007000	\$400	00A000		\$200		\$400				

This page EE592 only

For the following questions, determine the starting and ending addresses (in hex) that the assembler assigns for array storage. Note the use of DSM (Define Storage Modulo) and DS (Define Storage). The first one is done as an example (values are decimal).

11.

```
        org x:$00000a
hisvec   dsm           4
```

ANSWER: The array hisvec has a starting address of x:\$00000c and an ending address of x:\$00000f

12.

```
        org x:$000123
hervec   dsm           42
```

13.

```
        org x:$000246
yourvec   ds           64
```

15.

```
        org x:$000003
vec1     dsm           24
vec2     ds            16
vec3     dsm           10
```