NAME: _____ GRADE: _____

# EE492 / EE592 Real-Time Digital Signal Processing
# Quiz #4
# Open Books, Notes, Calculators (no programs, no graphing)

*Each question is worth 10 points unless otherwise noted.*

1. For the transmitter of Project #3, we used $f_s$ = 9600 Hz and a carrier with $f_c$ = 2400 Hz. The carrier was synthesized using a lookup table of length, $L$ = 256 with table increment, $\Delta$ = 64. For the same $f_s$ and $f_c$, is it possible to reduce the table length while keeping the table increment an integer (Y/N)? If yes, please give table length and increment.

2. In our asynchronous modem, why couldn't we have pre-loaded the coefficients of the matched filter? In other words, in the receiver .DAT file, why couldn't we simply include

```
MF1   dsm   4
      org   x:MF1
      dc -1, 0, 1.0, 0
```

3. Fill in the table

| *Parity Type* | *Data* | *Parity Bit* |
|---|---|---|
| | 1010101 | 1 |
| Odd | 0101010 | |
| | 1100110 | 0 |
| Odd | 0011001 | |
| | 1010101 | 0 |
| Even | 0101010 | |
| | 1100110 | 1 |
| Even | 0011001 | |

4. Rewrite the following code using the fewest number of lines (6 may be possible) Finally, any instructions which are not needed can be deleted.

```
move  #xtmp,r0
move  #ytmp,r4
move  x:(r0)-,x0
clr   a
move  y:(r4)-,y0
mac   x0,y0,a
move  x:(r0),x0
clr   b     (r0)+
move  y:(r4),y0
mac   x0,y0,b
rnd   a
move  a,x:(r0)
rnd   b
move  b,y:(r4)
move  a,b
```

---

Write a code for the following *non-real-time* programming tasks (Questions 5 – 6). The first programming task is done as an example.

5. Write a code which computes the following inner product

$$y = \mathbf{h}^T\mathbf{x}$$
$$= \begin{bmatrix} 0.4 & 0.3 \end{bmatrix} \begin{bmatrix} 0.2 \\ 0.1 \end{bmatrix}$$

*Solution:*

```
org   p:$100
move #0.4,x0
mpy #0.2,x0,a      ;a = 0.4*0.2
move #0.3,x0
mac #0.1,x0,a      ;a = 0.4*0.2 + 0.3*0.1 (ANSWER IN a)
```

6. For the data byte stored in bit positions 3 – 10 of x0, write a code which computes the even parity bit and stores this bit in position 2 of x0. Assume bit position 2 of x0 is initially clear.

Write a code for the following *real-time* programming tasks (Questions 7 – 9), assuming the Modified Pass Pack beginning on p. 307 of the text. Identify by file name (PASS.ASM or PASS.DAT) and line number where you would place your instructions. The first programming task is done as an example.

7. Write a code which multiplies the right channel by $g = 0.7$.

*Solution:*

In PASS.DAT insert at line 11 the following line:

```
g   equ   0.7
```

In PASS.ASM insert at line 83 the following line:

```
   move   #g,x1   ; move g to x1
```

In PASS.ASM replace lines 98 – 100 with the following lines:

```
   mpyr   x0,x1,a ; multiply right sample by 0.7, rounded result in a
   move   a,x0    ; move result to x0 for move to TX_BUFF_BASE in loop_1
```

8. Suppose we have an FIR filter, $\mathbf{h} = \begin{bmatrix} 0.5 & 0.5 & -0.5 & -0.5 \end{bmatrix}^T$. Using `PASS.ASM` and `PASS.DAT`, write a code which computes the filter output, $z$ and if $z \geq 0$, returns to the top of loop_1, else halts execution (`jmp *` or other).

9. (+25 points) Write a code which alternates between synthesizing a 440 Hz tone for 10 seconds and synthesizing an 880 Hz tone for 5 seconds (as close as possible to 440 and 880 Hz). Assume $f_s$ = 16 kHz; a table length, $L$ = 110; and use an integer table increment ($\Delta$). Output the tone on the left channel.

***This page for EE592 students only.***

10. In the receiver for Project #4, after the matched filter was loaded, the coefficients were rotated each sample period while the input was shifted into the input queue. Why? You may wish to draw a picture to explain.

11. What was the purpose of the frame synchronize routine at the end of the receiver algorithm?