1. Suppose I have an FIR filter given by

\[ H(z) = \frac{b_0 z^M + b_1 z^{M-1} + \cdots + b_m}{z^M} \]

A) If I multiply the numerator by \( z \) I get 

\[ H(z) = \frac{b_0 z^{M+1} + b_1 z^M + \cdots + b_m z}{z^{M+1}} \]

What are the implications for implementation on a real system?

B) If I multiply the denominator by \( z \) I get 

\[ H(z) = \frac{b_0 z^M + b_1 z^{M-1} + \cdots + b_m}{z^M} \]

What are the implications for implementation on a real system?

2. The magnitude plot for a low pass FIR filter with a sample frequency of 1000Hz is shown in figure below. Answer the following questions:

A) How many zeros are located on the unit circle?

B) Are there any real zeros? If so, at what frequency?

C) In the z-plane, what is the angle, \( \theta \), of the zero that corresponds to 100Hz. Put your answer in degrees.

3. An 800Hz signal is sampled at 1,200Hz and sent to a D to A converter. What is the fundamental frequency of the output.

4. A differentiator has a zero at the \( Z = +1 \) point since the frequency response is 0 at 0Hz. FIR differentiators also have linear phase. Why then do all even order FIR linear phase differentiators have zero gain at \( f_s/2 \) but odd order FIR linear phase differentiators do not.
5. If I have a sensor which puts out an analog signal in the range of 0 to 3 volts, what is the quantization noise level, in volts, that I can expect if I use an 8-bit A/D converter.

6. The unit step response of a difference equation is given by
   \[ y(nT)_{\text{step}} = \{0.4, 1.0, 1.18, 1.09, 0.991, 0.97, 0.987, 1.003, 1.005, 1.00, 1.00, 1.00, 1.00, ...\} \]
   From looking at the step response alone, what can you conclude about the frequency response of this difference equation?

7. A signal is sampled and passed through an A/D, filtered, and passed through a D/A converter. There is no anti-aliasing or anti-imaging filter. You can observe the input and output with an oscilloscope and you find the first two frequencies which alias as \( f_0 \) are at \( f_1 \) and \( f_2 \). What are the relationships between \( f_0, f_1, f_2 \), and \( f_s \)?

8. The Fourier series for a square wave with a 50% duty cycle and a frequency of \( f_0 \) as
   \[ f(kT) = \sum_{k=1}^{N} \frac{4}{k\pi} \sin(2\pi f_0 kT) \]
   Write the Nyquist frequency for this square wave as a function of \( N \).

9. Answer the questions below about the pole/zero plot.
   A) What is the minimum value of \( N \) to guarantee causality?

   B) Does this filter have linear phase? Explain why or why not?

   C) What is the phase shift of this filter at \( f_s/2 \)?
10. The C-code on the following page is an ARM Cortex M0 implementation of an FIR filter. Answer the following questions about the code:
   A) What is the filter order?
   B) If the 48 MHz clock is scaled to 24 MHz and MR0 is set for 2,176 counts what is the exact sample frequency?
   C) What is the purpose of line 42?
   D) How many multiplies are required for a single iteration of this implementation?
   E) What is the purpose of the code in lines 40-41?
   F) If you did not know in advance that this is an FIR filter, how could you deduce this from the C-code?
   G) Does this filter have linear phase or not. Explain.
   H) In line 34 what is the range of the integer xInt?
   I) In line 35 what is the range of the value of x[0]?
const float b[] = {0, 9.05067076689394e-05, 0.000370500415670684, 0.000562953433185262,
-3.17905965057806e-05, -0.00228322051866467, -0.00649627818802758,-.0113890190500555,
-0.0134404003840491, -0.00742126210324879, 0.0115541767399710, 0.0451003815523415,
0.0892824143491056, 0.134589281144634, 0.168763757675656, 0.181495997644637};

int yInt;

int main()
{
    int i, xInt;
    float x[31];
    float y;
    PDRUNCFG &= ~(1 << 4);
    SYSAHBCLKCTRL |= (1 << 13);
    SYSAHBCLKCTRL |= (1 << 8);
    SYSAHBCLKCTRL |= (1 << 7);
    IOCON_R_PIO0_11 &= ~0xFFFFF78;
    IOCON_R_PIO0_11 |= (1 << 1);
    AD0CR = 0x0001;
    NVIC_ISER |=(1 << 17);
    IOCONPIO1_9 |= 1;
    TMR16B1PR = 1;
    TMR16B1MR3 = 1024;
    TMR16B1MCR |=(1 << 10);
    TMR16B1MCR |=(1 << 9);
    TMR16B1PWM |= 1;
    TMR16B1TCR | = 1;
    TMR16B0PR = 1;
    TMR16B0MR0 = 2176;
    TMR16B0MCR | = 1;
    TMR16B0MCR | = (1 << 1);
    TMR16B0TCR | = 1;

    while (1)
    {
        AD0CR |= (1 << 24);
        while(AD0DR0 < 0x7FFFFFFF);
        xInt = AD0DR0;
        x[0] = ((float)((xInt >> 6) & 0x000003FF))/1024.0;
        y = b[15]*x[15];
        for(i=0; i<15; i++)
            y += b[i] * (x[i] + x[30-i]);
        yInt = (int)(512*y);
        for(i=30; i>0; i--)
            x[i] = x[i-1];
        while((TMR16B0IR & 1) == 0);
        TMR16B0IR | = 1;
    }
}

void TIMER16_1_IRQHandler()
{ TMR16B1IR = 8;
    TMR16B1MR0 = yInt;
}