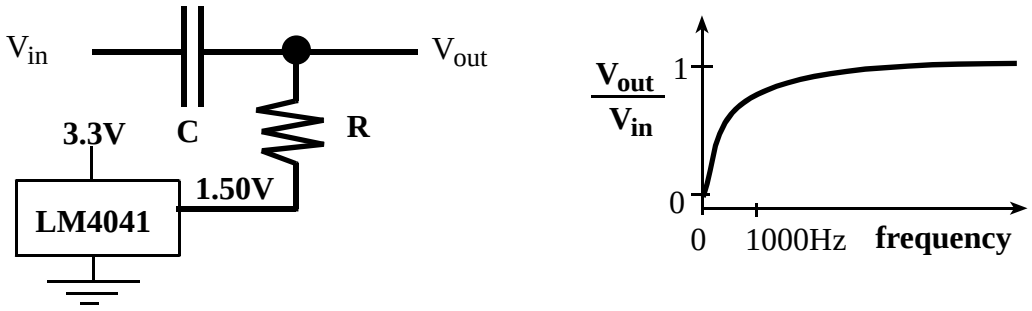
Jonathan W. Valvano
April 6, 2012, 10:00 to 10:50am

(20) Question 1. The cutoff frequency of a one-pole analog high pass filter is $1/(2\pi RC)$. Thus we need $1000 = 1/(2\pi RC)$ or $R = 1/(2000\pi C)$ Since C = 0.1 μF, R = 1600 Ω. We use a shunt diode to create the 1.50V offset. Any ceramic capacitor is ok; however C0G is much better than Z5U.



(15) Question 2. To take the **H(z)** transform for this filter, we transform both sides

$$\mathbf{y(n) = x(n) + y(n-1)}$$
$$\mathbf{Y(z) = X(z) + z^{-1}Y(z)}$$
$$\mathbf{(1-z^{-1})Y(z) = X(z)}$$
$$\mathbf{H(z) = Y/X = 1/(1-z^{-1})}$$

DC gain is $|\mathbf{H(z)}|$ at $\mathbf{z}=1$, so DC gain is infinite. This means the filter will overflow if the input has any DC component.

(25) Question 3. Write software to implement this 60 Hz digital reject filter.

Part a) Overflow cannot happen because 476+452+231 is less than 32768.

y(n) = x(n) + x(n-2) +(-476x(n-1)+ 452y(n-1)-231y(n-2))/256

4 additions
3 multiplications
5 memory reads
1 memory write
= 13 cycles

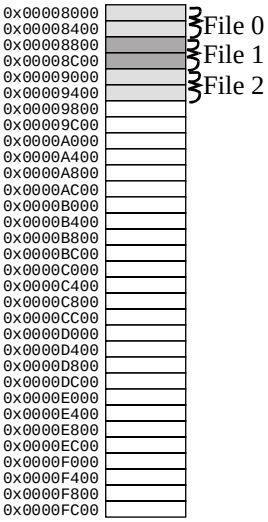
Part b) Show the C function that implements this filter. See book program 5.7.

```
long x[3]; // MACQ for the ADC input data
long y[3]; // MACQ for the digital filter output
long Filter(long input){
  x[2] = x[1]; x[1] = x[0];  // shift data
  y[2] = y[1]; y[1] = y[0];
  x[0] = input;              // 0 to 65535
  y[0] = x[0]+x[2]+(-476*x[1]+452*y[1]-231*y[2])>>8; // output
  return y[0];
}
```

(40) Question 4. You can implement this program using the example file on the web
http://users.ece.utexas.edu/~valvano/arm/Flash_811.zip
This solution allows 16 files to be written.

Part a) Erase the entire 32k ROM (sets all bytes to 0xFF). A 2k block with 0xFF in the last byte of the block is free. A block with 0 in the last byte of the block has been written (not free). This means 2047 out of 2048 bytes could have been used for user data.

Part b) Each file takes two blocks. I consecutively use blocks for each file write. 2000 out of 2048 bytes are used. A zero is placed in the last byte to signify a block is used.



Part c) Initialization erases entire disk (erase called 32 times)

```
int File_Init(void){ unsigned long addr;
  for(addr = 0x00008000;
      addr < 0x00010000;
      addr = addr+1024){
    if(Flash_Erase(addr)) return 1;
  }
  return 0;
}
```

Part d) Show the implementation of **File_Record**

```
#define Buf (*((unsigned char *)0x00008000))
// assumes the 2000-byte user block as space for 2048 bytes
int File_Record(unsigned char *pt){ // stores 2000 bytes
unsigned long addr; int i;
  i = 0;
  while(Buf[i+2047]==0){ // check last byte, 0 means used
    i = i+2048; // 2048 bytes per file
    if(i == 32768) return 1; // full
  }
  addr = 0x00008000+i; // address of first free block
  if(Flash_ProgramBlock(pt,addr)){ // first 1024
    return 1; // disk write error
  }
  pt[2047] = 0; // mark block used
  if(Flash_ProgramBlock(pt+1024,addr+1024)){ // second 1024
    return 1; // disk write error
  }
  return 0;
}
```