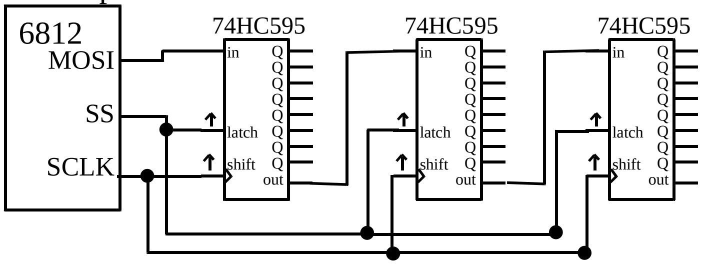
Jonathan W. Valvano October 4, 2006, 1 to 1:50pm

(25) Question 1. The objective of this question is to design a 24-bit parallel output interface

Part a) Build a 24-bit serial to parallel interface and use SPI.



Part b) Show the software device driver. The function names and parameters are given.

`// initialize 24-bit interface`
`void Dig24_Init(void){`
`unsigned char dummy;`
`  DDRM |= 0x38; // PM5(SCLK),`
`// PM4(MOSI), PM3(SS) outputs`
`   PTM |= 0x08;  // latch=1`
`/* bit SPICR1`
` 7 SPIE = 0   no interrupts`
` 6 SPE  = 1   enable SPI`
` 5 SWOM = 0   regular outputs`
` 4 MSTR = 1   master`
` 3 CPOL = 0   output on fall,`
` 2 CPHA = 0   clock normally low`
` 1 SSOE = 0   regular output`
` 0 LSBF = 0   MSB first */`
`  SPICR1 = 0x50;`
`  SPICR2 = 0x00; // MODFEN = 0`
`  SPIBR = 0x20;  // 4MHz`
`  dummy = SPISR;`
`  dummy = SPIDR;  // clear SPIF`
`}`
	`// output 24 bits`
	`void Dig24_Out(char data[3]){`
	`unsigned char dummy;`
	`  PTM &= ~0x08;    // latch=0`
	`  SPIDR = data[2]; // right most data`
	`  while((SPISR&0x80)==0){}; // SPIF`
	`  dummy = SPIDR;   // clear SPIF`
	`  SPIDR = data[1]; // data`
	`  while((SPISR&0x80)==0){}; // SPIF`
	`  dummy = SPIDR;   // clear SPIF`
	`  SPIDR = data[0]; // left most data`
	`  while((SPISR&0x80)==0){}; // SPIF`
	`  dummy = SPIDR;   // clear SPIF`
	`  PTM |= 0x08;     // latch=1`
	`}`

(25) Question 2. One way to manage free-space on a disk is to implement a **bit vector**.

Part a) Write a helper function that allocates a free block updating the disk copy of BitVector.

```
// allocate a free block, returns a block number of a free block
// Output: block number 2 to 255 if successful and 0 if full
unsigned char AllocateBlock(void){  unsigned short i,block=0;
unsigned short data,mask;
  eDisk_ReadBlock(BitVector,1);        // fresh RAM copy
  for(i=0;i<16;i++){
    data = BitVector[i]; // 0 means these 8 are allocated
    if(data){            // any free here?
      mask = 0x8000;
      while((data&mask)==0){
        mask = mask>>1;  // 0x8000, 0x4000,… 0x0002, 0x0001
        block++;
      }
      BitVector[i] &= ~mask;           // clear bit means allocated
      eDisk_WriteBlock(BitVector,1);   // update disk copy
      return block;
    }
    block += 16;
  }
  return 0;} // all full
```

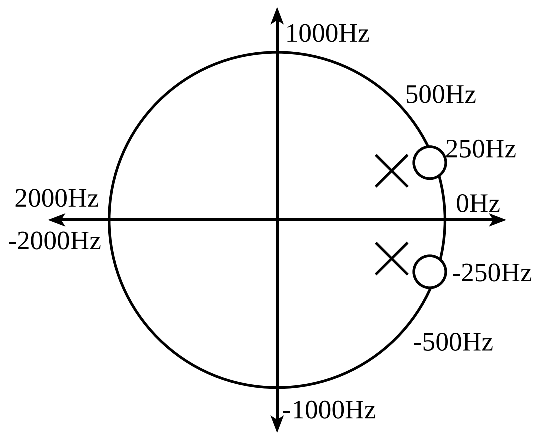
Part b) Write a helper function that deallocates a block updating the disk copy of BitVector.

```
// deallocate a free block
// Input: block number 2 to 255
unsigned short TheMask[16]={0x8000,0x4000,0x2000,0x1000,0x0800,0x0400,0x0200,
  0x0100,0x0080,0x0040,0x0020,0x0010,0x0008,0x0004,0x0002,0x0001};
void DeallocateBlock(unsigned char blockNum){ unsigned short i;
unsigned char data,mask;
  eDisk_ReadBlock(BitVector,1);   // fresh RAM copy
  i = blockNum/16;                // 0 to 15
  mask = TheMask[blockNum%16];    // 0x8000, 0x4000,… 0x0002, 0x0001
  BitVector[i] |= mask;          // set bit means free
  eDisk_WriteBlock(BitVector,1); // update disk copy
}
```

(10) Question 3. This system has no internal fragmentation because every byte on the disk can be used to store data. There is no extra storage used for the convenience of the OS. If the files could exist at any arbitrary size, then there would be internal fragmentation equal to about 16 bytes (1/2 a block) for each file.

This system has external fragmentation because files must be allocated contiguously. The amount of free storage (246 free blocks in the picture. 10 blocks allocated) is larger than the maximum file size that can be allocated (blocks 25 to 255 can be allowed for a file containing 231 blocks).

(20) Question 4. Show pole-zero plot for the design of a high-Q 250-Hz digital reject filter.



(20) Question 5. V_{AD} = 7.5 - 8V_t

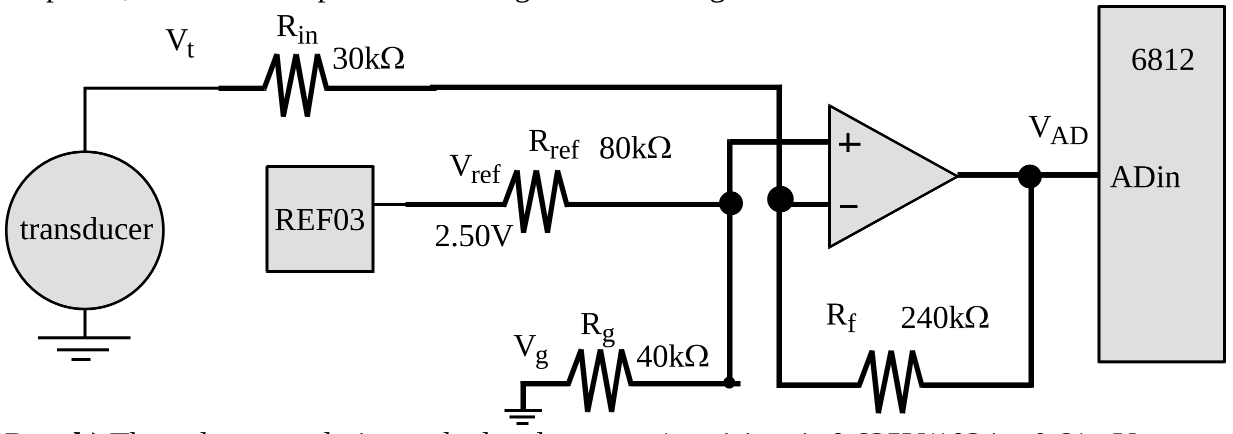
Part a) Step one, rewrite with reference chip voltage shown as a third input.

$$V_{AD} = 3\,V_{ref} - 8V_t$$

Step two, add a ground as a third input, with a gain such that the sum of the gains is 1.

$$V_{AD} = 3\,V_{ref} - 8V_t + 6\,V_g$$

Step three, choose a feedback resistor which is a common multiple of 3,6,8. R_f = 240kΩ.
Step four, select three input resistors to get the desired gains.



Part b) The voltage resolution, calculated as range/precision, is 0.625V/1024 = 0.61mV