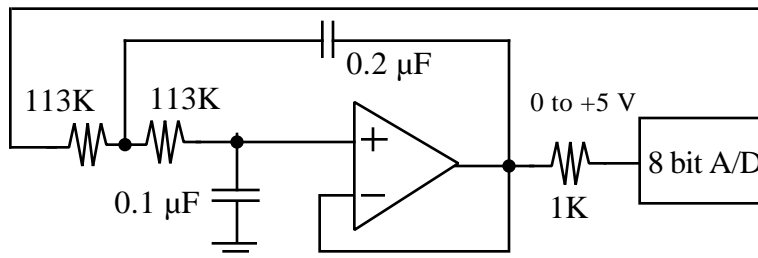
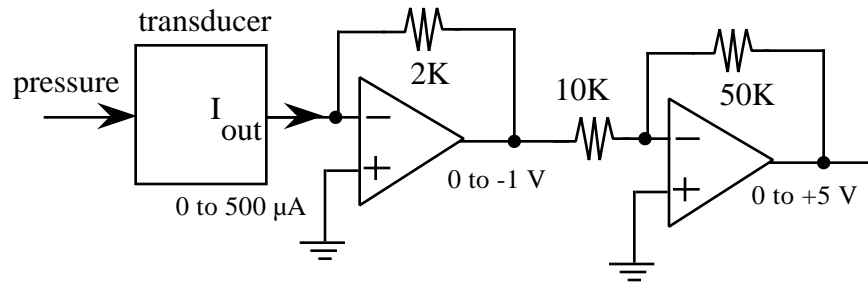


Jonathan W. Valvano May 15, 1998, 9am-12noon

(35) **Question 1.** The objective of this problem is to develop a pressure monitoring system.

(20) **Part a)** Show the analog interface between the transducer output (current) and the A/D input.



2 pole Butterworth
 0) Start with R's=10 K
 C's 70.7 and 141.4 μF
 1) frequency scale to 10 Hz
 $70.7 \mu F / (2 \cdot 10) = 1.13 \mu F$
 2) divide C's by 11.3
 multiply R's by 11.3

(4) **Part b)** To remove aliasing of high frequency signals (e.g., noise) that exists above $1/2 f_s$.

(3) **Part c)** Transducer sensitivity = $500 \mu A / 1000 \text{ dynes/cm}^2 = 0.5 \mu A / \text{dynes/cm}^2$.

(4) **Part d)** We make the analog circuit noise less than a 1/2 resolution, or $5v/512 = 10 \text{ mV}$.

(4) **Part e)** The system resolution is range/precision = $1000 \text{ dynes/cm}^2 / 256 = 4 \text{ dynes/cm}^2$.

(35) **Question 2.** Interface a 16 bit D/A converter directly to the MC68HC812A4 address/data bus.

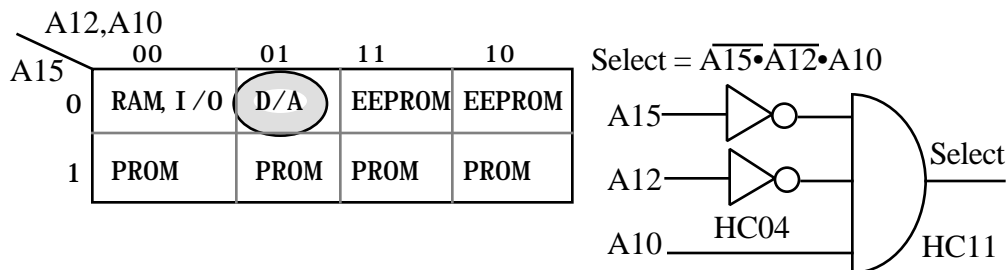
(5) **Part a)** Design the minimal cost, positive logic address decoder for this device

\$0000 to \$01FF	I/O registers	0000	000X	XXXX	XXXX
\$0800 to \$0BFF	internal RAM	0000	10XX	XXXX	XXXX
\$0C00 to \$0C01	your D/A	0000	1100	0000	000X
\$1000 to \$1FFF	internal EEPROM	0001	XXXX	XXXX	XXXX
\$8000 to \$FFFF	external PROM	1XXX	XXXX	XXXX	XXXX

We can use A10 to separate the D/A from the I/O registers, and internal RAM

We need A12 to separate the D/A from the internal EEPROM

We need A15 to separate the D/A from the external PROM



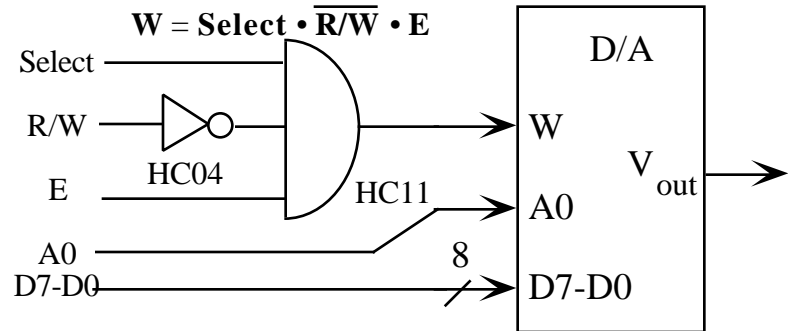
(5) **Part b)** write data available = $(62.5 + 13,125 + 14) = (62.5 + 40,125 + 25) = (102.5, 150)$

(5) **Part c)** write data required = $(W - t_{su}, W)$.

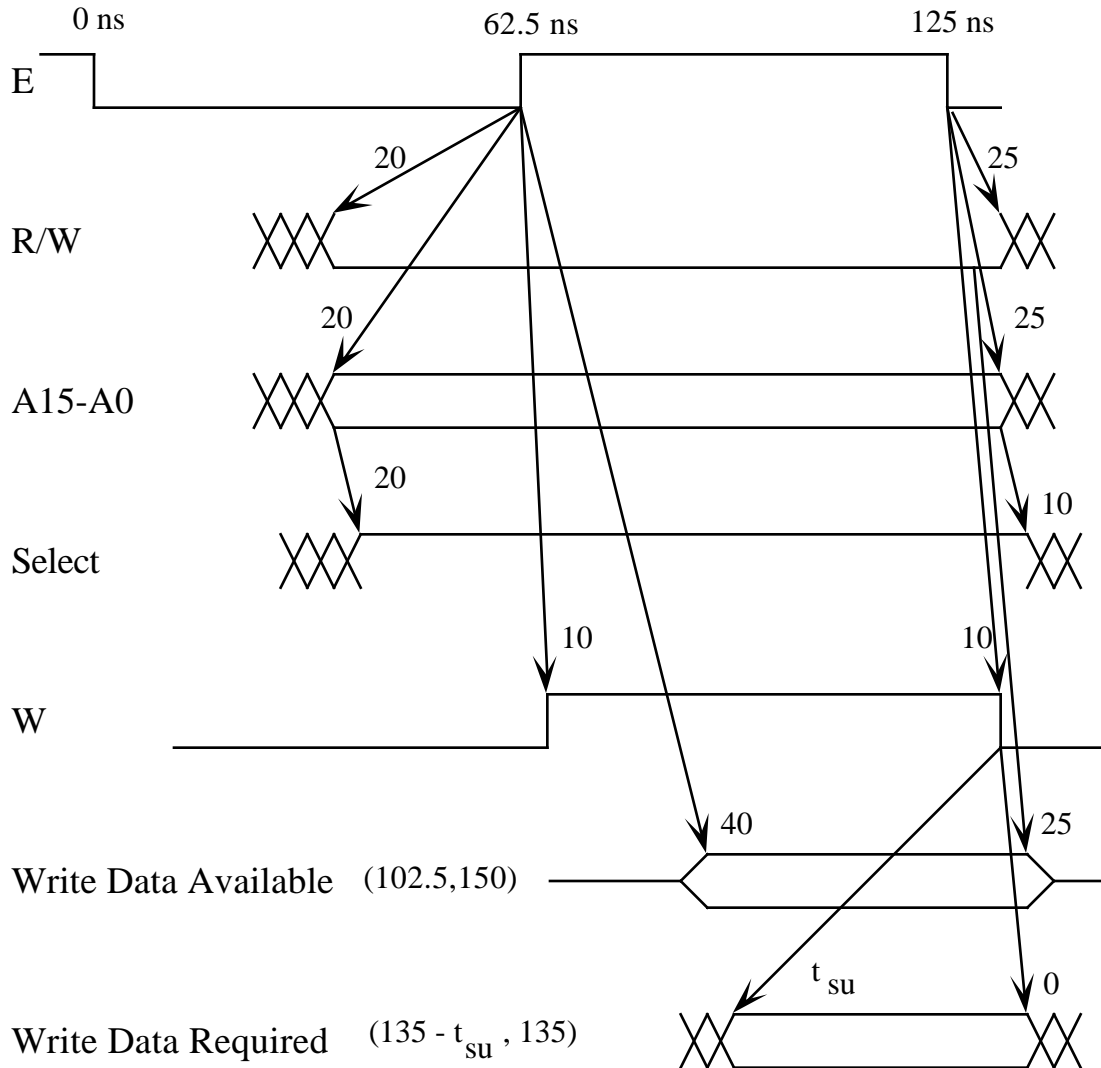
(5) **Part d)** W = Select, R/W, and E as inputs and W as output

Select	R/W	E	W	meaning
0	X	X	0	off, wrong address
1	0	0	0	
1	0	1	1	synchronized write access
1	1	0	0	off, read access
1	1	1	0	off, read access

(5) **Part e)** Show digital circuit for the interface between the 6812 and the D/A.



(5) Part f) The combined write cycle timing diagram.



(5) Part g) $135 - t_{su} = 102.5$, so $t_{su} = 32.5$ ns

(30) Question 3. You will write software to create a 3-phase digital waveform.

(5) Part a) The following factors determine the minimum allowable value for Period:

- 1) The 6812 E clock. The faster the computer, the more instructions/sec it executes;
- 2) Other interrupt threads. When the computer runs with interrupts disabled, it can not service this.
- 3) The number of cycles it takes to service the OC5 interrupt including pushing registers, and executing the TOC5Handler(). Related to this is the efficiency of the C compiler.

(25) Part b) Like the stepper motor but with six patterns: 5,1,3,2,6,4.

// The three phases are connected to PORTB bits 2, 1, 0

```
const struct State
{   unsigned char Out;           /* Output for this state */
    const struct State *Next; /* Next state */
};
typedef struct State StateType;
typedef StateType * StatePtr;
StateType motor[6]={ // circular linked list
    { 5, &motor[1]},
    { 1, &motor[2]},
    { 3, &motor[3]},
    { 2, &motor[4]},
    { 6, &motor[5]},
    { 4, &motor[0]}};
StatePtr Pt; /* Current State */
#define OC5 0x20
unsigned int sixth; // 1/6 of period in  $\mu$ sec
void ritual(void) {
asm(" sei "); // make atomic
  DDRB=0xFF; // 6812 only
  Pt=&motor[0];
  TIOS|=OC5; // enable OC5
  TSCR|=0x80; // enable
  TMSK2=0x33; // 1  $\mu$ s clock
  TMSK1|=OC5; // Arm output compare 5
  TFLG1=OC5; // Initially clear C5F
  TC5=TCNT+100; // First one right away
asm(" cli "); }
(10) Part c) Show the interrupt software which maintains the 10 by 7 matrix display.
#pragma interrupt_handler TOC5handler()
void TOC5handler(void) {
  PORTB=Pt->Out; // new motor output
  if(Pt->Out==5) sixth=(Period+3)/6;
  TC5=TC5+sixth; // Executed 6 times faster than period
  Pt=Pt->Next; // circulates around linked list
  TFLG1=OC5; // ack OC5F
}
```