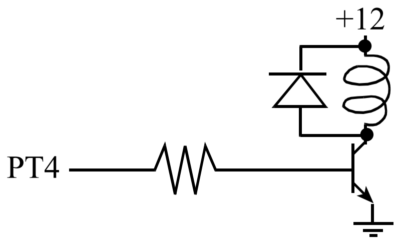
Jonathan W. Valvano April 10, 2006, 1:00pm-1:50pm.



(15) Question 1. The smallest possible $\mathbf{t_1}$ clock period is 500ns

Data available = $(t_2,t_1+t_2) = ([10,200],t_1+[10,200]) = (200,t_1+10)$ worst case

Data required = $(t_1/2-t_3,t_1/2+t_4) = (t_1/2-50,t_1/2+20)$

Make Data Available overlap Data Required

$200 \le t_1/2-50$ and $t_1/2+20 \le t_1+10$

$500 \le t_1$ and $20 \le t_1$

(15) Question 2. The resistor protects the TIP120 and 6812 if the motor were to short-circuit.

(15) Question 3. Design, equation and implementation

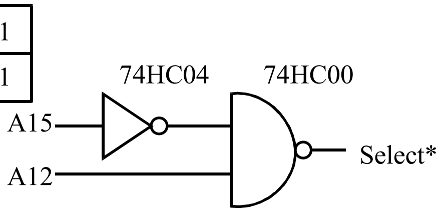
```
RAM        $6000-$67FF 0110,0XXX,XXXX,XXXX
YourDevice $7400-$77FF 0111,01XX,XXXX,XXXX
ROM        $C000-$FFFF 11XX,XXXX,XXXX,XXXX
```

Choose address lines A15,A12

Kmap, place 0 to activate, 1 to deactivate

A12 \ A15	0	1
0	1	1
1	0	1

Equation (either) $Select* = \overline{\overline{A15} \bullet A12} = A15 + \overline{A12}$



(5) Question 4. 1cm/1024

0.001cm

(10) Question 5. Give the proper values See Figure 7.40

CPHA = 0

CPOL = 0

If the memory interface were to be unsynchronized, the data out of the memory would conflict with the address out of the 6811 during the first half of the cycle (when E=0) during a read cycle.

(+5) Question 6. Answer A, B, C, D, E, or F

D

(20) Question 7a. Show the `InitFSM()` function

```
void InitFSM(void){
  asm sei              // make ritual atomic
  Pt = S0;             // Initial state
  DDRM   = 0x3E ;      // PM0 is input,PTM5-1 output
  TIOS   = 0x20 ;      // activate TC5 as output compare
  TSCR1 = 0x80 ;       // Enable TCNT
  TSCR2 = 0x02 ;       // prescale, 1MHz
  TIE    = 0x20 ;      // arm
  TFLG1 = 0x20 ;       // clear C5F
  PTM    = Pt->Out;          // output in first state
  TC5    = TCNT+Pt->Time;    // time to wait in first state
  asm cli              // enable
}
```

(20) Question 7b. Show the `output compare 5 ISR` that executes the finite state machine.

```
void interrupt 13 OC5handler(){
unsigned char in;      // 0 or 1
  TFLG1 = 0x20;        // acknowledge, clear C5F flag
  in = PTM&0x01;       // Input=0 or 1
  Pt = Pt->Next[in];   // Next state depends on the input
  PTM = Pt->Out;
  TC5 = TC5+Pt->Time;
}
```