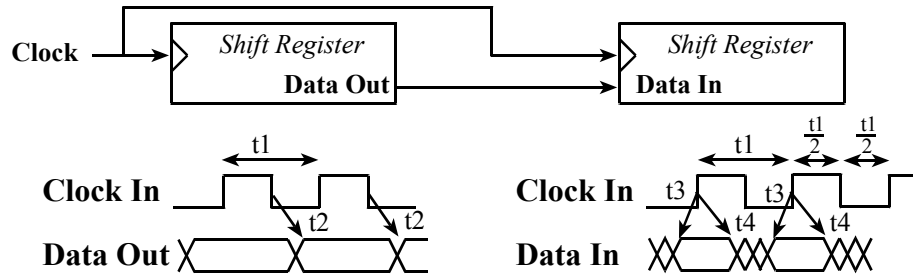


Place your answers on pages 3 and 4.

(15) Question 1. The output of one shift register is connected to the input of a second shift register. The two registers are controlled by the same 50% duty cycle **Clock**. The period of the clock is t_1 . The data is shifted out on the falling edge of **Clock**. The time $t_2 = [10\text{ns min}, 200\text{ns max}]$ is the delay from the falling edge of **Clock** until when the output is valid. The data is shifted into the second register on the rising edge of **Clock**. The time $t_3 = 50\text{ns}$ is the time before the rising edge of **Clock** that the data must be valid. The time $t_4 = 20\text{ns}$ is the time after that same rising edge of **Clock** that the data must continue to be valid. What is the smallest t_1 clock period that reliably transverses data from one shift register to the other?



(15) Question 2. The objective of this question is to interface a DC motor to the 6812 using one TIP120 Darlington NPN transistor. You may add simple components like capacitors, diodes and resistors, but no additional transistor devices other than the one TIP120. Show all connections between the 6812 PT4 and the motor. The DC motor requires a voltage between 10V and 12V at 500mA to spin. The motor has a 0.005mH inductance. You do not have to specify values of the resistors, capacitors, and diodes, just give the circuit.

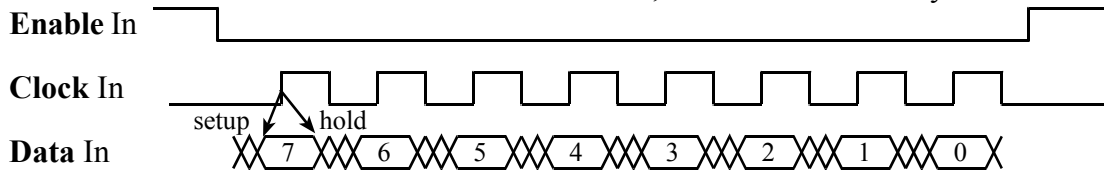
(15) Question 3. Design a minimal-cost negative-logic address decoder for *YourDevice* in the following system. The inputs are A15, A14, ...A0 and the output is **Select***. Negative logic means **Select*=0** when *YourDevice* should be activated, and **Select*=1** when *YourDevice* is deactivated.

RAM	\$6000-\$67FF	
<i>YourDevice</i>	\$7400-\$77FF	
ROM	\$C000-\$FFFF	

- Show
- 1) design steps,
 - 2) equation
 - 3) circuit.

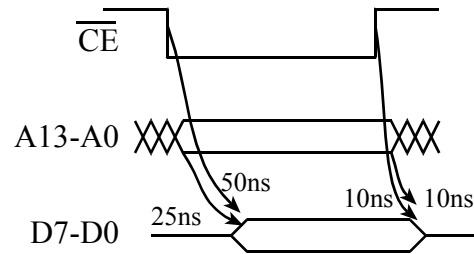
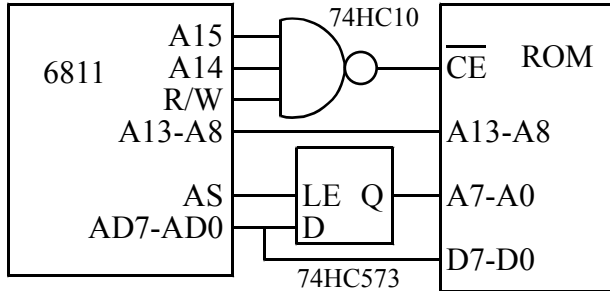
(5) Question 4. Assume you have a 6812-based system that measures distance using its 10-bit ADC, in such a way that the full-scale distance 0 to 1 cm is mapped in a linear manner to the 0 to +5V ADC range. The distance signal varies over time with frequency components from 0 to 25 Hz. What is the expected measurement resolution in cm for this system, assuming the errors are limited by the ADC?

(10) Question 5. A slave device will be interfaced to the master 6812 using SPI. There are three signals that will be outputs of the 6812 and inputs to the device (**Enable**, **Clock**, and **Data**). The timing of the external device is shown below. What CPHA, CPOL mode should you use?

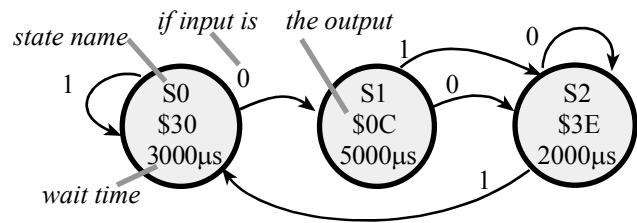


(+5 point extra credit) Question 6. The goal of the circuit is to interface a 16K ROM to a 6811 running at 2MHz. Full address decoding is supposed to place the ROM at addresses \$C000 to \$FFFF. The R/W signal is supposed to activate the ROM only for read cycles. The AS is supposed to latch the low address into the 74HC573 so that they are available throughout the cycle. The read access time of the ROM is 50ns from the fall of CE* and 25ns from address valid. This ROM interface does not operate properly. Give the correct reason why this interface is improperly designed. Place the letter code on the answer sheet.

- A) The beginning of RDA does not occur before the beginning of RDR
- B) The end of RDA does not occur after the end of RDR
- C) The address decoder is incorrect (i.e., it responds to addresses other than \$C000-\$FFFF)
- D) The E clock is missing, and thus it won't work because it is not synchronized
- E) The R/W signal is incorrectly used (i.e., it responds to write cycles rather than read cycles)
- F) None of the above, because this system actually operates correctly.



(40) Question 7. You will implement this Moore finite state machine. There is one digital input signal (connected to PTM bit 0) and five digital output signals (connected to PM5-1). The sequence is output, wait, input, go to next state, output, wait, input, go to next state, ... where the waiting occurs using **output compare interrupt 5**. You may assume the system is running at 4 MHz, i.e., the PLL was not activated. State S0 is the initial state. *For this question, don't worry about being friendly.* You will write the entire software system to run this FSM. You must use the following data structure that defines the FSM. After initialization, all input, output, and waiting occur in the output compare interrupt service routine. You cannot call any functions, unless you explicitly define those functions in your solution. The main program and FSM data structure will be as follows, and these cannot be changed.



```

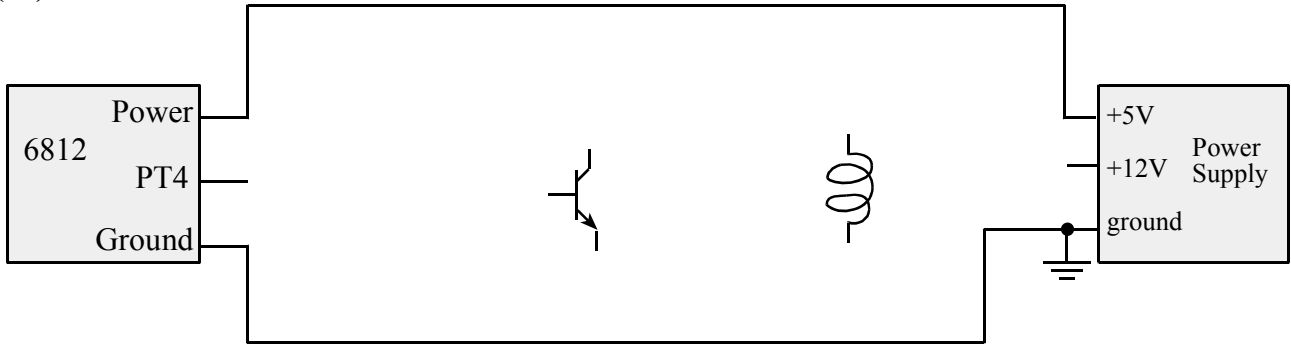
const struct State{
    unsigned short Time;           // Time in usec to wait
    unsigned char Out;            // Output to Port M, pins PTM5,4,3,2,1
    const struct State *Next[2];  // Next if input=0,1
};
typedef const struct State StateType;
#define S0 &fsm[0]
#define S1 &fsm[1]
#define S2 &fsm[2]
StateType fsm[3]={
    // no shifting required on the PTM output
    {3000,0x30,{S1,S0}}, // S0, make PTM5=1,PTM4=1,PTM3=0,PTM2=0,PTM1=0
    {5000,0x0C,{S2,S2}}, // S1, make PTM5=0,PTM4=0,PTM3=1,PTM2=1,PTM1=0
    {2000,0x3E,{S2,S0}} // S2, make PTM5=1,PTM4=1,PTM3=1,PTM2=1,PTM1=1
};
StateType *Pt; // Current State
void main(void){
    InitFSM(); // initialize Pt, PTM, and OC5 interrupts
    for(;;) {};
}
    
```

Other than the for(;;) statement in this main program, there can be NO backward jumps in this solution.

Jonathan W. Valvano First: _____ Last: _____
 April 10, 2006, 1:00pm-1:50pm. You can use the textbook, but no other materials. You have 50 minutes, so please allocate your time accordingly. **Please read the entire quiz before starting.** Only this piece of paper (pages 3 and 4) will be turned in.

(15) Question 1. What is the smallest possible t_1 clock period?

(15) Question 2. Show the interface circuit.



(15) Question 3. Show design steps, equation and implemented digital logic.

(5) Question 4. Measurement resolution

(10) Question 5. Give the proper values
 CPHA =
 CPOL =

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

(20) Question 7a. Show the `InitFSM()` function that initializes output compare 5 and the FSM. Additional globals are allowed. The first output occurs during this initialization.

```

void InitFSM(void) {
    asm sei                // make ritual atomic

    DDRM = ; // PM0 is input,PTM5-1 output
    TIOS = ; // activate TC5 as output compare
    TSCR1 = ; // Enable TCNT
    TSCR2 = ; // prescale
    TIE = ; // arm
    TFLG1 = ; // clear C5F
    PTM =
    TC5 =

}

```

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

```

void interrupt 13 OC5handler() {

```