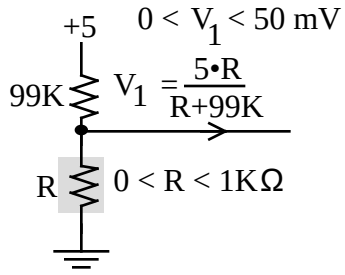
Jonathan W. Valvano First Name: _____ Last Name:_____
December 11, 1998, 2-5pm

This is an open book, open notes exam. You may put answers on the backs of the pages, but please don't turn in any extra sheets. You have 3 hours, so please allocate your time accordingly.

(20) Question 1. Design a wind direction measurement instrument using the 8-bit A/D converter. You are given a transducer that has a resistance that is linearly related to the wind direction. As the wind direction varies from 0 to 360 degrees, the transducer resistance varies from 0 to 1000 Ω. The frequencies of interest are 0 to 0.5 Hz, and the sampling rate will be 1 Hz.

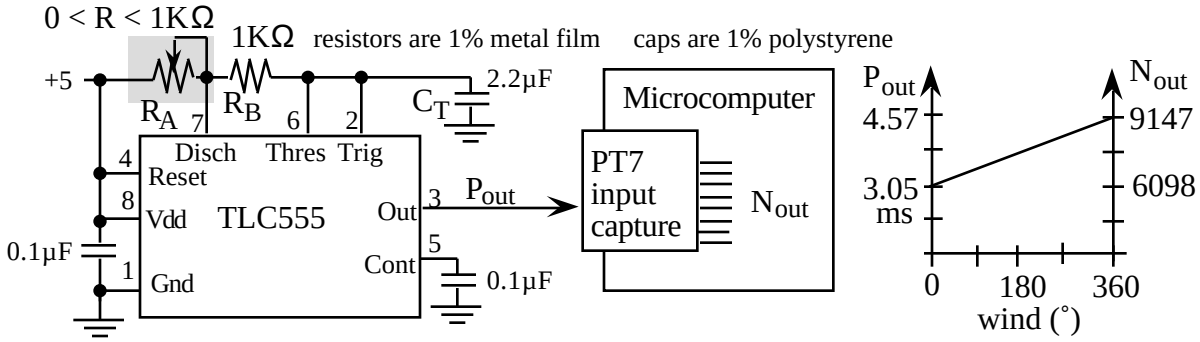
(10) Part a) Show the analog interface between the transducer and the A/D port channel 7. Only the +5 V supply can be used. Show the +5V power connections. Give chip numbers but not pin numbers. Specify the type and tolerance of resistors and capacitors. **YOU MUST USE A/D CHANNEL 7.**



PAD7

(10) Part b) Write the ritual and gadfly function/subroutine that measures the wind direction and returns a 16 bit unsigned result with units of degrees. I.e., the value varies from 0 to 359. (You do not have to write software that samples at 1 Hz, simply a function that measures wind direction once.) There are no interrupts, no FIFO queues, no global variables and no subfunctions. Simply write two functions, `void ritual(void)`, and `unsigned int MeasWind(void)`.

(20) Question 2. Design a wind direction measurement instrument using the input capture technique. Again, you are given a transducer that has a resistance that is linearly related to the wind direction. As the wind direction varies from 0 to 360 degrees, the transducer resistance varies from 0 to 1000 Ω. The frequencies of interest are 0 to 0.5 Hz, and the sampling rate will be 1 Hz. One way to interface the transducer to the computer is to use an astable multivibrator like the 555. The period of a 555 timer is $0.693 \bullet C_T \bullet (R_A+2R_B)$. In our circuit, R_A is the transducer, R_B is a fixed 1KΩ precision resistor, and C_T is a fixed precision 2.2µF capacitor. The output is connected the Input Capture port channel 7. The following hardware can not be modified. **YOU MUST USE CHANNEL 7.**



Write the ritual and gadfly function/subroutine that measures the wind direction and returns a 16 bit unsigned result with units of degrees. I.e., the value varies from 0 to 359. (You do not have to write software that samples at 1 Hz, simply a function that measures wind direction once.) There are no interrupts, no FIFO queues, no global variables and no subfunctions. Simply write two functions, `void ritual(void)`, and `unsigned int MeasWind(void)`.

(15) Question 3. Consider the 8K by 8 bit static RAM interface using the MCM60L64. Assume the MC68HC812A4 is running at 8 MHz in expanded narrow mode. The objective of this problem is to develop the specifications that allow the RAM interface to operate without cycle stretching. You may use timing equations or timing diagrams to justify your answers.

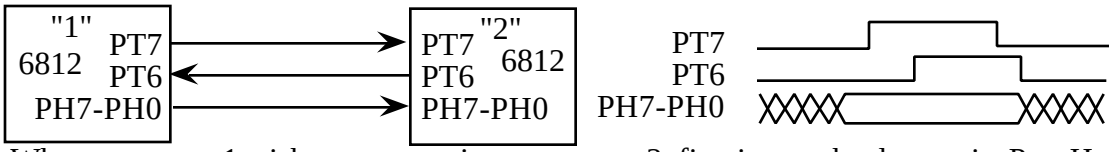
(10) Part a) Develop the equations that specify the maximum allowable t_{AVQV} and the maximum allowable t_{E1LQV}.

(5) Part b) Develop the equations that specify the maximum allowable t_{DVWH}.

(5) Question 4. Briefly explain the difference between the three modes
single chip,
expanded narrow, and
expanded wide
as they apply to memory interfacing. Give one sentence for each mode.

(10) Question 5. Briefly explain why it is important to choose the proper update rate for a fuzzy logic controller. In particular, explain what happens to a fuzzy logic controller if the controller is executed too infrequently. Similarly, explain what happens to a fuzzy logic controller if the controller is executed too frequently.

(30) Question 6. The objective of this problem is to develop a message passing facility that spans across two computers. A fully interlocked synchronization method will be implemented. The message is a simple 8-bit byte that is passed from the output port of computer 1 to the input port of computer 2. The following hardware connections are fixed. You may assume both computers initialize together where computer 1 initializes its PT7 to zero and computer 2 initializes its PT6 to zero.



When computer 1 wishes to transmit to computer 2, first it puts the data on its Port H outputs. Next computer 1 makes a rising edge on PT7 signifying new data is available. Computer 1 will then wait for a rising edge on PT6 signifying the data has been accepted. Next, computer 1 makes its PT7 low. Lastly, it waits for a low signal on its PT6 input line.

Part a) Write the software for the transmitting computer 1. The transmission will occur in the background using input capture interrupts. You may use a FIFO queue without writing the three routines, `InitFifo, PutFifo, and GetFifo`. You may ignore FIFO full errors. There are three routines you will write:

1) `void Ritual(void)` that clears the FIFO, makes the PT7 output low and arms/enables the input capture interrupts as appropriate.
2) `void PutMsg(unsigned char data)` function that is called by the main program (that you do not write). If a current message is in progress, then this data is entered in the FIFO. If there is no current message in progress, then one is started with the new data.
3) `void IC6Handler(void)` interrupt handler called by the input capture on PT6.

When computer 2 wishes to receive from computer 1, first it waits for a rising edge on its PT7 input. Then it gets the data from its Port H inputs. Next computer 2 makes a rising edge on its PT6 output signifying the data has been accepted. Computer 2 will then wait for a falling edge on its PT7 input. Lastly, it makes a low signal on its PT6 line.

Part b) Write the software for the receiving computer 2. The reception will occur in the background using input capture interrupts. You may use a FIFO queue without writing the three routines, `InitFifo, PutFifo, and GetFifo.` You may ignore FIFO full errors. There are three routines you will write:

1) `void Ritual(void)` that clears the FIFO, makes the PT6 output low and arms/enables the input capture interrupts as appropriate.
2) `unsigned char GetMsg(void)` function that is called by the main program (that you do not write). If the FIFO is empty, then this routine will call the `GetFifo()` routine over and over until data is available.
3) `void IC7Handler(void)` interrupt handler called by the input capture on PT7.