Jonathan W. Valvano First:_____ Last:_____
April 7, 2000, 11:00am-11:50am

This is an open book, open notes exam. You must put your answers on these pages only, you can use the back. You have 50 minutes, so please allocate your time accordingly. ***Please read the entire quiz before starting.***

(20) Question 1. The following IIR digital filter has a bug. It is supposed to implement the following
y(n) = 0.49•x(n)+0.51•y(n-1)
The ***8-bit*** input data (from `A2D(0)`) is bounded between 0 and 255. Since the filter gain is less than one, the y(n) values are also bounded between 0 and 255. Explain why the filter doesn't work, then ***edit*** it so it does work. No floating point calculations are allowed, and you must implement the IIR filter without approximations.

```
unsigned char x,y; // 8-bit unsigned numbers, 0 to 255
#define C5F  0x20
#pragma interrupt_handler TC5Handler()
void TC5Handler(void){
   TFLG1=C5F;         // ack interrupt
   TC5=TC5+8333;      // fs=240Hz
   x = A2D(0);        // new 8-bit data, 0 to 255
   y=(x/100)*49+(y/100)*51;}
```

(20) Question 2. This second implementation of the IIR filter has a bug too. It too is supposed to implement
y(n) = 0.49•x(n)+0.51•y(n-1)
This time a 12-bit ADC is used. The ***12-bit*** input data (from `A2D12(0)`) is bounded between 0 and 4095. Since the filter gain is less than one, the y(n) values are also bounded between 0 and 4095. Explain why the filter doesn't work, then ***edit*** it so it does work. No floating point or approximations are allowed.

```
unsigned short x,y; // 12-bit unsigned numbers (stored in 16-bit variables), 0 to 4095
#define C5F  0x20
#pragma interrupt_handler TC5Handler()
void TC5Handler(void){
   TFLG1=C5F;         // ack interrupt
   TC5=TC5+8333;      // fs=240Hz
   x = A2D12(0);      // new 12-bit data, 0 to 4095
   y=(49*x+51*y)/100;}
```

(20) Question 3. The following real time system calls `ProcessDAS()` 1000 times a second. All goes well as long as the time to execute `ProcessDAS()` is always less than 1 ms. If the function `ProcessDAS()` were to take more than 1 ms to execute then data will be lost. In this problem you will add debugging instruments to measure the execution speed of `ProcessDAS()` and count the number of times it takes more than 1 ms to execute. You may assume that the execution speed is always less than 64 ms.

```
#pragma interrupt_handler TC5handler()

void TC5handler(void){

   TFLG1=0x20;       // ack C5F

   TC5=TC5+1000;     // fs=1000Hz

   ProcessDAS();

}

void ritual(void) {

asm(" sei");     // make atomic

  TIOS|=0x20;    // enable OC5

  TSCR|=0x80;    // enable

  TMSK2=0x33;    // 1 us clock

  TMSK1|=0x20;   // Arm output compare 5

  TFLG1=0x20;    // Initially clear C5F

  TC5=TCNT+1000;

asm(" cli"); }
```

(40) Question 4. In this problem you will design an auto-ranging digital voltmeter. The overall range will be 0 to 2550 mV, but the resolution will vary depending on the actual voltage value. You will use a higher gain (better resolution) channel for the smaller voltages. In particular, there will be three possibilities

minimum voltage (mV)	maximum voltage (mV)	resolution (mV)	precision
0	25.5	0.1	256
25.5	255	1	230
255	2550	10	230

You will design three separate analog circuits, one for each range. Analog low pass filters are not required. The three analog circuits will get their inputs from the same unknown signal to be measured, V_{in}. But, they will deliver their outputs to three separate ADC channels on the microcomputer. The software, using MULT mode, will convert all three channels and use the digital results from the channel with the highest gain (better resolution) that has valid data (not saturated to 255.)

(20) Part a) Show the analog hardware interface between the V_{in} input and microcomputer ADC. Label all chip numbers, but not pin numbers. Show all connections to power supply voltages. Specify the type and tolerances of all resistors and capacitors.

(10) Part b) Show the ritual that initializes the system. In this problem, you will be writing the ADC conversion software, and not the real time DAS, so interrupts are not needed. You only need to initialize the ADC port.

(10) Part c) Show the gadfly function that measures the three channels, and sets the following two global variables to represent the unknown analog input.

```
unsigned char voltage; // unsigned 8 bit
char exponent;         // signed 8-bit
```

The operation of the system is illustrated in the following example table

V_{in}(mV)	`voltage`	`exponent`	meaning
0	0	-1	$0 \bullet 10^{-1}$
1.234	12	-1	$12 \bullet 10^{-1}$
12.34	123	-1	$123 \bullet 10^{-1}$
123.4	123	0	$123 \bullet 10^{-0}$
1234	123	1	$123 \bullet 10^{+1}$
2550	255	1	$255 \bullet 10^{+1}$