Jonathan W. Valvano First Name: ________________ Last Name:_____________________
December 17. 2001, 9 am to 12 noon

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.

(20) Question 1. Design an analog circuit with the following specifications

differential input	(V_1-V_2), neither V_1,V_2 is ground
transfer function	$V_{out} = 50\bullet(V_1-V_2) + 2.5$
constrained input	$-0.05 \le (V_1-V_2) \le 0.05V$
constrained output	$0 \le V_{out} \le +5V$
large input impedance	$Z_{in} \ge 1\ M\Omega$

Give chip numbers but not pin numbers. Specify all resistor values. You may use +12 and –12V analog supply voltages.

(20) Question 2. Consider a 128K by 8 bit static RAM interface between the MC68HC812A4 and 628128. The MC68HC812A4 is running at 8 MHz. Assume the same hardware as Lab 25. The objective of this problem is to develop the specifications that allow this RAM to operate without cycle stretching. Let $\mathbf{t_a}$ be the time delay from address valid to data out valid during a read cycle. $\mathbf{t_a}$ is also the time delay from chip enable low to data valid during a read cycle. Although the write cycle is important, we will neglect it for this problem.

Part a) Draw a combined read timing diagram assuming no cycle stretching. Show E, CSD (RAM chip enable), address, R/W (RAM WE), data available and data required. Clearly label $\mathbf{t_a}$.

Part b) Use the timing diagram to determine the maximum allowable access time $\mathbf{t_a}$. In other words, how fast would the RAM chip have to be to operate without cycle stretching?

(20) Question 3. The objective of this problem is to develop the fixed-point equations that implement the following PI position controller.

X(t) is the state variable (m)
X* is the desired state variable (m)
e(t)=X*-X(t) is the error (m)
V(t) is the actuator command (W)

$$V(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau$$

where K_P is 0.0512 (W/m)
K_I is 0.0408 (W/m-sec)

The state estimator gives you a digital sample, `x(n)`, as a decimal fixed-point number with Δ equal to 0.01 m. For example, if X(t) equals 1.234 m, then `x(n)` will be 123. `xstar`, the desired state variable is given (also as a decimal fixed point number with Δ equal to 0.01 m.) Your PID software will calculate a signed 16-bit integer output, `u(n)`, that will be fed to the actuator interface. `u(n)` is a decimal fixed-point number with Δ equal to 0.001 W. For example, if you want V(t) to be 1.2342 W, then you should set `u(n)` to be 1234.

The digital controller is executed every 0.1s (10Hz). Show the control equation to be executed in the periodic interrupt handler. In this problem you will convert from floating to fixed-point numbers, and convert from continuous to discrete time. Explain how you would deal with overflow/underflow. No C code is required, but please give very explicit fixed-point equations showing how to calculate `u(n)` in terms of `x(n)` and `xstar`. You may define additional variables as you need them.

(10) Question 4. Interrupts are a good method to create a real-time DAS. A typical approach is illustrated in the following software skeleton. Let $\mathbf{f_s}$ be the sampling rate.

```
#pragma interrupt_handler TC5handler()
void TC5handler(void){
  TFLG1 = 0x20;       // acknowledge, clear C5F
  TC5 = TC5+Rate;     // Executed periodically
  Fifo_Put(Adin());   // save new data
}
void main(void){
  Initialization();   // clear FIFO, arm TC5, enable interrupts
  while(1){
    while(Fifo_Get(&data)){};
    Process(data);
  }
}
```

The best case, average, and worst case execution times are given in the following table. Assume all other software times can be neglected. The goal of a real-time DAS is to execute `Adin()` every 1/$\mathbf{f_s}$.

Program	Best case (min)	typical (average)	Worst case (max)
`Initialization`	912 µs	912 µs	912 µs
`Adin`	25 µsec	25 µsec	25 µsec
`Process`	500 µs	1000 µs	5000 µs
`Fifo_Put`	15 µsec	15 µsec	15 µsec
`Fifo_Get`	20 µsec	20 µsec	20 µsec

What is the maximum sampling rate possible for this interrupt-based DAS? The system must be real-time continuously processing data with no delayed or lost data points.

(10) Question 5. The 6812 is running in expanded mode with 4 megabytes of extended program page ROM. Fill in the two boxes in the following software that reads physical location $15BCD.

```
PPAGE =              ;
data = *((char *)(              ));
```

Part b) The 6812 is running in expanded mode with 1 megabyte of extended data page RAM. Fill in the two boxes in the following software that reads physical location $15BCD.

```
DPAGE =              ;
data = *((char *)(              ));
```

Part c) How can there be RAM at $15BCD and ROM at $15BCD on the same system?

(20) Question 6. Develop a device driver for an interrupt-based square-wave generator. The output will be available on PT6. You will write one public function called `Square_Start` and an OC6 handler. The main program will call your public function to set the frequency of the 50% duty-cycle wave. The possible frequencies are 1, 2, 3, … 9999, 10000 Hz. Once `Square_Start` is called the wave is continuously generated using output compare 6 interrupts. Include code for enabling, arming and the interrupt vector. Private global variables and private functions can be added. You have a MC68HC912DG60 that includes TMSK2=0x36 (8µs) and 0x37 (16µs)

Part a) Show the code for the `Square.h` header file.

Part b) Show the code for the `Square.c` implementation file.