

Jonathan W. Valvano First Name: _____ Last Name: _____
December 15, 2005, 2-5pm

This is an open book, open notes exam. You must put answers on these papers; please don't turn in any extra sheets.

(10) Question 1. Design a differential analog amplifier with a gain of 100 and an input impedance larger than $10\text{ M}\Omega$. *Differential* means there are two inputs V_1 and V_2 , and one output V_{out} , such that $V_{\text{out}} = 100 \cdot (V_1 - V_2) + 2.5$. The input voltages are constrained to $-0.025\text{V} < (V_1 - V_2) < 0.025\text{V}$. This means the output voltage will be constrained to 0 to +5V. No analog filter is required in this question. You must build this interface with TLC2274s and one REF03, using just +5V power.

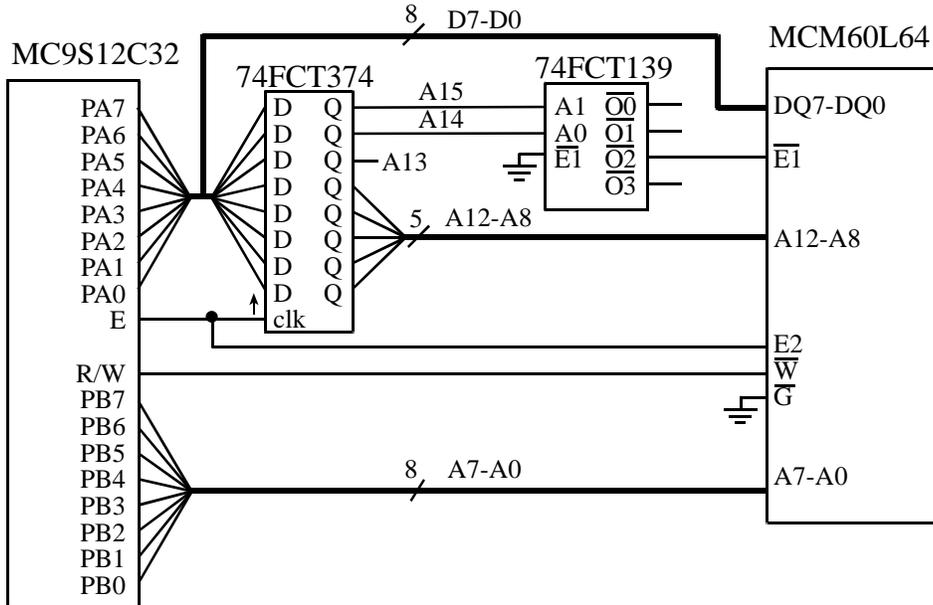
(5) Question 2. Design a two-pole analog low-pass filter with a cutoff frequency of 1125 Hz. Show the design steps, and specify resistor/capacitor values.

(5) Question 3. A 9S12C32 data acquisition system measures velocity with a range of 0 to 10 cm/sec, using an external 12-bit ADC. What number system (integer, binary fixed-point, decimal fixed-point, or floating point) would you propose to represent these velocities if the system required extensive mathematical calculations and very little human input/output? For full credit, give the value to be stored in memory when the velocity is 5 cm/sec. Give units for the number system.

(5) Question 4. A CAN system has a baud rate of 100,000 bits/sec, 29-bit ID, and four bytes of data per frame. Assuming there is no bit-stuffing, what is the maximum bandwidth of this network, in **BYTES/SEC**.

(5) Question 5. A different CAN system has a baud rate of 200,000 bits/sec and a message protocol with frame sizes that randomly vary from 75 to 100 bits. The receive process requires real-time solution, because CAN messages will stall on the network if the software does not read the message (making room in its receive queue) within a certain amount of time. Calculate this maximum allowable time between the Receive CAN Flag (RXF) being set and the execution of the CAN interrupt 38, which removes the receive message from the buffer (clearing RXF).

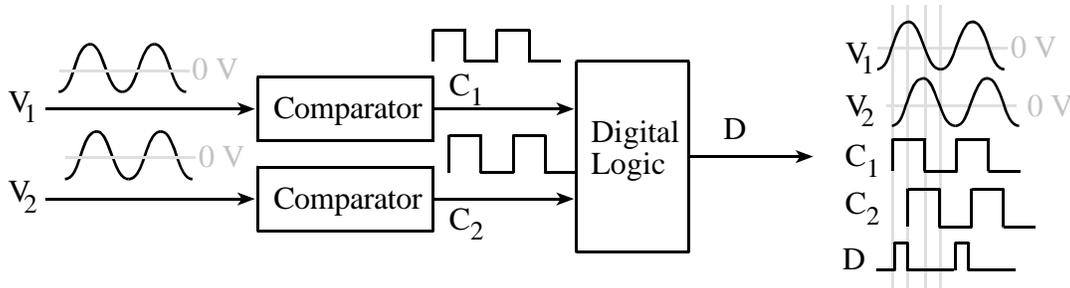
(10) Question 6. The following interface shows an external RAM interfaced to a 9S12C32. The 9S12C32 is running at 4 MHz (250ns bus cycle), with two memory stretches.



(5) Part a) The critical read timing occurs with the fall of E1. What is the worst-case read access time (maximum delay from the fall of E1 to the time read data will be available) possible for this RAM?

(5) Part b) The critical write timing occurs with the fall of E2. What is the worst-case write setup time (maximum time before fall of E2 before which the input data must be valid) possible for this RAM?

(20) **Question 7.** The goal of the system is to design a system to measure the phase between two sin waves, V_1 and V_2 . Both waves vary from -2 to +2 V around a DC value of 0 V and have a frequency of 100 Hz (period is 10ms). The phase lag from the first signal to the second signal will be limited to 0 to +90 degrees (pulse width of D varies from 0 to 2.5ms). Assume the E clock is 250ns.

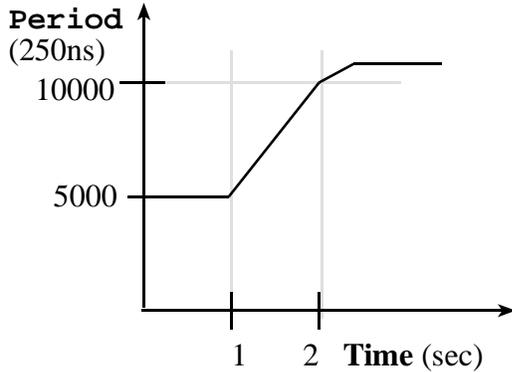


Part a) Show the hardware analog circuit connecting the two signals to a single input capture pin PT7=D. Show resistor values, capacitor values and chip numbers but not pin numbers.

Part b) Show the ritual that initializes the system.

Part c) Show the interrupt service routine that measures phase. Specify units of your measurement.

(10) **Question 8.** A DC motor will be controlled using a 9S12C32 system. The speed is measured using 16-bit input capture and has a measurement resolution of 250ns. The input capture device driver repeatedly updates a global variable, called **Period**. This 16-bit unsigned variable has units of 250ns and a range of 1000 to 65000. The 9S12C32 uses pulse-width modulation to control power to the motor. The controller software writes to a global variable, called **Duty**, which ranges from 0 (0%) to 10000 (100%). The following plot shows an experimental measurement obtained when **Duty** is changed from 2500 to 5000. The desired speed is stored in the global variable, **Desired**, which has the same units as **Period**. Design a fixed-point PI controller that takes **Period** and **Desired** as inputs and calculates **Duty** as an output. Show your work. How often should the controller be executed? Show just the equations (no software or hardware is required), calculating **Duty** as a function of **Period** and **Desired**.



(20) **Question 9.** Assume we have a memory manager that allocates RAM blocks. E.g., if we needed 200 bytes we'd call `pt=alloc(200)`. The goal of this question is to design a `fork()` function that creates a second copy of the executing thread. The new thread (child thread) is an exact copy of the calling thread (parent thread). The child thread inherits the local variables with current values from the parent thread. But, once created, the parent and child have separate stacks and separate TCBs. You may assume all of the Lab17 starter software is available (you can access/call any Lab 17 starter function/variable without explicitly copying it onto the exam paper). Basically, one thread calls `fork` but two threads return. Upon successful completion, `fork()` returns `0` to the child thread and returns the **thread ID** of the child thread to the parent thread. This example user program will spawn a child thread, where the parent thread runs the slow version of square-root and the child runs the fast version of square-root.

```
void Math(void){ unsigned char me,child;
unsigned short x, n, i;
  MathWork = 0;          // one thread here
  child = fork();
  if(child){            // two threads here
    me = OS_RunId(); // parent
    for(;;) {          // slow version of square-root
      for(n = 1; n <= NUM; n++){ x=0;
        while((x*x)<(16*n)){x++;}
      }
      MathWork++;      // calculation finished
    }
  }
}
else{
  me = OS_RunId(); // child
  for(;;) {        // fast version of square-root
    for(n = 1; n <= NUM; n++){ x=16*n;
      for(i=0;i<10;i++) x = ((x*x+16*n)/x)/2;
    }
    MathWork++;    // calculation finished
  }
}
```

