

Jonathan W. Valvano October 15, 2003, 1 to 1:50pm

**(20) Question 1.** The baud rate of **100 bits/sec** yields a bit time of **10ms**.

**(10) Part a)** Latency is defined as the time from RDRF being set to the read SC0DRL. This time needs to be less than 10 bit times, which is 100 ms.

**(10) Part b)** RxFifo\_Put returns a false if data is lost

```
unsigned short CharLost=0;
#pragma interrupt_handler SciHandler
void SciHandler(void){
    if(SC0SR1 & RDRF){
        if(!RxFifo_Put(SC0DRL)){ // clears RDRF
            CharLost++;          // lost data
        }
    }
}
```

**(25) Question 2.** The only writes to these two globals occur during the first microseconds of the first launch to server. There are no synchronization/corruption problems caused by reading the globals. Therefore no semaphores are needed, and the program will run as is.

**(20) Question 3.** The thread switch system with a **status** field added to implement blocking.

**(10) Part a)** The error occurs because the instructions `sts 2,x lds 2,x` access the **Next** field instead of accessing the **StackPt** field as intended.

**(10) Part b)** The correction is to change the accesses to **StackPt**

```
void threadSwitchISR(void){
    asm(" ldx _RunPt\n"
        " sts 4,x");
    RunPt = RunPt->Next;
    PORTJ = RunPt->Id;
    TC3 = TCNT+TimeSlice;
    TFLG1 = 0x08;
    asm(" ldx _RunPt\n"
        " lds 4,x");
}
```

**(20) Question 4.** Lab 17 measured a time-jitter. It was usually a small number.

**(10) Part a)** We can define time-jitter,  $\delta t$ , as the difference between when a periodic task is supposed to be run, and when it is actually run. The goal of a DAS is to start the ADC at a periodic rate,  $\Delta t$ . Let  $t_n$  be the  $n$ th time the ADC is started. In particular, the goal to make  $t_n - t_{n-1} = \Delta t$ . The jitter is defined as the constant,  $\delta t$ , such that

$$\Delta t - \delta t < t_i - t_{i-1} < \Delta t + \delta t \quad \text{for all } i.$$

**(10) Part b)** Let  $dV/dt$  be the maximum slew rate of the input.  $\delta V = (dV/dt) * \delta t$

**(15) Question 5.** There is a read-modify-write critical section involving the semaphore counter. If the semaphore is equal to 1, and two threads try and call **OS\_wait**, the proper action is to let the first one pass and make the second one spin. If **OS\_wait** does not disable interrupts, then is it possible for both threads to proceed.