Jonathan W. Valvano October 8, 2010, 2:00pm-2:50pm.

(10) Question 1. Debugging dump.
Part a) This debugging monitor is *minimally intrusive*.
Part b) When defining intrusiveness, it is better to quantify exactly how much change the instrument affects the system being tested. This code only takes 20 cycles to execute. As a fraction, this is only 20/8000.

(bonus) For an inductor, V = L dI/dt (FYI, the capacitor is I = C dV/dt)

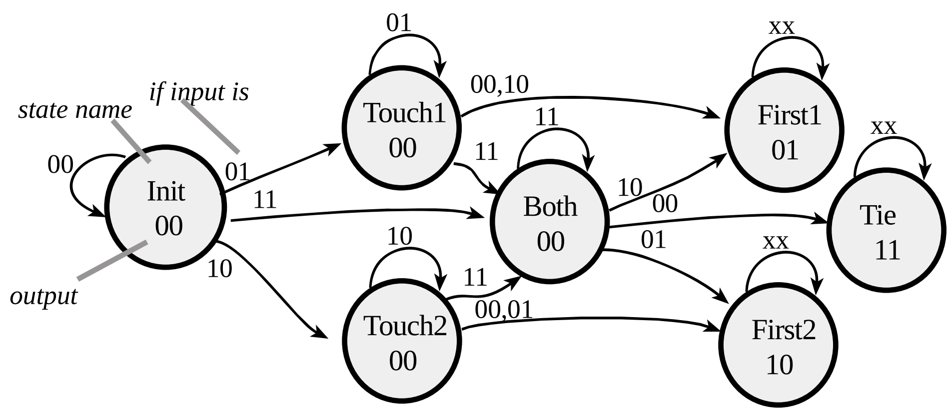
(15) Question 2. Critical sections may occur when there is a nonatomic read-modify-write access to a shared global. One cannot tell if software like `bReg = ~bOC1;` will be atomic without seeing the assembly code the compiler produces. The machine is not atomic with respect to lines of C, rather most assembly instructions are atomic.

```
short bOC1=0; // true if first ISR occurred first
short bReg=0; // true if first call to function occurred first
interrupt 9 void IC1handler(void){
  if((bReg==0)&&(bOC1==0)) bOC1 = 1; // interrupt first
  TFLG1 = 0x02;       // acknowledge OC1
  Stuff1();
}
void RegularFunction(void){
asm sei   // make atomic, to remove critical section
  if((bReg==0)&&(bOC1==0)) bReg = 1; // function first
asm cli
  Stuff2();
}
```

Solution that does NOT disable interrupts

```
short bOC1=0;    // true if first ISR occurred first
short bReg=0;    // true if first call to function occurred first
char bLooking=1; // true until first occurrence
interrupt 9 void IC1handler(void){
  if(bLooking){
   bLooking = 0; // stop looking
   bOC1 = 1;     // interrupt was first
  }
  TFLG1 = 0x02;       // acknowledge OC1
  Stuff1();
}
void RegularFunction(void){
asm   clra            ;new value for bLooking
asm   ldx  #bLooking  ;pointer to bLooking
asm   minm 0,x        ;in either case bLooking is now 0, carry set if used to be 1
asm   bcc  skip       ;skip if second program to execute
asm   movw #1,bReg    ;regular function first
asm skip:
  Stuff2();
}
```

(20) Question 3. Draw a Moore FSM graph to solve the following problem.



(20) Question 4. The goal is to eliminate jitter. The proper solution occurs when one interrupt is time-shifted by 500 µs from the other interrupt.

Part a) Let t be an arbitrary time reference in ms. The OC1 interrupts occur (every 1 ms) at t+1, t+2, t+3, … The OC2 interrupts occur (every 2 ms) at t+1.5, t+3.5, t+5.5, …

```
void Question5_Init(void){
  TSCR1 = 0x80;    // Enable TCNT 8 MHz in run mode
  TSCR2 = 0x03;    // divide by 8 TCNT prescale, 1us
  TIOS |= 0x06;    // activate TC1,TC2 as output compares
  TIE |= 0x06;     // arm OC1, OC2
  TC1 = TCNT+1000; // First OC1 in 1ms
  TC2 = TC1+500;   // First OC2 in 0.5ms after the OC1 interrupt
  TFLG1 = 0x06;    // clear both flags
  asm cli
}
```

Part b) If you acknowledge using `TFLG1 |= 0x02;`, or `TFLG1 |= 0x04;` it will clear all 8 bits of `TFLG1`, which is very BAD. If you use code like `TC1 = TCNT+1000;`, then the interrupt period will not be accurate.

```
interrupt 9 void TC1handler(void){
  TFLG1 = 0x02;     // acknowledge OC1
  Task1();
  TC1 = TC1+1000;   // execute Task 1 every 1ms
}
interrupt 10 void TC2handler(void){
  TFLG1 = 0x04;     // acknowledge OC2
  Task2();
  TC2 = TC2+2000;   // execute Task 2 every 2ms
}
```

(10) Question 5. n*1024 must be less than 65535 to avoid overflow. Choose smallest error

```
      out = (12*in)/17;
```
or
```
      out = (12*in+8)/17;
```

(15) Question 6. 100 mA will require the 2N2222 (because it can handle up to 500 mA of I_{CE}). We could have used any NPN with I_{CE} > 100mA, e.g., TIP120, IRF540. The V_{CE} on voltage of the 2N2222 is 0.3V. Because the current gain is 100 (h_{fe}) the base current needs to be 100mA/100 = 1mA. The I_{OH} of the 9S12 can supply this 1mA (I_{OH} can be up to 10 mA). Because the V_{OH} of the 9S12 is 4.2V (or greater) and the V_{BE} if the 2N2222 is 0.6V (or less), the resistor from the 9S12 to the 2N2222 base must be less than (4.2-0.6V)/1mA = 3.6/0.001 = 3.6 kΩ. I suggest making R much less than 3.6 kΩ (e.g., 2 kΩ) because it will force the NPN into saturation, independent of the V_{OH} of the 9S12, the V_{BE} of the 2N2222, the h_{fe} of the 2N2222, and the resistance of the coil. Therefore, when the digital output is high, the voltage across the motor will be 5.7V. You might have been tempted to use a higher voltage supply, like the "Bad solution" and use a series resistor (R2) to drop the voltage down to 5V. There are two fundamental problems with the "Bad solution". First, the solution wastes power, the power delivered into R2 is lost as heat. Second, the resistance of the coil is a function of the mechanical load on the electromagnet. The coil resistance can not be assumed to be constant.