

**(50) Question 1.** Frequency measurement system.

Part a) These additional global variables are private because they should not be accessed by the client (main).

```
long Count;           // private, used by key wakeup to count cycles
unsigned short Time; // private, used by TOF to create 1 minute delay
```

Part b) Show the ritual that initializes your frequency measurement software system.

```
void Freq_Init(void){
    Frequency = 0;
    Count = 0;
    Time = 0;
    KWIEH = 0x80;    // arm key wakeup interrupts, PH7
    DDRH = 0x7F;    // PH7 is an input
    TSCR = 0x80;    // enable TCNT
    TMSK2 = 0x80;   // arm TOF, 8 MHz
    TFLG2 = 0x80;   // initially clear TOF
    asm(" cli");    // enable interrupts
```

Part c) The interrupt service routines for key wakeup H and TOF.

```
#pragma interrupt_handler KeyWakeupHHandler
void KeyWakeupHHandler(void){ // every falling edge
    KWIFH = 0x80; // acknowledge
    Count++; }
#pragma abs_address:0xffce
void (*KeyH_vector[])() = { KeyWakeupHHandler };
#pragma end_abs_address
#pragma interrupt_handler TOFhandler
void TOFhandler(void){ // every 8192 us, when TCNT=0
    if(++Time == 7324 ){ // every 1 minute
        Frequency = Count; // new measurement
        Count = 0; Time = 0; }
    TFLG2 = 0x80;} // acknowledge TOF
#pragma abs_address:0xffde
void (*TOF_vector[])() = { TOFhandler };
#pragma end_abs_address
```

Part d) Because the global variable is 32 bits, the compiler can not create code that implements an atomic read of its value. So, the main program will fetch the value in four separate 8-bit reads or two separate 16-bit reads. Since interrupts are enabled it might fetch part of the data, then if the 1-minute TOF occurs, the TOF ISR will update the global. When the main program resumes it will fetch the rest of the value. At this point, the main program has garbage data (some bytes from one measurement, and the rest of the bytes from a subsequent measurement.) This is the danger of public globals. We can repair this VERY simple problem by disabling interrupts during the time when the public global is being read. In a more complex system, the public global may be read in 100's of places, making the repair extremely difficult. There needs to be a public function that retrieves the most recent measurement. This function can disable interrupts while the 32-bit private global is accessed.

**(12) Question 2.** For Circuit A, the answer is the  $I_{OL}$  of the 6812, which is 1.6 mA.

For Circuit B, the answer is the  $I_{OH}$  of the 6812, which is 0.8 mA.

For Circuit C, the answer is the  $I_{OL}$  of the 7406, which is 40 mA.

Circuit D will not work, because the open collector 7406 will not source current.

**(38) Question 3.** Part a) Show the ritual that initializes the traffic light system.

```
void Ritual(void){
    DDRA = 0x3F;} // PA7,6 inputs, PA5-0 are outputs
```

Part b) Show the main program that runs the controller in the foreground.

```
void main(void){unsigned char input;
unsigned char currentState;
    Ritual();
    currentState = goN;
    while(1){
        PORTA = fsm[currentState].out; // output
        Wait1Sec(fsm[currentState].time); // wait
        input = PORTA>>6; // input will be 0,1,2,3
        currentState = fsm[currentState].next[input];} // next
```