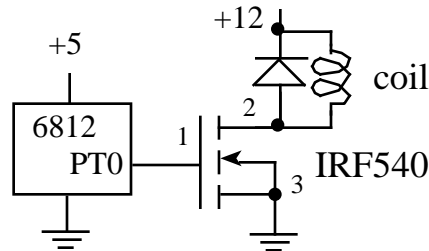


EE345M Fall 2001 Quiz 1 Solution

(25) Question 1. Of all the devices in the textbook, only the MOSFET IRF540 can sink 5A. The diode voltage is chosen to match the back EMF voltage. The diode is chosen to handle the back EMF voltage, typically selected from 1N4002 (100V) through 1N4007 (1000V). Since the MOSFET is voltage-controlled, it can be driven directly from the microcomputer output. The book and old quiz solutions have some circuits with a diode to ground. I was wrong, this second diode is never needed.



(10) Question 2. The 6812 I_{OL} is 1.6 mA. The 6812 V_{OL} is 0.4V.

(15) Question 3.

Part a) These two functions are friendly because they do not undo each other's action. In particular, it does not matter in what order they are executed.

Part b) Because the compiler produces atomic code for the read-modify-write sequence, there are no critical sections.

Part c) Intrusiveness is a measure of how much the measurement process itself modifies the operation being measured. It takes 4 cycles for the `jsr`, 4 cycles for the `bset`, and 5 cycles for the `rts`. Altogether, the debugging instrument executes in about 1.5 μ s. Assuming the software being debugged runs much slower than 1.5 μ s, these debugging instruments are minimally intrusive because the time to execute them is very small. Invasiveness is a measure of how much the debugging process changes the environment, or the way the system runs. If PORTJ were being used for actual system functions, then this code would be invasive (or just plain stupid).

(50) Question 4. To change the range to 36 μ s to 128ms, the resolution is increased to 2 μ s. TOF interrupts are used to detect a period above 256 ms. Overflow if there are two TOF interrupts without any IC interrupts.

```
// PT1/IC1 input = external signal
// rising edge to rising edge
// resolution = 2us
// Range = 36  $\mu$ s to 128 ms,
// TOF overflow checking
// IC1 interrupt each period,
unsigned int Period; // units of 2us
unsigned int First; // TCNT first edge
unsigned char Done; // Set each rising
unsigned int count;
void Ritual(void){
    asm(" sei"); // make atomic
    TIOS &= 0xFD; // PT1 input capture
    DDRT &= 0xFD; // PT1 is input
    TSCR = 0x80; // enable TCNT
    TMSK2= 0xB4; // 2us clock, arm TOF
    TCTL4 = (TCTL4&0xF3)|0x04; // rising
    First = TCNT; // first will be wrong
    Done=0; // set on subsequent
    TFLG1 = 0x02; // Clear C1F
    TMSK1 |= 0x02; // Arm IC1
    count = 0; // error if becomes 2
    TFLG2=0x80; // clear TOF
    asm(" cli");
}
#pragma interrupt_handler TIC1handler()
void TIC1handler(void){
    Period=TC1-First; // 2us resolution
    First=TC1; // Setup for next
    TFLG1=0x02; // ack by clearing C1F
    count = 0; // overflow if 2
    Done=0xFF;
}

#pragma abs_address:0xffec
void (*TC1_vector[])() = {
    TIC1handler};
#pragma end_abs_address

#pragma interrupt_handler TOFhandler()
void TOFhandler(void){
    TFLG2=0x80; // acknowledge TOF
    count++;
    if(count==2){ // period above 256 ms
        count=0;
        Period=0xFFFF;
        Done=1;
    }
}
#pragma abs_address:0xffde
void (*TOF_vector[])() = { TOFhandler};
#pragma end_abs_address
```

EE345M Fall 2001 Quiz 1 Solution

Here is a second, better answer that works for all periods

```
// PT1/IC1 input = external signal
// rising edge to rising edge
// resolution = 2us
// Range = 36 us to 128 ms,
// OC0 overflow checking
// IC1 interrupt each period,
unsigned int Period; // units of 2us
unsigned int First; // TCNT first edge
unsigned char Done; // Set each rising
void Ritual(void){
    asm(" sei"); // make atomic
    TIOS &= 0xFD; // PT1 input capture
    TIOS |= 0x01; // PT0 output compare
    DDRT &= 0xFD; // PT1 is input
    TSCR = 0x80; // enable TCNT
    TMSK2= 0x34; // 2us clock
    TCTL4 = (TCTL4&0xF3)|0x04; // rising
    First = TCNT; // first will be wrong
    Done=0; // set on subsequent
    TFLG1 = 0x03; // Clear C1F, C0F
    TMSK1 |= 0x03; // Arm IC1, OC0
    TC0 = TCNT; // error if match
    asm(" cli");
}
#pragma interrupt_handler TIC1handler()
void TIC1handler(void){
    if(done == 1){
        done = 0; // try to restart
    }
    else{
        Period=TC1-First; // 2us resolution
        Done=0xFF;
    }
    First=TC1; // Setup for next
    TFLG1=0x02; // ack by clearing C1F
    TC0 = TC1; // overflow if match
}
#pragma abs_address:0xffec
void (*TC1_vector[])() = {
    TIC1handler};
#pragma end_abs_address

#pragma interrupt_handler OC0handler()
void OC0handler(void){
    TFLG1=0x01; // acknowledge OC0
    Period=0xFFFF; // larger than 128ms
    Done=1;
}
#pragma abs_address:0xffee
void (*OC0_vector[])() = { OC0handler};
#pragma end_abs_address
```