

Russell Bjella  
ASEN 5519 – Microavionics  
Lab #3  
9/27/2016

#### Questions (4519/5519)

1. `MOVF 0x0020,0`  
`BCF WREG,0,0`  
`BSF WREG,2,0`  
`BSF WREG,3,0`  
`BTG WREG,5,0`  
`BTG WREG,6,0`
2. `MOVFF 0x200 0x2A0`  
`MOVFF 0x201 0x2A1`  
`MOVFF 0x202 0x2A2`  
`MOVFF 0x203 0x2A3`  
`MOVFF 0x204 0x2A4`  
`MOVFF 0x205 0x2A5`  
`MOVFF 0x206 0x2A6`  
`MOVFF 0x207 0x2A7`  
`MOVFF 0x208 0x2A8`  
`MOVFF 0x209 0x2A9`
3. The X option for the LIST compiler directive turns macro expansion on or off. Having it set to "ON" allows you to observe what each macro is doing in the listing file, but setting it to "OFF" can reduce clutter in listing files for programs with many macros.
4. Forgot to do this until it was too late. Sorry about that.
5. It is slower. This is because in addition to the 2 millisecond waits, there are additional commands being executed leading to a total delay of greater than 1 second.
6. Switch bounce is when the switch bounces back to a closed position after being released due to the inherent "springiness", registering a false "push" of the switch. One possible solution is to add a short delay after each switch check, preventing the PIC from seeing these bounces.
7. 5.6: `MOVLW B'00001111'`  
`MOVWF TRISB`  
5.22: Loop1 is executed 50 times.  
5.25: It repeats infinitely because it's storing the result of the decrement in WREG, not in REG2. So, every time it reenters the loop, it subtracts one from 50 and stores the result in WREG, leaving REG2 at 50, thus never reaching zero.  
5.29: There are 600 total instruction cycles (12 per loop times 50 loops), multiplied by 0.4 microseconds per instruction cycle (4 clock cycles per instruction cycle) = 240 microseconds.

#### Questions (5519 only)

1. `STACK EQU 0x0E00 ;top of stack`
2. As the RTG spins, the internal switches are open most of the time but close once per revolution, setting the output to binary high. There is a 5 millisecond delay to prevent switch bounce. There is a 90 degree phase shift between channels A and B.
3. A
4. 5.28:

- 5.30: 1.6 microseconds, since it takes 4 fewer instruction cycles.
- 5.31:  $150 \text{ microseconds} / 0.4 \text{ microseconds} = 375$  instruction cycles, divide by 12 and round up to 32, so set value to 0x20.