ECE 3724 Test #1 – Spring 2005/Reese -- there 5 pages (3 pages front/back)

Student ID: _______________________ (no names please)

You may NOT use a calculator. You may use only the provided reference materials. For problems in part II, assume initial memory contents as shown below at the start of EACH instruction. Explain what register/memory location is modified, and give the FINAL HEX value of the modified register/memory location, and the final status of the C, Z flags. Recall that the 'd' bit in a machine word indicating a destination is '0' if the destination is W, is '1' if the destination is the file register. For the 'a' bit, use the assumptions we have used in class ('a' bit is '0' if address in ACCESS RAM, 'a' bit is '1' if not in ACCESS RAM).

Part I: (20 pts)

a. The PIC18 has something called the ACCESS bank. These locations are:

1. The locations in bank 0 and the locations of bank 15.
2. The last 128 locations of bank 0, and last 128 locations of bank 15.
3. The first 128 locations of bank 0, and the first 128 locations of bank 15.
4. The first 128 locations of bank0, and the last 128 locations of bank 15.
5. The last 128 locations of bank0, and the first 128 locations of bank 15.
6. Only the locations of bank 0.

b. Give the machine code in **HEX** for the instruction:

```
movff   0x150, 0x2A3
```

c. The machine code 0x9745 represents instruction? (use 'w' or 'f' for the destination, and 'ACCESS' or BANKED to represent the value of the *a* bit)

d. For a 25 MHz clock, how long does it take to execute the following instructions? (give the answer in microseconds)

```
movf    0x013,w
addwf   0x020,f
```

e. Circle the true statement (non-volatile means contents are retained when power is removed; volatile means contents are lost when power is removed).

1. On the PIC18, program memory is volatile and data memory(file registers) is volatile.
2. On the PIC18, program memory is non-volatile and data memory(file registers) is volatile
3. On the PIC18, program memory is volatile and data memory(file registers) is non-volatile.
4. On the PIC18, program memory is non-volatile and data memory(file registers) is non-volatile.

Location	Contents
0x048	0xAF
0x049	0x00
0x04A	0x4F
0x04B	0x1D

Assume the W register has the value 0xE3 in it, and that initial values of C, Z are both ‘0’.

Part II. (35 pts) Assume the above memory contents, W register value, initial C,Z values at the START of each instruction.

a. bcf 0x04A, 3

Circle one: W dest. Reg. file dest.

New value (hex) ____ C_flag :____ , Z flag:__

b. iorlw 0x48

Circle one: W dest. Reg. file dest.

New value (hex) ____ C_flag :____ , Z flag:__

c. rlcf 0x04B, f

Circle one: W dest. Reg. file dest.

New value (hex) ____ C_flag :____ , Z flag:__

d. addwf 0x04B,f

Circle one: W dest. Reg. file dest.

New value (hex) ____ C_flag :____ , Z flag:__

e. subwf 0x049, f

Circle one: W dest. Reg. file dest.

New value (hex) ____ C_flag :____ , Z flag:__

(45 pts) PART III. Convert the following C code fragments to PIC18 assembly.

Variable locations are: *i* is data location 0x000, *j* is data location 0x001, *k* is data location 002. Assume the BSR has a 0x0 value in it initially.

If you use a temporary memory location, use temp and assume it is in bank 0. When writing code, you **must use** symbolic names for variable names, register names, and bit names for (i.e, use: bsf STATUS, C instead of bsf 0xFD8, 0x0). You do not have to show the CBLOCK declaration for variables.

Hint: A common mistake in these problems is to write code that modifies variables to the right of the '=' sign (i.e, for 'a = b – c;' the code you write somehow modifies *b*, or *c*, as well as *a*). This in incorrect; make sure that your code only modifies variables to the left of the '=' sign.

Also, recall that 'k++' is the same as 'k=k+1;', 'j- - ' is the same as 'j = j – 1', that " i = = j" is true if *i* is equal to *j*, that "i != j" is true if *i* is not equal to *j*, "<<" is a left shift, ">>" is a right shift, '|' is bitwise logical OR, '&' is a bitwise logic AND, '^' is a bitwise logical XOR.

```
unsigned char i,j,k;
```

a. (6 pts)

```
k =  (i >> 1) + 50;
```

b. (10 pts)

```
if ( (i == 0) && (j != 0) {
    k--; j++;
}
```

c. (7 pts)

```
k =  i – j – 1;
```

d. (10 pts)

```
while ( i < k) {
   k = k << 2;
}
```

e. (5 pts) Write a sequence of assembly code instructions that will clear the MSB (most significant bit) and LSB (least significant bit) of variable k.

f. (7 pts) Write a PIC18 instruction sequence that does

k = (i | j) & 0xA0