

DO NOT MAKE THE MISTAKE OF ONLY STUDYING THE QUESTIONS IN THIS SAMPLE TEST!!!
On test #2, you are responsible for ALL MATERIAL up until the first PIC hardware lecture.

ECE 3724/CS 3124 Test #2 Sample – Summer 2004- Reese

You may NOT use a calculator. You may use only the provided reference materials. .

Part I: (70 pts)

- a. (10 pts) Write a PIC18 assembly code fragment to implement the following:
int k;

k = k - 1;

```
movlw 0      ; w = 0;
decf k,f     ;decrement LSByte
subwfb k+1,f ;decrement MSByte
```

- b. (10 pts) Write a PIC18 assembly code fragment to implement the following:
signed int j, k;
char i;

```
do {
    i++;
} while (j > k);
```

```
loop_top
    incf i,f;           ;i++, a byte variable
    movf j,w           ;w = LSByte of j
    subwf k,f          ;k-j, LSByte
    movf j+1,w         ;w = MSByte of j
    subwfb k+1,w       ;k-j; MSByte;
    bov v_1            ;branch if v_1
    bn loop_top        ;loop if v=0, N=1
    bra loop_exit      ; exit otherwise
V_1   bnn loop_top     ;loop if V=1, N=0
loop_exit
    ...rest of code....
```

- c. (10 pts) Write a PIC18 assembly code fragment to implement the following:
int k;

k = k << 2;

```
bcf STATUS, C
rlcf k,f
rlcf k+1,f ; k = k << 1
bcf STATUS,C
rlcf k,f
rlcf k+1,f ; k = k << 1
```

d. (10 pts) Implement the 'FILLSTR' subroutine in PIC18 assembly language.

```

/* get string */
fillstr (s,c)
unsigned char *s;
unsigned char c
{
  while (*s != 0) {
    *s = c;
    s++;
  }
}

char *mystr = "Hello";
main ()
{
  fillstr(mystr,0x20);
}

// 0x20 = space char

```

```

;parameter block for fillstr
CBLOCK 0x20
s:2, c ; s contains pointer to character string, c is fill character
ENDC

```

```

movf    s,w
movwf   FSR0L
movf    s+1,w
movwf   FSR0H ; FSR0 = s
loop
movf    INDF0,w ; *s == 0?
bz      exit ; exit if zero
movf    c,w
movwf   POSTINC0 ; *s = c, s++
bra     loop
exit
return

main
movlw   low mystr
movwf   s;
movlw   high mystr
movwf   s+1;
movlw   0x20 ; space character
movwf   c
call    fillstr
...rest of code...

```

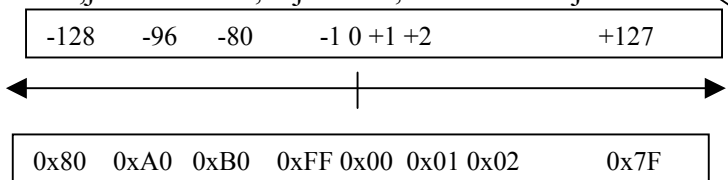
```

; same as
movwf INDF0 ; *s=c
infsnz FSR0L,f
incf FSR0H,f ; FSR0++

```

e. (10 pts) For each one of the following comparisons, indicate if the comparison is true or false.

- unsigned char i,j; i = 0x80; j= 0x34; i > j ??? TRUE FALSE
- signed char i, j i = 0xB0; j= 0xA0; i > j ??? TRUE FALSE
- signed char i,j i = 0xFF; j= 0x01; i > j ??? TRUE FALSE



For the 0xB0, 0xA0 case, visualize the number line. The value 0xB0 is to the right of 0xA0, so 0xB0 is greater than 0xA0. Or convert to decimal; 0xA0 = -96, 0xB0 = -80

- f. (10 pts) Write a PIC18 assembly code fragment to implement the following:
signed int j, k;
unsigned char i;

```
if (j != k) {  
    i = i >> 1;  
}
```

```
movf k,w          ;w = LSByte of j  
subwf j,f         ;j-k, LSByte  
bnz if_body      ;branch if Z=0  
movf k+1,w       ;w = MSByte of j  
subwfb j+w,w     ;k-j; MSByte;  
bz end_if        ;branch if Z = 1  
if_body          ; reach here if either subtraction nonzero  
bcf STATUS,C  
rrcf i,f         ; i = i >> 1  
end_if  
...rest of code....
```

- g. (10 pts) Write a PIC18 assembly code fragment to implement the following:

signed char i;

```
i = i >> 1;
```

```
;This is a signed right shift, the sign bit must  
; be kept the same  
bcf STATUS,C  
btfsc i,7        ;check sign bit  
bsf STATUS,C     ;set carry, i is negative  
rrcf i,f         ; do shift
```

Part II: (40 pts) Answer 7 out of the next 9 questions. Cross out the 2 questions that you do not want graded.

1. Give the machine code for the fragment below:

here
bra here

The bra will branch one instruction word backwards, or -1 (bra -1), so machine code is: 0xD7FF

2. The value 0xE2 is a two's complement, 8-bit number. What is the decimal value?.

This is a negative number, as the MSB is set. So, magnitude is $0x00 - 0xE2 = 0x1E = 16+14 = 20$.
Final answer is -20.

3. The value of -128 in 8 bits two's complement is 0x80. What is the value as a 16-bit number?

In 16-bits, must use sign extension, answer is 0xFF80.

4. Give the result of the operation $0xB3 + 0x9A$, and the V, N, C, Z flag settings.

$0xB3 + 0x9A = 0x4D$, C = 1, V = 1 (-N + (-N) = +N, wrong), N = 0, Z = 0

5. Give the result of the operation $0x40 - 0xA3$, and the V, N, C, Z flag settings

$0x40 - 0xA3 = 0x9D$, C = 0 (borrow), V = 1 (+N - (-N)) = (+N + (+N)), answer should be positive, but get a negative value, so answer is wrong.
N = 1, Z = 0

6. What range of two's complement numbers can be encoded in 5 bits?

+15 to -16 ($+2^{N-1} - 1$ to -2^{N-1})

7. When would a 'goto' instruction have to be used instead of a *bra* instruction?

If the jump target address was further away than +1023 or -1024 instruction words from the location of the *bra* instruction.

8. What are the table read/write instructions useful for?

Table reads are used to read a byte from program memory using the TBLPTR register. The value is read into the TABLAT register. They are useful for reading or writing values from/to program memory.

9. What is the limitation of the stack on the PIC18?

The stack can only store 31 return addresses. If 31 CALLS are made without a return, then the next call will cause stack overflow as there is no place to store the return address.