EECS 281: Homework #4 **Due: Tuesday, February 22, 2005**

Name: ________________________________ Email: ________________________________

(0) Practice the Wakerly problems 2.1 (a,b,e), 2.7a, 2.9a, 2.10a, 2.12a, 2.37 but do not hand these in. See Wakerly website solutions http://www.ddpp.com/
(1) Using standard C++ precision and data types, convert the following into 8051 assembler syntax in column 2 (read as31 manpage) and convert the columns 3 to 6 in various base 2 binary complements in base 2 format.

	8051 definition	big endian two's compl.	little endian two's compl.	big endian one's compl.	big endian signed magnitude
unsigned char x='a';	.byte 'a'	01100001			
unsigned char x=0;					
signed char x=−1;					
unsigned char x=0x255;					
unsigned char x=255;					
unsigned char x=256;					
unsigned char x=0255;					
signed char x=255;					
signed char x=−'a';					
unsigned char x=127;					
signed char x=127;					
unsigned char x=128;					
signed char x=128;					
signed char x=−128;					
unsigned char x=0128;					
unsigned char x=−64;					
signed char x=013;					
signed short x=013;					
signed short x='a';					
signed short x=−'a';					
unsigned short x=256;					

Using C++ convert the following using the following values:

register unsigned char u, a=0x85, b=0xa7, c=03;

register signed char s, w=0x81, x=0xa6, z=-1;

State if **overflow or carry** has occurred. Assign the 8051 registers as follows: u=A, a=R0, b=R1, c=R2, s=R3, w=R4, x=R5; z=R6; You can double check your work by using the C compiler:

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv) {
  unsigned char u, a=0x85, b=0xa7, c=03; signed char s, w=0x81, x=0xa6, z=-1;

  u= ~a; printf("u=hex=0x%x=Octal=0%o=decimal=%d a=0x%x=%d\n", u, u, u, a, a, a);
}
```

	two's complement big endian	8051 instructions
a = 0x85;	10000101b	mov r1,#0x85
b = 0xa7;		
z = −1;		
u = $\sim a$;		mov a,r1; cpl a
u = $-a$;		
u = a & b;		
u = a & w;		
u = a | b;		
u = a | b & c;		
u = a ˆ b;		
u = a ˆ 'C';		
u = a + 'C';		
u = $\sim a$ + 1;		
u = a − b;		
u = a << 2;		
u = a >> 2;		
s = $\sim w$;		
s = − w;		
s = w + x;		
s = w - x;		
s = w ˆ x;		
s = $-z$ ˆ ∼ a;		

3.Using the 8051 instruction, assemble the instruction **by pencil and paper** into hex, and then execute it showing the clock time in machine cycles:

Mem. Addr.	Machine instructions	Assembly	Clock Time	PC	OV	CY	AC	Reg. A	Reg. R1
0x100		MOV A,#0xC8	0	0x100	0	0	0	0xff	0xff
		MOV R1,#0x88							
		ADD A,R1							
		MOV DPL,A							
		CLR A							
		ADDC A,#0x10							
		MOV DPH,A							
		Total Time=							