EE 472 – Embedded Systems

Dr. Shwetak Patel
Assistant Professor
Computer Science & Engineering
Electrical Engineering


Agenda

- Lab1
- Finish C programming
  - Bitwise operations
- Lab time

Questions?

- Pointers
- Function pointers
- Arrays
- Structs

Bitwise Operations

- & bitwise AND
- | bitwise OR
- ^ bitwise XOR
- << left shift
- >> right shift
- ~ ones complement

Bitwise Operations

- Important in embedded systems
- Manipulating registers
  - Setting and observing
- Sometimes faster for certain arithmetic operations

Bitwise Operations

- & bitwise AND
- | bitwise OR
- ^ bitwise XOR
- << left shift
- >> right shift
- ~ ones complement

Class Exercise

```
/* Use of bitwise logical operators  */

#define  Bit_Zero   0x01
#define  Bit_One    0x02
#define  Bit_Two    0x04
#define  Bit_Three  0x08
#define  Bit_Four   0x10
#define  Bit_Five   0x20
#define  Bit_Six    0x40

// etc etc etc ..


int x = 0x0B;           // x  = [... 0 0 0 0 1 0 1 1]

int y = 011;            // y  = [... 0 0 0 0 1 0 0 1]

int z = x << 2;         // z  = [... 0 0 1 0 1 1 0 0]

int z1 = y & Bit_Three; // z1 = [... 0 0 0 0 1 0 0 0]

int z2 = z | Bit_Four;  // z2 = [... 0 0 1 1 1 1 0 0]
```

XOR

- Can hide two pieces of data in one variable

- variable2 = variable1 ^ Feature
- variable3 = variable2 ^ Feature
- (variable3 and variable1 are the same now)

New swap function

- Use XOR to redo swap without a temp variable

XOR Swap

```
void swap(int *a, int *b){

   *a = *a ^ *b;   //hide *b in *a
   *b = *a ^ *b;   //pull the original *a out
   *a = *a ^ *b;   //pull the original *b out
}
```


Class Exercise

- Write a C function using bitwise logical operators to print the binary value of a 16 bit unsigned integer

Example solution

```
void bin_display(unsigned int x){

    unsigned int I,j;

    for(i=0;i<16;i++){

          j = 1<<(15-i);

          if(j & x){ printf("1"); }

          else { printf("0"); }
    }
}
```

Questions?

- Next time:
  - Software Design
  - UML

Class Exercise

```
Convert 0x1A to decimal: __26_

Convert 0xA2 to decimal:  __162___

Convert 020 to Hex:    _____10_______

Convert 0x20 to Octal: _____40________

/* Give the binary value of the least significant
8 bits of the following values:   */

int aa = 0xC5;

int ab = 017 + 1;

int a = x | y;

int b = Bit_Two | Bit_Five | Bit_Seven;

int c = ~((z | Bit_Zero) << 1);

int d = !(z1 | Bit_Two);
```

Class Exercise

```
Convert 0x1A to decimal: __26_

Convert 0xA2 to decimal:  __162___

Convert 020 to Hex:    _____10_______

Convert 0x20 to Octal: _____40________

/* Give the binary value of the least significant
8 bits of the following values:   */

int aa = 0xC5;     // aa = [  1 1 0 0 0 1 0 1  ]

int ab = 017 + 1;  // ab = [  0 0 0 1 0 0 0 0  ]

int a = x | y;     // a  = [  0 0 0 0 1 0 1 1  ]

int b = Bit_Two | Bit_Five | Bit_Seven;
                   // b  = [  1 0 1 0 0 1 0 0  ]

int c = ~((z | Bit_Zero) << 1);
                   // c  = [  1 0 1 0 0 1 0 1  ]

int d = !(z1 | Bit_Two);
                   // d  = [  0 0 0 0 0 0 0 0  ]
```