#### **Embedded Software**

(bit-wise operations)





- ▶ 강지수
  - 고속컴퓨팅시스템연구실 (하1104)
  - jikakr@gmail.com
- 질문
  - I-Class 질의응답 이용
  - 연구실 방문
    - 사전 약속 필수 (이메일)

# Outline

- Overview
- Bit operators
- Bit manipulation
- Programming exercises
  - If you cannot finish this within 2 hours, you need to write programs as your homework.



- Bit manipulation is used to control the machine at low level
  - Writing device drivers
  - Pixel-level graphic programming
- High-level programmer may not use this programing technique, but device driver programmer needs to handle this skillfully



#### Hex and binary

- Hexadecimal is convenient way to represent binary
  - 0: 0000
  - 1: 0001
  - **9**: 1001
  - A: 1010
  - F: 1111
- 0xAF: 10101111
- 0xBA: 10111010
- 0x1AAF: 0001 1010 1010 1111



# Bit operators

#### Bit operators

Operator	Meaning
&	Bitwise AND
	Bitwise OR
^	Bitwise exclusive OR
~	Complement
<<	Shift left
>>	Shift right



#### Bitwise AND operator (&)

Bit1	Bit2	Bit1 & Bit2
0	0	0
0	1	0
1	0	0
1	1	1

#### Bitwise OR operator ( | )

Bit1	Bit2	Bit1   Bit2
0	0	0
0	1	1
1	0	1
1	1	1



#### Bitwise Exclusive OR (^)

Bit1	Bit2	Bit1 ^ Bit2
0	0	0
0	1	1
1	0	1
1	1	0

#### Ones complement (~)

Bit	~Bit
0	1
1	0

Left and right shift operator (<< , >>)e.g.

	c = 0x1C	00011100
c << 1	(c << 1) = 0x38	00111000
c >> 2	(c >> 2) = 0x07	00000111



#### Bit values

Bit	Binary Value	Hex Constant
7	1000000	0x80
6	01000000	0x40
5	00100000	0x20
4	00010000	0x10
3	00001000	0x08
2	00000100	0x04
1	0000010	0x02
0	0000001	0x01



#### Bit manipulation

 Bitwise operators may be used to manipulate the contents of registers effectively;

```
1> Testing the third bit;
                 if (*pTimerstatus & 0x08)
2> Setting the forth bit;
                  *pTimerstatus |= 0x10;
                   int *pTimerstatus;
```

# 4

3. Clearing the second bit;

\*pTimerstatus &= 
$$\sim$$
(0x04)

4. Toggling the seventh bit;

\*pTimerstatus ^= 0x80;



Bitmask: a constant often used along with bitwise operators

e.g. #define TIMER\_COMPLETE (0x80)

Bitmask Macro

#define BIT(X) (1<<(X))
#define TIMER\_STATUS BIT(5)</pre>

00100000

0x20



\*pTimerstatus &=  $\sim$ (0x20)

\*pTimerstatus &= ~TIMER\_STATUS



#### Assignments

- I-Class에 다음 슬라이드의 과제 제출
- 제출 파일
  - 소스 코드(.c / .cpp)
  - 보고서(.pdf) 실행 결과 화면 포함
- 파일명
  - 2주차\_학번\_이름
- 제출 기한
  - 다음주 화요일 자정까지



## Programming exercises (1)

- 8개의 LED를 갖는 LEDs 라는 char 변수에
  - 이진수 00000001을 대입하면, 0번째 LED 가 켜지고, 이진수 00000010을 대입하면 1번째 LED 가 켜지고, ... 이진수 1000000을 대입하면 7번째 LED 가 켜지는 변수 LEDs 를 가정하자.
    - char LEDs;
    - LEDs=0x01;
    - LEDs=0x02;
  - 0부터 7까지 숫자를 입력받아서, LEDs 값을 다음 과 같이 바꾸어주는 함수를 작성하시오.
    - int LED\_function(int input);
    - LED\_function(2) 함수를 호출하면, 0x04 를 return 함



- 상기 프로그램을 어떻게 check 하나?
  - int return\_value;
  - return\_value=LED\_function(2);
  - printf("return\_value:%d\n",return\_value);
    - 2가 출력됨



### Programming exercises (2)

- 상기 프로그램을 활용하여,
  - 1. LEDs의 초기값을 이진수로 0000001을 대입하고,
  - 2. LEDs 값을 왼쪽으로 1만큼 계속 shift하고,
  - 3. LEDs 값이 이진수로 10000000 으로 대입되었을 때
  - 4. 역으로 오른쪽으로 1만큼 계속 shift하는
  - 5. 과정이 총 4번 반복되는 프로그램을 작성하시오.
  - 정답은 ?
    - 출력 순서 ?
      - 1, 2, 4, 8,..., 128, 64 ....
    - <반드시 shift 연산자를 사용할 것>



# Programming exercises (3)

- char 변수를 2진수로 변환하였을때, 1의 개수를 세고, 이를 다 왼쪽으로 shift 하였을때의 값을 출력하시오.
  - Input: 01010110, 십진수: 86, 16진수: 0x56
  - Output: 11110000, 십진수: 240, 16진수: 0xF0
  - How to check?
    - printf 문
    - <정답> 위의 예의 출력 값은 ?
      - 1의 개수? 4
      - Shift 시의 값은? 240
  - 조교가 char 변수를 임의로 입력해서 제대로 출력 되는 지 체크할 테니, scanf 함수를 사용해서 입력 받도록 할 것