



# Lesson 14: Bitwise Operations

Jiun-Long Huang

National Chiao Tung University



# Bitwise AND

## ◆ Bitwise AND

&	0	1
0	0	0
1	0	1

00001111

& 00111100

-----

00001100 ← Set to 0

← Keep the same

# Bitwise OR

## ◆ Bitwise OR

	0	1
0	0	1
1	1	1

00001111

| 00111100

-----

00111111

← Keep the same

← Set to 1

# Bitwise XOR

## ◆ Bitwise XOR (Exclusive OR)

$\wedge$	0	1
0	0	1
1	1	0

```
00001111
^ 00111100
-----
00110011
^ 00111100
-----
00001111
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    char text[]="ABCD";
```

```
    char key='1';
```

```
    int i;
```

```
    for(i=0;i<strlen(text);i++)
```

```
        text[i]^=key;
```

```
    printf("Encryped Text: %s\n", text);
```

```
    for(i=0;i<strlen(text);i++)
```

```
        text[i]^=key;
```

```
    printf("Decryped Text: %s\n", text);
```

```
    return 0;
```

```
}
```

Encryped Text: psru  
Decryped Text: ABCD

# Bitwise Complement

## ◆ Complement

$\sim$	0	1
	1	0

$\sim$  00001111

-----

11110000

# Set Bit

## ◆ Set bit j of i to 1

- $i \mid= 1 \ll j$

```
#include <stdio.h>
int main(void)
{
    unsigned char i=2,j=4;
    i|=1<<j;
    printf("%d",i);
    return 0;
}
```

```
i=00000010
1<<j=00010000
```

```
      00000010
|   00010000
-----
      00010010
```

# Clear Bit

## ◆ Clear bit j of i to 0

- $i \&= \sim(1 \ll j)$

```
#include <stdio.h>
int main(void)
{
    unsigned char i=6,j=1;
    i&=~(1<<j);
    printf("%d",i);
    return 0;
}
```

```
i=00000110
1<<j=00000010
~(1<<j)=11111101

    00000110
& 11111101
-----
    00000100
```



# Test Bit

## ◆ Test the value of bit j of i

- if  $(i \& 1 \ll j) \dots$

```
#include <stdio.h>
int main(void)
{
    unsigned char i=6,j=1;
    if (i&1<<j)
        printf("1\n");
    else
        printf("0\n");
    return 0;
}
```

```
i=00000110
1<<j=00000010
```

```
    00000110
& 00000010
-----
    00000010
```