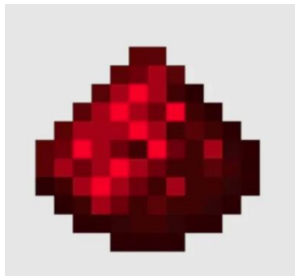
Introduction to Computer System

EPISODE 2.1-2.3

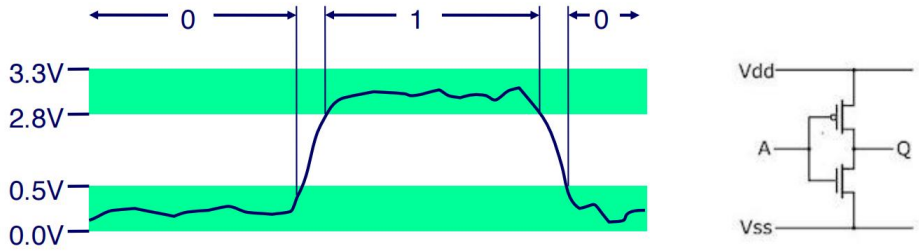
-ljh

为什么使用二进制?

· 运算简单什么的都是次要的
· 根本原因在于，对于可以记录信息的事物来讲，有两种状态的事物最多。
· 首先要记录信息，每个单位至少要有两种可能的状态。1的多少次方都是1。
· 而相比于有三种或更多种状态的元件，有两种状态的元件显然多得多，也容易获取得多。比如开关，比如电位的高低，光照的有无。
· 可能有人会问电位为什么只能区分高低，不能分多个吗？我的评价是确实可以，比如下面这个例子↓



■ **Why Computers Use Binary?**



Binary is the most practical system to use!

· 但有没有想过，现实里的电位可以像游戏这样区分得这么精确吗？更多时候还是下面这个例子↓

十六进制表示法

- 课本上的内容尽量不说了
- 为什么16进制要以0x开头?
- 虽然说只是规定的，但有一个比较容易被接受的说法是，由于C语言的变量命名规则，变量名只能由数字、字母和下划线组成，且必须由字母开头，所以标识符的开头最好是数字，避免被整体识别为一个变量。八进制（Octal）的开头是O，所以用了数字0这个和O相似的数字来表示，十六进制沿用了此方法，在0后面加上一个字母来与八进制区分，十六进制（hexadecimal）的前几个字母中x这个字母不太常用，所以使用了0x。

基本逻辑运算

And

- A&B = 1 when both A=1 and B=1

&	0	1
0	0	0
1	0	1

Or

- A|B = 1 when either A=1 or B=1

|	0	1
0	0	1
1	1	1

Not

- ~A = 1 when A=0

~	
0	1
1	0

Exclusive-Or (Xor)

- A^B = 1 when either A=1 or B=1, but not both

^	0	1
0	0	1
1	1	0

· 异或（和同或）是基本逻辑运算吗?

· C++逻辑运算的提前终止原则

补码表示法

Encoding Example (Cont.)

```
x =      15213: 00111011 01101101
y =     -15213: 11000100 10010011
```

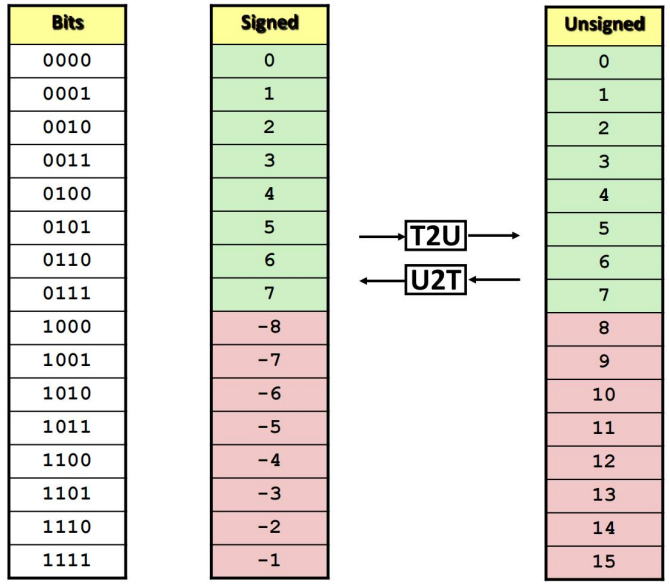
Weight	15213		-15213	
1	1	1	1	1
2	0	0	1	2
4	1	4	0	0
8	1	8	0	0
16	0	0	1	16
32	1	32	0	0
64	1	64	0	0
128	0	0	1	128
256	1	256	0	0
512	1	512	0	0
1024	0	0	1	1024
2048	1	2048	0	0
4096	1	4096	0	0
8192	1	8192	0	0
16384	0	0	1	16384
-32768	0	0	1	-32768
Sum		**15213**		**-15213**

$$B2T(X) = -x_{w-1} \cdot 2^{w-1} + \sum_{i=0}^{w-2} x_i \cdot 2^i$$

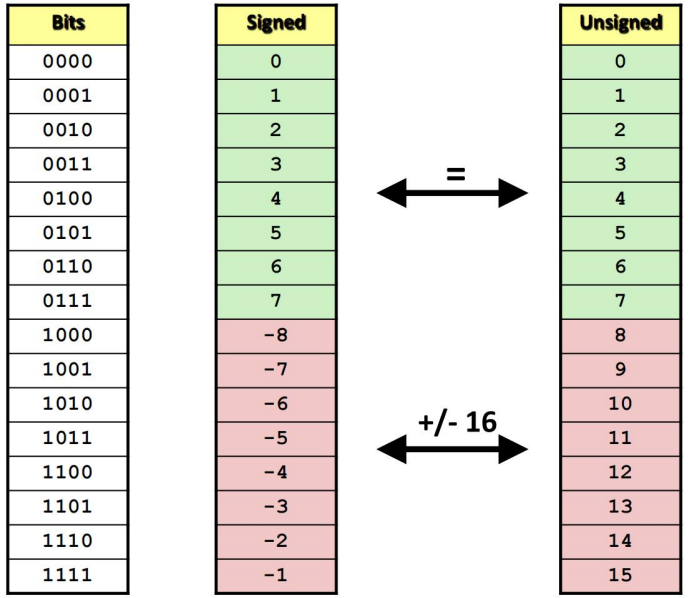
- 补码书上都讲了
- 此外还有原码
- 原码的+0和-0总感觉很怪吧

数位转换

Mapping Signed ↔ Unsigned



Mapping Signed ↔ Unsigned



· 好像没什么特别的，都是课件里的东西。

数位的切割和扩展

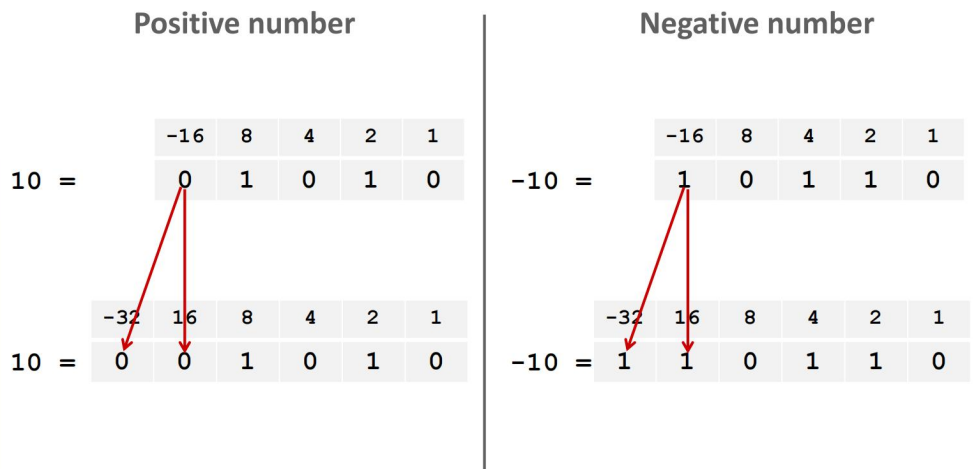
Summary:
Expanding, Truncating: Basic Rules

- **Expanding (e.g., short int to int)**
  - Unsigned: zeros added
  - Signed: sign extension
  - Both yield expected result

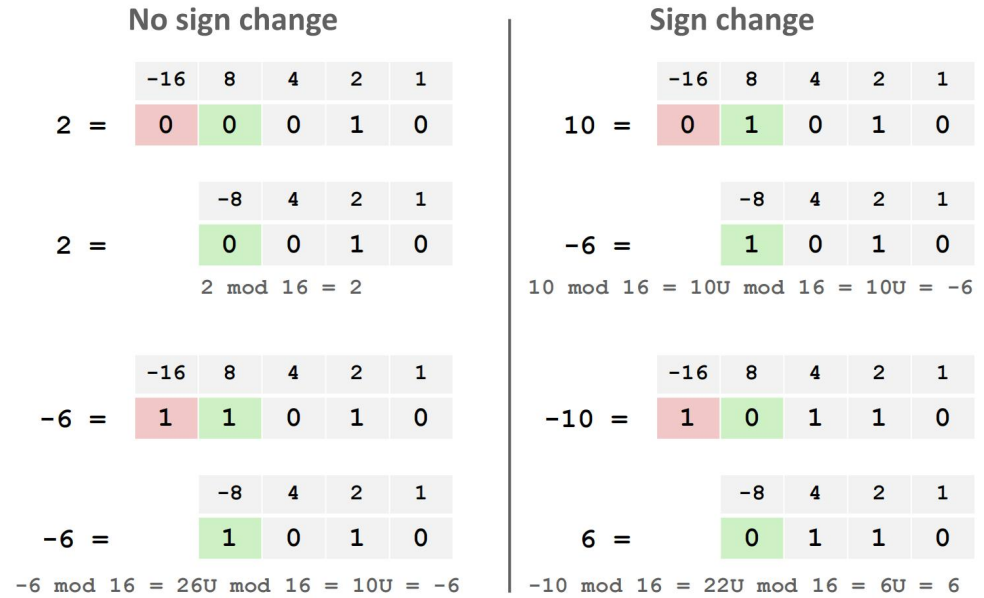
- **Truncating (e.g., unsigned to unsigned short)**
  - Unsigned/signed: bits are truncated
  - Result reinterpreted
  - Unsigned: mod operation
  - Signed: similar to mod

· 似乎都是课上的东西了啊?

Sign Extension: Simple Example



Truncation: Simple Example



各种运算

- 不是感觉没什么好讲的啊
- 乘法是移位和加法的结合，因为比较快
- 整数运算实际上是模运算