CS 33: Introduction to Computer Organization

TA: Aalisha Dalal
LA: Jonathan Myong
Office Hours: Friday, 9:30-11:30AM

Quick Introduction

Motivation for the Course

- At the granular level, everything in a computer is done using digital circuits which is done using bitwise operations.
- Foundation for Operation Systems.
- Prevent security attacks
- To write efficient code by using parallelization

Focus of the Discussion

- Bits and Integers
- Worksheet Problems

Bitwise vs. logical Operators in C

Bitwise operator	**Logical Operator**
&, |, ~, ^	&&, ||, !
Bitwise operation on datatypes	Evaluates expression to true or false with short-circuiting behavior

How are negative numbers stored?

Most popular approach is using Two's complement form

Example: Two's complement of 133

Dividing by Power of 2 - Shift Operator

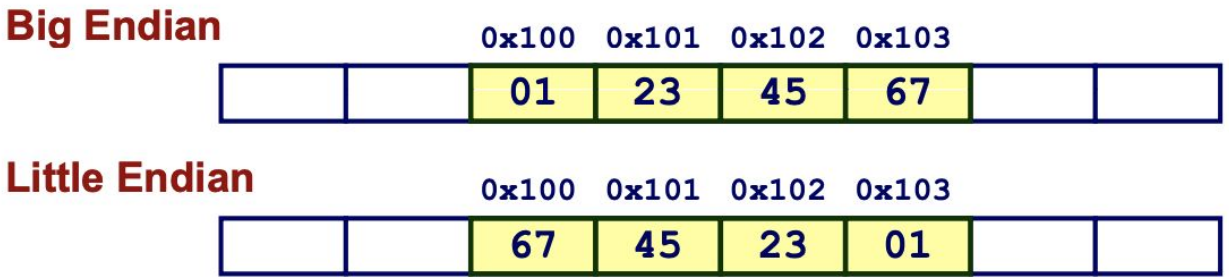
For positive numbers: $\lfloor u/2^k \rfloor$

For negative numbers: $\lceil u/2^k \rceil$

How do we achieve this?

Big Endian vs. Little Endian

- **Example**
  - Variable x has 4-byte value of 0x01234567
  - Address given by &x is 0x100



Signed & Unsigned Int

- Easy to make mistakes

```
unsigned i;
for (i = cnt-2; i >= 0; i--)
  a[i] += a[i+1];
```

- Can be very subtle

```
#define DELTA sizeof(int)
int i;
for (i = CNT; i-DELTA >= 0; i-= DELTA)
  . . .
```

Integer Puzzles

Initialization

```
int x = foo();
int y = bar();
unsigned ux = x;
unsigned uy = y;
```

```
x < 0                ⇒  ((x*2) < 0)
ux >= 0
x & 7 == 7           ⇒  (x<<30) < 0
ux > -1
x > y                ⇒  -x < -y
x * x >= 0
x > 0 && y > 0       ⇒  x + y > 0
x >= 0               ⇒  -x <= 0
x <= 0               ⇒  -x >= 0
(x|-x)>>31 == -1
ux >> 3 == ux/8
x >> 3 == x/8
x & (x-1) != 0
```

Appendix

C Data Type	Typical 32-bit	Typical 64-bit	x86-64
`char`	1	1	1
`short`	2	2	2
`int`	4	4	4
`long`	4	8	8
`float`	4	4	4
`double`	8	8	8
`long double`	–	–	10/16
pointer	4	8	8

Questions