Computer Storage & Representing Numbers

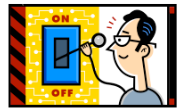
CE 311 K - Introduction to Computer Methods

Daene C. McKinney

Introduction

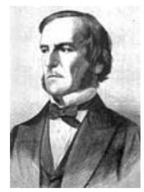
- Computer Storage
- Binary Numbers
- Bits & Bytes
- Computer Storage and Number Representation

Computer Storage

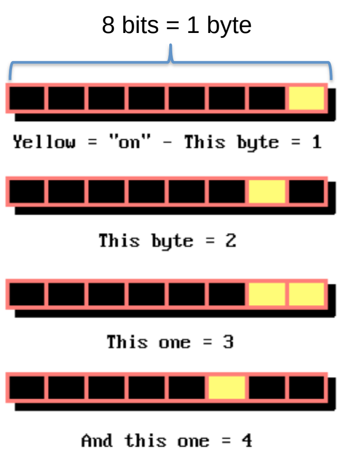


- **<u>Numbers and letters</u>** - not stored using symbols we recognize
- **<u>Bit</u>** - Smallest data item in computer: 0 – 1; on - off
- Bits are bundled in 8s to form bytes
  - comes from "bite," as in the smallest amount of data a computer could "bite" at once

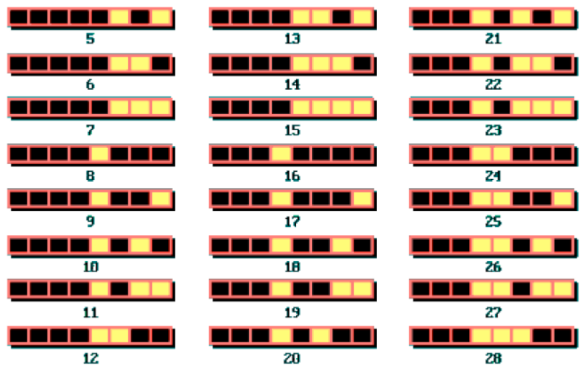
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1 bit	0	1														
2 bits	0	0	1	1												
	0	1	0	1												
4 bits	0	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1
	0	0	0	1	1	1	0	0	1	0	1	0	1	1	0	1
	0	0	1	1	1	0	0	0	1	1	0	1	0	0	1	1
	0	1	1	1	0	0	0	1	0	0	0	0	1	1	1	1



Some Binary Numbers



Some More Binary Numbers



How Many Bytes is ...?

A single text character	1 byte
A typical text word	10 bytes
A typewritten page	2 kilobytes (KB)
A short novel	1 megabyte (MB)
A photograph (low-med res)	2 megabytes
The complete works of Shakespeare	5 megabytes
The complete works of Beethoven	20 gigabytes
50,000 trees made into paper and printed	1 terabyte (TB)
An academic research library	2 terabytes
All U.S. academic research libraries	2 petabytes
All hard disk capacity 1995	20 petabytes
All printed material in the world	200 petabytes

Computer Memory

- Bit = 0 or 1
- Byte (B) = 8 bits
- Word = a few bytes
- Kilobyte (kB) = 2^{10} bytes = 1,024 bytes
- Megabyte (MB) = 2^{20} bytes = 1,048,576 bytes
- Gigabyte (GB) = 2^{30} bytes
- Terabyte (TB) = 2^{40} bytes
- Petabyte (PB) = 2^{50} bytes

  – Facebook > 1 petabytes of users' photos
  – Android "Data" on Star Trek > 80 petabytes

Bits and Bytes

1 byte (= 8 bits): 0 - 255

0 = 00000000
1 = 00000001
2 = 00000010
...
254 = 11111110
255 = 11111111

2 bytes (= 16 bits): 0 - 65,535

0 = 0000000000000000
1 = 0000000000000001
2 = 0000000000000010
...
65534 = 1111111111111110
65535 = 1111111111111111

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$

End here **8 bits = 1 byte** Start here

Representing Integers

High bit **8 bits = 1 byte** Low bit

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$

Example: 101101 (in binary, base-2) = 45 (in decimal, base-10)

$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
		1	0	1	1	0	1
		+32	+0	+8	+4	+0	+1

Sometimes we write: $(101101)_2 = (45)_{10}$

Example

- Convert binary (base 2) to decimal (base 10)
  $(1010010)_2 = (\_\_\_\_\_\_)_{10}$?

$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
1	0	1	0	0	1	0

Example

- Decimal (base 10) to binary (base 2)
- $(92)_{10} = (\_\_\_\_\_\_\_\_)_2$?

$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$

Binary Math

- Binary addition:

```
  010
+ 111
  ---
 1001
```

Start at the right,

First digit:	0 + 1 = 1	
Second digit:	1 + 1 = 10	(carry 1)
Third digit:	0 + 1 + 1 = 10	(carry 1)
Last digit:	0 + 0 + 1 = 1	

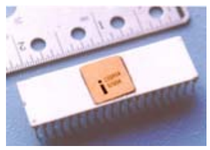
Answer = 1001

Translate to decimal: 2 + 7 = 9.

Chips & CPUs

- **Intel 4004 (1971)**
  - **add 4 bits**
  - **calculator**

- **Intel 8080 (1974)**
  - **first microprocessor**
  - **computer on chip**



- **Intel 8088 (1979)**
  - **IBM PC**

Name	Date	Transistors	Microns	Clock speed	Data width	MIPS
8080	1974	6,000	6	2 MHz	8 bits	0.64
8088	1979	29,000	3	5 MHz	16 bits 8-bit bus	0.33
80286	1982	134,000	1.5	6 MHz	16 bits	1
80386	1985	275,000	1.5	16 MHz	32 bits	5
80486	1989	1,200,000	1	25 MHz	32 bits	20
Pentium	1993	3,100,000	0.8	60 MHz	32 bits 64-bit bus	100
Pentium II	1997	7,500,000	0.35	233 MHz	32 bits 64-bit bus	~300
Pentium III	1999	9,500,000	0.25	450 MHz	32 bits 64-bit bus	~510
Pentium 4	2000	42,000,000	0.18	1.5 GHz	32 bits 64-bit bus	~1,700

Computer Storage

- **Words**:
  - Groups of one or more bytes
- **Word Size**:
  - 4 byte (4*8 = 32-bit)
  - 8 byte (8*8 = 64-bit)
  - Determines min and max values that can be stored

Word Size	Min Integer Value	Max Integer Value
2 Bytes (16 bit)	-32,767	+32,767
4 Bytes (32 bit)	-2,147,483,647	+2,147,483,648

Why 32767???

16 bits: +/- 2^{15} = 32768

High bit

16th bit	15th bit					...	9th bit
+/- sign	2^{14} =16348	2^{13} =8192	2^{12} =4096	2^{11} =2048	2^{10} =1024	2^{9} =512	2^{8} =256
8th bit			...	4th bit	3rd bit	2nd bit	1st bit
2^{7} =128	2^{6} =64	2^{5} =32	2^{4} =16	2^{3} =8	2^{2} =4	2^{1} =2	2^{0} =1

Low bit

Add 'em up = 32767

Hexadecimal (base-16)

- Binary: 0,1
- Decimal: 0,1,2,3,4,5,6,7,8,9
- Hex: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Example: 30A1

16^6	16^5	16^4	16^3	16^2	16^1	16^0
			3	0	A=10	1
			12,288	+0	+160	+1

= 12,449

Floating Point Representation

$m * (b)^e$

Single-precision floating-point number

Bit	31	30 --- 23	22 --- 0
	Sign	Exponent	Mantissa

- Floating point numbers are represented by
  - sign, mantissa, and exponent
- # bits used for each determines precision
  - Single precision: 32 bit (4 byte)
  - Double precision: 64 bit (8 byte)

$0.3x10^{-5}$

Summary

- Computer Storage
- Binary Numbers
- Bits & Bytes
- Computer Storage and Number Representation