Chapter 1

HOW COMPUTERS MANIPULATE DATA

Coming up: Analog vs. Digital
Digital Information

- Computers store all information digitally:
  - numbers
  - text
  - graphics and images
  - video
  - audio
  - program instructions

- In some way, all information is *digitized* - broken down into pieces and represented as numbers.

Coming up: Representing Text Digitally
Representing Text Digitally

- For example, every character is stored as a number, including spaces, digits, and punctuation.
- Corresponding upper and lower case letters are separate characters.

```
Hi, Heather.
```

Coming up: Binary Numbers
Once information is digitized, it is represented and stored in memory using the *binary number system*

- A single binary digit (0 or 1) is called a *bit*
- Devices that store and move information are cheaper and more reliable if they have to represent only two states
- A single bit can represent two possible states, like a light bulb that is either on (1) or off (0)
- Permutations of bits are used to store values

Coming up: Bit Permutations
## Bit Permutations

<table>
<thead>
<tr>
<th>1 bit</th>
<th>2 bits</th>
<th>3 bits</th>
<th>4 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>00</td>
<td>000</td>
<td>0000</td>
</tr>
<tr>
<td>1</td>
<td>01</td>
<td>001</td>
<td>0001</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>010</td>
<td>0010</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>011</td>
<td>0011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100</td>
<td>0100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>101</td>
<td>0101</td>
</tr>
<tr>
<td></td>
<td></td>
<td>110</td>
<td>0110</td>
</tr>
<tr>
<td></td>
<td></td>
<td>111</td>
<td>0111</td>
</tr>
</tbody>
</table>

Each additional bit doubles the number of possible permutations
Bit Permutations

- Each permutation can represent a particular item
- There are $2^N$ permutations of $N$ bits
- Therefore, $N$ bits are needed to represent $2^N$ unique items

How many items can be represented by

- 1 bit? $2^1 = 2$ items
- 2 bits? $2^2 = 4$ items
- 3 bits? $2^3 = 8$ items
- 4 bits? $2^4 = 16$ items
- 5 bits? $2^5 = 32$ items

Coming up: Java and Unicode
Java and Unicode

• How do we map from numbers to characters?

• In Java we use the Unicode specification which maps each character to a 16-bit number.

• So, how many possible characters can we have? $2^{16} = 65536$

• ASCII is an older set that was 8-bits and thus could represent only $2^8 = 256$

• Note: The creators of Unicode started with ASCII, so the 256 ASCII character codes are a subset of Unicode

Coming up: Java and Unicode
Java and Unicode

- See: http://www.alanwood.net/demos/ansi.html

- Unicode also includes some non-printable characters like null, tab, line feed, delete, …

- Why 65,000 characters? We only have 26 letters!

- Unicode is International… and includes our alphabet, but many other countries (Russian, Asian, Arabic, etc…).
- For our alphabet we need both upper and lower case representations!
Binary – How does the computer see numbers?

- Computers represent information digitally, but only using a series of 1s and 0s.
- Binary = Base 2
- Decimal = Base 10
- Hexadecimal = Base 16
In decimal (base 10) we represent a number between 0-9 with one digit. To get any higher we use another position.

23

Place 0 means multiple by \(<base>^0\)
In this case \(10^0 = 1\)
Place 1 means multiply by \(<base>^1\)
In this case \(10^1 = 10\)

So \(23 = 10^1 \times 2 + 10^0 \times 3 = 23\)
Binary Conversions

5037 = 10^3\times5 + 10^2\times0 + 10^1\times3 + 10^0\times7

Now, what about Binary, which is Base 2?
Available digits then are 1 and 0 only.

Binary 101 is what in decimal?

2^2\times1+2^1\times0+2^0\times1 = 4*1 + 2*0 + 1*1 = 5 in decimal =
Binary Conversions

What is 111 in decimal?
A. 111
B. 8
C. 7
D. 12

Coming up: Binary Conversions
Binary Conversions

What is 001 in decimal?
A. 001
B. 4
C. 2
D. 10
Binary Conversions

What is 10 in decimal?
A. 001
B. 4
C. 2
D. 10
Joke

- There are only 10 kinds of people in this world. Those who know binary and those who don’t.
Hexadecimal – base 16

- Base sixteen means we need 16 digits. 0-9 is 10 digits, how do I get more? A,B,C,D,E,F are valid “digits” in Hex. A=10, B=11, C=12, D=13, E=14, F=15

- So, a hex number looks like:
  - 3A or FFF or A2C4B

- What is 1A in decimal?
  - \(16^1 \times 1 + 16^0 \times 10 = 26\) decimal

- Normally hexadecimal numbers are preceded by “0x” which means it is a hex number.

Coming up: What is 0x20 in decimal?
What is 0x20 in decimal?

- A. 20
- B. 16
- C. 32
- D. 18

(Recall: A=10, B=11, C=12, D=13, E=14, F=15)
What is 0x2 in decimal?

- A. 2
- B. 16
- C. 32
- D. 4

(Recall: A=10, B=11, C=12, D=13, E=14, F=15)
What is 0xF1 in decimal?

- A. $16^1 \times 15 + 1$
- B. $16^2 \times 10 + 16$
- C. $16^1 \times 16 + 1$
- D. $16^0 + 16$

(Recall: A=10, B=11, C=12, D=13, E=14, F=15)

Coming up: How to convert from decimal to binary
How to convert from decimal to binary

- Given a number (24) find the largest place value that is lower than the number

\[
\begin{align*}
2^6 &= 64 \\
2^5 &= 32 \\
2^4 &= 16 \\
2^3 &= 8 \\
2^2 &= 4 \\
2^1 &= 2 \\
2^0 &= 1
\end{align*}
\]

- Next divide the number by the place value to determine the digit for that position

\[
\begin{align*}
24/16 &= 1 \\
8/8 &= 1
\end{align*}
\]

- Repeat process with remainder (8 in this example)

\[
\begin{align*}
8/8 &= 1
\end{align*}
\]

- So I need a 1 in the 16 position and 8 position:

\[
= 11000
\]
Convert 9 into binary

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

- 8 is the largest place value that fits inside 9, so
- $9/8 = 1$
- Remainder is 1
- 1 is the largest place value that fits in 1, so
- $1/1 = 1$
- Remainder is 0

1001 = 9

Coming up: Convert 7 into binary?
Convert 7 into binary?

- A. 1101
- B. 110
- C. 111
- D. 101

Coming up: Convert 35 to hexadecimal
Convert 35 to hexadecimal

- $16^3 = 4096$
- $16^2 = 256$
- $16^1 = 16$
- $16^0 = 1$

- 16 is the largest placevalue that fits inside 35, so
- $35/16 = 2$
- Remainder is 3
- $3/1 = 3$
- Remainder is 0
  - $0 \times 23 = 35$

Coming up: Conclusions
Convert 42 to hexadecimal

- $16^3 = 4096$
- $16^2 = 256$
- $16^1 = 16$
- $16^0 = 1$

Coming up: Conclusions
But wait, we’re programmers…

- Writing a program to manipulate numbers in other bases.
Conclusions

You should understand the math to do conversions to/from binary/decimal/hexadecimal

We’ll use this later in a project and lab. You may even see a question on it on the exam.