How we represent bits, numbers, letters?

Communicating in the Blink of an Eye

Lawrence Snyder
University of Washington, Seattle
Key principle: *Information is the presence or absence of a phenomenon at given place/time*

Turn signal is an example

- Phenom: Flashing light
- Present: Flashing
- Absent: Off
- Info: Present == intention to turn in specific direction
- Place (side of car)
- Time: now
The **Presence and Absence** of a phenomenon at a specific place and time abbreviated: PandA

- Phenomena: light, magnetism, charge, mass, color, current, ...

- Detecting depends on phenomenon – but the result must be discrete: was it detected or not; there is no option for “sorta there”

- Place and time apply, but usually default to “obvious” values; not so important to us

- Many alternatives ...
Alternatives ...

- "Presence and absence" is too long, use 0, 1
- At the coffee shop ... record passersby:

  == going right
  == wearing purple

- In multi-state cases, pick one for present, all others are absent
- Two states, means this is a binary system
A Curious Story...

The Diving Bell and the Butterfly
Jean-Dominique Bauby
Asking Yes/No Questions

- A protocol for Yes/No questions
  - One blink == Yes
  - Two blinks == No
- PandA implies that this is not the fewest number of blinks ... really?
Asking Letters

In English ETAOINSHRDLU...
How many questions to encode:

*Plus ça change, plus c'est la même chose?*

Asking in Frequency Order:

ESARINTULOMDPCFBVHGGJQZYXKW

9 12
How many questions to encode: 
*Plus ça change, plus c'est la même chose?*

- Asking in Frequency Order: 
  ESARINTULOMDPCFBVHGJQZYXKW
- Asking in Alphabetical Order: 
  ABCDEFGHIJKLMNOPQRSTUVWXYZ

12 16
Compare Two Orderings

- How many questions to encode:
  
  \[ \text{Plus ça change, plus c'est la même chose?} \]

- Asking in Frequency Order: \(247\)
  
  ESARINTULOMDPCFBVHGJQZYXKW

- Asking in Alphabetical Order: \(324\)
  
  ABCDEFGHIJKLMNOPQRSTUVWXYZ
An Algorithm – A Brief Comment

- Spelling by going through the letters is an algorithm
- Going through the letters in frequency order is a program (also, an algorithm but with the order specified to a particular case, i.e. FR)
- The nurses didn’t look this up in a book ... they invented it to make their work easier; they were thinking computationally, though they probably didn’t know it
PandA is a *binary representation* because it uses 2 patterns.

Bit – it’s a contraction for “binary digit”

Information exists even if the phenom is absent.

*Sherlock Holmes’s* *Mystery of Silver Blaze* -- a popular example where “absent” gives information … the dog didn’t bark, that is the phenomenon wasn’t detected.

Memory -- a position in space/time capable of being set and detected in 2 patterns.
A byte is eight bits treated as a unit

- Adopted by IBM in 1960s
- A standard measure ever since
- Bytes encode the Latin alphabet using ASCII -- the American Standard Code for Information Interchange
<table>
<thead>
<tr>
<th>ASCII</th>
<th>0000</th>
<th>0001</th>
<th>0010</th>
<th>0011</th>
<th>0100</th>
<th>0101</th>
<th>0110</th>
<th>0111</th>
<th>1000</th>
<th>1001</th>
<th>1010</th>
<th>1011</th>
<th>1100</th>
<th>1101</th>
<th>1110</th>
<th>1111</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0010</td>
<td>!</td>
<td>&quot;</td>
<td>#</td>
<td>$</td>
<td>%</td>
<td>&amp;</td>
<td>'</td>
<td>(</td>
<td>)</td>
<td>*</td>
<td>+</td>
<td>,</td>
<td>-</td>
<td>.</td>
<td>/</td>
<td></td>
</tr>
<tr>
<td>0011</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0100</td>
<td>@</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
<td>G</td>
<td>H</td>
<td>I</td>
<td>J</td>
<td>K</td>
<td>L</td>
<td>M</td>
<td>N</td>
<td>O</td>
</tr>
<tr>
<td>0101</td>
<td>P</td>
<td>Q</td>
<td>R</td>
<td>S</td>
<td>T</td>
<td>U</td>
<td>V</td>
<td>W</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0110</td>
<td>`</td>
<td>a</td>
<td>b</td>
<td>c</td>
<td>d</td>
<td>e</td>
<td>f</td>
<td>g</td>
<td>h</td>
<td>i</td>
<td>j</td>
<td>k</td>
<td>l</td>
<td>m</td>
<td>n</td>
<td>o</td>
</tr>
<tr>
<td>0111</td>
<td>p</td>
<td>q</td>
<td>r</td>
<td>s</td>
<td>t</td>
<td>u</td>
<td>v</td>
<td>w</td>
<td>x</td>
<td>y</td>
<td>z</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>6 0</td>
<td>6 1</td>
<td>6 2</td>
<td>6 3</td>
<td>N I</td>
<td>N L</td>
<td>S S</td>
<td>E S</td>
<td>H S</td>
<td>H J</td>
<td>Y S</td>
<td>P D</td>
<td>F V</td>
<td>R I</td>
<td>S 2</td>
<td>S 3</td>
</tr>
<tr>
<td>1001</td>
<td>D C</td>
<td>P 1</td>
<td>P Z</td>
<td>S E</td>
<td>C C</td>
<td>M M</td>
<td>S P</td>
<td>E P</td>
<td>Q B</td>
<td>Q O</td>
<td>Q A</td>
<td>C S</td>
<td>S T</td>
<td>O S</td>
<td>P M</td>
<td>A P</td>
</tr>
<tr>
<td>1010</td>
<td>A o</td>
<td>i</td>
<td>c</td>
<td>£</td>
<td>ã</td>
<td>¥</td>
<td>†</td>
<td>$</td>
<td>℃</td>
<td>£</td>
<td>℃</td>
<td>«</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>1011</td>
<td>±</td>
<td>2</td>
<td>3</td>
<td>µ</td>
<td>‰</td>
<td>′</td>
<td>″</td>
<td>™</td>
<td>™</td>
<td>™</td>
<td>™</td>
<td>™</td>
<td>™</td>
<td>™</td>
<td>™</td>
<td>™</td>
</tr>
<tr>
<td>1100</td>
<td>À</td>
<td>Á</td>
<td>Â</td>
<td>Ä</td>
<td>Å</td>
<td>Æ</td>
<td>Ç</td>
<td>Ê</td>
<td>Ë</td>
<td>Ê</td>
<td>Ë</td>
<td>Ê</td>
<td>Ë</td>
<td>Ê</td>
<td>Ë</td>
<td>Ê</td>
</tr>
<tr>
<td>1101</td>
<td>ë</td>
<td>Ë</td>
<td>Ë</td>
<td>Ë</td>
<td>Ë</td>
<td>Ë</td>
<td>Ë</td>
<td>Ë</td>
<td>Ë</td>
<td>Ë</td>
<td>Ë</td>
<td>Ë</td>
<td>Ë</td>
<td>Ë</td>
<td>Ë</td>
<td>Ë</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1111</td>
<td>ð</td>
<td>į</td>
<td>ø</td>
<td>ò</td>
<td>ò</td>
<td>ò</td>
<td>ò</td>
<td>ò</td>
<td>ò</td>
<td>ò</td>
<td>ò</td>
<td>ò</td>
<td>ò</td>
<td>ò</td>
<td>ò</td>
<td>ò</td>
</tr>
</tbody>
</table>
UTF-8

Uniform Transformation Format for bytes (UTF-8) is universal ... all characters have a place: 1,2,3,4 B
UTF-8

Uniform Transformation Format for bytes (UTF-8) is universal ... all characters have a place: 1,2,3,4 B

- 100,000 characters

Can you read this?
Encoding Information

- Bits and bytes encode the information, but that’s not all
  - Tags encode format and some structure in word processors
  - Tags encode format and some structure in HTML
  - In the *Oxford English Dictionary* tags encode structure and some formatting
  - Tags are one form of meta-data: *meta-data* is information about information
byte (balt). *Computers.* [Arbitrary, prob. influenced by bit sb.° and bite sb.] A group of eight consecutive bits operated on as a unit in a computer. 1964 Blaauw & Brooks in *IBM Systems Jnl.* III. 122 An 8-bit unit of information is fundamental to most of the formats [of the System/360]. A consecutive group of $n$ such units constitutes a field of length $n$. Fixed-length fields of length one, two, four, and eight are termed bytes, halfwords, words, and double words respectively. 1964 *IBM Jnl. Res. & Developm.* VIII. 97/1 When a byte of data appears from an I/O device, the CPU is seized, dumped, used and restored. 1967 P. A. Stark *Digital Computer Programming* xix. 351 The normal operations in fixed point are done on four bytes at a time. 1968 *Dataweek* 24 Jan. 1/1 Tape reading and writing is at from 34,160 to 192,000 bytes per second.
Positional Notation

- Binary numbers, like decimal numbers, use *place notation*

\[
\begin{align*}
1101 &= 1 \times 1000 + 1 \times 100 + 0 \times 10 + 1 \times 1 \\
 &= 1 \times 10^3 + 1 \times 10^2 + 0 \times 10^1 + 1 \times 10^0
\end{align*}
\]

except that the base is 2 not 10

\[
\begin{align*}
1101 &= 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 \\
 &= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0
\end{align*}
\]

1101 in binary is 13 in decimal
Binary is just like decimal except that it uses base 2 rather than base 10 ...
Representing 160 in Binary

Given a binary number, add up the powers of 2 corresponding to its 1s

\[
\begin{align*}
1 \times 2^7 &= 1 \times 128 &= 128 \\
0 \times 2^6 &= 0 \times 64 &= 0 \\
1 \times 2^5 &= 1 \times 32 &= 32 \\
0 \times 2^4 &= 0 \times 16 &= 0 \\
0 \times 2^3 &= 0 \times 8 &= 0 \\
0 \times 2^2 &= 0 \times 4 &= 0 \\
0 \times 2^1 &= 0 \times 2 &= 0 \\
0 \times 2^0 &= 0 \times 1 &= 0 \\
\end{align*}
\]

\[= 160\]
Representing 76 in Binary

Given a binary number, add up the powers of 2 corresponding to 1s

\[
\begin{align*}
0 \times 2^7 &= 0 \times 128 &= 0 \\
1 \times 2^6 &= 1 \times 64 &= 64 \\
0 \times 2^5 &= 0 \times 32 &= 0 \\
0 \times 2^4 &= 0 \times 16 &= 0 \\
1 \times 2^3 &= 1 \times 8 &= 8 \\
1 \times 2^2 &= 1 \times 4 &= 4 \\
0 \times 2^1 &= 0 \times 2 &= 0 \\
0 \times 2^0 &= 0 \times 1 &= 0 \\
\hline
= 76
\end{align*}
\]
So Husky purple is (160,76,230) which is

\[
\begin{align*}
1010 & 0000 \\
0100 & 1100 \\
1110 & 0110 \\
\end{align*}
\]

\[
\begin{align*}
160 & \\
76 & \\
230 & \\
\end{align*}
\]

Suppose you decide it’s not “red” enough

- Increase the red by 16 = 1 0000

\[
\begin{align*}
1010 & 0000 \\
+ & 1 0000 \\
\hline
1011 & 0000
\end{align*}
\]

Adding in binary is pretty much like adding in decimal
**Adding In Binary ... like Decimal**

- Increase by 16 more

  \[
  \begin{array}{c}
  \text{Carries} \\
  00110 000 \\
  1011 0010 \\
  + 1 0100 \\
  1100 0110
  \end{array}
  \]

The rule: When the “place sum” equals the radix or more, subtract radix & carry

*Check it out online: searching binary addition hits 19M times, and all of the p.1 hits are good explanations*
Find Binary From Decimal

What is 230? Fill in the Table:

<table>
<thead>
<tr>
<th>Num Being Converted</th>
<th>230</th>
<th>230</th>
<th>102</th>
<th>38</th>
<th>6</th>
<th>6</th>
<th>6</th>
<th>2</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place Value</td>
<td>256</td>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Subtract</td>
<td>102</td>
<td>38</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Num</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Place number to be converted into the table; fill place value row with decimal powers of 2

<table>
<thead>
<tr>
<th>Num Being Converted</th>
<th>Place Value</th>
<th>256</th>
<th>128</th>
<th>64</th>
<th>32</th>
<th>16</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subtract</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Num</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Rule: Subtract PV from the number; a positive result gives new number and “1”; otherwise, “0”

<table>
<thead>
<tr>
<th>Num Being Converted</th>
<th>230</th>
<th>230</th>
<th>256</th>
<th>128</th>
<th>64</th>
<th>32</th>
<th>16</th>
<th>8</th>
<th>4</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subtract</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Num</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Rule: Subtract PV from the number; a positive result gives new number and “1”; otherwise, “0”

<table>
<thead>
<tr>
<th>Num Being Converted</th>
<th>230</th>
<th>230</th>
<th>102</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place Value</td>
<td>256</td>
<td>128</td>
<td>64</td>
</tr>
<tr>
<td>Subtract</td>
<td>102</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Num</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Find Binary From Decimal

Rule: Subtract PV from the number; a positive result gives new number and “1”; otherwise, “0”

<table>
<thead>
<tr>
<th>Num Being Converted</th>
<th>230</th>
<th>230</th>
<th>102</th>
<th>38</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place Value</td>
<td>256</td>
<td>128</td>
<td>64</td>
<td>32</td>
</tr>
<tr>
<td>Subtract</td>
<td>102</td>
<td>38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Num</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Rule: Subtract PV from the number; a positive result gives new number and “1”; otherwise, “0”

<table>
<thead>
<tr>
<th>Num Being Converted</th>
<th>230</th>
<th>230</th>
<th>102</th>
<th>38</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place Value</td>
<td>256</td>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
</tr>
<tr>
<td>Subtract</td>
<td>102</td>
<td>38</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Num</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Rule: Subtract PV from the number; a positive result gives new number and “1”; otherwise, “0”

<table>
<thead>
<tr>
<th>Num Being Converted</th>
<th>230</th>
<th>230</th>
<th>102</th>
<th>38</th>
<th>6</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place Value</td>
<td>256</td>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>Subtract</td>
<td>102</td>
<td>38</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Num</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Find Binary From Decimal

© 2010 Larry Snyder, CSE
Rule: Subtract PV from the number; a positive result gives new number and “1”; otherwise, “0”

<table>
<thead>
<tr>
<th>Num Being Converted</th>
<th>230</th>
<th>230</th>
<th>102</th>
<th>38</th>
<th>6</th>
<th>6</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place Value</td>
<td>256</td>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>Subtract</td>
<td>102</td>
<td>38</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Num</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Rule: Subtract PV from the number; a positive result gives new number and “1”; otherwise, “0”

<table>
<thead>
<tr>
<th>Num Being Converted</th>
<th>230</th>
<th>230</th>
<th>102</th>
<th>38</th>
<th>6</th>
<th>6</th>
<th>6</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place Value</td>
<td>256</td>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Subtract</td>
<td>102</td>
<td>38</td>
<td>6</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Num</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Rule: Subtract PV from the number; a positive result gives new number and “1”; otherwise, “0”

<table>
<thead>
<tr>
<th>Num Being Converted</th>
<th>230</th>
<th>230</th>
<th>102</th>
<th>38</th>
<th>6</th>
<th>6</th>
<th>6</th>
<th>2</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place Value</td>
<td>256</td>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Subtract</td>
<td>102</td>
<td>38</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Num</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Rule: Subtract PV from the number; a positive result gives new number and “1”; otherwise, “0”

<table>
<thead>
<tr>
<th>Num Being Converted</th>
<th>230</th>
<th>230</th>
<th>102</th>
<th>38</th>
<th>6</th>
<th>6</th>
<th>6</th>
<th>2</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place Value</td>
<td>256</td>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Subtract</td>
<td>102</td>
<td>38</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Binary Num</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Read off the result: 0 1110 0110
Final Fact: Bits Are IT

- We ALL KNOW computers represent data by binary numbers
- NOT QUITE TRUE
- Computers represent information by bits
  - ASCII, numbers (yes, in binary), metadata +
    computer instructions, color, sound, video, etc.
- Fundamental Fact –
  - Bits can represent ALL information
  - Bits have no inherent meaning ... you don’t know
    what 1100 0100 1010 1110 means ... it could be
    anything