Number systems

- Last lecture
  - Course overview
  - The Digital Age
- Today's lecture
  - Binary numbers
  - Base conversion
  - Number systems
  - Two-complement
  - A/D and D/A conversion

The basics: Binary numbers

- Bases we will use
  - Binary: Base 2
  - Octal: Base 8
  - Hexadecimal: Base 16
- Positional number system
  - \(101_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0\)
  - \(E_{16} = 6 \times 16^1 + 3 \times 16^0\)
- Additon and subtraction
  - \(101\)
  - + 1010
  - _______
  - 10101

Binary \(\rightarrow\) hex/decimal/octal conversion

- Conversion from binary to octal/hex
  - Binary: 10011110001
  - Octal: 10 011 110 001 = 23618
  - Hex: 100 1111 0001 = 4F116
- Conversion from binary to decimal
  - \(101_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 5_{10}\)
  - \(63.48_{10} = 6 \times 8^1 + 3 \times 8^0 + 4 \times 8^{-1} = 51.5_{10}\)

Decimal \(\rightarrow\) binary/octet/hex conversion

- Bases we will use
  - Binary: Base 2
  - Octal: Base 8
  - Hexadecimal: Base 16
- Positional number system
  - \(N=56_{10} = 1\ 1\ 1\ 0\ 0\)
  - \(Q=N/2=56/2=11100/2=1100\) remainder 0
- Why does this work?
  - \(N=56_{10}=11100_2\)
  - Each successive divide liberates an LSB

Digital

- Digital = discrete
  - Binary codes (example: BCD)
  - Decimal digits 0-9
  - DNA nucleotides
- Binary codes
  - Represent symbols using binary digits (bits)
- Digital computers:
  - I/O is digital
    - ASCII, decimal, etc.
  - Internal representation is binary
    - Process information in bits

Decimal

- Conversion from decimal to binary
  - \(N=56_{10} = 1\ 1\ 1\ 0\ 0\)
- Conversion from decimal to octal
  - \(N=56_{10} = 1\ 1\ 1\ 0\ 0\)
- Conversion from decimal to hexadecimal
  - \(N=56_{10} = 1\ 1\ 1\ 0\ 0\)

Number systems

- How do we write negative binary numbers?
- Historically: 3 approaches
  - Sign-and-magnitude
  - Ones-complement
  - Two's-complement
- For all 3, the most-significant bit (msb) is the sign digit
  - 0 = positive
  - 1 = negative
- Learn two's-complement
  - Simplifies arithmetic
  - Used almost universally
Sign-and-magnitude

- The most-significant bit (msb) is the sign digit
  - 0 = positive
  - 1 = negative
- The remaining bits are the number’s magnitude
- Problem 1: Two representations for zero
  - 0 = 0000 and also –0 = 1000
- Problem 2: Arithmetic is cumbersome

<table>
<thead>
<tr>
<th>Add</th>
<th>Subtract</th>
<th>Compare and subtract</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0100</td>
<td>0100 – 4 1100 1100</td>
</tr>
<tr>
<td>+3</td>
<td>+0111</td>
<td>=7 0111 = 0000</td>
</tr>
<tr>
<td>–3</td>
<td>–0111</td>
<td>≠ 1111 = 1001</td>
</tr>
</tbody>
</table>

Ones-complement

- Negative number: Bitwise complement positive number
  - 0011 = 3
  - 1100 = –3
- Solves the arithmetic problem
  
<table>
<thead>
<tr>
<th>Add</th>
<th>Invert, add, add carry</th>
<th>Invert and add</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0100</td>
<td>0100</td>
</tr>
<tr>
<td>+3</td>
<td>+0111</td>
<td>=7 0111</td>
</tr>
<tr>
<td>–3</td>
<td>–1011</td>
<td>≠ 1101</td>
</tr>
</tbody>
</table>

- Remaining problem: Two representations for zero
  - 0 = 0000 and also –0 = 1111

Twos-complement

- Negative number: Bitwise complement plus one
  - 0011 = 3
  - 1101 = –3
- Number wheel
- Only one zero!
- msb is the sign digit
  - 0 = positive
  - 1 = negative

<table>
<thead>
<tr>
<th>Add</th>
<th>Subtract</th>
<th>Compare and subtract</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0100</td>
<td>0100 – 4 1100 1100</td>
</tr>
<tr>
<td>+3</td>
<td>+0011</td>
<td>=7 0011 = 0000</td>
</tr>
<tr>
<td>–3</td>
<td>–1010</td>
<td>≠ 1010 = 1001</td>
</tr>
</tbody>
</table>

Twos-complement (con't)

- Complementing a complement = the original number
- Arithmetic is easy
  - Subtraction = negation and addition
  - Easy to implement in hardware

<table>
<thead>
<tr>
<th>Add</th>
<th>Invert and add</th>
<th>Invert and add</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0100</td>
<td>0100</td>
</tr>
<tr>
<td>+3</td>
<td>+0011</td>
<td>=7 0011</td>
</tr>
<tr>
<td>–3</td>
<td>–1010</td>
<td>≠ 1010</td>
</tr>
<tr>
<td>drop carry</td>
<td>= 0000</td>
<td>–1 1111</td>
</tr>
</tbody>
</table>

Miscellaneous

- Twos-complement of non-integers
  - 1.6875₁₀ = 01.1011₂
  - –1.6875₁₀ = 10.0101₂
- Sign extension
  - Write +6 and –6 as twos complement
  - 0110 and 1010
  - Sign extend to 8-bit bytes
  - 00000010 and 11111010
- Can’t infer a representation from a number
  - 11001 is 25 (unsigned)
  - 11001 is –9 (sign magnitude)
  - 11001 is –6 (ones complement)
  - 11001 is –7 (twos complement)

Twos-complement overflow

- Summing two positive numbers gives a negative result
- Summing two negative numbers gives a positive result
Twos-complement overflow (cont’d)

- Correct results
  - 1111 -1
  - 0011 +3
  - 1010 -6
  - 0010 +2
  - 1001 -7
  - 0101 +5

- Incorrect results
  - 0110 +6
  - 1001 -7
  - 1110 +6
  - 1010 -6
  - 0101 +3

- Overflow condition
  - Carry from 2sb-msb and
  - carry from msb-Cout are different

<table>
<thead>
<tr>
<th>2sb-msb</th>
<th>msb-Cout</th>
<th>Overflow</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Gray and BCD codes

<table>
<thead>
<tr>
<th>Decimal Symbols</th>
<th>Gray Code</th>
<th>Decimal Symbols</th>
<th>BCD Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000</td>
<td>0</td>
<td>0000</td>
</tr>
<tr>
<td>1</td>
<td>0001</td>
<td>1</td>
<td>0001</td>
</tr>
<tr>
<td>2</td>
<td>0010</td>
<td>2</td>
<td>0010</td>
</tr>
<tr>
<td>3</td>
<td>0011</td>
<td>3</td>
<td>0011</td>
</tr>
<tr>
<td>4</td>
<td>0100</td>
<td>4</td>
<td>0100</td>
</tr>
<tr>
<td>5</td>
<td>0101</td>
<td>5</td>
<td>0101</td>
</tr>
<tr>
<td>6</td>
<td>0110</td>
<td>6</td>
<td>0110</td>
</tr>
<tr>
<td>7</td>
<td>0111</td>
<td>7</td>
<td>0111</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>8</td>
<td>1000</td>
</tr>
<tr>
<td>9</td>
<td>1001</td>
<td>9</td>
<td>1001</td>
</tr>
</tbody>
</table>

The physical world is analog

- Digital systems need to
  - Measure analog quantities
  - Speech waveforms, etc
  - Control analog systems
  - Drive motors, etc

- How do we connect the analog and digital domains?
  - Analog-to-digital converter (ADC or A/D)
    - Example: CD recording
  - Digital-to-analog converter (DAC or D/A)
    - Example: CD playback

Sampling

- Quantization
  - Conversion from analog to discrete values

- Quantizing a signal
  - We sample it

Conversion

- Encoding
  - Assigning a digital word to each discrete value
  - Encoding a quantized signal
    - Encode the samples
    - Typically Gray or binary codes