2進数とその簡単な計算

情報ネットワーク工学入門

只木進一（理工学部）

10進数とその演算

- {0,1,2,3,4,5,6,7,8,9}の10種類の記号
- 加法
  - 10×10通りの加法規則と桁上がり
- 乗法
  - 10×10通りの乗算規則
- 減法・徐法
  - 加法・乗法の逆演算

2進数とその演算

- {0,1}の2種類の記号
- 加法・乗法
  - 2 × 2の演算規則
- 減法・除法
  - 補数を使った加算への置き換え
- 規則が単純
- 論理回路で容易に実装可能

コンピュータ内でのデータの取り扱い

- 2進数一けた[0,1]をbitと呼ぶ
- 2進数8桁[0,255]をbyteと呼ぶ
  - ASCIIコード：7bitで数字やアルファベットを表現
  - 日本語コード：JIS、SJIS、EUCは2バイト
  - 多言語混在：UTF-8など

10進数⇔2進数

$$53 = 32 + 16 + 4 + 1 = 2^5 + 2^4 + 2^2 + 2^0$$

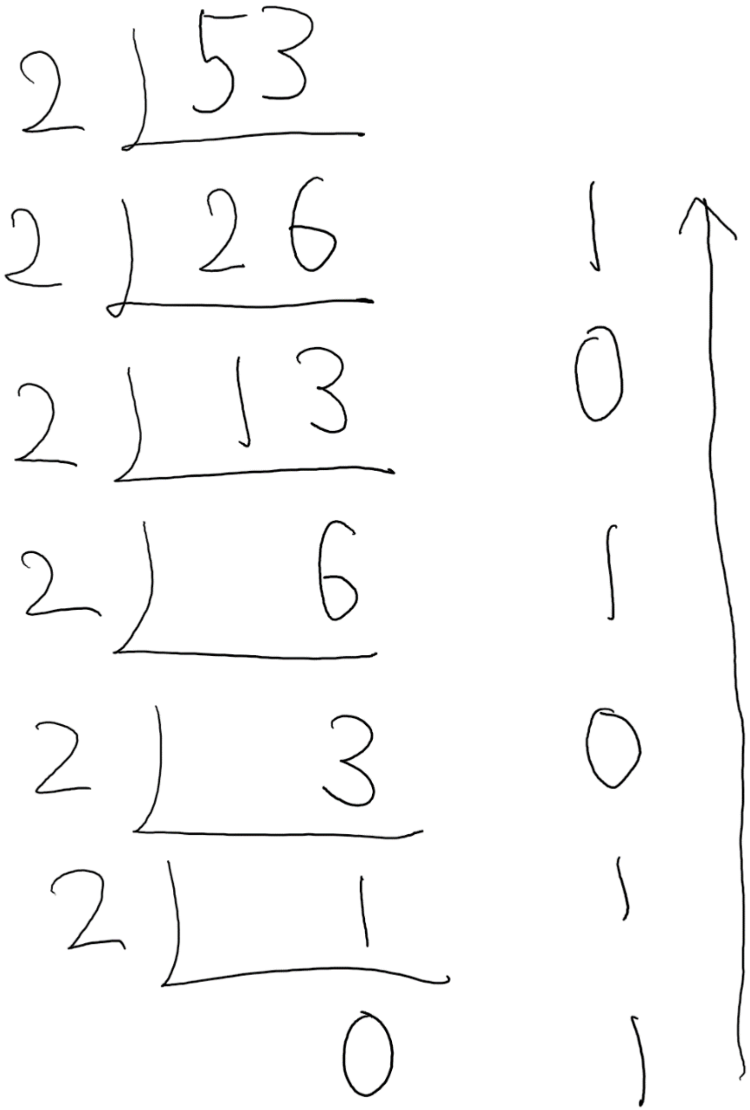
$$= (00110101)_2$$

$$130 = 128 + 2 = 2^7 + 2^1$$

$$= (10000010)_2$$

$$163 = 128 + 32 + 2 + 1 = 2^7 + 2^5 + 2^1 + 2^0$$

$$= (10100011)_2$$



2で割った商と余りを求める
これを0になるまで繰り返す

余りを下から上に読む

$$53 = (00110101)_2$$

2^nはある程度覚えよう

$$2^0 = 1$$
$$2^1 = 2$$
$$2^2 = 4$$
$$2^3 = 8$$
$$2^4 = 16$$
$$2^5 = 32$$
$$2^6 = 64$$
$$2^7 = 128$$
$$2^8 = 256$$
$$2^9 = 512$$
$$2^{10} = 1024$$

なぜ、コンピュータは2進数を使うのか

- 素子が簡単にできる
  - 状態はオンとオフの二つ
- 演算規則が簡素

a	b	$a+b$
0	0	0
0	1	1
1	0	1
1	1	10

a	b	$a \times b$
0	0	0
0	1	0
1	0	0
1	1	1

二進数の計算の例
加法・乗法

$$(101)_2 + (11)_2 = (1000)_2$$

$$\begin{aligned}(101)_2 \times (11)_2 &= (101)_2 \times (1)_2 + (101)_2 \times (10)_2 \\ &= (101)_2 + (1010)_2 = (1111)_2\end{aligned}$$

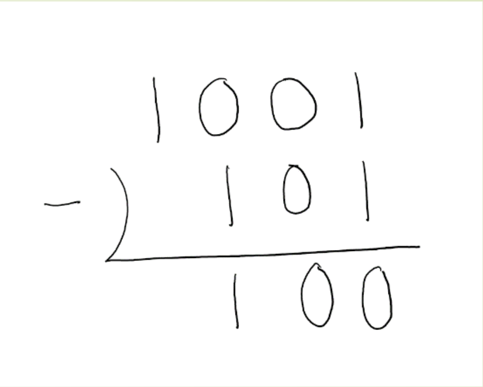
$$\begin{array}{r} 101 \\ +\ \ 11 \\ \hline 1000 \end{array}$$

$$\begin{array}{r} 1011 \\ +\ \ 110 \\ \hline 10001 \end{array}$$

$$\begin{array}{r} 101 \\ \times\ \ 11 \\ \hline 101 \\ +\ 101 \\ \hline 1111 \end{array}$$

$$\begin{array}{r} 101 \\ \times\ 101 \\ \hline 101 \\ +\ 101 \\ \hline 11001 \end{array}$$

減法



- 8ビットと考える[0,256)

$$9-5=(0001001)_2-(00000101)_2$$
$$=(00000100)_2=4$$

- 引き算は、上の桁から「借りる」操作が必要
  - 処理が複雑になる

減算：続き

- 5に対して2の補数を計算
  - ビットを反転して1を加える:$((256-1)-5)+1$
  - $(11111010)_2+(00000001)=(11111011)_2$

減算：続き

- 加算して8ビット部分を計算
- $9 + ((256 - 1) - 5) + 1 = 256 + (9 - 5)$
- $(00001001)_2 + (11111011)_2 = (100000100)_2$
- 8bit部分
  - $(00000100)_2 = 4$

減算：続き 5-9

- $9 = (00001001)_2$に対する「2の補数」
  - $(11110110)_2 + (00000001)_2 = (11110111)_2$
- $5 - 9 = (00000101)_2 + (11110111)_2 = (11111100)_2$
- これは、4に対する「2の補数」
- 「2の補数」は対応するマイナスの数

徐算

- 2進のため、順次、減算を行う
- 減算の際に、補数を利用する
- 例：$65 \div 11 = (01000001)_2 \div (00001011)_2$
- $(01000001)_2 = (01000001)_2 \times (00000101)_2 + (00001010)_2 = 11 \times 5 + 10$

$$
\begin{array}{r}
101 \\
1011 \overline{)01000001} \\
1011 \\
\hline
10101 \\
1011 \\
\hline
1010
\end{array}
$$

負の数

- 8bitのうち、最上位を符号として扱う
- 例：$0 - 1 = (11111111)_2$
  - 1の「2の補数」に相当
- プログラミング言語では（java)
  - int型：32bit
    - 最上位は符号bit
    - $[-2^{31}, 2^{31} - 1]$

小数

- $(0.101)_2 = 2^{-1} + 2^{-3} = \frac{1}{2} + \frac{1}{8} = \frac{5}{8} = 0.625$
- コンピュータは、浮動小数(floating point numbers)として保持している
  - $(0.101)_2 = 2^{-1} \times (1 + (0.01)_2)$

接頭辞：3桁毎

- $1\mathrm{k} = 10^3$、$1\mathrm{M} = 10^3\mathrm{k}$、$1\mathrm{G} = 10^3\mathrm{M}$、$1\mathrm{T} = 10^3\mathrm{G}$、$1\mathrm{P} = 10^3\mathrm{T}$
- $1\mathrm{m} = 10^{-3}$、$1\mu = 10^{-3}\mathrm{m}$、$1\mathrm{n} = 10^{-3}\mu$

接頭辞：3桁毎

- 2進の場合には、1000の代わりに $2^{10} = 1024$を使う
- 正確に2^{10}毎の場合
  - 1ki (kilobinary), 1Mi (Megabinary)などと使う

10進数、2進数、8進数、16進数

- n進数：使える記号がn個
- 10進数：{0,1,2,3,4,5,6,7,8,9}
  - 9+1=10
- 2進数：{0,1}
  - 1+1=10
- 8進数：{0,1,2,3,4,5,6,7}
  - 7+1=10
- 16進数：{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}
  - F+1=10

16進数

- 16進数は良く利用される
- 文字コード
  - 1Byte⇔8bit⇔[0,255]⇔[00,ff]
  - 日本語は2Byte
  - http://www.unicodetables.com/
- MACアドレス
  - 8bit×6, 16進で表記

16進の例

- 「佐」のUnicodeは4F50
  - 0x4F $= 4 \times 16 + 15 = 64 + 15 = 79$
  - 0x50 $= 5 \times 16 + 0 = 80$
  - “0x”は16進であることを表す記号