

計算機システムの基礎 (第4回配布)

第2章 データ表現

(1) 数値データ (numeric data)

整数(integer), 浮動小数点数(floating point number)等

(2) 非数値データ (nonnumeric data)

文字コード、論理データ、制御コードなど

(中間試験は [6/3 or] 6/10の予定です)

担当: 福井大学 大学院工学研究科
 情報・メディア工学専攻
 森 眞一郎 (moris@u-fukui.ac.jp)

2. 基数の変換

2進数と10進数の相互変換

10進数⇒2進数

1) 基本形 (整数部 と 小数部を別々に計算)

例えば 26.875d を 2進数に変換

整数部	小数部
26/2 = 13...0	0.625 * 2 = 1.25
13/2 = 6...1	0.25 * 2 = 0.50
6/2 = 3...0	0.50 * 2 = 1.00 (最下位桁)
3/2 = 1...1	
1/2 = 0...1 (最上位桁)	

検算!

11010b = 16+8+2 = 26 (正解!) 0.101b = 0.5 + 0.125 = 0.625 (正解!)

2) 引き算の利用(整数部 と 小数部を別々に計算)

27 - 16 = 11 (16d=1000b)	0.875 - 0.5 = 0.375 (0.5d = 0.1b)
11 - 8 = 3 (8d= 1000b)	0.375 - 0.25 = 0.125 (0.25d = 0.01b)
3 - 2 = 1 (2d= 10b)	0.125 - 0.125 = 0 (0.125d = 0.001b)
1 - 1 = 0 (1d= 1b)	

便利な表?

0.5	2
0.25	4
0.125	8
0.0625	16
0.03125	32
	64
	128
	256
	512
	1024

2. 基数の変換

2進数と10進数の相互変換

2進数表現での少数表現の限界 ... 有限桁の2進数で表現できない数値が存在!

0.725 * 2 = 1.45	
0.45 * 2 = 0.9	
0.9 * 2 = 1.8	
0.8 * 2 = 1.6	0.8 * 2 = 1.6
0.6 * 2 = 1.2	0.6 * 2 = 1.2
0.2 * 2 = 0.4	0.2 * 2 = 0.4
0.4 * 2 = 0.8	0.4 * 2 = 0.8

0.725d = 0.1011100

循環小数表現となり、有限桁で表すことが不可能

有限ビットで表現すると、真の値 との差が 「打ち切り誤差」となる

少数点以下 mビットで打ち切った場合の 打ち切り誤差の絶対値は 2^{-m} 未満

(例えば少数点以下2ビットで打ち切ると、表現可能な小数部は
 0.00 0.25 0.50 0.75 の4種類で 誤差の絶対値は 0.25 未満)

2. 基数の変換

2進数と16進数の相互変換

2進数 nビット で 0~2ⁿ⁻¹までの
 2ⁿ通りの数値を表現可能

2進数3桁 は 8進数1桁 に相当する表現が可能

2進数4桁 は 16進数1桁 に相当する表現が可能

10進	2進	8進	16進	10進	2進	8進	16進
0	0000	0	0	8	1000	10	8
1	0001	1	1	9	1001	11	9
2	0010	2	2	10	1010	12	A
3	0011	3	3	11	1011	13	B
4	0100	4	4	12	1100	14	C
5	0101	5	5	13	1101	15	D
6	0110	6	6	14	1110	16	E
7	0111	7	7	15	1111	17	F

3F40h = 0011_1111_0100_0000b
 (0x3F40 = 0b0011_1111_0100_0000)

3	F	4	0
0	0	1	1
1	1	1	1
1	1	0	1
0	0	0	0
0	0	0	0

1011_1110b = BEh
 (0b1011_1110 = 0x3BE)

1	0	1	1	1	1	1	0
B		E					

1111_1111b = FFh
 (= 15d*16d + 15d = 255d)

1	1	1	1	1	1	1	1
F		F					

3. 負数の表現と加減算

符号付の2進数表現

n bitで表現された2進数の最上位ビット(Most Significant Bit:MSB)を符号ビットとし、**MSBをSとすると S=0のとき正、S=1のとき負の数**をあらわす。

下位 n-1ビットの表現法として、例えば以下の3つが考えられる

符号+絶対値

下位 n-1ビットで、数値の絶対値を表現する

2の補数

符号を含めた n bitで2の補数表現とする

1の補数

符号を含めた n bitで1の補数表現とする

実数の表現(2進数で符号+絶対値表記の例)

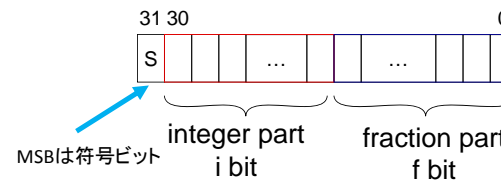
固定小数点表現

- 整数部(integer part) i桁と小数部(fraction part)j桁

で実数を表現

10110.1 -1011.01 1.01101
(i=5,f=1) (i=4,f=2) (i=1,f=5)

32bit固定小数点表現の例



特に
i=31,f=0 の場合は 整数
逆に
i=0, f=31の場合は 純小数

教科書P.44 図4 は整数 or 純小数のみに言及

実数の表現(10進数)

固定小数点表現 (小数点の位置は絶対位置)

- 日常的に接している実数表現
- 整数部 (integer part) i桁と小数部(fraction part) f桁で表現
12345.6 -9234.56 5.23456
(i=5,f=1) (i=4,f=2) (i=1,f=5)

浮動小数点表現 (小数点の位置は相対位置)

- 1.23456x10⁴=12.3456x10³=123.456x10² (=123456.0x10⁻¹)
- 正規化表現 [(-1)^S × M × B^E形式] (科学記数法[有効桁数が重要])
- 整数部1桁(0以外の数値)+小数部k桁 (以下はk=5の例)
1.23456x10⁴ -9.23456x10³ 5.23456x10⁰

この例では有効桁数は10進6桁

3. 負数の表現と加減算

補数(Compliment)

r 進数 n 桁で数を表すとき, r 進数の数 a に対して
 $r^n - a$ を a の **r の補数** (基数の補数)
 $r^n - a - 1$ を a の **r-1 の補数** (減基数の補数)
 と呼ぶ。

10進数2桁 で考えると

13 の 10の補数 は $10^2 - 13 = 87$ 9の補数は 86
 08 の 10の補数 は $10^2 - 8 = 92$ 9の補数は 91
 49 の 10の補数 は $10^2 - 49 = 51$ 9の補数は 50

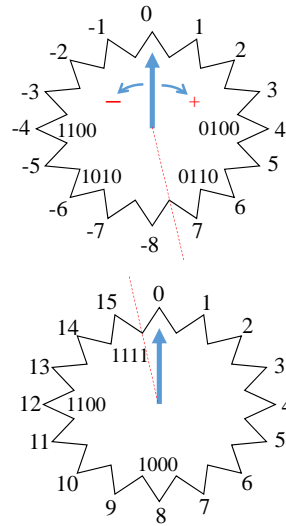
10進数3桁 で考えると

913 の 10の補数 は $10^3 - 913 = 087$ 9の補数は 086
 001 の 10の補数 は $10^3 - 1 = 999$ 9の補数は 998
 002 の 10の補数 は $10^3 - 2 = 998$ 9の補数は 997

符号付きの2進数表現

4bitの例

10進数	2の補数	10進数	2の補数
--	--	-8	1000
+7	0111	-7	1001
+6	0110	-6	1010
+5	0101	-5	1011
+4	0100	-4	1100
+3	0011	-3	1101
+2	0010	-2	1110
+1	0001	-1	1111
0	0000	--	--



3. 負数の表現と加減算

補数(Compliment)と引き算の関係

r 進数 n 桁で数を表すとき, r 進数の数 a に対して
 $r^n - a$ を a の **r の補数** (基数の補数)
 $r^n - a - 1$ を a の **r-1 の補数** (源基数の補数)
 と呼ぶ。

いま, a の r の補数を a' とすると, b から a を引くという作業は

$$b - a = b + (r^n - a) - r^n = b + a' - r^n \quad (= b + a' \pmod{2^n})$$

となり, b に a の r の補数を加える 計算を行ない, 桁あふれを無視する動作と等価

計算結果が $\frac{r^n}{2}$ から $r^n - 1$ までの数字の場合, 計算結果は r の補数として表現された負の数と考えればよい。

[r=2のときの別説明]

$$X - Y = X + (-Y) = X + (\overline{Y} + 1) \pmod{2^N}$$

↑ Yの1の補数

3. 負数の表現と加減算

補数(Compliment)と引き算の関係

いま, a の r の補数を a' とすると, b から a を引くという作業は

$$b - a = b + (r^n - a) - r^n = b + a' - r^n \quad (= b + a' \pmod{2^n})$$

となり, b に a の r の補数を加える 計算を行ない, 桁あふれを無視する動作と等価

計算結果が $\frac{r^n}{2}$ から $r^n - 1$ までの数字の場合, 計算結果は r の補数として表現された負の数と考えればよい。

10進数表現での減算の例

(注: 10の補数表現を用いて

2桁で表現できる数の範囲は -50 ~ 49 まで)

$$30 - 24 \Rightarrow 30 + 76 = 106 \Rightarrow 6$$

$$22 - 24 \Rightarrow 22 + 76 = 98 \Rightarrow -2$$

$$99 - 24 \Rightarrow 99 + 76 = 175 \Rightarrow 75 \Rightarrow -25$$

$$50 - 24 \Rightarrow 50 + 76 = 126 \Rightarrow 26 \text{ [本来は -74]} \\ \text{(オーバーフロー発生)}$$

2進数での例

(注: 2の補数表現を用いて

4桁で表現できる数の範囲は -8 ~ 7 まで)

$$0110 - 0011 \Rightarrow 0110 + 1101 = 10011 \Rightarrow 0011 (3)$$

$$0010 - 0011 \Rightarrow 0010 + 1101 = 1111 (-1)$$

$$1000 - 0011 \Rightarrow 1000 + 1101 = 10101 \Rightarrow 0101 \text{ [本来は -11]} \\ \text{(オーバーフロー発生)}$$

3. 浮動小数点数

浮動小数点表現の基本概念

- 正規化表現 $[(-1)^s \times M \times 2^E]$ 形式

(科学記数法[有効桁数が重要])

- 整数部1桁(0以外の数値) + 小数部k桁 (以下はk=5の例)

$$1.00111 \times 2^4$$

$$-1.10100 \times 2^3$$

$$1.01011 \times 2^{-120}$$

32bit浮動小数点表現の例

