第2章「数の表現」

- 2進数，16進数，基数変換
- 補数（2の補数と1の補数）
- 負の整数の表現
- 文字の表現

2進法

・ 数を0と1のみで表現

10進	2進
0	0
1	1
2	10
3	11
4	100
5	101

10進	2進
6	110
7	111
8	1000
9	1001
10	1010
11	1011

10進	2進
12	1100
13	1101
14	1110
15	1111
16	10000
17	10001

例） 21 ＝ 10101, 129 ＝ 10000001, 63 ＝ 111111

ビットとバイト

- ビット＝2進一桁；bit (binary digit)
  - ビット数，ビット幅，最上位ビット，最下位ビット
- バイト＝2進8桁，つまり8ビット；byte
  - 16ビット＝2バイト，32ビット＝4バイト，64ビット＝8バイト

- ワード（語）と
  - 長さは計算機（MPU）に依存
  - 例：16ビットMPUなら2バイトが1ワード，32ビットなら4バイトが1ワード）

16進法

・ 0－9とA－Fまでの文字を数字とする

10進数	2進数	16進数
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A

10進数	2進数	16進数
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	10000	10
17	10001	11
18	10010	12
19	10011	13
20	10100	14
21	10101	15

r進法

- 2進法の‘2’，10進法の‘10’，16進の’16’のことを「基数」と呼ぶ
- 基数をrとする数の表記法 = r進法
  - r進n桁で表現できる整数の範囲： $0 \leq x \leq r^n - 1$

（整数の表記法）

$$a_{n-1} \cdots a_2 a_1 a_0 = \sum_{i}^{n-1} a_i r^i$$

例）2進 11101 = 10進 29

$$1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

小数

（小数の表記法）

$$\cdots a_3 a_2 a_1 a_0 . a_{-1} a_{-2} \cdots = \sum_i a_i r^i$$

例1）2進 11101.011 = 10進 $\boxed{29.375}$

$$1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3}$$

例2）16進 A.B = 10進 $\boxed{10.6875}$

$$\mathrm{A} + \mathrm{B} \cdot 16^{-1} = 10 + \frac{11}{16} = 10.6875$$

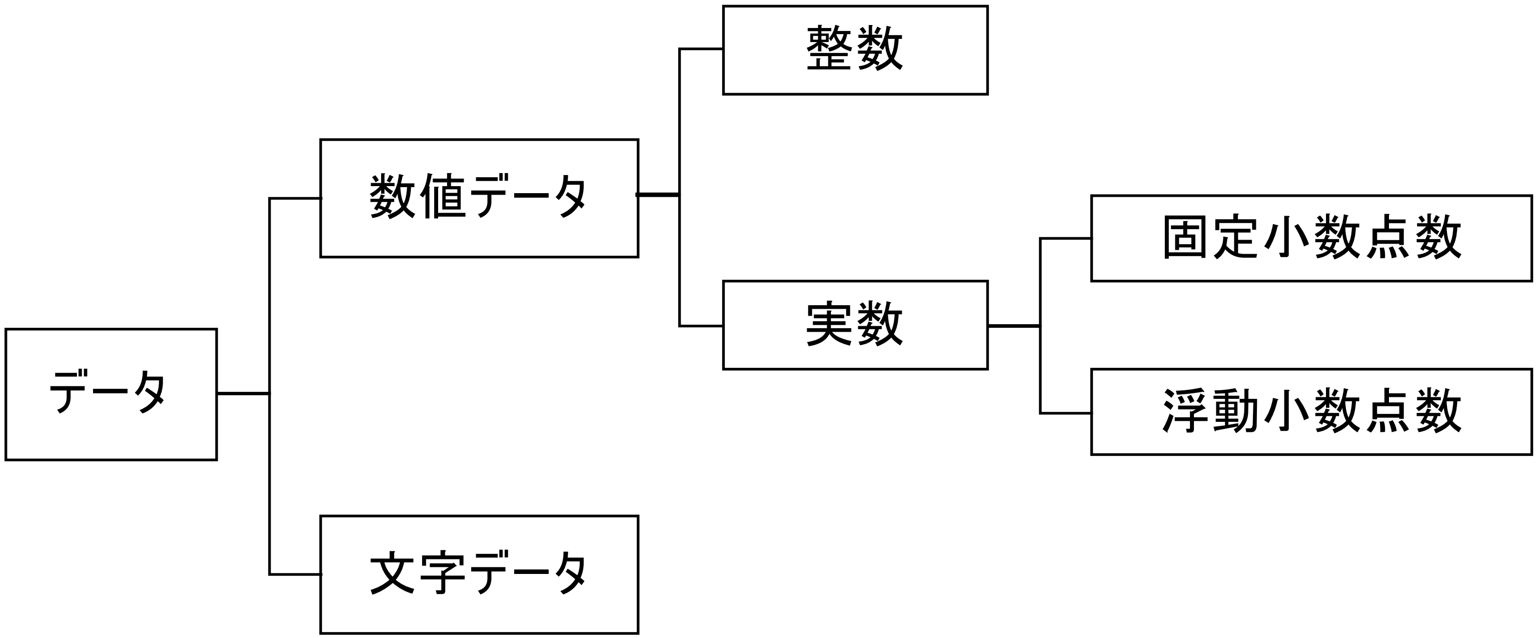
基数変換

・基数変換＝与えられた数の基数を入れ替えて表現を変換すること

2進	10進	16進
11001	25	19
0.1	0.5 (2^{-1})	0.8
0.01	0.25 (2^{-2})	0.4
10.0111	2.4375	2.7
10001100.1000111	140.5546875	8C.8E

計算機内部のデータ表現

- 計算機内部ではいかなる情報も0と1の2状態の集合で表現される
- 計算機で扱われるデータの種類



文字の表現

- アルファベットを基本，拡張する形で漢字が表現
- アルファベット：ASCIIコードで表現
  - 1バイト＝1文字，半角文字
- 漢字
  - 2バイト＝1文字
  - JIS，シフトJIS，EUC，Unicodeなど複数の規格がある
  - 半角カナはASCIIコードの拡張になっており扱いが別

例）「漢字」に相当するコード

JIS	1B 24 42 34 41 3B 7A 1B 28 42
シフトJIS	8A BF 8E 9A
EUC	B4 C1 BB FA

ASCIIコード

(American National Standard Code for Information Interchange)

ビット番号	b4	b3	b2	b1	行＼列								
b8						0	0	0	0	0	0	0	0
b7						0	0	0	0	1	1	1	1
b6						0	0	1	1	0	0	1	1
b5						0	1	0	1	0	1	0	1
						0	1	2	3	4	5	6	7
下位ビット	0	0	0	0	0	NUL	TC7(DLE)	SP	0	@	P	`	p
	0	0	0	1	1	TC1(SOH)	DC1	!	1	A	Q	a	q
	0	0	1	0	2	TC2(STX)	DC2	″	2	B	R	b	r
	0	0	1	1	3	TC3(ETX)	DC3	#	3	C	S	c	s
	0	1	0	0	4	TC4(EOT)	DC4	$	4	D	T	d	t
	0	1	0	1	5	TC5(ENQ)	TC8(NAK)	%	5	E	U	e	u
	0	1	1	0	6	TC6(ACK)	TC9(SYN)	&	6	F	V	f	v
	0	1	1	1	7	BEL	TC10(ETB)	'	7	G	W	g	w
	1	0	0	0	8	FE0(BS)	CAN	(	8	H	X	h	x
	1	0	0	1	9	FE1(HT)	EM	)	9	I	Y	i	y
	1	0	1	0	10	FE2(LF)	SUB	*	:	J	Z	j	z
	1	0	1	1	11	FE3(VT)	ESC	+	;	K	[	k	{
	1	1	0	0	12	FE4(FF)	IS4(FS)	,	<	L	¥	l	|
	1	1	0	1	13	FE5(CR)	IS3(GS)	−	=	M	]	m	}
	1	1	1	0	14	SO	IS2(RS)	.	>	N	^	n	‾
	1	1	1	1	15	SI	IS1(US)	/	?	O	_	o	DEL

パリティビットまたは0

例）“A” = 65, “a” = 97

整数の表現

- すべての整数は2進数で表現され格納される
- 1つの数を一定のビット数で表現
  - 8ビット，16ビット，32ビットなど
  - 表現できる数の範囲はこのビット数で決まる

  例）10進“91”→ 1011011

  - 正の整数しか扱わない場合はこれでよい
- 負の整数も扱う場合は特別な表現方法が必要に
  - 絶対値表現（最上位ビットで符号を表現；“−”を1に）
  - 補数による表現

ビッグエンディアン（big endian）
リトルエンディアン（little endian）

補数

- 「xに対する補数y」とは，ある決まったzに対して
  x+y = z となるような yを言う
  - zは，基数rと桁数nで決まる

- r進数の数に対し，「rの補数」と「r−1の補数」を使用
  - 例） 10進の数123の補数（10進，指定桁数3）

10の補数		9の補数	
	$\begin{array}{r} 1000 \\ -\ \ 123 \\ \hline 877 \end{array}$		$\begin{array}{r} 999 \\ -\ \ 123 \\ \hline 876 \end{array}$

2進数の補数

- 1の補数と2の補数がある
  - 例）桁数7の場合の, 1101101の「1の補数」と「2の補数」

1の補数

$$
\begin{array}{r}
1111111 \\
-\ 1101101 \\
\hline
0010010
\end{array}
$$

各桁の1と0を入れ替えたものと同じ

2の補数

$$
\begin{array}{r}
10000000 \\
-\ 1101101 \\
\hline
0010011
\end{array}
$$

1の補数に1を加えたものと同じ

2の補数による負の数の表現(1/2)

- 負の数xを，その絶対値|x|の2進表現の2の補数で表現する方法
- 例）8ビット（8桁）で数を表現する場合，−126を126＝01111110 の2の補数 10000010 で表現する
- 最上位ビットを見ると「符号」が分かる
- 負の数を表現するために当然正の数の表現可能な範囲が狭くなる
  - 上の例では，255⇔−1，254⇔−2，･･･，128=−128
- nビットで表現できる数の範囲：$-2^{n-1} \sim 2^{n-1}-1$

2の補数による負の整数の表現(2/2)

- 2の補数で負数を表現すると，減算を2進数の加算で処理できる利点がある

例）8ビットで表現する場合の100−90

100 = 01100100

−90 = 10100110　　（90 = 01011010）

```
   01100100    (100)
+  10100110    (−90)
-----------
  100001010
   00001010    (10)
```

最上位ビットを無視

付録：補助単位

単位記号	大きさ	近似
T（テラ）	10^{12}	2^{40}
G（ギガ）	10^{9}	2^{30}
M（メガ）	10^{6}	2^{20}
k（キロ）	10^{3}	2^{10}
m（ミリ）	10^{-3}	－
μ（マイクロ）	10^{-6}	－
n（ナノ）	10^{-9}	－
p（ピコ）	10^{-12}	－