

基礎コンピュータ工学

第2章 情報の表現

(パート4：小数や文字の表現)

<https://github.com/tctsigemura/TecTextBook>

本スライドの入手：



2進数の和差の計算（復習）

2進数の場合は以下のようになる。

- 1より大きくなる時に**桁上げ**が発生する。

010	001	010	011	011
+ 001	+ 001	+ 011	+ 001	+ 011
-----	-----	-----	-----	-----
011	010	101	100	110

- 桁借り**では2借りてくる。

011	010	101	100	110
- 001	- 001	- 011	- 001	- 011
-----	-----	-----	-----	-----
010	001	010	011	011

2進数の和差の計算（復習）

10進数の計算と2進数の計算をなさい。

3+8

10進

3

+ 8

2進

0011

+ 1000

5+7

10進

5

+ 7

2進

0101

+ 0111

11-8

10進

11

- 8

2進

1011

- 1000

12-7

10進

12

- 7

2進

1100

- 0111

負数の表現 (復習)

- 2の補数による負数の表現

2の補数 ($2^n - x$) を負数の表現に使用する.

4ビット2進数の2の補数 ($2^4 - x = y$)

もとの数 (x)	補数へ変換			補数 (y)
0	1	0000 ₂	− 0000 ₂	= 1 0000 ₂
1	1	0000 ₂	− 0001 ₂	= 1 1111 ₂
2	1	0000 ₂	− 0010 ₂	= 1 1110 ₂
3	1	0000 ₂	− 0011 ₂	= 1 1101 ₂
4	1	0000 ₂	− 0100 ₂	= 1 1100 ₂
5	1	0000 ₂	− 0101 ₂	= 1 1011 ₂
6	1	0000 ₂	− 0110 ₂	= 1 1010 ₂
7	1	0000 ₂	− 0111 ₂	= 1 1001 ₂
8	1	0000 ₂	− 1000 ₂	= 1 1000 ₂

負数の表現（復習）

- 2 の補数の求め方

ビット反転 + 1

$$x = +3_{10} = 0011_2 \text{ (もとの数)}$$

$$y = -3_{10} = 1100_2 + 1 = 1101_2 \text{ (2 の補数)}$$

元に戻すのもビット反転 + 1

$$y = -3_{10} = 1101_2 \text{ (2 の補数)}$$

$$y = +3_{10} = 0010_2 + 1 = 0011_2 \text{ (もとの数)}$$

- 表現できる数値の範囲

$$4 \text{ ビット} : -8 \sim +7 \text{ } (-2^3 \sim + (2^3 - 1))$$

$$n \text{ ビット} : -2^{n-1} \sim + (2^{n-1} - 1)$$

- 正負の判定

最上位ビットが

0 : 正の値を表現している.

1 : 負の値を表現している.

負の数を含む計算

2の補数表現の負数は符号無し2進数と同じ手順で計算できる！！

- 最上位ビットからの桁上げは無視する.

0010	(+2)	1011	(-5)	1101	(-3)			
+	1111	(-1)	+	0101	(+5)	+	1101	(-3)
<hr/>			<hr/>			<hr/>		
0001	(+1)	0000	(+0)	1010	(-6)			

- 仕組み

- **正の数と負の数の和** ($-b$ を2の補数 $(2^n - b)$ と表現する)
正の値 a と負の値 $-b$ の和を計算し 2^n (最上位の桁上げ) を無視する
$$a + (-b) = a + (2^n - b) = 2^n + a - b = a - b$$
- **負の数と負の数の和** ($-a, -b$ を2の補数で表現する)
 2^n (最上位からの桁上げ) を一つ無視すると
$$(-a) + (-b) = (2^n - a) + (2^n - b) = 2^n - (a + b)$$

負の数を含む計算

2の補数表現の負数は符号無し2進数と同じ手順で計算できる！！

- 最上位ビットの桁借りは**制限なし**とする.

0010	(+2)	0000	(+0)	1101	(-3)
- 1111	(-1)	- 0101	(+5)	- 1010	(-6)
<hr/>		<hr/>		<hr/>	
0011	(+3)	1011	(-5)	0011	(+3)

- 仕組み

- **正の数と負の数の差** ($-b$ を2の補数 $(2^n - b)$ と表現する)
正の値 a と負の値 $-b$ の差を計算し -2^n (最上位の桁借り) を許す
$$a - (-b) = a - (2^n - b) = -2^n + a + b = a + b$$
- **負の数と負の数の差** ($-a, -b$ を2の補数で表現する)
 2^n (最上位からの桁上げ) を一つ無視すると
$$(-a) - (-b) = (2^n - a) - (2^n - b) = (-a) + b$$

負数を含む計算（問題 1/2）

問題 1 1：次の計算を 2 進数と 10 進数でしなさい。
(ただし, 2 進数は 2 の補数表現形式になっている)

1)
$$\begin{array}{r} 0011\ 0010_2 \\ +\ 0011\ 0010_2 \\ \hline \boxed{}_2 \end{array} \rightarrow \begin{array}{r} \boxed{}_{10} \\ +\ \boxed{}_{10} \\ \hline \boxed{}_{10} \end{array}$$

2)
$$\begin{array}{r} 1111\ 1111_2 \\ +\ 1111\ 1111_2 \\ \hline \boxed{}_2 \end{array} \rightarrow \begin{array}{r} \boxed{}_{10} \\ +\ \boxed{}_{10} \\ \hline \boxed{}_{10} \end{array}$$

負数を含む計算（問題 2/2）

3)

$$\begin{array}{r} 0110\ 0100_2 \\ +\ 1001\ 1100_2 \\ \hline \end{array} \rightarrow \begin{array}{r} \boxed{}_{10} \\ +\ \boxed{}_{10} \\ \hline \boxed{}_{10} \end{array}$$

4)

$$\begin{array}{r} 1111\ 0000_2 \\ +\ 1110\ 1111_2 \\ \hline \end{array} \rightarrow \begin{array}{r} \boxed{}_{10} \\ +\ \boxed{}_{10} \\ \hline \boxed{}_{10} \end{array}$$

5)

$$\begin{array}{r} 0001\ 0000_2 \\ -\ 1110\ 1111_2 \\ \hline \end{array} \rightarrow \begin{array}{r} \boxed{}_{10} \\ -\ \boxed{}_{10} \\ \hline \boxed{}_{10} \end{array}$$

2進数による小数の表現

固定小数点方式（小数点の位置を約束する.）

$$00.00_2 = 0.0_{10}$$

$$00.01_2 = 0.25_{10}$$

$$00.10_2 = 0.5_{10}$$

$$00.11_2 = 0.75_{10}$$

$$01.00_2 = 1.0_{10}$$

$$01.01_2 = 1.25_{10}$$

$$01.10_2 = 1.5_{10}$$

$$01.11_2 = 1.75_{10}$$

$$10.00_2 = 2.0_{10}$$

...

$$11.11_2 = 3.75_{10}$$

桁の重み

- 小数点から左に進むと 2 倍

$$001.0000_2 = 1.0_{10}$$

$$010.0000_2 = 2.0_{10}$$

$$100.0000_2 = 4.0_{10}$$

- 小数点から右に進むと $1/2$ 倍

$$000.1000_2 = 0.5_{10}$$

$$000.0100_2 = 0.25_{10}$$

$$000.0010_2 = 0.125_{10}$$

$$000.0001_2 = 0.0625_{10}$$

固定小数点方式 2 進数 → 10 進数

桁の重みを合計する.

$$\begin{aligned} 10.01_2 &= 1 \times 2 + 0 \times 1 + 0 \times 1/2 + 1 \times 1/4 \\ &= 2 + 0 + 0 + 1/4 \\ &= 2 + 0.25 \\ &= 2.25 \end{aligned}$$

問題 1 2 : 2 進数を 10 進数に変換しなさい.

- 1) 0101.1010_2
- 2) 0011.0011_2
- 3) 0100.0101_2
- 4) 1010.1111_2

10進数 → 固定小数点方式2進数

2進数	×2は	10進数
0.101 ₂	左シフトと同じ	0.625
✓		×
1.010 ₂		<u>1.250</u>

2進数	×2は	10進数
0.010 ₂	左シフトと同じ	0.250
✓		×
0.100 ₂		<u>0.500</u>

2進数	×2は	10進数
0.100 ₂	左シフトと同じ	0.500
✓		×
1.000 ₂		<u>1.000</u>

10進数で計算したとき，小数点を横切って整数部に出てきた数を小数点の右に順番に並べると 0.101₂ になる．

固定小数点方式 2 進数 → 10 進数

問題 13 : 10 進数を 2 進数に変換しなさい.
但し 2 進数は, 小数点以下 4 桁, 全体で 6 桁とする.

1) 0.75_{10}

2) 0.5625_{10}

3) 2.5_{10}

4) 1.1875_{10}

文字の表現

ASCIIコード (*American Standard Code for Information Interchange*)
1963年にアメリカ規格協会 (ANSI) が定めた情報交換用の文字コード.

(上位3ビット)

	0	1	2	3	4	5	6	7
0	NUL	DLE	(SP)	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(8	H	X	h	x
9	HT	EM)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[k	{
C	FF	FS	,	<	L	\	l	
D	CR	GS	-	=	M]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

(下位4ビット)

文字の表現

JIS (Japan Industrial Standard : 日本工業規格) 8ビットコード

JIS 8ビットコードは、ASCIIコードに半角カタカナを追加したもの。記号、数字、英字の部分は、ほぼ、同じ並びになっている。

(上位4ビット)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	DLE		0	@	P	`	p				ー	タ	ミ		
1	SOH	DC1	!	1	A	Q	a	q				。	テ	ム		
2	STX	DC2	"	2	B	R	b	r				「	イ	ツ	メ	
3	ETX	DC3	#	3	C	S	c	s				」	ウ	テ	モ	
4	EOT	DC4	\$	4	D	T	d	t				、	エ	ト	ヤ	
5	ENQ	NAK	%	5	E	U	e	u				・	オ	ナ	コ	
6	ACK	SYN	&	6	F	V	f	v				ヲ	カ	ニ	ヨ	
7	BEL	ETB	'	7	G	W	g	w				ア	キ	ヌ	フ	
8	BS	CAN	(8	H	X	h	x				イ	ク	ネ	リ	
9	HT	EM)	9	I	Y	i	y				ウ	ケ	ノ	ル	
A	LF	SUB	*	:	J	Z	j	z				エ	コ	ハ	レ	
B	VT	ESC	+	;	K	[k	{				オ	サ	ビ	ロ	
C	FF	FS	,	<	L	¥	l					セ	シ	フ	ワ	
D	CR	GS	-	=	M]	m	}				ユ	ス	ヘ	シ	
E	SO	RS	.	>	N	^	n					ヨ	セ	ホ		
F	SI	US	/	?	O	_	o	DEL				ッ	ソ	マ		

(下位4ビット)

はASCIIコード表と異なる部分

補助単位

1,000m を 1km, 1,000g を 1kg, 0.001l を 1ml, 0.001m を 1mm
ここで, k や m は**補助単位**と呼ばれる.

一般的に			記憶容量		
値	記号	読み方	値	記号	読み方
10^3	<i>k</i>	キロ	2^{10}	<i>Ki</i>	キビ
10^6	<i>M</i>	メガ	2^{20}	<i>Mi</i>	メビ
10^9	<i>G</i>	ギガ	2^{30}	<i>Gi</i>	ギビ
10^{12}	<i>T</i>	テラ	2^{40}	<i>Ti</i>	テビ

- 通常は 10^3 毎に補助単位がある.
- コンピュータの記憶容量では 2^{10} 毎に補助単位がある.
 $2^{10} = 1,024 = 1Ki$
 $10^3 = 1,000 = 1k$