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進数の計算をしなさい．

3+8

10 進	２進
3	0011
+ 8	+ 1000

5+7

10 進	２進
5	0101
+ 7	+ 0111

11-8

10 進	２進
11	1011
- 8	- 1000

12-7

10 進	２進
12	1100
- 7	- 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	$1\boxed{0000}_2 - \boxed{0000}_2$	=	$1\boxed{0000}_2$
1	$1\boxed{0000}_2 - \boxed{0001}_2$	=	$\boxed{1111}_2$
2	$1\boxed{0000}_2 - \boxed{0010}_2$	=	$\boxed{1110}_2$
3	$1\boxed{0000}_2 - \boxed{0011}_2$	=	$\boxed{1101}_2$
4	$1\boxed{0000}_2 - \boxed{0100}_2$	=	$\boxed{1100}_2$
5	$1\boxed{0000}_2 - \boxed{0101}_2$	=	$\boxed{1011}_2$
6	$1\boxed{0000}_2 - \boxed{0110}_2$	=	$\boxed{1010}_2$
7	$1\boxed{0000}_2 - \boxed{0111}_2$	=	$\boxed{1001}_2$
8	$1\boxed{0000}_2 - \boxed{1000}_2$	=	$\boxed{1000}_2$

負数の表現（復習）

- ２の補数の求め方
  ビット反転＋１
  $x = +3_{10} = 0011_2$（もとの数）
  $y = -3_{10} = 1100_2 + 1 = 1101_2$（２の補数）
  元に戻すのもビット反転＋１
  $y = -3_{10} = 1101_2$（２の補数）
  $y = +3_{10} = 0010_2 + 1 = 0011_2$（もとの数）
- 表現できる数値の範囲
  ４ビット：$-8 \sim +7$ $(-2^3 \sim +(2^3 - 1))$
  n 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)
------          ------          ------
  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)
------            ------            ------
  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１１：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) $0011\ 0010_2 + 0011\ 0010_2 = \boxed{}_2$ → $\boxed{}_{10} + \boxed{}_{10} = \boxed{}_{10}$

2) $1111\ 1111_2 + 1111\ 1111_2 = \boxed{}_2$ → $\boxed{}_{10} + \boxed{}_{10} = \boxed{}_{10}$

負数を含む計算（問題 2/2）

3) $0110\ 0100_2 + 1001\ 1100_2 = \boxed{}_2$ → $\boxed{}_{10} + \boxed{}_{10} = \boxed{}_{10}$

4) $1111\ 0000_2 + 1110\ 1111_2 = \boxed{}_2$ → $\boxed{}_{10} + \boxed{}_{10} = \boxed{}_{10}$

5) $0001\ 0000_2 - 1110\ 1111_2 = \boxed{}_2$ → $\boxed{}_{10} - \boxed{}_{10} = \boxed{}_{10}$

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１２：2 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 は左シフトと同じ	10 進数
0.101_2 ↙ 1.010_2		$0.625 \times 2 = \underline{1}.250$
0.010_2 ↙ 0.100_2		$0.250 \times 2 = \underline{0}.500$
0.100_2 ↙ 1.000_2		$0.500 \times 2 = \underline{1}.000$

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 $0.\underline{101}_2$ 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）

（下位 4 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

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)

(下位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			。	ア	チ	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	ユ		
6	ACK	SYN	&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
7	BEL	ETB	’	7	G	W	g	w			ァ	キ	ヌ	ラ		
8	BS	CAN	(	8	H	X	h	x			ィ	ク	ネ	リ		
9	HT	EM	)	9	I	Y	i	y			ゥ	ケ	ノ	ル		
A	LF	SUB	∗	:	J	Z	j	z			ェ	コ	ハ	レ		
B	VT	ESC	+	;	K	[	k	{			ォ	サ	ヒ	ロ		
C	FF	FS	,	<	L	¥	l	|			ャ	シ	フ	ワ		
D	CR	GS	−	=	M	]	m	}			ュ	ス	ヘ	ン		
E	SO	RS	.	>	N	ˆ	n				ョ	セ	ホ	゛		
F	SI	US	／	?	O	_	o	DEL			ッ	ソ	マ	゜		

░ はASCIIコード表と異なる部分

補助単位

1,000m を 1km，1,000g を 1kg，0.001 l 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	*k*	キロ	2^{10}	*Ki*	キビ
10^6	*M*	メガ	2^{20}	*Mi*	メビ
10^9	*G*	ギガ	2^{30}	*Gi*	ギビ
10^{12}	*T*	テラ	2^{40}	*Ti*	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．
  $2^{10} = 1,024 = 1Ki$
  $10^3 = 1,000 = 1k$