プロセッサと機械語

2年次前期　（第2回）

中島 克人
情報メディア学科
nakajima@im.dendai.ac.jp

2進法（教科書 p.5）

h_2	h_1	h_0

$_2$ 3桁の2進数

$$= h_2 \times 2^2 + h_1 \times 2^1 + h_0 \times 2^0$$
$$= h_2 \times 4 + h_1 \times 2 + h_0 \times 1$$

h_i : 0, 1 の2種類の**数字**

例：

1	0	1

$$(101)_2 = (1 \times 4 + 0 \times 2 + 1 \times 1)_{10} = 5_{10}$$

16進法（教科書 p.6）

h_2	h_1	h_0

$_{16}$ 3桁の16進数

$$= h_2 \times 16^2 \times h_1 \times 16^1 \times h_0 \times 16^0$$
$$= h_2 \times 256 \times h_1 \times 16 \times h_0 \times 1$$

h_i : 0 ～ 9, A, B, C, D, E, F の16種類の**数字**

例：

1	3	5

$$(135)_{16} = (1 \times 256 + 3 \times 16 + 5 \times 1)_{10} = 309_{10}$$

0	A	E

$$(0AE)_{16} = (0 \times 256 + 10 \times 16 + 14 \times 1)_{10} = 174_{10}$$

2進法と16進法の関係

- 2進数を下位桁から4桁ずつ区切り，それぞれを16進数字（0～F）に置き換えることにより，16進数が得られる
- 逆に，16進数のそれぞれの桁を2進数に置き換えることにより，2進数が得られる

4n桁の2進数 ⇔ n桁の16進数

例：

A	E
(1010)	(1110)
16	1

$$(AE)_{16} = (10 \times 16 + 14 \times 1)_{10} = 174_{10}$$

1	0	1	0	1	1	1	0
128	64	32	16	8	4	2	1

$$(10101110)_2 = (1 \times 128 + 0 \times 64 + 1 \times 32 + 0 \times 16 + 1 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1)_{10} = 174_{10}$$
$$= (1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1)_{10} \times 16 + (1 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1)_{10} = 174_{10}$$

2進法の加減算 （教科書 1.3 2進法の加減算 p.7）

2進数1桁の加算	2進数1桁の減算
$0_2 + 0_2 = 0_2$	$0_2 - 0_2 = 0_2$
$0_2 + 1_2 = 1_2$	$0_2 - 1_2 = \dot{1}1_2$
$1_2 + 0_2 = 1_2$	$1_2 - 0_2 = 1_2$
$1_2 + 1_2 = 10_2$	$1_2 - 1_2 = 0_2$

$\dot{1}$：借り(borrow)

2進法の加減算 （教科書 p.7）

問題1.6 をやって見よう！

$101010_2 - 10011_2 =$

問題1.7 をやって見よう！

$A7B_{16} - 28D_{16} =$

負数の表現 （教科書 1.4 負数の表現 p.7）

- 補数表示
  - 桁数を限定し，マイナス符号を付けずに表現
  - 10進数3桁の例：

補数表現		通常の表記
499	←	499
…	←	…
1		1
0	←	0
999	←	−1
…	←	…
501		−499
500	←	−500

負数の表現法：符号と絶対値表現

- 減算例：$001_2 - 110_2 = -101_2$
- 負数を符号と絶対値で表現

内部表現（4ビット）

1	1	0	1
符号	絶対値		

- 減算方法：

$0001_2 - 0110_2 =$（左右の大小比較後，左右を入れ替えて）

$-(110_2 - 001)_2$

$= -101_2 = 1101_2$

- 上記の左右の大小比較や入れ替え無しにそのまま演算したい

⇒ 内部表現に **2の補数表現** を用いる

負数の表現法：2の補数表現

- 限られた桁数（ビット数）で正負両方の数を表現する
- 4ビットの例：

10進数	2進数
7	0111
6	0110
:	: : : :
1	0001
0	0000
−1	1111
−2	1110
:	: : : :
−7	1001
−8	1000

- 最上位ビットは符号を表すが，下位ビットにはどういう規則性か？

負数の表現法：2の補数表現

- 下記のように作られる表現を「2の補数表現」という

10進数	2進数
7	0111
3	0011
1	0001
0	0000
−1	1111
−3	1101
−7	1001
−8	1000

1の補数を作る
（ビット反転）

1000
1100
1110

1を足す

演習：次の8ビット2進数の2の補数を求めよ

(1) 01100100_2

(2) 00110010_2

演習：2の補数表現

10進数7の2の補数を6ビットで表現せよ

10進数	2進数
7	000111
−7	111001

1の補数を作る
（ビット反転）

111000

1を足す

10進数7の2の補数を8ビットで表現せよ

10進数	2進数
7	00000111
−7	11111001

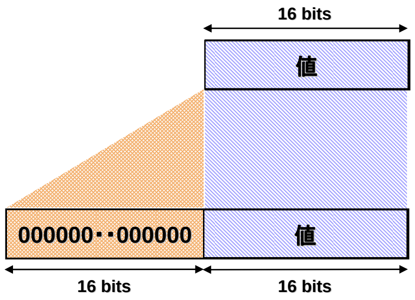
1の補数を作る
（ビット反転）

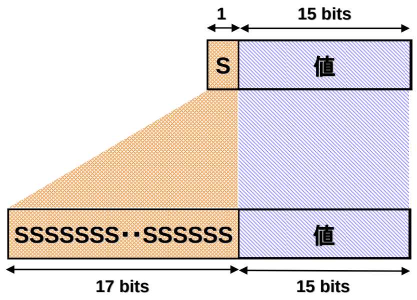
11111000

1を足す

ゼロ拡張と符号拡張

- ゼロ拡張：符号なし整数
- 符号拡張：符号付き整数





演習：次の8ビット2進数を12ビット表現にせよ

(1) 01100100_2

(2) 10011100_2

2進数の補数表現

- 長所

  加算の演算対象が正か負かを意識せずに，各桁で下記の1桁加算ルールを適用すれば正しい結果が得られる．

2進数1桁の加算

$0_2 + 0_2 = 0_2$

$0_2 + 1_2 = 1_2$

$1_2 + 0_2 = 1_2$

$1_2 + 1_2 = 10_2$

従って，減算A－Bは，加算A＋（Bの2の補数）に置き換えることができる．

→ 計算機は加算機能だけあれば良いことになる．

2進数の加算（8ビット整数の例）

- 正数＋正数

	00100010	34
+	00010111	23
	00111001	57

- 正数＋負数

	00010111	23
+	11011110	-34
	11110101	-11

2進数の減算 ⇨ 負数を加算

演習：次の加減算を8ビットの2進数で！

● 正数＋負数

42 +（－15）

00101010	42
+ 11110001	-15

● 正数－負数

53 －（－22）

00110101	53
+ 00010110	22

2の補数表現（特殊な値）

● 0000の2の補数は?

10進数	2進数		
0	0000	（ビット反転）→	1111
0	????	← 1を足す	

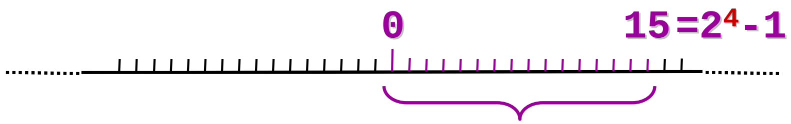
● 1000の2の補数は？

－8	1000	（ビット反転）→	0111
8	????	← 1を足す	

例外：対応する正数はない（overflow）

2進数の加減算

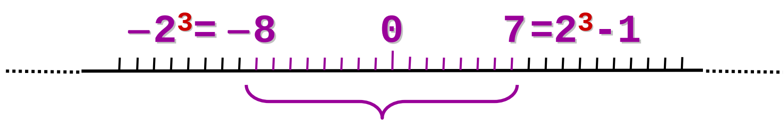
● 4ビット整数の表現範囲（符号なし整数）... 0～15



● 4ビット整数の表現範囲（正負整数）... －8～0～7

10進数	2進数
－1	1111
－2	1110
:	: : : :
－7	1001
－8	1000

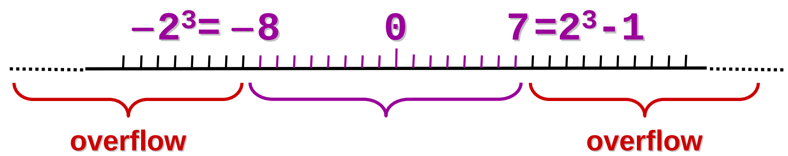
10進数	2進数
7	0111
6	0110
:	: : : :
1	0001
0	0000



2進数の加減算

● 4ビット整数（正負整数）での加減算

10進数	2進数
7	0111
6	0110
:	: : : :
1	0001
0	0000
－1	1111
－2	1110
:	: : : :
－7	1001
－8	1000

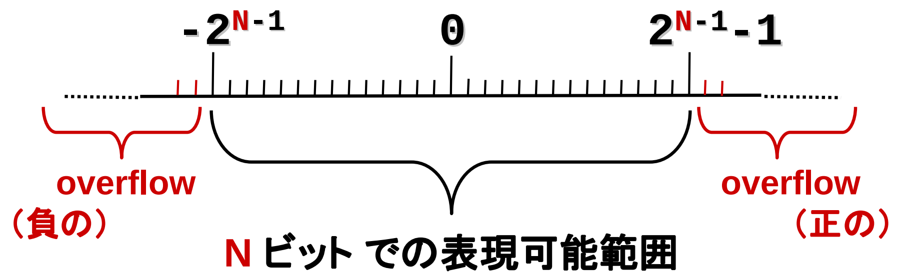


a+b>7 となるような演算をしたら，結果は4ビットで正しく表現できない！ . . . overflow と称する
（オーバーフロー）

－a－b<－8 となるような演算をしたら，結果は4ビットで正しく表現できない！ . . . overflow と称する

2進数の加減算

- Nビット整数の表現範囲 -2^{N-1}～0～$2^{N-1}-1$



宿題：2進数の負数表現，および，加減算を行え

- 授業HP「第2回宿題」のPDFを各自印刷し，次回授業の冒頭に副手に提出すること