コンピュータ基礎
数、データの表現

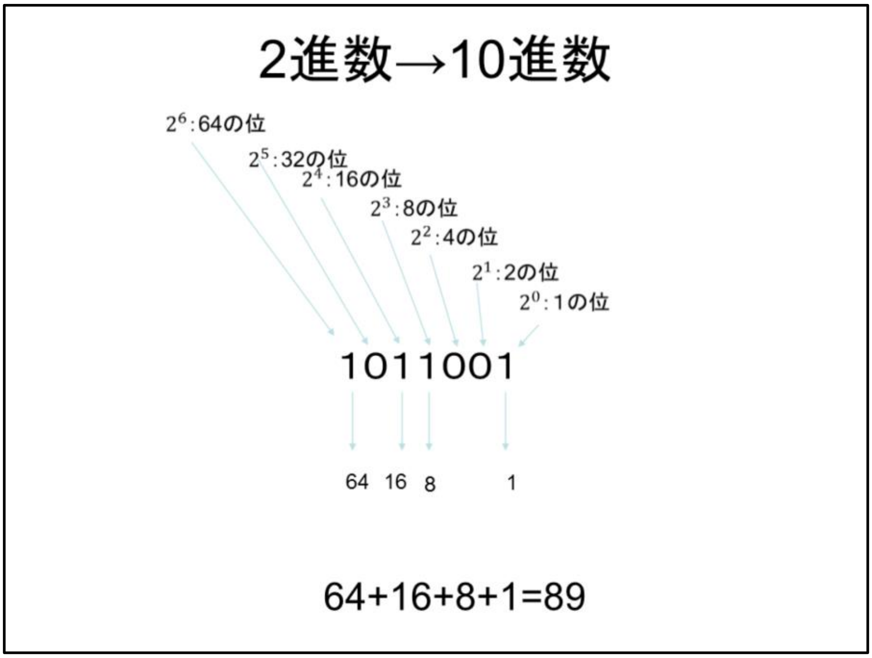
天野
hunga@am.ics.keio.ac.jp

すいません。今日はCoolChipsのパネルディスカッションのため早く終わります。

数の表現

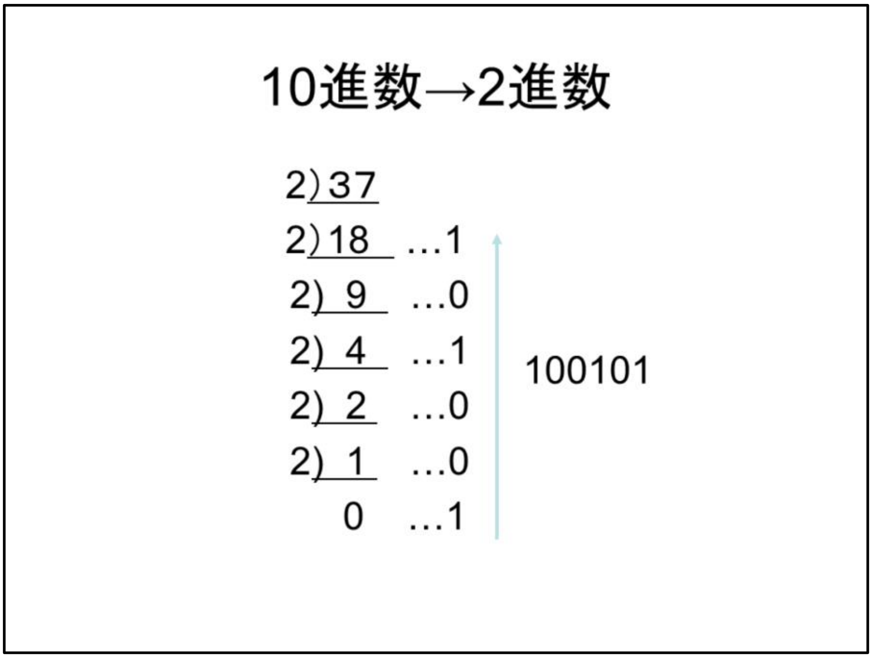
- コンピュータの内部では2進数で表現
- 1・0パターンに意味を持たせるのは使う方（プログラマ）の役割
  - 数
    - 符号なし数
    - 符号付数（2の補数表現）
    - 固定小数
    - 浮動小数
  - 機械語命令
  - 文字コード
    - ASCIIコード
    - JISコード、シフトJISコード
    - Unicode

ご存知と思いますが、コンピュータの内部では、数は1，0から成る2進数で表現します。ここで注意しておきたいのは、1，0のパターンに意味を持たせるのは使う側だということです。コンピュータ内部には数を識別する機能はないので、解釈の仕様によって、様々な表現の数、機械語命令、文字コードなど様々な種類のデータを表します。



最も簡単なのは、符号なし数です。これは中学で習った2進数そのものです。2進数は、各桁は１か０で、２の0乗の位、2の１乗の位、、、、と順番に桁の重みを表します。

この例では2の０乗の位＝１、２の３乗の位＝８、２の４乗の位＝１６、２の６乗の位＝６４が1なので、全てを足すと８９になります。



逆に変換する場合、2で割っていき、余りを記録して行きます。この場合2で6回割ったとき、3回割ったとき、1回目に割ったときが1なので、100101になります。

2進数と16進数

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

注意！ 16進表現は2進数の短縮表現としてのみ考えるべし！
マイナスの16進数とか考えない

２進数は、ちょっとした大きさの数を表すときも、桁数が長くなってしまいます。そこで、これを４桁ずつ区切って読みます。０－９までは普通の１０進数と同じですが、1010は10ではなくてAを使います。以降、B,C,D,E,Fと使います。すなわち16進数は一桁を０－Fで表すのです。ここで注意です。16進数はあくまで2進数の短縮表現として考えましょう。マイナスの16進数というのは定義可能ですが、あまり使いませんし、頭が混乱してしまいます。

2の補数表現

2の補数	10進数
1000	−8
1001	−7
1010	−6
1011	−5
1100	−4
1101	−3
1110	−2
1111	−1

2の補数	10進数
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7

さて、次にマイナスの数を含む符号付き数を考えます。現在のコンピュータでは符号付き数として、2の補数表現を使います。2の補数とは、元の数と足した時に、桁あふれを除いて全て0になる数のことです。例えば、0001（1）の、２の補数は、1111です。1111＋0001＝10000になるからです。0011（3）の、２の補数は、1101です。0011＋1101＝10000になるからです。

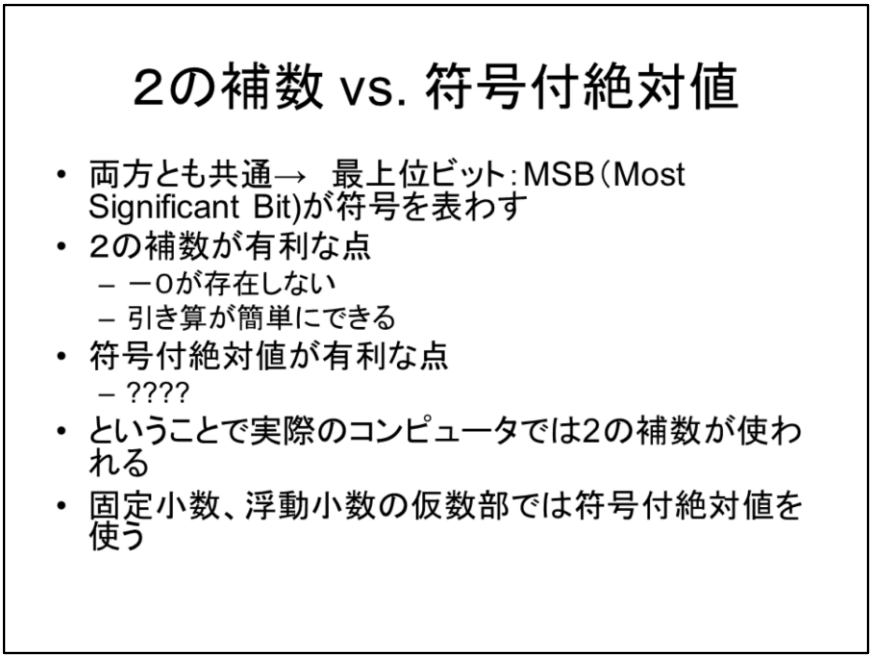
2の補数の作りかたは簡単です。まず1と0を反転します。これで足して全ての桁が1になる1の補数ができます。これに1を足すと、桁溢れを除いて全ての桁が0になる2の補数になります。２の補数を使うと引き算が簡単にできます。引く数の２の補数を足してやれば良いのです。この例では８－５を計算します。５の「２の補数」は0101→1010→1011です。これを1000（８）に足して桁あふれを無視すれば、0011（３）を求めることができます。

符号＋絶対値表現

絶対値	10進数
1111	−7
1110	−6
1101	−5
1100	−4
1011	−3
1010	−2
1001	−1
1000	−0

絶対値	10進数
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7

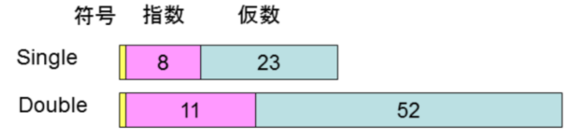
2の補数表現を符号＋絶対値表現と勘違いしている人も居ます。符号＋絶対値表現も浮動小数点表記の一部に使われることもあるのですが、普段使うのは圧倒的に2の補数表現です。なぜでしょう？これには次の理由があります。



両者とも、最上位ビット（MSB:Most Significant Bit)が、符号を表し、これが1の場合、負の数になります。しかし、２の補数は一0が存在しない、引き算が簡単にできるという利点があるのに対して、符号付き絶対値表現は特に有利な点が見当たりません。というわけで、実際のコンピュータでは２の補数が使われますが、例えば浮動小数の仮数の表現には符号付き絶対値を使います。

浮動小数点数

– (仮数) × 2 （指数）
– 倍精度 64bit, 単精度 32bit.
– IEEE Standardでフォーマットと丸めの方法を決めている



浮動小数点数は、表現できる範囲を広げるため、仮数×２の指数乗で表します。これは10進数の指数表現と同じです。この表現は、コンピュータ毎に別なものを使うとデータの交換ができなくなるため、IEEE Standardという国際標準に基づきます。単精度（C言語だとfloat)は32ビット、倍精度（C言語だとdouble)は64ビットで表します。このスライドで示すように指数部と仮数部を分割しています。浮動小数点表記では符号を独立させているため、仮数部は絶対値表現を使いますが、一番上の桁の1を省略します。また指数部は下駄履き(バイアス）表現を使います。また、浮動小数は、答えがぴったり仮数部の桁に収まらない場合にどのようにするか（丸め）という問題があって結構面倒です。ここでは深く突っ込まないことにします。

文字コード

ASCIIコードの一部、左の16進数で文字を表す

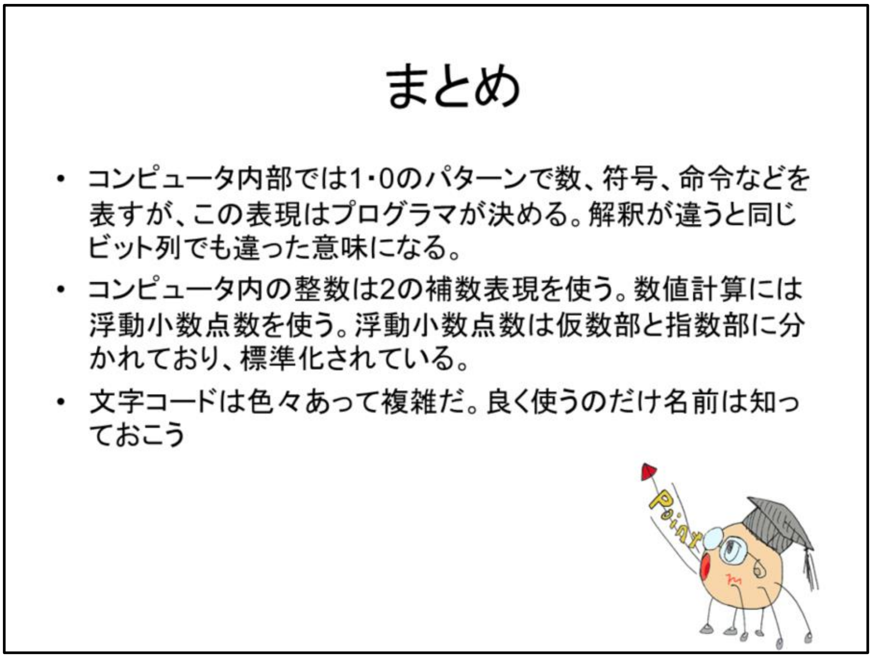
30	0	38	8	40	@	48	H	50	P	58	X
31	1	39	9	41	A	49	I	51	Q	59	Y
32	2	3A	:	42	B	4A	J	52	R	5A	Z
33	3	3B	;	43	C	4B	K	53	S	5B	[
34	4	3C	<	44	D	4C	L	54	T	5C	¥
35	5	3D	=	45	E	4D	M	55	U	5D	]
36	6	3E	>	46	F	4E	N	56	V	5E	^
37	7	3F	?	47	G	4F	O	57	W	5F	_

文字コードはコンピュータ上で文字を表すコードです。昔から使われているコードがASCIIコードで7ビット（8ビット）を使って英数、特殊文字を表現します。ここではその一部を示します。

その他の文字コード

- Unicode：様々な言語の文字コードを表現するために拡張されたコード。結構複雑
- JIS-コード（ISO 2022JP）：ASCIIコードを含む日本語用の文字コード、カタカナ、ひらがな、漢字を表現できる
- Shift-JISコード：JISコードをさらに拡張した日本語用コード、Windowsなどで標準的に用いられる

ASCIIコードでは日本語のかなや漢字が表現できないため、様々なコード化手法があります。どのような集合の記号に対して符号化するか、どのように符号化するかによって方式の違いが出ます。JISとShift-JISは対象となる記号は同じで、コード化が違います。Unicodeは様々な言語に対応するため、JISやShift-JISとは対象とする記号の集合が違います。さらには、符号化方式によってutf-8とかutf-16とかあります。



ではインフォ丸によるまとめです。今日は内容が少なくてすいません。では小テストをやってくださいませ。