第1章「序論」

- 計算機の歴史
- 計算機を「理解する」とは？
- 配布資料ダウンロード
  - http://www.vision.is.tohoku.ac.jp/jp/course/2018

計算機とは？

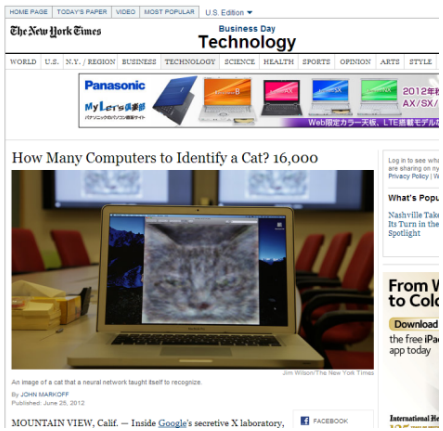
- 計算機＝あらかじめ与えた指令の列を，順番にこなしてゆく機械
  - 指令の列＝プログラム
- 講義の主な対象
  - 演算処理装置，CPU (Central Processing Unit)，MPU (Micro Processing Unit)
- 例）
  - 「計算機」然としたもの：PC，大型計算機，並列計算機，スーパーコンピュータ
  - 身近な計算機：携帯電話，（携帯）ゲーム機
  - 機械の制御：自動車（例：エンジン），航空機…

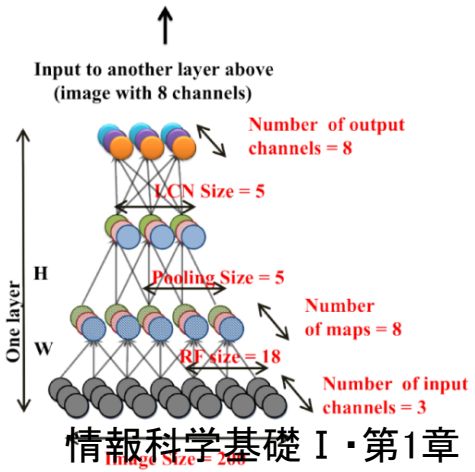
プログラム

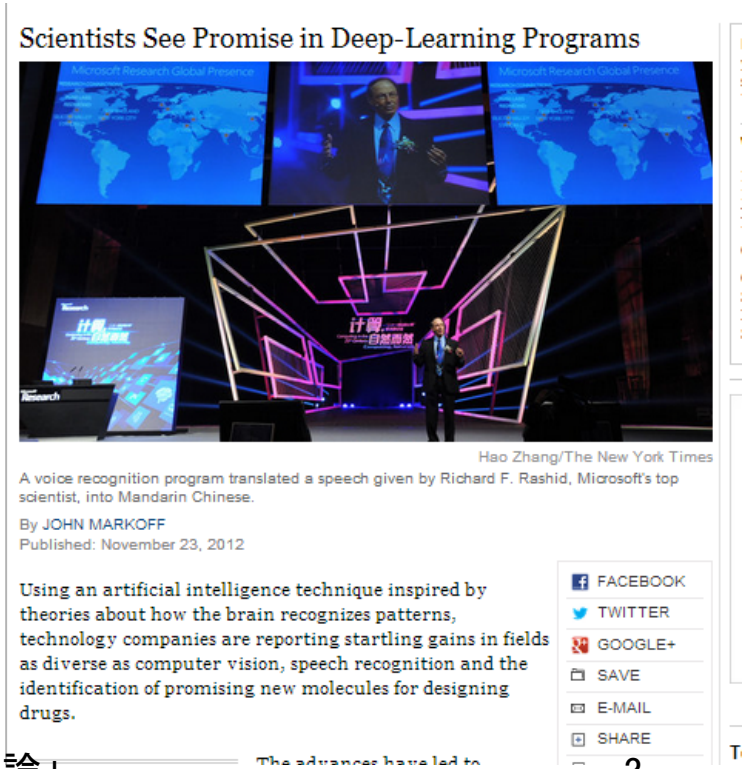
- プログラムは用途・目的に応じて作成
  - スマートフォン
  - 自動車のエンジン
- 計算機と生物の脳の違い
  - 脳＝超並列，パタンマッチングが得意，論理計算苦手
  - 計算機＝直列，論理計算が得意，パタンマッチングは苦手．だが，どんどん接近











計算機を「理解する」

- 計算機は膨大な技術の集積
- 独立した層状の技術
  - 0層：アプリケーションソフトウェア
    - ワープロソフトなど
  - 1層：プログラミング
    - ソフトウェアを作る
  - 2層：オペレーティングシステム(Operating System; OS)
    - Microsoft Windows, UNIX, …
  - 3層：アーキテクチャ
  - 4層：論理回路
  - 5層：回路素子・デバイス（主に半導体）

ソフトウェア

ハードウェア

計算機の歴史

- 電子計算機以前
  - リレー式計算機；素子にリレーを使用；1960年頃まで
  - MARKI（ハーバード大・IBM;1944）など
- 電子計算機の登場
- 第１世代：素子が真空管
  - ENIAC (1946), EDSAC (1949)
  - UNIVAC I (1951)，最初の商用大型計算機
- 第２世代：トランジスタが素子に使われ始めた
- 第３世代：集積回路の時代
  - IBM System/360（1964）
- 第4世代：大規模集積回路；マイクロコンピュータの時代
  - 用途がいわゆる「計算機」だけではなくなる；ロケットから携帯電話まで



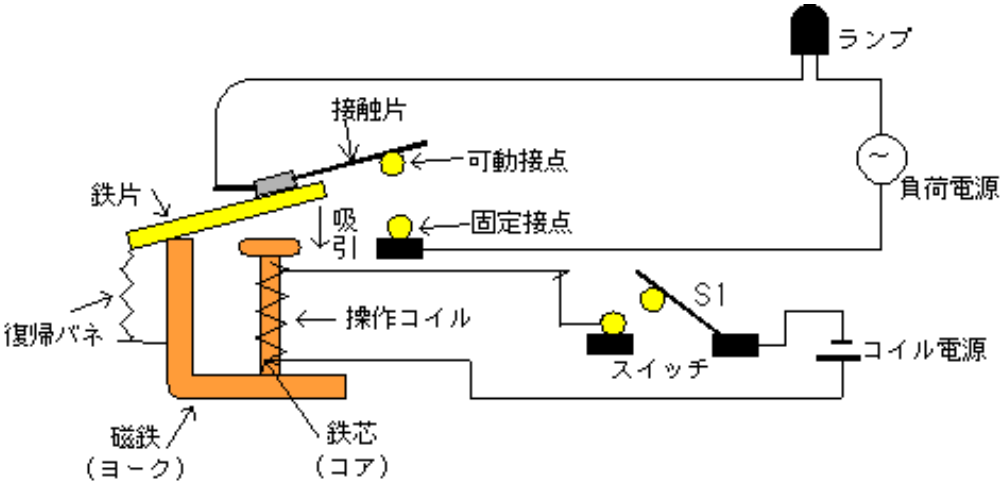
機械式（手回し）計算機（～1960代末）



リレー計算機(FACOM138A)



ENIAC



リレーの原理

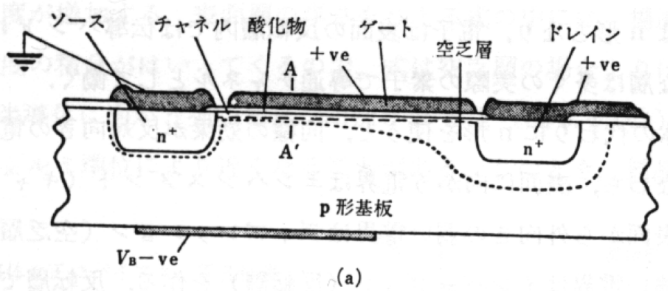
http://www.omron.co.jp/ecb/products/pry/ry_tech/b.html

素子の変遷

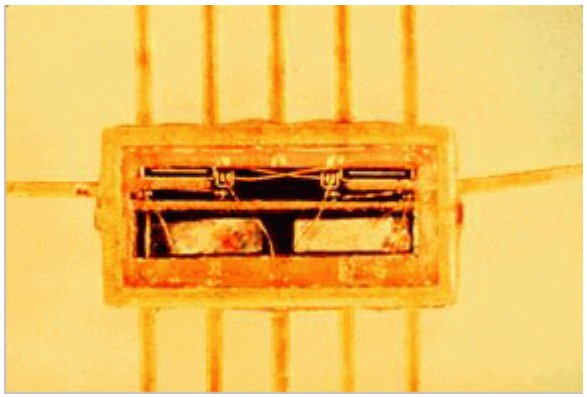
素子＝計算機を構成する論理的な最小単位



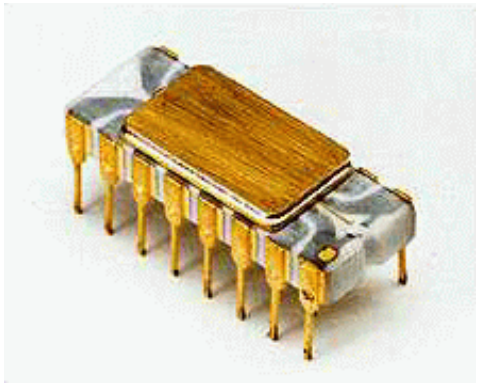
真空管

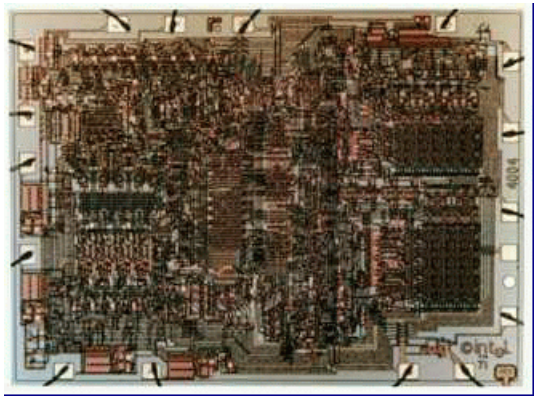


トランジスタ



初の集積回路
（Texas Instruments;1958）





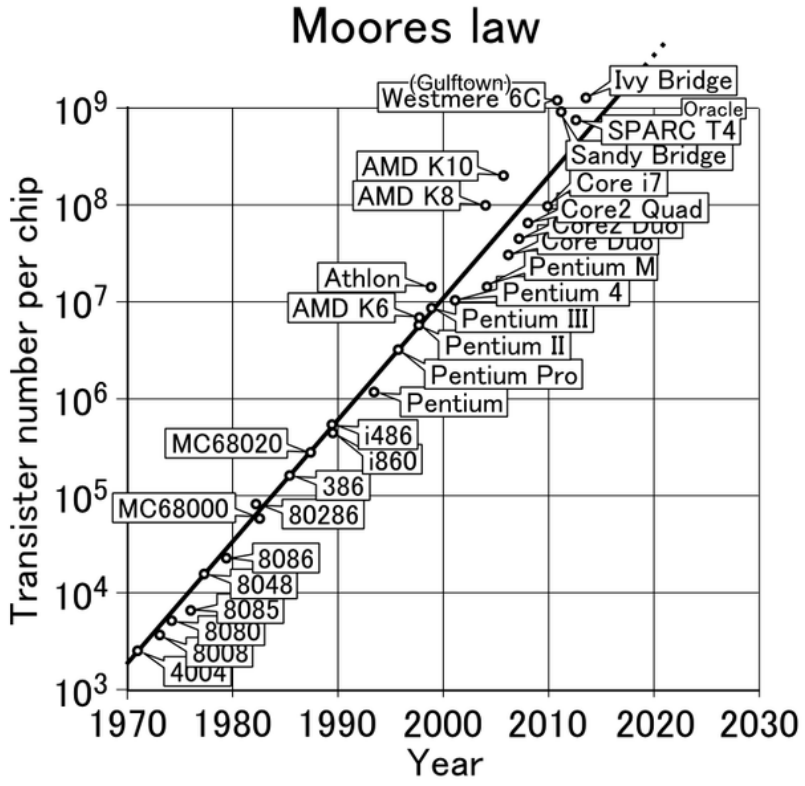
初のMPU
（Intel 4004；1971）

MPUの発展

- より高機能に，より高速に
  - 集積回路の集積度は2年で2倍（ムーアの法則）

表．Intel製PC用CPUの発展

名称	年	ビット数(バス幅)	製造プロセス（μm）	トランジスタ数	動作周波数	MIPS
4004	1971	4	10	2300	740kHz	0.06
8008	1972	8	10	3500	500kHz	0.05
8086	1978	16	3	3万	5–10MHz	0.33–0.75
80386	1985	32	1	27.5万	16M–33M	6–11.4
Pentium	1993	64	0.6–0.8	310万	60M–200M	100–
Pentium 3	1999	–	0.25	950万	450M–600M	–
Pentium 4	2000	–	0.13–0.18	4200万	1.4G–2.4G	–10000
Xeon(の1つ)	2006	–	0.065	2億9,100万	2–3G	–



(MIPS=Million Instructions Per Second)

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	
9	
10	
11	

10進	2進
12	
13	
14	
15	
16	
17	

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 [　　]

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

例2）16進 A.B = 10進 [　　]

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

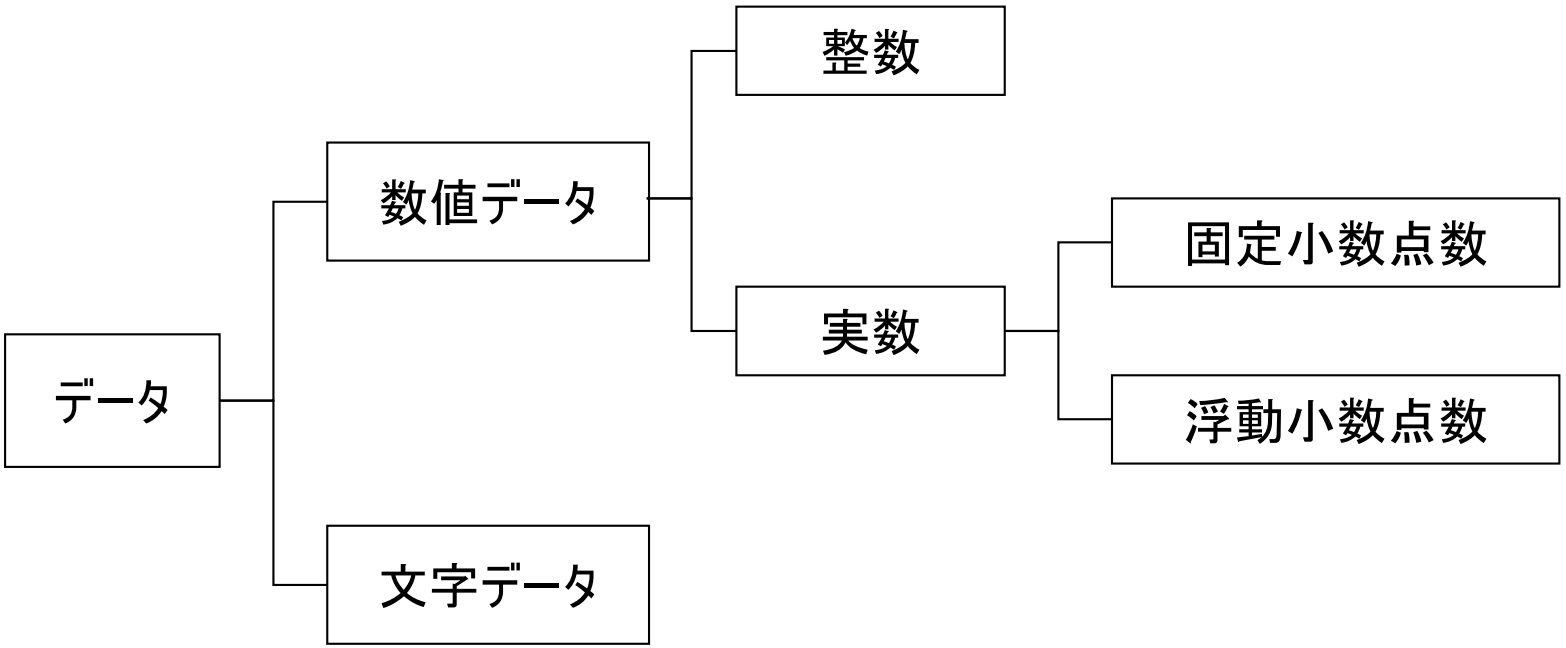
基数変換

・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
		0.4
	2.4375	2.7
10001100.1000111	140.5546875	

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

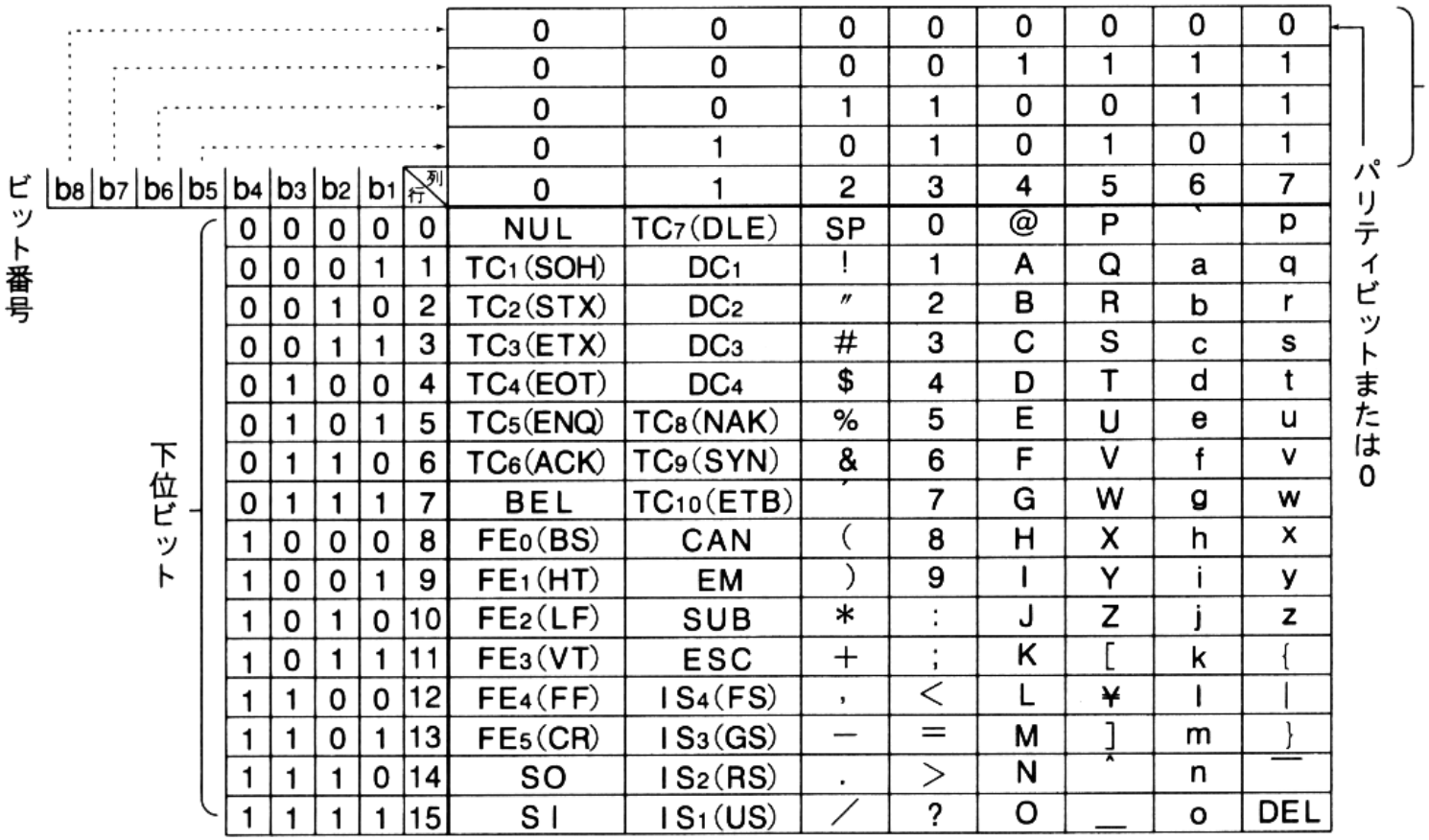
- アルファベットを基本，拡張する形で漢字が表現
- アルファベット：ASCIIコードで表現
  - 1バイト＝1文字，半角文字
- 漢字
  - 2バイト＝1文字
  - JIS，シフトJIS，EUC，Unicodeなど複数の規格がある
  - 半角カナはASCIIコードの拡張になっており扱いが別

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 b8	b7	b6	b5	列 / 行	0	0	0	0	0	0	0	0
					0	0	0	0	1	1	1	1
					0	0	1	1	0	0	1	1
					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

$$\begin{array}{rr} & 1000 \\ - & 123 \\ \hline & 877 \end{array}$$

9の補数

$$\begin{array}{rr} & 999 \\ - & 123 \\ \hline & 876 \end{array}$$

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

```
  1111111
- 1101101
---------
```



各桁の1と0を入れ替えたものと同じ

2の補数

```
  10000000
- 1101101
---------
```

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} ～$2^{n-1}-1$

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 = (90 = 01011010)

```
   01100100    (100)
+              (−90)
------------
  100001010
```

最上位ビットを無視 → 00001010 (10)

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}	–