情報基礎

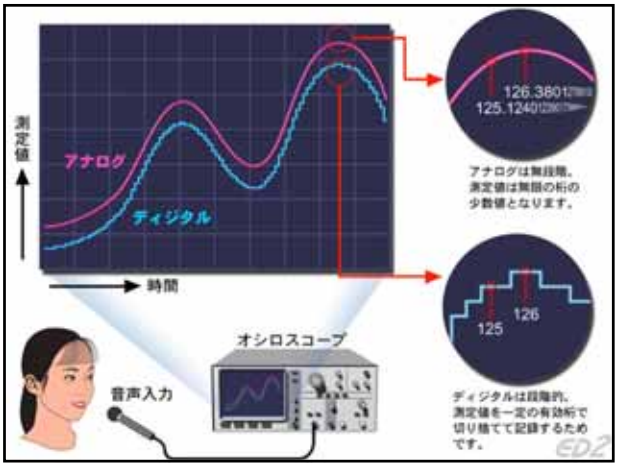
情報の符号化 (1)
2進数・16進数

今日やること

- コンピュータでは全ての情報は2進数で扱う
- 数値の表現方法
  - 2進数・16進数
  - 符号付きの整数
  - 小数点のついた数，非常に大きい（小さい）数

アナログとデジタル

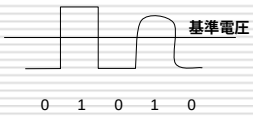
- アナログ (analog): 情報をコップに注いだ水の量や電圧などの物理量で表現
  - **連続量**
- デジタル (digital): 情報を**数値化**して表現
  - **離散量**
  - 自然界の量 を数値で厳密に表現しようとすれば、無限の桁数が必要
  - 「少数点以下3桁で四捨五入」あるいは「切り捨て」といったような丸めが行われる
- コンピュータでは全ての情報をデジタル化して扱う
  - 内部では2進数を用いる



2進数・16進数

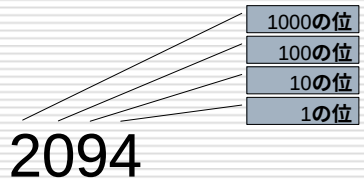
2進数

- 各桁が 0 or 1
- 0と1を電圧の高低で表せる
  - 0V = 0, 5V = 1 など
- 電気回路で扱いやすい
- ノイズに強い



10進数(復習)

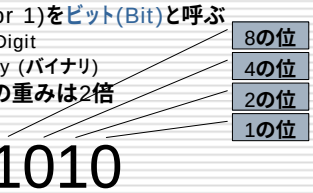
- 各桁は 0〜9 までの10通り
- 一桁上がると，桁の重みは10倍



2094 = 2*1000 + 0*100 + 9*10 + 4*1

2進数

- 各桁は 0 と 1 の2通り
- 2進数の1桁(0 or 1)をビット(Bit)と呼ぶ
  - Bit = Binary Digit
  - 2進数 = Binary (バイナリ)
- 一桁上がると桁の重みは2倍



1010 = 1*8 + 0*4 + 1*2+0*1
= 10

演習

- 以下の2進数を10進数に変換せよ
  - 1000 =
  - 11001 =

2進数の加算

1101

+) 101

10進数と2進数の対応

埋めてみよう

10進数	2進数	10進数	2進数	10進数	2進数
0	00000	6	00110	12	
1	00001	7	00111	13	
2	00010	8	01000	14	
3	00011	9	01001	15	
4	00100	10	01010	16	
5	00101	11	01011	17	

基数の表記法

- 10進法以外の記法で数を表すと混乱する
- 基数(n進法のnのこと)を右下に小さく書く

$1010_{(10)}$ 10進法の1010

$1010_{(2)}$ 2進法の1010 (10進法の10)

- 読むときはイチゼロイチゼロと読む
  - センジュウと読んではいけない!

10進2進変換

- $11_{(10)}$を2進数に変換してみよう

2) 11… 1 ←最下位桁

2) 5…

2) 2…

←最上位桁

10進2進変換

- 11 = 8 + 2 + 1 と分解してもよい
  - 2の累乗の和で表す
  - $8 + 2 + 1 = 2^3+2^1+2^0 = 1011_{(2)}$

演習

- 以下の10進数を2進数に変換せよ
  - 30 =
  - 63 =

nビットで表せる数

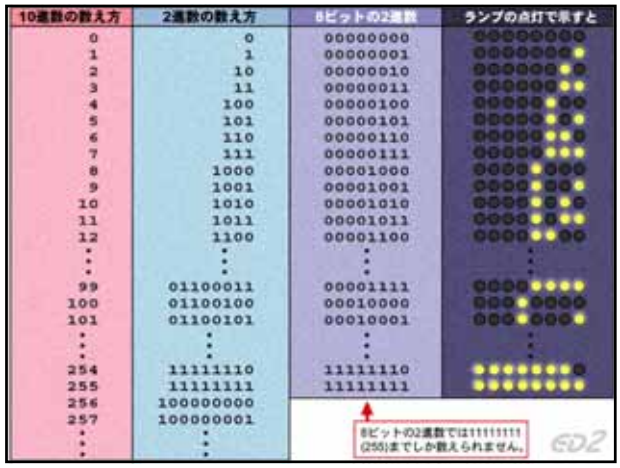
- 10進数n桁の組み合わせは10^n通り
  - 2桁だと00～99の100通り
- 2進数n桁(nビット)の組み合わせは2^n通り
  - 2桁だと 00, 01, 10, 11 の4通り
- nビットで $0 \sim 2^n-1$ まで表せる

nビットで表せる数

- nビットで 0 ～ 2^n-1 まで表せる

ビット数	表せる最大の数
2ビット	3
4ビット	15
8ビット	255
10ビット	1,023
16ビット	65,535
32ビット	(約43億) 4,294,967,295

- 最近のコンピュータは一度に32ビットの演算ができるものが主流
  - 32ビットCPU

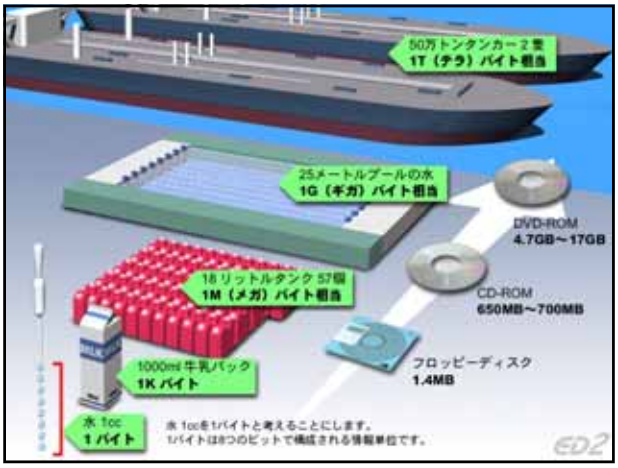


ビットとバイト

- 1ビットという単位では細かすぎて使いにくい
- 8ビットをまとめて1バイト (Byte) として使う
  - 1バイト = 8ビット
- 1バイトで 256通り表せる
  - アルファベット(26×2=52文字)＋数字(10文字)＋各種記号を表すのに十分なサイズ
- 1bit = 1b, 1byte = 1B と書くこともある
  - 1B = 8b
- コンピュータではデータをバイト単位で扱う場合が多い

バイトとキロバイト

- SI単位系では…
  - 1k (キロ) = 1000 = 10^3　　1M (メガ) = 1000k = 10^6
  - 1G (ギガ) = 1000M = 10^9　　1T (テラ) = 1000G = 10^{12}
- コンピュータの世界では1KB(キロバイト) = 1024バイト
  - 1024は2進数できりがよい (1024 = 2^{10})
  - $1024_{(10)} = 10000000000_{(2)}$
  - 1KB = 2^{10}B = 1024 B
  - 1MB = 2^{20}B = 1024 KB = 1,048,576 B
  - 1GB = 2^{30} B = 1024 MB = 1,073,741,824 B
- 1k (キロ) = 1000, 1K (ケー) = 1024 として区別することもある



kilo = 1000 or 1024?

- モノによって“キロ”が何を意味するか異なる
  - メモリの容量　　kilo = 1024
    - メモリ128MBのパソコン
  - ファイルのサイズ　kilo = 1024
    - 100KBのファイル
  - ハードディスクの容量　　kilo = 1(
    - 40GBのハードディスク
  - 通信速度　　kilo = 1000
    - ADSL 8M bps (1秒間に $8*10^6$ビットの通信速度)

演習

- 次のものを表現するために必要な最小ビット数を求めよ
  - 性別(男･女)
  - 信号の状態(赤･黄･青)
  - 試験の成績(0～100)
  - 640x480ピクセルの白黒ディスプレイ
    - 各ピクセル 1bit
  - 640x480ピクセルのフルカラーディスプレイ
    - 各ピクセル 24bit (Red, Green, Blue各8bit)

16進数(1)

- 2進数では桁数が多すぎて表記上不便
  - $50000_{(10)}$ = $1100001101010000_{(2)}$
  - 16進数で表記するのが一般的
- 2進数を4桁ごとに区切り，それぞれを16進数1桁で表す
  - 1100|0011|0101|0000
  - 16進数 = Hexadecimal
- 4ビットで 0～15 までの値を表現できる
  - 10～15 も1桁で表さないといけない
  - A～F を使う
- 表記法はいろいろ
  - $12AB_{(16)}$　0x12AB　$12AB　12ABH

16進数(2)

10進	2進	16進
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7

10進	2進	16進
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

16進-10進変換表

HI	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

16進数(3)

1100|0011|0101|0000

C　3　5　0

- 1桁上がるごとに桁の重みは16倍
  - 1の位, 16の位, 256の位, 4096の位, 65536の位, ...

$C350_{(16)} = 12*4096 + 3*256 + 5*16+0*1$

$= 50000$

演習

- $30_{(16)}$を10進数に変換せよ
- $ABCD_{(16)}$を10進数に変換せよ

16進数の加算

3A93
+) 290F

16進数の減算

3A93
-) 290F

演習

- 次の計算をせよ．結果は16進数で示せ．
  - 0xBAD + 0xBED =
  - 0xDEAD - 0xBEEF =

符号付きの整数の表現

符号付きの整数の表現

- いままでの数値はすべて<符号なし>(0以上)
  - 符号付の数を扱うには?
- いくつかの方式
  - 符号ビットとして1ビット使う方法
    - 符号ビット(0:正, 1:負) + 数値
  - 1の補数を用いる方法
    - 全ビットを反転する
  - ⇒正の0と負の0ができてしまう
- 現在では, 通常 2の補数表現 が使われる
  - 演算回路が大幅に簡略可能

まずは10進数で考える

- 2桁の10進数(0〜99)を使って符号付の数を表す(100の補数)
  - 100の補数: 足したら100になる数
  - 負の数 -x を 100 - x で表現する
- 約束
  - 0〜49は0〜49を表し,
  - 50〜99は-50〜-1を表すことにすると,
  - 2桁の10進数で-50〜49までを表せる

	0	----------	49	50	----------	99
表す数	0	----------	49	-50	----------	-1

- 100の補数での計算例
  - 1 + (-2) = 1 + 98 = 99 (= -1)
  - -1 + (-2) = 99 + 98 = 197　下2桁(97) = -3
- 正の数と負の数を区別せずに扱える
  - <符号なしの数>用の回路で符号付の数を扱える(回路が簡単)

2の補数

- 2の補数:負の数 -x は 2^n - x で表現する(nは使用するビット数)
- 8ビットの2の補数
  - 2^8=256

	00000000		01111111 10000000		1111111
	0		127 128		255
表す数	0		127 -128		-1

- 16ビットの2の補数
  - 2^{16}=65536

	0	32767 32768		65535
表す数	0	32767	-32768	-1

2の補数

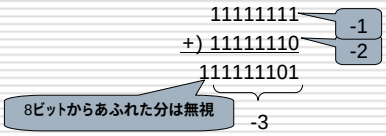
- 2の補数では正負の反転は全ビットを反転(0と1を入れ替え)して+1すればよい
- -12を8ビットの2の補数で表してみる
  - $12_{(10)}$ = $00001100_{(2)}$
  - 全ビットを反転すると
  - +1 すると (-12の2の補数表現)
  - 逆に ↑ を全ビット反転して+1すると00001100に戻る
- 最上位ビットが符号を表す(0なら正, 1なら負)

2の補数で表せる値の範囲

- 8ビットの2の補数では127+1= -128 になってしまう (オーバーフロー)
- 計算で用いる値を表すために十分なビット数を使って計算する必要がある
- 数を符号あり･なしのどちらで扱うかは予め決めておく
  - 10000000 = 8ビット符号ありだと -128
    8ビット符号なしだと +128

2の補数での加減算

- 5 + (-12) を計算してみる
  - 00000101 + 11110100 = 11111001
  - 11111001が$-7_{(10)}$の2進表現になっている
    - 11111001の全ビットを反転して+1すると
      $00000111 = 7_{(10)}$
- -1 + (-2)



演習

(以下すべて8ビットで計算せよ)

- -2, -3, -4 を2の補数で表現せよ
- 2の補数表現で 11001011(2) の表す値はいくつか?
- 2の補数を使って-3 - 4を計算せよ

小数点のある数･非常に大きい(小さい)数の表現

2進数の小数 (1)

- 10進数では

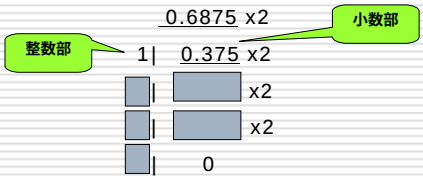
$$0.123_{(10)} = 1*\frac{1}{10} + 2*\frac{1}{100} + 3*\frac{1}{1000}$$

- 2進数では

$$0.101_{(2)} = 1*\frac{1}{2} + 0*\frac{1}{4} + 1*\frac{1}{8}$$
$$= 0.5 + 0 + 0.125$$
$$= 0.625_{(10)}$$

2進数の小数 (2)

- 10進数から2進数への変換
  - $0.6875_{(10)}$を2進数に変換



$0.6875_{(10)} = 0.1011_{(2)}$

演習

- 以下の10進数を2進数に変換せよ
- 0.875
- 10.125
- 0.1

2進数の小数 (3)

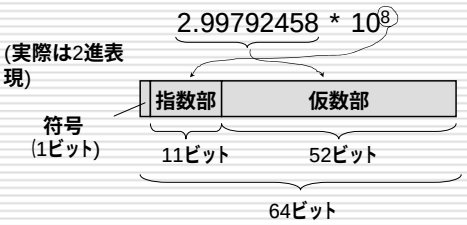
- 10進数では切りの良い小数も2進数にすると循環小数になる場合が多い
  - $0.1_{(10)} = 0.0001100110011001100110011\ldots_{(2)}$
- コンピュータは有限の精度でしか計算できない
  - 有限のビット数（32ビット，64ビットなど）で計算するため
- コンピュータでは実数を正確に扱えない
  近似値になる
  - 表面上正確に計算できているように見えても，丸められているだけの場合が多い
  - $0.1_{(10)}$を小数点以下32ビットで切り捨てた場合
    .09999999860301613807678222656 25
  - 64ビットだと
    .09999999999999999996747393482543486697977641597390174865722656 25

浮動小数点(1)

- 科学技術分野では数を指数形式で表す
  - 光の速さ　　$2.99792458 * 10^{8}$ [m/sec]
  - 水素原子の質量　　$1.7*10^{-24}$ [g]
- 同様に2進数で表現するのが浮動小数点形式
- 通常コンピュータで実数を扱うときはこれ
  - (もちろん近似値!)

浮動小数点(2)

- 例えば次のように表す
  (IEEE754規格の倍精度浮動小数点形式)



浮動小数点(3)

- 扱える値の範囲(IEEE754倍精度浮動小数点)
  - 絶対値の大きいほう　約 $\pm 1.8*10^{308}$
  - 絶対値の小さいほう　約 $\pm 4.9*10^{-324}$
    - これより小さい値は0になってしまう
      (アンダーフロー)
- 精度
  - 10進数で約16桁

まとめ

- 2進数，16進数
  - 10進数との変換
  - 加減算
- ビット，バイト，キロバイト，メガバイト，…
- 2の補数表現
- 小数，浮動小数点
  - 実数は正確に扱えない