第5章　演算アーキテクチャ

5.1.1　10進数の表現

2進表現

- 2進数nビットで表現できるデータの個数pは，
  - $p=2^n$
- p個のデータを表現するために必要なビット数nは，
  - $n=\log_2 p$

【例】

- 8種類の数字（文字）を表現するために必要なビット数は，
  - $n=\log_2 8=3$
- 10種類の数字（文字）を表現するために必要なビット数は，
  - $n=\log_2 10=3.22\cdots$
  - ビット長は整数なので，上記を切り上げて，n=4

文字	2進表現
a	000
b	001
c	010
d	011
e	100
f	101
g	110
h	111

文字	2進表現
a	0000
b	0001
c	0010
d	0011
e	0100
f	0101
g	0110
h	0111
i	1000
j	1001

10進数の表現

- 10進数
  - 我々が日常生活で用いる.

- 2進数
  - コンピュータが内部で用いる.

10進数を2進数へ変換する場合,
数値によっては, 誤差が生じてしまう.

金銭を扱うようなソフトウェアでは, 我々が10進数で計算した結果と,
コンピュータが「0, 1」を用いて計算した結果が, 異なってはいけない.

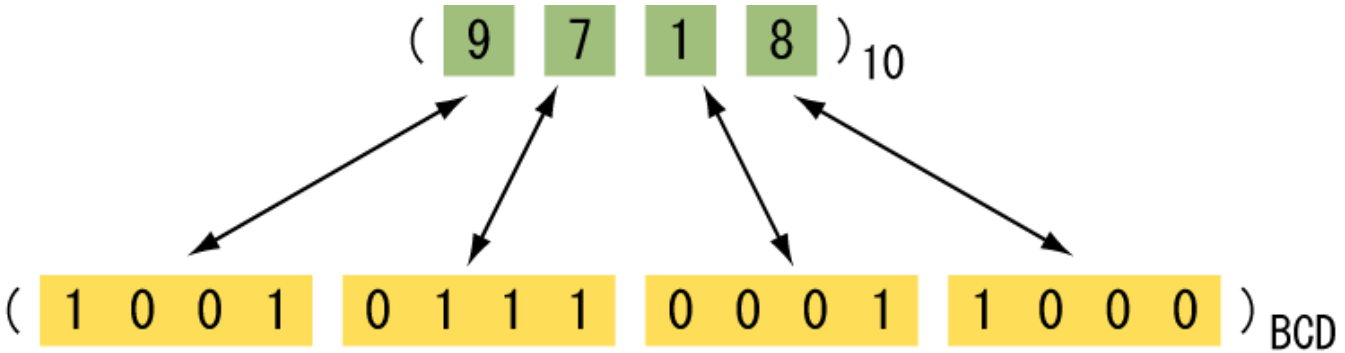
「0, 1」を用いて, 10進数値を,
“正確に”表現するための方法も必要である.

10進数の2進表現

10進数の2進表現

- 10進数を2進表現する方法
  - 10進数1桁（10種類の数字）を，2進数4ビットで表現する．
  - たとえば，2進数16ビットで，10進数4桁，すなわち，$(0000)_{10}$から$(9999)_{10}$を表現することができる．
  - BCDコード（BCD code ; Binary Coded Decimal code），3増しコード（excess-3 code），グレイコード（gray code）などがある．

【例】



BCDコード，3増しコード，グレイコード

BCDに3を足したコード

10進数	BCDコード	3増しコード	グレイコード
0	0000	0011	0000
1	0001	0100	0001
2	0010	0101	0011
3	0011	0110	0010
4	0100	0111	0110
5	0101	1000	0111
6	0110	1001	0101
7	0111	1010	0100
8	1000	1011	1100
9	1001	1100	1101

10進数のデータが1異なる場合，
対応するコードは1個所だけ異なる．

5.1.3　実数の表現

固定小数点数表現と浮動小数点数表現

- 固定小数点数表現
  - 小数点の位置を固定して，数を表現する．
  - 一般に，整数は，固定小数点数で表現される．

- 浮動小数点数表現
  - 数によって小数点の位置を変えて，数を表現する．
  - 一般に，実数は，浮動小数点数で表現される．

浮動小数点数表現

- 2進数$(R)_2$の浮動小数点数表現

$$R = m \times 2^e$$

m ： 仮数部
e ： 指数部

浮動小数点数表現では，
仮数部 m と指数部 e の2つの数値で，1つの2進実数を表す．

正規化数

- 2進実数は，仮数と指数の組み合わせによって，様々な浮動小数点数の形で表現され得る．

$(11.0001)_2$

$= 1100.01 \times 2^{-2}$

$= 110.001 \times 2^{-1}$

$= 11.0001 \times 2^{0}$

$= 1.10001 \times 2^{1}$

$= 0.110001 \times 2^{2}$

$= 0.0110001 \times 2^{3}$

正規化数の
仮数部 m と指数部 e を用いて，
2進実数を表す．

正規化数
仮数部において，
小数点の左側が1桁であり，
かつその値が0でないような数値を，
正規化数という．

（注）本講義では，上記を正規化数の定義とする．

浮動小数点数の精度

正規化数　$R = m \times 2^e$
　　m　：　仮数部
　　e　：　指数部

- 「仮数mの絶対値」の最小値　：　1.00 … 00 = 1
- 「仮数mの絶対値」の最大値　：　1.11 … 11 < 2.00 … 00
- 「仮数mの絶対値」の範囲　　：　$1 \leqq |m| < 2$

仮数mを何ビットで表現しようと，「仮数mの絶対値」の範囲は一定である．
ただし，仮数部のビット数を増やすと，表現できる仮数の個数が増える．
すなわち，表現の精度が良くなる．

浮動小数点数の範囲

正規化数　$R = m \times 2^e$
　　m　：　仮数部
　　e　：　指数部

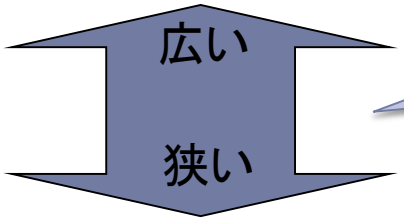
- 指数eの範囲（eを符号ビットなしのqビットの正整数とした場合）
  - $0 \leqq e \leqq 2^q-1$
- 2^eの範囲
  - $1 \leqq 2^e \leqq 2^{2^q-1}$
- 仮数mの範囲（前述）
  - $1 \leqq m < 2$
- 実数Rの範囲
  - $1 \leqq R < 2^{2^q}$

指数部のビット数を増やすと，
表現できる実数の範囲が広くなる．

浮動小数点数と固定小数点数の範囲比較

- 浮動小数点数（仮数部24ビット，指数部8ビット）で表現する実数Rの範囲

$$1 \leqq R < 2^{256}$$



浮動小数点数表現の方が，
表現範囲が格段に広い．

- 固定小数点数（整数部32ビット，小数部0ビット）で表現する整数Nの範囲

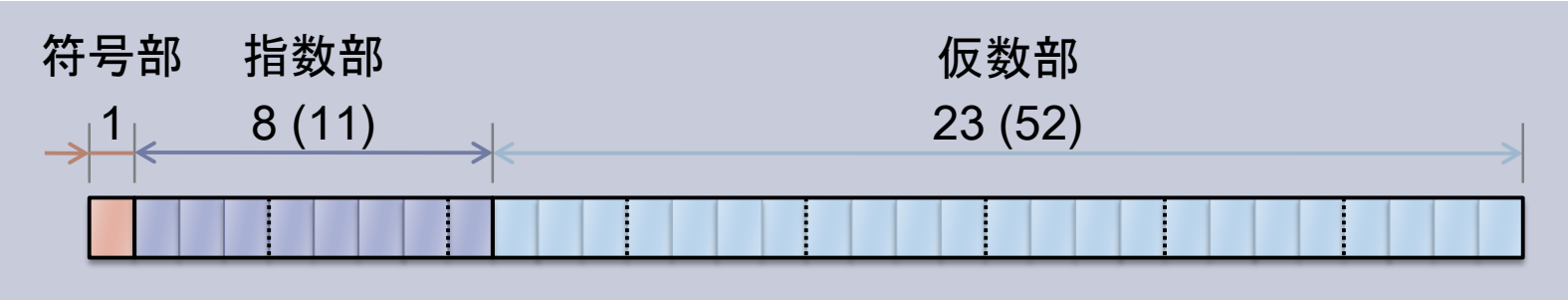
$$0 \leqq N < 2^{32}$$

単精度と倍精度

- 単精度
  - 仮数部と指数部を（合わせて）1ワード（現時点においては通常32ビット）で表現する浮動小数点数表現.
- 倍精度
  - 仮数部と指数部を（合わせて）2ワードで表現する浮動小数点数表現.

ANSI/IEEE標準規格

- ANSI（米国規格協会）/IEEE（国際電気電子学会）標準規格
  - 仮数は，符号-絶対値表現を用い，けち表現する．（後述）
  - 指数は，バイアス表現する．（後述）

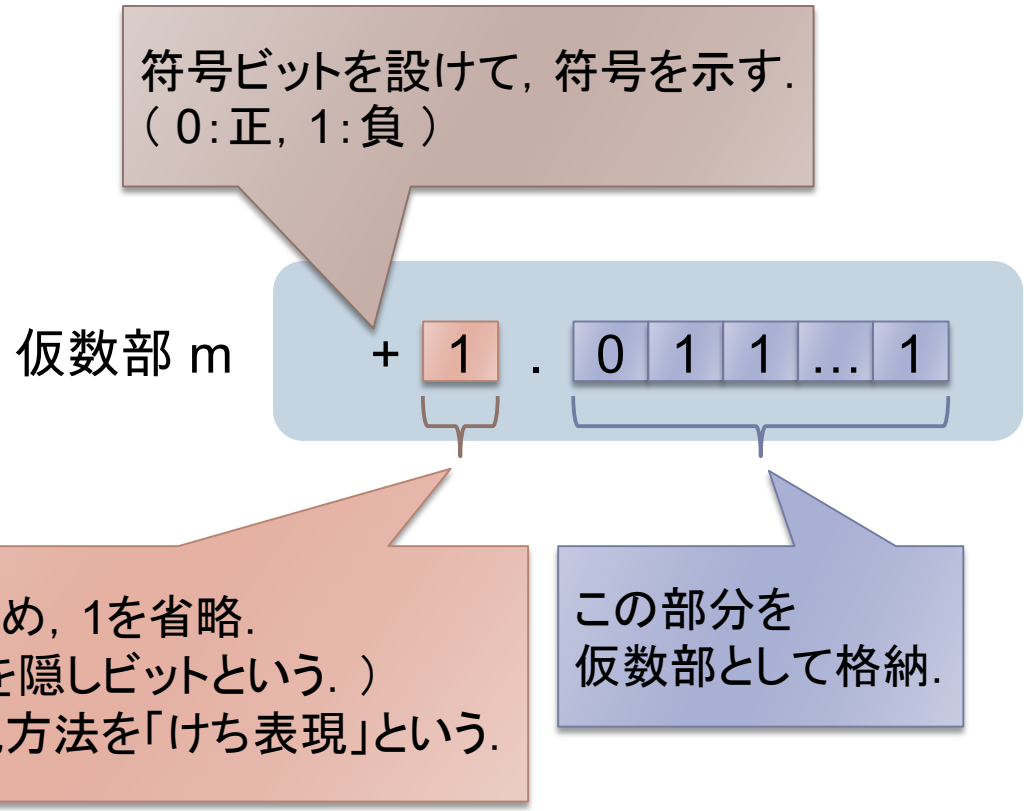


単精度ビット数（倍精度ビット数）

（注） 浮動小数点数表現は，コンピュータの機種に依存している．
そのため，本講では，ANSI/IEEEの標準規格に基づいて説明を行う．

仮数の表現

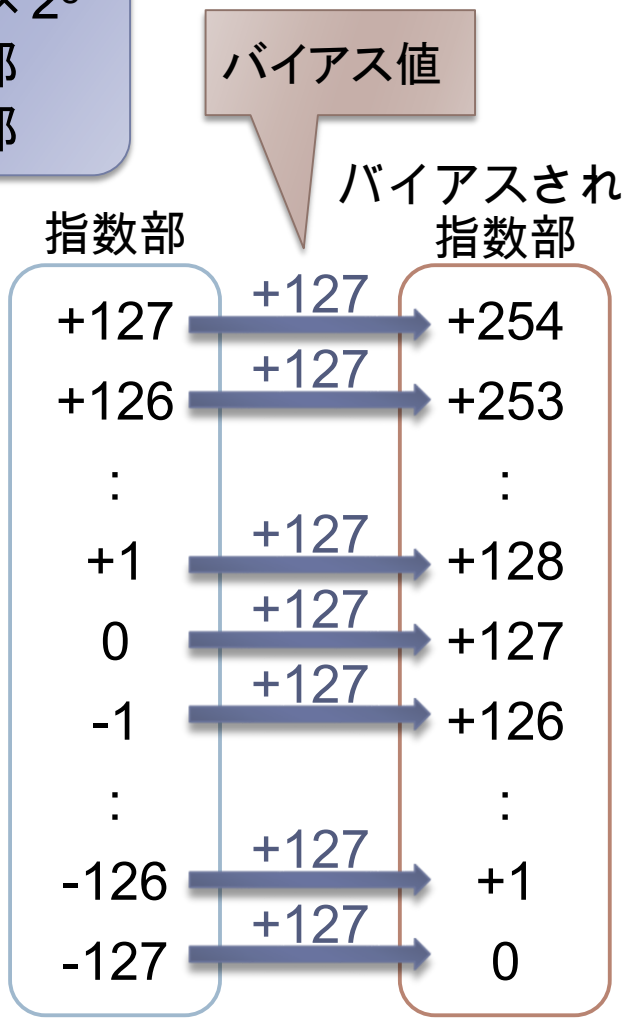
正規化数　$R = m \times 2^e$
　　m　：　仮数部
　　e　：　指数部



指数の表現

正規化数　$R = m \times 2^e$
m　：　仮数部
e　：　指数部

- ANSI/IEEE標準規格
  - 単精度のバイアス値：127
  - 倍精度のバイアス値：1023



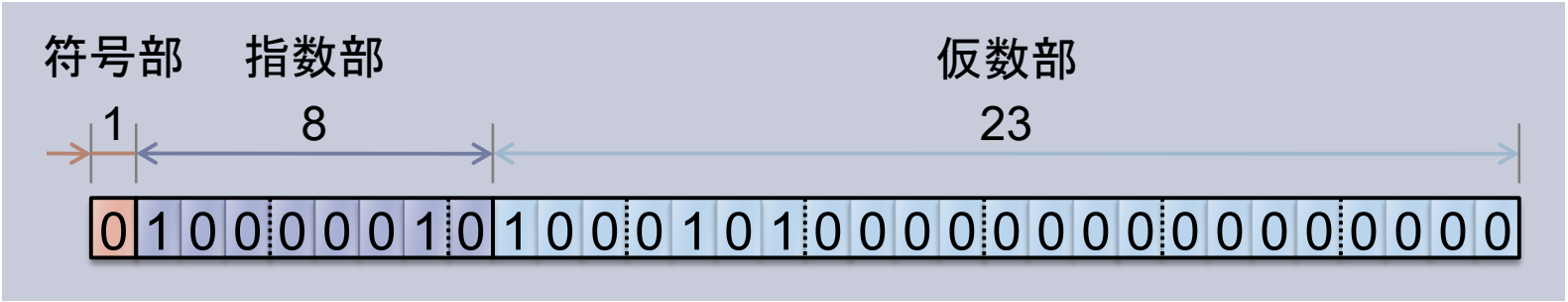
指数部は，
正負の値を
取り得る．

指数部にバイアス値を加え，
ゼロ以上の正整数に変換し，
これを指数部として格納．
これを，「バイアス表現」，
あるいは，「げた履き表現」
という．

ANSI/IEEE標準規格の具体例

【例】　$(12.3125)_{10}$のANSI/IEEE標準規格（単精度）による表現

- 10進数から2進数へ変換
  - $(12.3125)_{10}=(1100.0101)_2$
- 正規化
  - $(1100.0101)_2 =(1.1000101)_2 \times 2^3$
- 仮数部23ビット（けち表現）
  - 10001010000000000000000
- 指数部（バイアス値 127）
  - $(3)_{10} + (127)_{10}= (130)_{10} =(10000010)_2$
- 符号ビット
  - 0



演習問題

- 問題1
  - ANSI/IEEE標準規格に基づき，$(-0.75)_{10}$の単精度の浮動小数点数表現を，以下の手順に従って求めよ．

1. $(-0.75)_{10}$は，符号は　(a)　で，絶対値は（　(b)　$)_{10}$である．
2. （　(b)　$)_{10}$を2進数で表すと，（　(c)　$)_2$である．
3. （　(c)　$)_2$を，正規化すると，（　(d)　$)_2 \times 2^{(e)}$である．
4. したがって，仮数部には，隠しビットを除いた23ビット

が格納される．

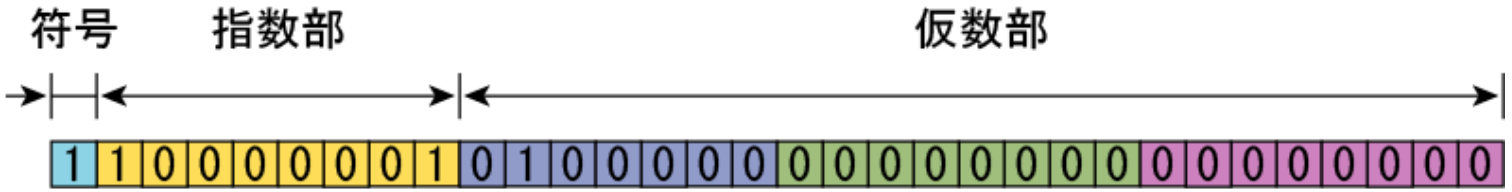
5. 一方，指数部を$(127)_{10}$でバイアスすると（　(f)　$)_{10}$になり，これを8ビットの符号なし整数で表すと

になる．

6. よって，$(-0.75)_{10}$の単精度の浮動小数点表現は，以下のようになる．

演習問題

- 問題2
  - ANSI/IEEE標準規格に基づき，以下に示される単精度の浮動小数点数が表現している10進数の値を，以下の手順に従って求めよ．



1. 符号ビットは1であり，　(a)　であることを示している．
2. 仮数部に格納された値は（　(b)　）$_{10}$であり，これに隠しビットによる（　(c)　）$_{10}$を加えると（　(d)　）$_{10}$になる．
3. 指数部に格納された値は（　(e)　）$_{10}$であり，これからバイアス値$(127)_{10}$を引くと（　(f)　）$_{10}$になる．
4. よって，求める値は，
   （　(d)　）$_{10}\times\ 2^{(f)}$ =（　(g)　）$_{10}$
   になる．

5.1.4　文字データの表現

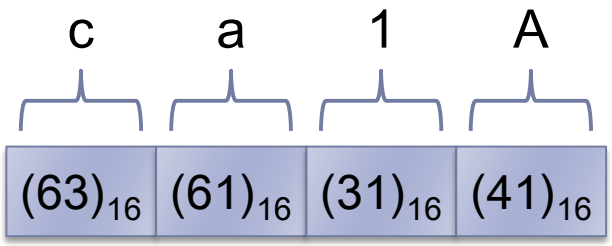
文字データの表現

- 文字コード
  - 英数字などの文字情報そのものを，表現あるいは識別するコード.
  - 基本的には1文字を1バイト（8ビット）で表現.
    （1バイトコードという）
  - 主要な文字コードに，ASCII，EBCDICなどがある.

- 日本語文字コード
  - 多種多様な日本語文字（ひらがな，カタカナ，漢字など）を，表現あるいは識別するコード.
  - 1文字を2バイト（16ビット）で表現.
    （2バイトコードという）
  - 主要な日本語文字コードにシフトJIS，EUC，Unicodeなどがある.

ASCII（アスキー）

- ASCII
  (American Standard Code
  for Information Interchange)
  - 米国規格協会が制定した1バイトで
    1文字を表すコード.
    （1バイト中の7ビットを使用している）
  - JIS（日本工業規格）でも
    JIS X0201として定められている.



数値の1と，文字の1は，
コンピュータの中では
異なる2進数値で表現されている.

ASCII コード表

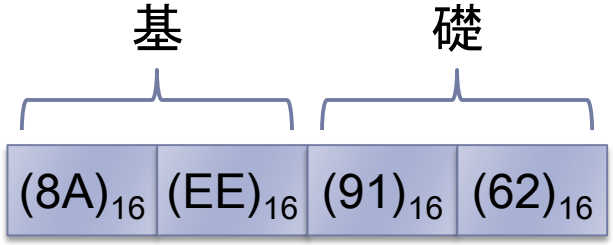
下位4ビット	上位3ビット 0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P	`	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	“	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	‘	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	¥	l	|
D	CR	GS	−	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

EBCDIC（イビシディック）

- EBCDIC (Extended Binary Coded Decimal Interchange Code)
  - IBMが開発した1バイトで1文字を表すコード.
    （8ビットすべてを使用している）

シフトJISコード

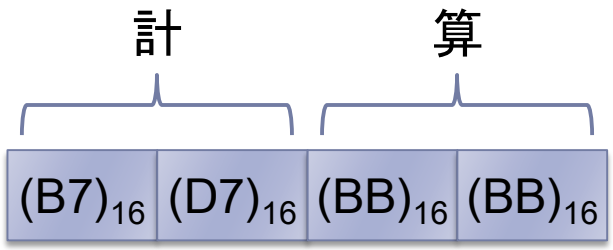
- シフトJISコード
  - 日本工業規格（Japan Industrial Standard）
  - 上位バイトが $(20)_{16}$～$(7E)_{16}$の範囲にある場合には，1バイトコードのASCIIとして扱う．
  - 上位バイトが $(81)_{16}$～$(9F)_{16}$，$(E0)_{16}$～$(FC)_{16}$の範囲にある場合には，その下位バイトと合わせた2バイトを，日本語文字コードとして扱う．
  - 主に，パソコン用として使われている．



制御コード	$(00)_{16}$～$(1F)_{16}$, $(7F)_{16}$
ASCII文字	$(20)_{16}$～$(7E)_{16}$
漢字	$(8140)_{16}$～$(9FFC)_{16}$, $(E040)_{16}$～$(FCFC)_{16}$ ただし，第1バイトは$(81)_{16}$～$(9F)_{16}$，$(E0)_{16}$～$(FC)_{16}$， 第2バイトは$(40)_{16}$～$(7E)_{16}$ ，$(80)_{16}$～$(FC)_{16}$，

EUC

- EUC（Extended UNIX Code）
  - AT&Tが定めたコード.
  - 上位バイトが $(20)_{16}$～$(7E)_{16}$の範囲にある場合には，1バイトコードのASCIIとして扱う.
  - 上位バイトが $(A1)_{16}$～$(FE)_{16}$の範囲にある場合には，その下位バイトと合わせた2バイトを，日本語文字コードとして扱う.
  - 主に，UNIX用として使われている.



制御コード	$(00)_{16}$～$(1F)_{16}$, $(7F)_{16}$
ASCII文字	$(20)_{16}$～$(7E)_{16}$
漢字	$(A1A1)_{16}$～$(FEFE)_{16}$ ただし，第1・第2バイトとも$(A1)_{16}$～$(FE)_{16}$

UNICODE

- Unicode
  - ユニコードコンソーシアム（The Unicode Consortium）が定めたコード.
  - 2バイトコードで全世界の文字を表現.
  - 上位バイトによって,
    - アルファベット（Alphabet） (A) : アルファベット，ギリシア文字など
    - イデオグラフ（Ideograph） (I) : 漢字など
    - オープン（Open） (O) : 現在未定義
    - 制限（Region） (R) : 限定使用の文字など

    の領域に分けて，コード化.
  - I領域に属している漢字は，日本，中国，台湾，韓国で統一してコード化しており，字体が微細な部分で異なる漢字を同一文字コードに割り当てている.