コード

コンピュータ基礎 (3)

菊池浩明

講義概要

- 教科書
  - 2章データ表現
  - 4. 浮動小数点数
  - 6. コード
    - » 10進コード
    - » 文字コード
    - » ハミングコード

浮動小数点 (floating-point format)

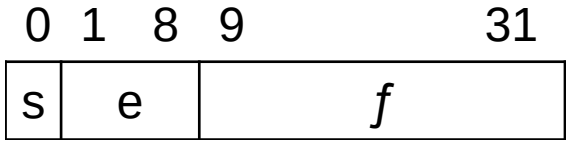
- 定義
  - 小数点の位置を固定しない小数の表現.
  - 符号s＋仮数f　　＋指数e
    (sign)　(mantissa)　(exponent)
  - 例）-0.125×10^{7}
    s = 1 (負),
    f = .125 = $.001_{(2)}$ = 1/8 （Mで表す時もある）
    e = 7 = $0111_{(2)}$
  (実際はもう少し複雑）

IEEE 754

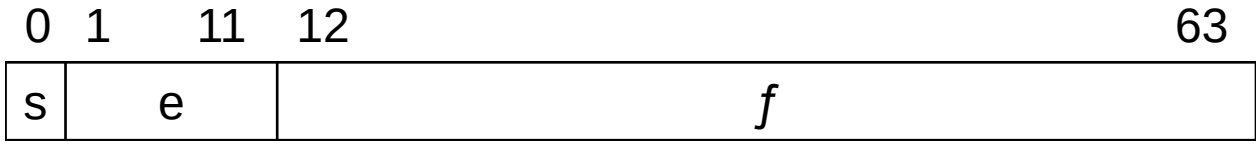
- 浮動小数点の表示規格
  - IEEE (Institute of Electrical and Electronics Engineering; 米国電気電子学会）
  - 短精度 (float)
    - » s = 1bit, e = 8bit, f = 23 bit: 計32 bit (4 byte)
  - 倍精度 (double)
    - » s = 1bit, e = 11bit, f = 52 bit: 計64 bit (8 byte)

IEEE 754

- 単精度
  - 32ビット



- 倍精度
  - 64ビット



正規化

- 仮数部の正規化
  - $0.15625_{(10)} = 0.00101_{(2)}$

    $= 0.00101 \times 2^{0}$; f =00101, e=0

    $= 1.01 \times 2^{-3}$; f =101, e = -3

    $= 1.01 \times 2^{-3}$; f = 01, e = -3

    (MSBは必ず1に正規化されるので省略;「ケチ表現」)

浮動小数点数の演算

- 加算
  - $300_{(10)} + 0.2_{(10)} = 3 \times 10^{2} + 2 \times 10^{-1}$
    $= 3000 \times 10^{-1} + 2 \times 10^{-1}$;2と-1の最小値に揃える
    $= 3.002 \times 10^{2}$
- 乗算
  - $300_{(10)} \times 0.2_{(10)} = 3 \times 10^{2} \times 2 \times 10^{-1}$
    $= 3 \times 2 \times 10^{2-1}$; 仮数部は乗算，指数部は加算
    $= 6 \times 10^{1}$
  - 指数部の計算を簡単化するためバイアス表示を導入

負数の表現（指数部のみ）

一般の整数					IEEE 754	
10進数	2進数	絶対値(符号なし)	符号	2の補数	符号	バイアス表示
FF	11111111	255	−	−1	+	最大 +128
FE	11111110	254		−2		+127
80	10000000	128		最小 −128		+1
7F	01111111	127	+	最大 +127	−	0
1	00000001	1		1		−126
0	00000000	0		0		最小 −127

バイアス表示の理由

- 理由1
  - 指数部の計算の効率化
- 理由2
  - 「クリーンゼロ」
    $0 = 0 \times 10^{-1023}$ の時，s=f=e=0となる
  - アンダーフローしても安全
  - オーバーフロー e=1024, f=0

IEEE 754による表現

- 符号部s, 仮数部f, 指数部eによる浮動小数点数x

$$x = (-1)^s \, 2^{e-127} \times (1.f)$$

  - -127 = $7F_{(16)}$ は指数のバイアス表示
  - 1.f の1はケチ表現

例1

- 例) -14.625をIEEE754単精度で表せ.
  - -14.625 = $-1110.101_{(2)}$
    $= -1.110101 \times 2^3$
  - s = 1 (-1)
  - $f = 110101_{(2)} = 35_{(16)}$
  - e= 7F+3 = $82_{(16)}$ = 1000010
  - 1 1000010 110101 00000 00000 00000 00000 000

演習

- 1)　s = 1, f=101, e = 80　で表される浮動小数点xを10進実数で示せ.
  - x = (-1)s x 1.f x 2^{80-7F} =-1 x 1.101 x 2
    = $_{(2)}$ = $_{(10)}$
- 2) 10進数 14.8125　を浮動小数点表記せよ.
  - $14.8125_{(10)} = 1110.1101_{(2)}$
    $= 1.1101101 \times 2^{3}$
    s =　, f =　, e = 7f +3 = $_{(16)}$

実際のメモリ上の値

- 実行例(Intel , Gcc)
  - -3.25
  - 0 0 50 c0　　f=5 0 0, s=1, e = 80
  - 3.25
  - 0 0 50 40　　f=5 0 0, s=0, e = 80
  - 14.8125
  - 0 0 6d 41　　f=6d 0 0, s=0, e = 81

特殊な浮動小数点数

- 特殊な数
  - s= 0, f =0, e = 0　　x=0「クリーン0」
    - » s = 1, f =0, e = 0　　（未定義？）
    - » s=0, f =10, e = 0　　禁止
  - s = 0, f =0, e= FF　　オーバーフロー
  - s = 0, f ≠0, e= FF　　非数 (NaN;
    - » 1/0や, √-1 など　　Not a Number)
- 通常の数値の範囲: -126 ≤ e ≤ +127

1　　FE

コード

文字コード

ASCII, Unicode

コード

- code (符号)
  - 情報の表現方法
  - 符号化 encode, 復号化 decode
  - 例）値$12_{(10)}$を2進数4bitで1100で符号化する

big endian v.s. little endian

- Big endian
  - X = $1234_{(16)}$

アドレス	値
2000	12
2001	34

  - ネットワークでの通信．Motrora, Sun SPARCなど

- Little Endian
  - X = $1234_{(16)}$

アドレス	値
2000	34
2001	12

  - Intel x86

卵を大きい方から割る人々．
「ガリバー旅行記」

http://hareenlaks.blogspot.jp/2010/12/big-endian-

10進数のコード

- BCDコード (Binary Coded Decimal)
  - 10進数1ケタを2進数4bit で表現する.
  - 例）$23_{(10)}$ = 0010 0011
- ゾーン10進数
  - 10進数1ケタを，1byteで表現. 符号は最後の1byteの上位4bit (1100 = 正，1101 = 負）
  - 例）$-23_{(10)}$ = 00110010 11010011
- パック10進数
  - BCDコードに，符号を加える.
  - 例）$+23_{(10)}$= 0010 0011 1100

例

- -1024をパック10進数で表せ.

- BCDコード，ゾーン10進数，パック10進数で表現された値がある．どれがどれか．
  1. 1001 0011 0001 1100
  2. 1001 0011 0001
  3. 0011 1001 0011 0011 1100 0011

文字コード

- ASCIIコード
  - _____ national Standard Code for Information Interchange (情報交換の為の米国標準コード)，7 bitの英数文字コード

L	0	1	2	3	4	5	6	7
0	NUL		SP	0	@	P	`	p
1			!	1	A	Q	a	q
2			"	2	B	R	b	r
3			#	3	C	S	c	s
4			$	4	D	T	d	t
5			%	5	E	U	e	u
6			&	6	F	V	f	v
7			'	7	G	W	g	w
8	BS		(	8	H	X	h	x
9			)	9	I	Y	I	y
A	LF		*	:	J	Z	j	z
B		ESC	+	;	K	[	k	{
C			,	<	L	¥	l	|
D	CR		-	=	M	]	m	}
E			.	>	N	^	n	~
F			/	?	O	_	o	

例

- 文字列の表現
  - "ABC" = 41 ＿ ＿$_{(16)}$ (3 byte)
  - "123" = 31 ＿ ＿$_{(16)}$
  - 大文字から小文字の変換
    "A" = 41 + $20_{(16)}$ = 61 = "a"
    "D" = 44 + $20_{(16)}$ = 64 = "d"
  - 8を表す文字列 = 8 + $30_{(16)}$ = $38_{(16)}$
  - 次の符号はどんな文字列か？
    31 73 74 46 4D 53

改行コード

- 改行に関する制御コード
  - LF (Line Feed) 0a
  - CR (Carriage Return) 0d
- OSの違い

http://www.kurzweilai.net/passing-of-the-typewriter

改行	OS
LF	Linux, Mac OS X
CR + LF	Windows, Network標準形
CR	Mac OS 9

日本語文字コード

コード	符号語数	特徴	代表例
JIS (Japanese Industrial Standard) X 208	可変長	7 bit 文字コード, モードにより英数漢字を分ける	電子メール
Shift JIS (sjis)	2バイト固定長	8 bit 2byte コード	PC (Windows, MacOS)
EUC (Extended Unix Code)	2バイト固定長	8 bit 2 byteコード (2バイト目に制約)	Linuxなど
Unicode (UTF-16)	2バイト固定長	多国語(日中韓の漢字を同一コードで統一)	Javaの内部コード
UTF-8	1〜3バイト可変長	8 bitコードの組み合わせ	Web

文字コード表示例

- 文字列 "OhOh明治¥n"

	BOM	"O"	"h"	"O"	"h"	ESC$ B	"明"	"治"	CR LF
JIS		4f	68	3f	68	1b 24 42	4c 40	3c 23	0d 0a
SJIS		4f	68	4f	68		96 be	8e a1	0d 0a
EUC		4f	68	4f	68		cc c0	bc a3	0d 0a
UTF16	ff fe	4f 00	68 00	4f 00	68 00		0e 66	bb 6c	0d 00 0a 00
UTF-8	ef bb bf	4f	68	4f	68		e6 98 8e	e6 b2 bb	0d 0a

JIS

- 概要
  - JIS X 0208（第1水準2965文字，第2水準 3390文字）とASCIIを混在して使う符号化形式 (ISO-2022-JP). 電子メールなどの標準
- エスケープシーケンス

1B 28 42	ESC (B	ASCII
1B 28 4A	ESC (J	JIS X 0201 ラテン文字
1B 24 40	ESC $ @	JIX X 0208 1978版
1B 24 42	ESC $ B	JIS X 0208 1983版

  - 例)

A	7				計		算					;	a
41	37	1B	24	42	37	57	3B	3B	1B	28	4A	3B	a

- 問題点
  - statefull (現在の状態によって符号化が変わる）

Shift_JIS

- 概要
  - JIS X 0201の隙間にJIS X 0208を押し込んだ符号化形式. PCでの＿＿＿＿＿＿スタンダード.
- 特徴
  - 2バイトの固定長コード. 第一バイト目に 81~9F, E0~EFが来たら漢字（2バイトコード）
- 問題点
  - 第2バイト目に7F以降(ASCII)が使われていて, 処理系では誤動作する可能性.

EUC (Extended Unix Code)

- 概要
  - Unix系のOSの業界団体が定めたコード．2バイトの固定長符号.
- 特徴
  - A1~FEで始まるデータは2バイトコード（7F以下は必ずASCII).
- 問題点
  - 第1バイトと2バイトが同じ範囲の値を取るので，区切りの判別が困難.

Unicode (UTF-16)

- 概要
  - ベンダーのコンソーシアムで策定された16ビットの国際文字コード仕様．漢字の統合などが行われたが，現在はISOと統合してUTF-16, UTF-8などに拡張された．
- 特徴
  - 符号語の単位でUTF-8, UTF-16, UTF-32など複数の符号化がある．

UTF-8

- ASCIIと互換性がある，8ビット単位のUnicode符号化方式．ISO/IEC 10646やRFC 3629でも定義されている．

符号値域	長	UTF-8の符号化バイト列
0000 0000-0000 007F	1	0xxxxxxx (ASCII)
0000 0080-0000 07FF	2	110xxxxx 10xxxxxx
0000 0800-0000 FFFF	3	1110xxxx 10xxxxxx 10xxxxxx
0001 0000-0010 FFFF	4	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx

00	1 byte (ASCII)
01	
02	
03	
04	
05	
06	
07	
08	多バイト符号語の2バイト目以降
09	
0A	
0B	
0C	2 byte
0D	
0E	3 byte
0F	4 byte

例

ef	bb	bf	48	61	72	72	79	e3	81	af	31	39	38	30	e5	b9	b4
			1	1	1	1	1	3a	b	b	1	1	1	1	3a	b	b
BOM			H	a	r	r	y	は			1	9	8	0	年		

50	6f	74	74	65	72	e5	ae	b6	e3	81	ae	e9	95	b7	e7	94	9f

BOM

- BOM (Byte Order Mark)
  - UTF-16などで多バイトコードのエンディアンを示すために先頭につけられたコード.
    - » FE FF = Big Endian
    - » FF FE = Little Endian
  - UTF-8では，Uncodeを示すために使われる.
    - » FE BB BF

Unicodeの問題点

- 表記の問題
  - 明治大学先端メディアサイエンス 2014
  - 明治大学先端メディアサイエンス 2014
- 漢字の統合
- 円記号の問題

機種依存文字

- 概要　(環境依存文字）
  - JIX X 0208の空き領域にベンダーが独自に定義した文字「外字」
  - Windows ①②，ⅢⅣ，㎞，㎏，㈱
  - Macintosh (月)(火)
  - メールなどで送ると文字化けを引き起こすので極力避ける．授業名「プログラミング演習Ⅱ」

漢字の違い

- 簡体字(中国本土）simplified
- 繁体字(香港，台湾）traditional

日本語	簡体字	繁体字
豊	丰	豐
万	万	萬
東	东	東
	习	
	业	
	广	廣

文字コードの変換

- エディターによる自動変換
  - 「文字コード指定保存」（Tera Pad）

- 変換ツール
  - nkf (Network Kanji Filter)
    - » nkf –j < input > output
    - » -j (jis), -s (shift jis), -e (EUC), -w (Utf-8), -w16 (Utf-16)

誤り訂正コード

パリティコード

ハミングコード

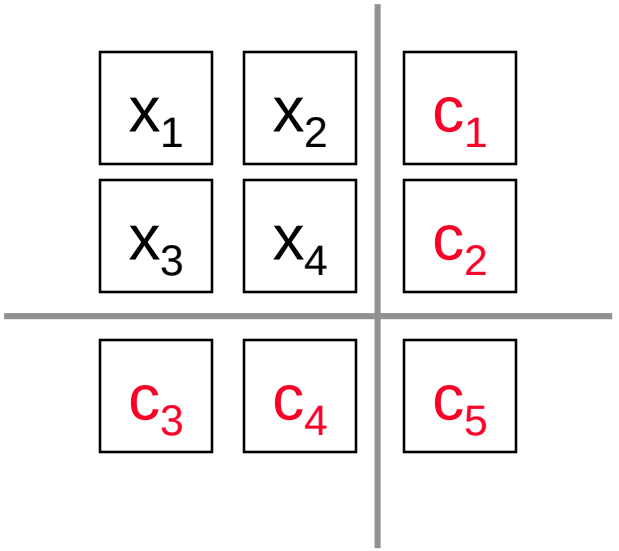
誤り訂正・検出技術

- 誤り訂正符号
  - パリティ検査符号
    - » 通信，記憶装置
  - ハミング符号
    - » 基本・理論的整理
  - BCH符号
    - » Reed-Solomon符号 (CD)
- 誤り検出符号
  - 巡回符号
    - » 実用的（容易な復号回路）
    - » CRCチェック（パケットの通信誤り）

水平垂直パリティ検査符号

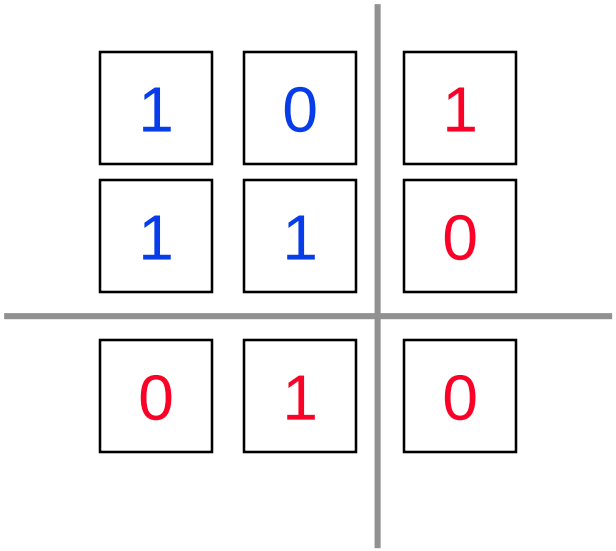
- 検査方程式
  - $c_1=x_1+x_2$
  - $c_2=x_3+x_4$
  - $c_3=x_1+x_3$
  - $c_4=x_2+x_4$
  - $c_5=c_1+c_2+c_3+c_4$
- (9,4)符号
- 符号語

  $\mathbf{w}=(x_1, x_2, x_3, x_4, c_1, c_2, c_3, c_4, c_5)$



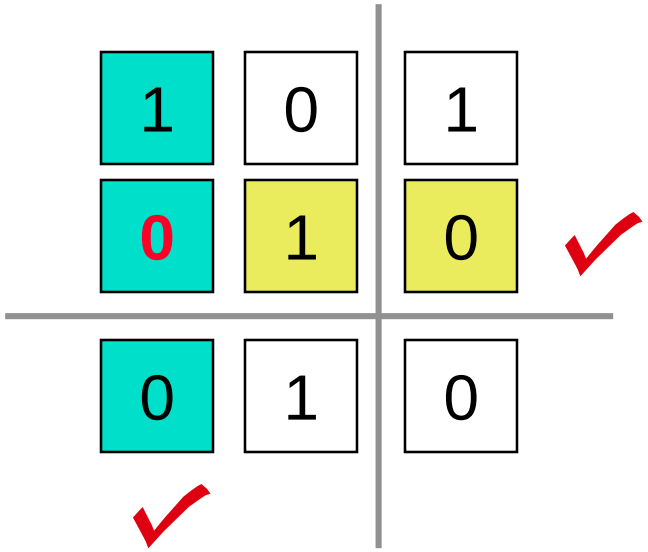
誤り訂正の原理

- 正しい受信語



y=(1,0,1,1, 1,0,0,1,0)

- 誤りのある受信語



y=(1,0,**0**,1, 1,0,0,1,0)

w=(1,0,**1**,1, 1,0,0,1,0)

Richard Wesley Hamming

- ハミング符号
  - 誤り訂正，検出符号
  - 1915-1998
  - Bell Lab.
  - Turing Prize in 1968
  - IEEE, Hamming Medal

(7,4)ハミング符号

- 水平垂直パリティ検査符号
  - $c_1=x_1+x_2 \pmod 2$
  - $c_2=x_3+x_4$
  - $c_3=x_1+x_3$
  - $c_4=x_2+x_4$
  - $c_5=c_1+c_2+c_3+c_4$
- (9,4)符号
  - $\mathbf{w}=(x_1, x_2, x_3, x_4, c_1, c_2, c_3, c_4, c_5)$
- ハミング符号
  - $c_1=x_1+x_2+x_3$
  - $c_2=x_2+x_3+x_4$
  - $c_3=x_1+x_2+x_4$
- (7,4)符号
  - $\mathbf{w}=(x_1, x_2, x_3, x_4, c_1, c_2, c_3)$
  - 符号化率 $\eta=4/7$

情報ビットの符号化

- 符号語
  - 情報ビット

    x_1,x_2,x_3,x_4= (1,0,1,0)
  - 検査ビット

    $c_1=x_1+x_2+x_3$
    $=1+0+1=0$

    $c_2=0+1+0=1$

    $c_3=1+0+0=1$
  - 符号語

    w=(1,0,1,0, 0,1,1)

- 検査方程式

$$\begin{cases} c_1=x_1+x_2+x_3 \\ c_2=\quad x_2+x_3+x_4 \\ c_3=x_1+x_2\quad +x_4 \end{cases}$$

$$\begin{cases} x_1+x_2+x_3\quad +c_1\quad =0 \\ \quad x_2+x_3+x_4\quad +c_2\quad =0 \\ x_1+x_2\quad +x_4\quad +c_3=0 \end{cases}$$

符号語 C

$x_1x_2x_3x_4$	c_1
0 0 0 0	0
0 0 0 1	0
0 0 1 0	1
0 0 1 1	1
0 1 0 0	1
0 1 0 1	1
0 1 1 0	0
0 1 1 1	0
1 0 0 0	1
1 0 0 1	1
1 0 1 0	0
1 0 1 1	0
1 1 0 0	0
1 1 0 1	0
1 1 1 0	1
1 1 1 1	1

- 検査方程式

$$\begin{cases} c_1 = x_1 + x_2 + x_3 \\ c_2 = x_2 + x_3 + x_4 \\ c_3 = x_1 + x_2 + x_4 \end{cases}$$

$$\begin{cases} x_1 + x_2 + x_3 + c_1 = 0 \\ x_2 + x_3 + x_4 + c_2 = 0 \\ x_1 + x_2 + x_4 + c_3 = 0 \end{cases}$$

符号語 C

$x_1x_2x_3x_4$	c_1	c_2
0 0 0 0	0	0
0 0 0 1	0	1
0 0 1 0	1	1
0 0 1 1	1	0
0 1 0 0	1	1
0 1 0 1	1	0
0 1 1 0	0	0
0 1 1 1	0	1
1 0 0 0	1	0
1 0 0 1	1	1
1 0 1 0	0	1
1 0 1 1	0	0
1 1 0 0	0	1
1 1 0 1	0	0
1 1 1 0	1	0
1 1 1 1	1	1

- 検査方程式

$$\begin{cases} c_1 = x_1 + x_2 + x_3 \\ c_2 = x_2 + x_3 + x_4 \\ c_3 = x_1 + x_2 + x_4 \end{cases}$$

$$\begin{cases} x_1 + x_2 + x_3 + c_1 = 0 \\ x_2 + x_3 + x_4 + c_2 = 0 \\ x_1 + x_2 + x_4 + c_3 = 0 \end{cases}$$

符号語 C

$x_1x_2x_3x_4$	c_1 c_2 c_3	
0 0 0 0	0 0 0	$=w_0$
0 0 0 1	0 1 1	$=w_1$
0 0 1 0	1 1 0	$=w_2$
0 0 1 1	1 0 1	$=w_3$
0 1 0 0	1 1 1	$=w_4$
0 1 0 1	1 0 0	$=w_5$
0 1 1 0	0 0 1	$=w_6$
0 1 1 1	0 1 0	$=w_7$
1 0 0 0	1 0 1	$=w_8$
1 0 0 1	1 1 0	$=w_9$
1 0 1 0	0 1 1	$=w_{10}$
1 0 1 1	0 0 0	$=w_{11}$
1 1 0 0	0 1 0	$=w_{12}$
1 1 0 1	0 0 1	$=w_{13}$
1 1 1 0	1 0 0	$=w_{14}$
1 1 1 1	1 1 1	$=w_{15}$

- 検査方程式

$$\begin{cases} c_1 = x_1 + x_2 + x_3 \\ c_2 = \quad x_2 + x_3 + x_4 \\ c_3 = x_1 + x_2 \quad + x_4 \end{cases}$$

$$\begin{cases} x_1 + x_2 + x_3 \quad + c_1 \quad = 0 \\ \quad x_2 + x_3 + x_4 \quad + c_2 \quad = 0 \\ x_1 + x_2 \quad + x_4 \quad + c_3 = 0 \end{cases}$$

誤りが発生したら？

- 単一誤り
  - 符号語　　w_3=(0,0,1,1, 1,0,1)
  - 誤りベクトル　　e =(0,0,1,0, 0,0,0)
  - 受信語　　y =(0,0,0,1, 1,0,1)
- 検査方程式

$$x_1+x_2+x_3 \quad +c_1 \qquad =0+0+0 \quad +1 \qquad =1 =s_1$$

$$x_2+x_3+x_4 \quad +c_2 \qquad = \ +0+0+1 \ +0 \qquad =1 =s_2$$

$$x_1+x_2 \quad +x_4 \qquad +c_3 \quad =0+0 \quad +1 \quad +1 =0 =s_3$$

- シンドローム

単一誤りのシンドローム

誤りベクトル $x_1x_2x_3x_4c_1c_2c_3$	シンドローム $s_1s_2s_3$
1 0 0 0 0 0 0	1 0 1
0 1 0 0 0 0 0	1 1 1
0 0 1 0 0 0 0	1 1 0
0 0 0 1 0 0 0	0 1 1
0 0 0 0 1 0 0	1 0 0
0 0 0 0 0 1 0	0 1 0
0 0 0 0 0 0 1	0 0 1

- シンドローム

$s_1 = x_1+x_2+x_3 \quad +c_1$

$s_2 = \quad x_2+x_3+x_4 \quad +c_2$

$s_3 = x_1+x_2 \quad +x_4 \quad +c_3$

誤り訂正

誤りベクトル $x_1x_2x_3x_4c_1c_2c_3$	シンドローム $s_1s_2s_3$
1 0 0 0 0 0 0	1 0 1
0 1 0 0 0 0 0	1 1 1
0 0 1 0 0 0 0	1 1 0
0 0 0 1 0 0 0	0 1 1
0 0 0 0 1 0 0	1 0 0
0 0 0 0 0 1 0	0 1 0
0 0 0 0 0 0 1	0 0 1

- シンドローム
  - 全て異なる
  - 誤りパターンと一対一に対応
- 例
  - **y**=(0,0,0,0,1,1,0)
  - **s**=(1,1,0)
  - **e**=(0,0,1,0,0,0,0)
  - **w**=(0,0,1,0,1,1,0)

演習

- (6,3) ハミング検査符号
  - » $x_1+x_2 \quad +c_1=0$
  - » $x_1+x_2+x_3 \quad +c_2=0$
  - » $x_1 \quad +x_3 \quad +c_3=0$
  - 次の受信語のシンドロームを求め，誤りを正せ
    - » $\mathbf{y}_1=(0,1,0,1,1,0)$
    - » $\mathbf{y}_2=(0,1,0,1,0,1)$
    - » $\mathbf{y}_3=(0,0,1,1,1,0)$
    - » $\mathbf{y}_4=(0,0,0,0,1,1)$

(ヒント) 符号語　C

$x_1x_2x_3$	$c_1c_2c_3$
0 0 0	0 0 0
0 0 1	0 1 1
0 1 0	1 1 0
0 1 1	1 0 1
1 0 0	1 1 1
1 0 1	1 0 0
1 1 0	0 0 1
1 1 1	0 1 0

- 検査方程式

$$\begin{cases} x_1+x_2 \quad +c_1=0 \\ x_1+x_2+x_3 \quad +c_2=0 \\ x_1 \quad +x_3 \quad +c_3=0 \end{cases}$$

(ヒント) 単一誤りの全シンドローム

$x_1x_2x_3$	$c_1c_2c_3$	$s_1s_2s_3$
1 0 0	0 0 0	1 1 1
0 1 0	0 0 0	1 1 0
0 0 1	0 0 0	0 1 1
0 0 0	1 0 0	1 0 0
0 0 0	0 1 0	0 1 0
0 0 0	0 0 1	0 0 1

- 検査方程式

$$\begin{cases} x_1+x_2 \quad +c_1=0 \quad =s_1 \\ x_1+x_2+x_3 \quad +c_2=0 \quad =s_2 \\ x_1 \quad +x_3 \quad +c_3=0 \quad =s_3 \end{cases}$$

- 誤り訂正

$s=(0,1,1)$

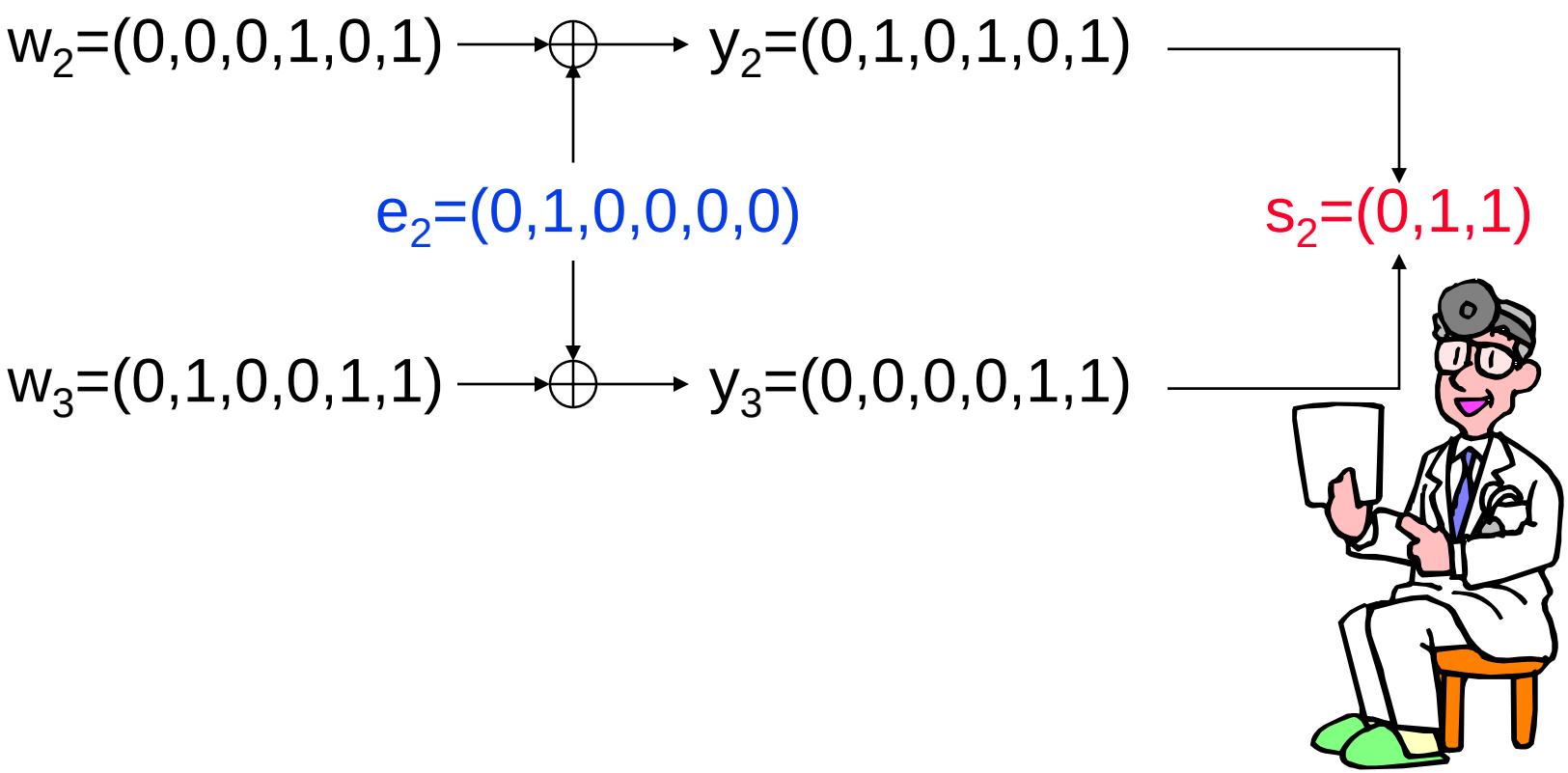
$e_2=(0,1,0,0,0,0)$

$y_2=(0,1,0,1,0,1)$

$w_1=(0,0,0,1,0,1)=y_2+e_2$

シンドロームの性質

- 誤りが同じ時，シンドロームも同じ



宿題

- 2章
  - 問4 (文字コード)
  - 問6
  - 問7
  - 問8
  - 問9
  - 問10

まとめ

- 浮動小数点数は，符号sと（　　）fと指数( )の3つの要素で表現される．IEEE754では指数部は2の補数ではなく（　　）で表現される．
- 日本語文字コードには，7ビットの（　　），PCのディファクト標準である（　　），多国語に対応した可変長の（　　）がある．
- （　　）符号は代表的な誤り訂正符号である．検査方程式で符号語を検査した結果（　　）により誤り位置を検出する．