

文字の表現

樋口さぶろお

龍谷大学工学部数理情報学科

情報処理の基礎 L05(2014-10-22 Wed)

今日の目標

- 文字コードを用いた符号化, 復号化の仕組みを説明できる.
- 文字コード表を参照して, 文字列とビットパターンとを互いに書き替えられる.
- 文字セットと, 文字列の長さから, 必要なビットパターンの長さを求められる.



<http://hig3.net>

略解 I

L04-S3 110000

L04-S5 011111

L04-S6

- 1
- -31

L04-S8 10011100

L04-S9 -3

ここまで来たよ

- 1 復習:負の整数の 2 進法表現・2 の補数
 - 2 の補数
 - 演算装置:減算器
 - オーバーフロー
- 2 文字の表現
 - ビット長と場合の数
 - 文字コードによる符号化
 - 16 進法

方法2: 負の数からビットパターンを求める方法 (左列から右列)

b : ビットパターン. 長さを固定したときには, 左側を0で埋める.

$s(b)$: b の表す符号付き整数

x の2の補数 $b = \text{Comp}_n(x)$ の定義

- ① x のすべてのビットを, ビットごとに (bitwisely) 反転する.

$$\bar{x} = \overline{X_{n-1}} \cdots \overline{X_0}$$

- ② \bar{x} が普通の2進法の整数だと思って $1_{(2)}$ を加えて得られた整数の2進法表示が $b = \text{Comp}_n(x)$.

「2の補数をとる」 = (-1) 倍

$$s(\text{Comp}_n(b)) = -s(b).$$

$m =$	b_1	$-m =$	b_2
$s(b_1)$		$s(b_2)$	

筆算してみよう

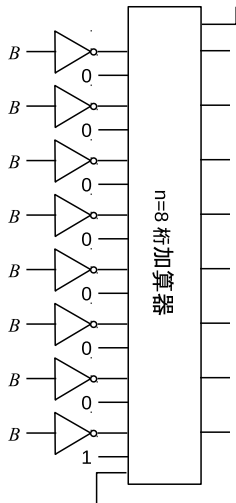
+)						
<hr/>							

+)						
<hr/>							

ここまで来たよ

- 1 復習:負の整数の2進法表現・2の補数
 - 2の補数
 - 演算装置:減算器
 - オーバーフロー
- 2 文字の表現
 - ビット長と場合の数
 - 文字コードによる符号化
 - 16進法

(-1)倍する論理回路

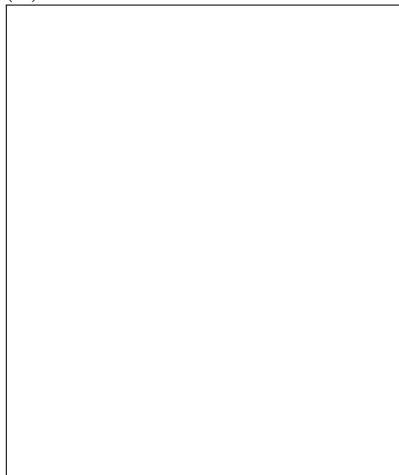


L03 加算器

減算器

(-1) 器と加算器を流用しちゃえ!

$$s(a) - s(b) = s(x)$$

 A_{n-1} \vdots A_1 A_0 B_{n-1} \vdots B_1 B_0 X_{n-1} \vdots X_1 X_0

ここまで来たよ

- 1 復習:負の整数の2進法表現・2の補数
 - 2の補数
 - 演算装置:減算器
 - オーバーフロー
- 2 文字の表現
 - ビット長と場合の数
 - 文字コードによる符号化
 - 16進法

オーバーフロー

演算の結果が

ことによる計算の「誤り」の

こと。
例. $n = 8$.

$127 + 1 = ?$

+)						
<hr/>							

$127 + 127 = ?$

+)						
<hr/>							

```
#include <stdio.h>

int main(){
    int a,b;

    printf("int型変数は%luビット\n",sizeof(int)*8);

    printf("a?\n");
    scanf("%d",&a);

    printf("b?\n");
    scanf("%d",&b);

    printf("a+b=%d\n",a+b);
    return 0;
}
```

こういう入出力に対してこうなる

ここまで来たよ

- 1 復習:負の整数の2進法表現・2の補数
 - 2の補数
 - 演算装置:減算器
 - オーバーフロー
- 2 文字の表現
 - ビット長と場合の数
 - 文字コードによる符号化
 - 16進法

ビット長と場合の数

- n ビットのビットパターンで, $m = 2^n$ 通りの場合を区別できる
- $\log_2 m$ ビットのビットパターンで, m 通りの場合を区別できる

何ビットあれば区別できる?

英語のアルファベットは $26 \times 2 + (\text{数字}) + (\text{記号})$.

日本語の漢字 (JIS 第 1 水準) は 3000.

⋮

L05-Q1

Quiz(int 型変数のビット長)

あるコンピュータのCの(符号付き)int 型変数のビット長は16ビットであるという. このint 型変数で表現できる整数の範囲を求めよう. 32ビット, 64ビットの場合は? 符号なしの整数については?

ここまで来たよ

- 1 復習:負の整数の2進法表現・2の補数
 - 2の補数
 - 演算装置:減算器
 - オーバーフロー
- 2 文字の表現
 - ビット長と場合の数
 - 文字コードによる符号化
 - 16進法

コンピュータ上での文字の表現

- 文字: ここでは色サイズフォントなどは考えない
- 文字列: 文字のならば, テキスト

個々の文字に一定の長さのビットパターン=正の整数 (文字コード) を対応させる.

文字 (列) $\xrightarrow{\text{符号化}}$ 整数 (列), ビットパターン
 $\xleftarrow{\text{復号化}}$

ASCII コード

ASCII=American Standard Code for Information Interchange

世の中で支配的な文字コード (週刊アスキーの語源)

文字の種類

- 0-31,127 制御文字 印刷方法通信方法の指示
 - ▶ 例: LF=改行, DEL=削除, 過去の遺物的なもの
- 32. SP=空白=スペース
- 33- 印刷可能文字
 - ▶ 記号
 - ▶ 数字
 - ▶ 英字
- 128- ASCII 拡張コードの印刷可能文字. 追加の記号や, é, ü, ç, æ.

正確には, 7ビットの 0-127 が ASCII コード (左端ビットが 0 なので 7 ビットコードとみなされる), 128-255 は ASCII 拡張コード.

左端ビットは, もともとの ASCII コードは , 拡張コードは .

ASCIIコード表 (0-127)

10進	16進	文字										
0	00	NUL	16	10	DLE	32	20	SP	48	30	0	
1	01	SOH	17	11	DC1	33	21	!	49	31	1	
2	02	STX	18	12	DC2	34	22	"	50	32	2	
3	03	ETX	19	13	DC3	35	23	#	51	33	3	
4	04	EOT	20	14	DC4	36	24	\$	52	34	4	
5	05	ENQ	21	15	NAK	37	25	%	53	35	5	
6	06	ACK	22	16	SYN	38	26	&	54	36	6	
7	07	BEL	23	17	ETB	39	27	'	55	37	7	
8	08	BS	24	18	CAN	40	28	(56	38	8	
9	09	HT	25	19	EM	41	29)	57	39	9	
10	0A	LF	26	1A	SUB	42	2A	*	58	3A	:	
11	0B	VT	27	1B	ESC	43	2B	+	59	3B	;	
12	0C	FF	28	1C	FS	44	2C	,	60	3C	<	
13	0D	CR	29	1D	GS	45	2D	-	61	3D	=	
14	0E	SO	30	1E	RS	46	2E	.	62	3E	>	
15	0F	SI	31	1F	US	47	2F	/	63	3F	?	

64	40	@	80	50	P	96	60	'	112	70	p
65	41	A	81	51	Q	97	61	a	113	71	q
66	42	B	82	52	R	98	62	b	114	72	r
67	43	C	83	53	S	99	63	c	115	73	s
68	44	D	84	54	T	100	64	d	116	74	t
69	45	E	85	55	U	101	65	e	117	75	u
70	46	F	86	56	V	102	66	f	118	76	v
71	47	G	87	57	W	103	67	g	119	77	w
72	48	H	88	58	X	104	68	h	120	78	x
73	49	I	89	59	Y	105	69	i	121	79	y
74	4A	J	90	5A	Z	106	6A	j	122	7A	z
75	4B	K	91	5B	[107	6B	k	123	7B	{
76	4C	L	92	5C	\	108	6C	l	124	7C	
77	4D	M	93	5D]	109	6D	m	125	7D	}
78	4E	N	94	5E	~	110	6E	n	126	7E	~
79	4F	O	95	5F	_	111	6F	o	127	7F	DEL

証拠

```
#include <stdio.h>

int main(){
    int x;

    printf("%d\n", 'a');

    printf("%c\n", 97);

    scanf("%d",&x);

    printf("%c\n", x);

    return 0;
}
```

文字列の符号化

文字列の例: “Adelie Penguin!”

文字	A	d	e	l	i	...
文字コード	65	100	101	108	105	...
ビットパターン	01000001	01100100	01100101	01101100	01101001	...

e		P	e	n	...	n	!
101	32	80	101	110	...	110	33
01100101	00100000	01010000	01100101	01101110	...	01101110	00100001

Adelie Penguin! ↔ 010000010110010001100101.....0110111000100001

(ビット)

日本語の文字コード

JIS 第1水準で 2965 個, 第2水準で 3390 個, これ以外に漢字以外の文字もある. これらを 8 ビットで表すことはとてもできない. 16 ビット ($2^{16} = 65536$) で記述するのが普通.

ASCII コードを拡張するように定められている. 何種類かあって統一されていない.

Web ブラウザで 表示 > エンコーディング

- JIS コード ISO-2022-JP
- EUC コード
- ShiftJIS コード CP932 MS-KANJI
- UTF-8

文字化け (上品な言葉ではない)

ある文字コードで符号化されたものを, 別の文字コードだとして復号化すると, もちろん最初とは異なった文字列になる

L05-Q2

Quiz(文字コード)

- ① アルファベット 3 文字の秘密の単語 w を決めよう (意味のある英単語でなくてもよい)
- ② ビット長 8 ビットの ASCII コードで, ビットパターンに符号化しよう. 長さ 24 の 01 の列 b になるはず.
- ③ b をチーム内の人への配布資料の余白に書いてに伝えよう.
- ④ チーム内の人から書いてくれた b を, 単語 w に戻そう.
- ⑤ チーム内の人から求めた w が正しいか, 配布資料を受け取ってチェックしよう. チェック責任者の学籍番号と名前を書こう.

ここまで来たよ

- 1 復習:負の整数の2進法表現・2の補数
 - 2の補数
 - 演算装置:減算器
 - オーバーフロー
- 2 文字の表現
 - ビット長と場合の数
 - 文字コードによる符号化
 - 16進法

L05-Q3

Quiz(16 進 10 進変換)

- ① 10 進法で書かれた次の整数をそれぞれ 16 進法で表そう.

11, 16, 50, 100, 1000

- ② 16 進法で書かれた次の整数をそれぞれ 10 進法で表そう.

A, 11, 16, 100, 1AB

2 進法と 16 進法の変換

もちろん 10 進法を経由して変換すればいいが、もっと楽な方法がある。

2 進法と 16 進法の変換

16 進法の 1 桁と、2 進法の 4 桁との間で変換すればいい。

例



なぜなら $16 = 2^4$.

$$H_1 H_{0(16)} = 2^{4 \times 1} \times H_1 + 2^{4 \times 0} H_0$$

$$B_7 B_6 B_5 B_4 B_3 B_2 B_1 B_{0(2)} = \sum_i 2^i B_i =$$

$$2^4 \underbrace{(2^3 B_7 + 2^2 B_6 + 2^1 B_5 + 2^0 B_4)}_{H_1} + 2^0 \underbrace{(2^3 B_3 + 2^2 B_2 + 2^1 B_1 + 2^0 B_0)}_{H_0}.$$

L05-Q4

Quiz(16 進 2 進変換)

- ① 2 進法で書かれた次の整数をそれぞれ 16 進法で表そう. 10110011, 1001000.
- ② 16 進法で書かれた次の整数をそれぞれ 2 進法で表そう. $4f$, 101.

来週はこんな非参照 Quiz

L05-Q5

Quiz(文字の個数と文字コードのビット長)

中国の漢字は、数え方によるが20万字程度あるらしい。これらすべてを表す文字コードは、1文字あたりのビット長として最低どれだけ必要になるか答えよう。

L05-Q6

Quiz(文字の個数と文字コードのビット長)

日本語の文庫本1冊のテキストを符号化するとおよそ何ビットになるか。図や絵は考えない。

情報処理技術者試験

<https://www.jitec.ipa.go.jp/>

- IT パスポート いつでも
- 基本情報技術者 3年4月, 3年10月, 4年4月
- 応用情報技術者, …

4年4月までに基本情報技術者に合格するとよいですね。
IT系企業では取得プレッシャーがあることも。

- この科目は前半が午前基礎理論, 後半が午前コンピュータシステム, 技術要素。
- 計算機基礎実習 II は午後。

過去問題集 <https://moodle.media.ryukoku.ac.jp/> → 基本情報…

連絡

- 次回も 7-002 講義室. 座席指定あり.
- 最初のころはいろいろ変更あるかも. メールに注意. 実習室のときはいちおうイヤフォン持ってきて.
- 配布資料は 1-503 向かいの引出, <http://hig3.net> で再配布しています.
- Quiz の略解は <http://hig3.net> で配布しています.
- 予習問題, 成績や略解は <http://hig3.net> → RaMMoodle から
- 大注意: 一度解答して, 再度解答を開始して, そのままブラウザ閉じると, 白紙答案 0 点になります.
- 非参照非相談テストの答案や成績や略解は <http://hig3.net> → RaMMoodle から

Schedule

- プチテスト 2014-11-12 水 3. 非参照. 30 ピーナッツ.
- 予習問題 月 23:59 まで.
- 樋口のオフィスアワー 木 6(1-539), 金昼 (7-002/1-502).
- 2014-10-01 水昼 からチューターやっています (1-614).
- 2014-10-28 火 4, (29 水 14:00-17:00) 数理情報学科特別講義.
- 2014-11-06 木 数学検定団体受検申込締切.
<http://www.math.ryukoku.ac.jp/suken/> で受付中.
- 2014-12-06 土 34 数学検定団体受検.