Chương 02

BIỂU DIỄN THÔNG TIN TRONG MÁY TÍNH

Nội dung

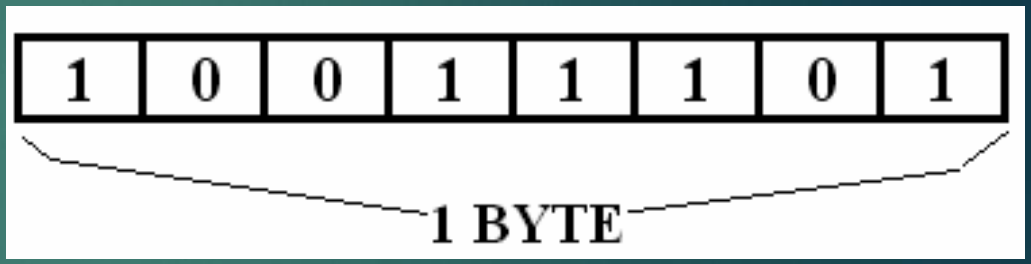
A. Cách biểu diễn thông tin
B. Biểu diễn số nguyên
C. Biểu diễn số thực
D. Biểu diễn ký tự

A. Cách biểu diễn thông tin

- Thông tin trong máy tính được biểu diễn dạng nhị phân
- Ví dụ:
  - 5 bit biểu diễn được 32 trạng thái.
  - 5 bit có thể dùng để biểu diễn 26 chữ cái A..Z.

1	1	0	0	0	0	0	1	1	0
1	0	0	0	0	0	1	1	1	1
0	1	0	1	1	1	0	0	0	0
0	0	1	1	0	1	1	0	0	1
1	1	1	0	1	1	0	1	0	0
0	1	0	1	1	0	0	0	0	0
1	1	1	0	0	1	0	1	0	0
1	1	1	0	1	1	0	1	0	0
0	1	1	0	1	0	1	1	0	1
1	1	0	1	0	1	1	0	0	1

- Đơn vị thông tin
  - BIT
    - Chỉ nhận giá trị 0 hoặc 1
  - 1Byte = 8 BIT
  - 1KB = 2^{10} Bytes
    = 1024 Bytes
  - 1MB = 1024 KB
  - 1GB = 1024 MB
  - …



B. Biểu diễn số nguyên

I. Biểu diễn số nguyên không dấu
II. Biểu diễn số nguyên có dấu

I. Số nguyên không dấu

1. Nguyên tắc tổng quát
2. Ví dụ
3. Biểu diễn số nguyên không dấu 8 bit

1. Nguyên tắc tổng quát

- Dùng n bit biểu diễn số nguyên không dấu A:

  $a_{n-1}a_{n-2}\ldots.a_2a_1a_0$

- Giá trị của A được tính như sau:

$$A = \sum_{i=0}^{n-1} a_i * 2^i$$

- Dải biểu diễn của A: $0 \div 2^{n-1}$
  - Số 8 bit có giá trị : 0 ÷ 255
  - Số 16 bit có giá trị : 0 ÷ 65 535
  - Số 32 bit có giá trị : 0 ÷ 4 294 967 295

2. Ví dụ

- Biểu diễn các số nguyên không dấu sau đây bằng 8-bit:
  A = 41 ; B = 150
- Giải:

A = 41 = 32 + 8 + 1 = $2^5 + 2^3 + 2^0$

41 = 0010 1001

B = 150 = 128 + 16 + 4 + 2

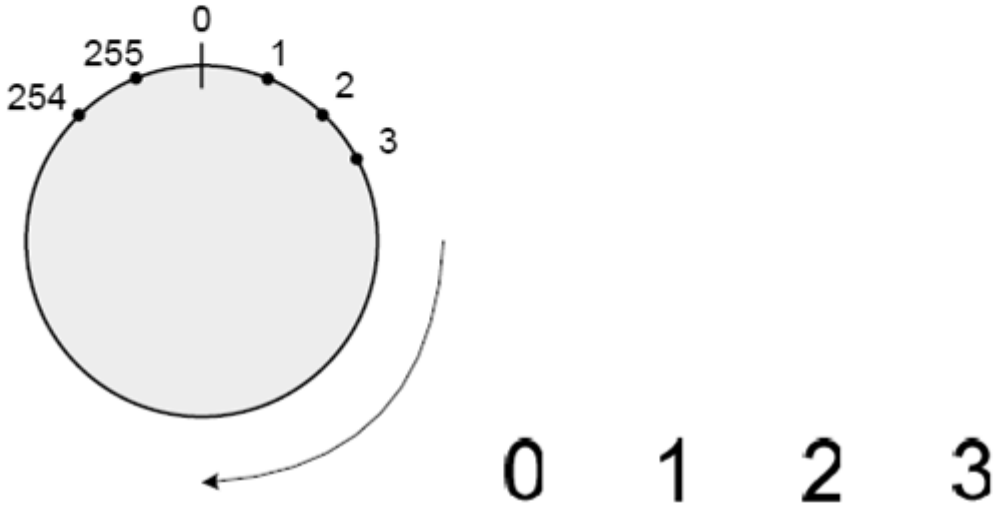
= $2^7 + 2^4 + 2^2 + 2^1$

150 = 1001 0110

3. Biểu diễn số nguyên không dấu 8 bit

- Trục số học:

- Trục số học máy tính:



- Biểu diễn số nguyên không dấu 8 bit: 0 đến 255
  - 0000 0000 = 0
  - 0000 0001 = 1
  - 0000 0010 = 2
  - 0000 0011 = 3
  - …
  - 1111 1111 = 255

Chú ý:

1111 1111
+ 0000 0001
1 0000 0000

Vậy: 255 + 1 = 0?

do tràn nhớ ra ngoài

II. Số nguyên có dấu

1. Nguyên tắc tổng quát
2. Ví dụ
3. Biểu diễn số nguyên có dấu 8 bit

1. Nguyên tắc tổng quát

- Dùng n bit biểu diễn số nguyên có dấu A:

  $a_{n-1}a_{n-2}\ldots a_2a_1a_0$

- Với A là số dương: bit a_{n-1} = 0, các bit còn lại biểu diễn độ lớn như số không dấu
- Với A là số âm: được biểu diễn bằng số bù hai của số dương tương ứng

- Giá trị của A được xác định như sau:

$$A = -a_{n-1}2^{n-1} + \sum_{i=0}^{n-2} a_i * 2^i$$

- Dải biểu diễn: $-2^{n-1} \div 2^{n-1}-1$
  - Số 8 bit có dấu có giá trị : -128 ÷ +127
  - Số 16 bit có dấu có giá trị : -32768 ÷ +32767

2. Ví dụ

- Biểu diễn các số nguyên có dấu sau đây bằng 8-bit:
  A = +58 ; B = -80
- Bài giải
  A = +58
  = 32 + 16 + 8 + 2
  = $2^6 + 2^5 + 2^4 + 2^1$
  = 0011 1010

B = -80

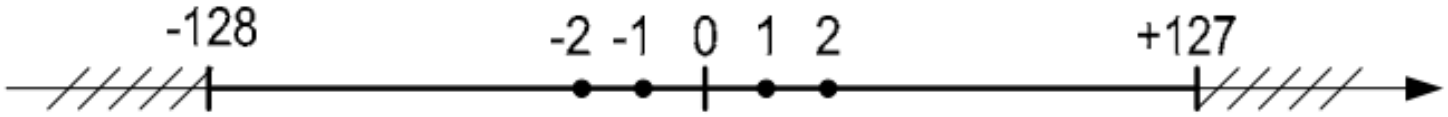
Ta có + 80 = 64 + 16
= $2^7 + 2^5$
= 0101 0000

Số bù một = 1010 1111
\+ 1

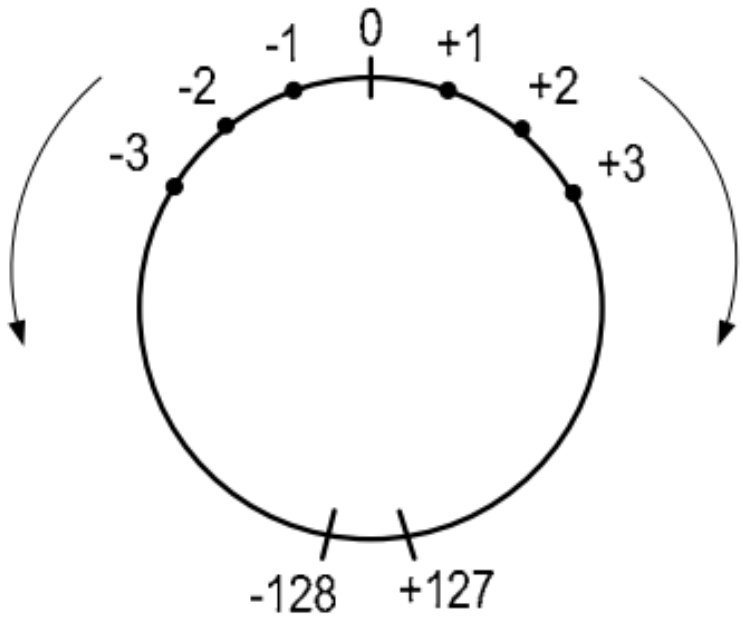
Số bù hai = 1011 0000

B = -80 = 1011 0000

- Trục số học:



- Trục số học máy tính:



3. Biểu diễn số nguyên có dấu 8 bit

- Biểu diễn số nguyên có dấu 8 bit: -128 đến +127

0000 0000 = 0

0000 0001 = +1

0000 0010 = +2

0000 0011 = +3

…

0111 1111 = +127

1000 0000 = - 128

1000 0001 = - 127

…

1111 1110 = -2

1111 1111 = -1

Chú ý:

+127 + 1 = -128

-128 - 1 = +127

do tràn xảy ra

C. Biểu diễn số thực

I. Biểu diễn số thập phân theo mã BCD (Binary Coded Decimal)
II. Biểu diễn số dấu chấm động

I. Biểu diễn số thập phân theo mã BCD

1. Nguyên tắc tổng quát
2. Ví dụ
3. Các kiểu lưu trữ

1. Nguyên tắc tổng quát

- Dùng để biểu diễn một cách chính xác số thập phân (không làm tròn số)
- Dùng 4 bit để mã hóa các chữ số từ 0 đến 9

Số thập phân	Số nhị phân	Số thập phân	Số nhị phân
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

- Còn 6 tổ hợp chưa sử dụng:
  1010, 1011, 1100, 1101, 1110, 1111

2. Ví dụ

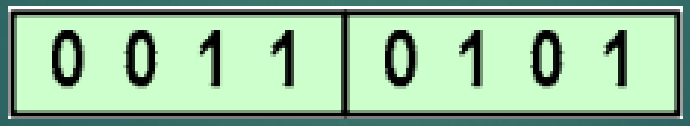
- 35 = $0011\ 0101_{(BCD)}$
- 61 = $0110\ 0001_{(BCD)}$
- 35.61= $0011\ 0101.0110\ 0001_{(BCD)}$

- 10.87 = $0001\ 0000.1000\ 0111_{(BCD)}$
- 96.40 = $1001\ 0110.0100\ 0000_{(BCD)}$

3. Các kiểu lưu trữ

- BCD không nén (Unpacked BCD):Mỗi số BCD4-bit được lưu trữ trong 4-bit thấp của mỗi byte.
  - Ví dụ: Số 35 được lưu trữ như sau:

- BCD nén (Packed BCD): Hai số BCD được lưu trữ trong 1 byte.
  - Ví dụ: Số 35 được lưu trữ như sau:



II. Biểu diễn số dấu chấm động

1. Nguyên tắc chung
2. Chuẩn IEEE754

1. Nguyên tắc chung

- Một số thực X được biểu diễn theo kiểu số dấu phẩy động như sau:

$$X = (-1)^S M * R^E$$

  - S: dấu
  - M: định trị (Mantissa)
  - R: cơ số (Radix)
  - E: phần mũ (Exponent)
- Ví dụ: $2009 = (-1)^0 * 2.009 * 10^3$

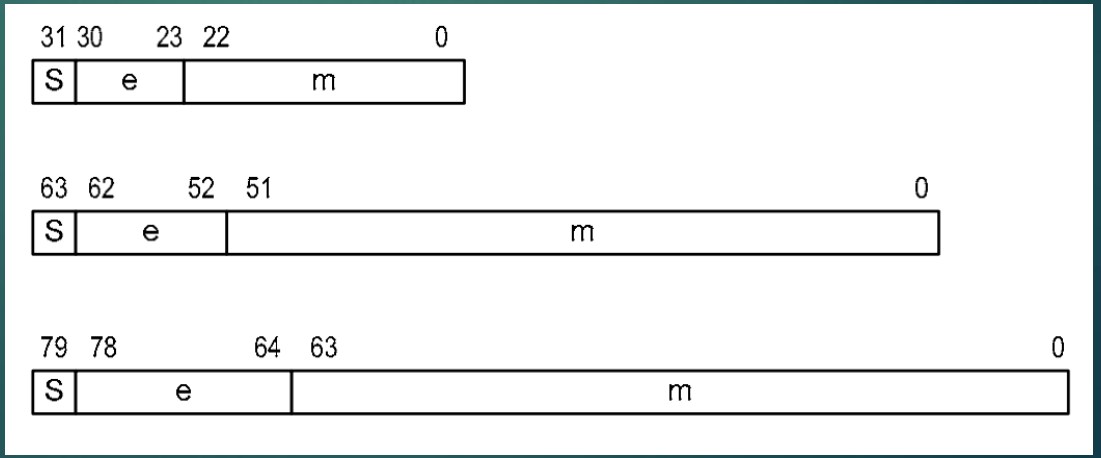
2. Chuẩn IEEE754

a. Tổng quan
b. Dạng 32 bit
   - Ví dụ
   - Các quy ước đặc biệt
   - Dải biểu diễn

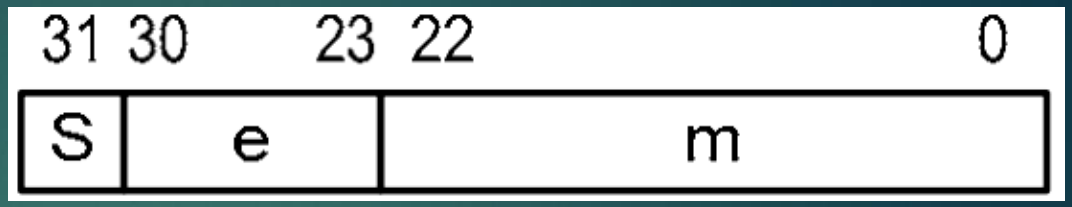
a. Tổng quan

Được sử dụng rộng rãi trong khoa học máy tính hiện nay

- Dùng 1 bit cho phần dấu: 0-dương, 1-âm
- Cơ số R=2
- Các dạng biểu diễn chính:
  - 32 bit
  - 64 bit
  - 80 bit



b. Dạng 32 bit



- S: bit dấu
- e (8 bit): là mã excess-127 của phần mũ E
  - e = E + 127
  - Giá trị 127 được gọi là độ lệch (bias)
- m (23 bit) là phần lẻ của phần định trị M
  - M=1.m
- Công thức xác định số thực

$$X = (-1)^S M * R^E = \mathbf{(-1)^S 1.m * 2^{e-127}}$$

Ví dụ

Xác định giá trị của số thực được biểu diễn bằng 32-bit như sau:
1100 0001 0101 0110 0000 0000 0000 0000

- S = 1 số âm
- e = $1000\ 0010_{(2)}$ = 130 E = 130-127=3
- Vậy X = -1.10101100 * 2^3 = -1101.011 = -13.375

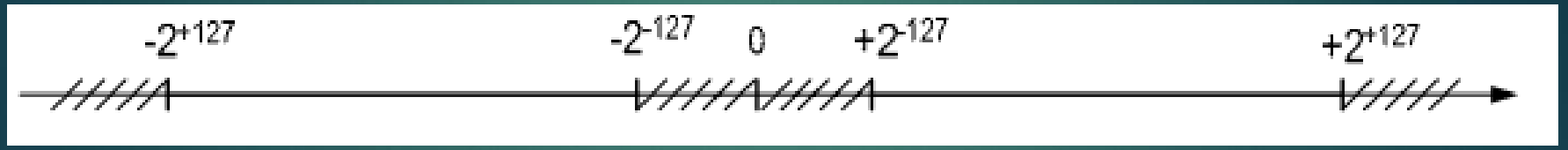
- 0011 1111 1000 0000 0000 0000 0000 0000 = ?
  = +1.0

Các quy ước đặc biệt

- Các bit của e bằng 0, các bit của m bằng 0
  x000 0000 0000 0000 0000 0000 0000 0000
  thì X = ± 0
- Các bit của e bằng 1, các bit của m bằng 0
  x111 1111 1000 0000 0000 0000 0000 0000
  thì X = ±∞
- Các bit của e bằng 1, còn m có ít nhất một bit bằng 1, thì nó không biểu diễn cho số nào cả (NaN - not a number)

Dải giá trị biểu diễn

- 2^{-127} đến 2^{+127}
- 10^{-38} đến 10^{+38}

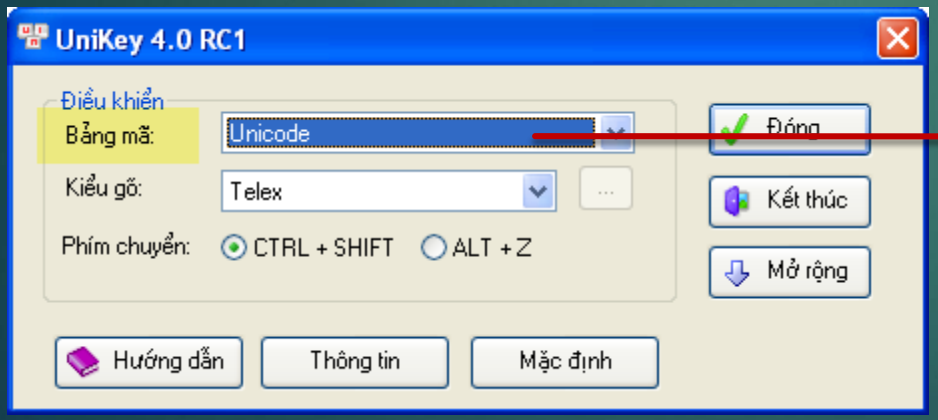


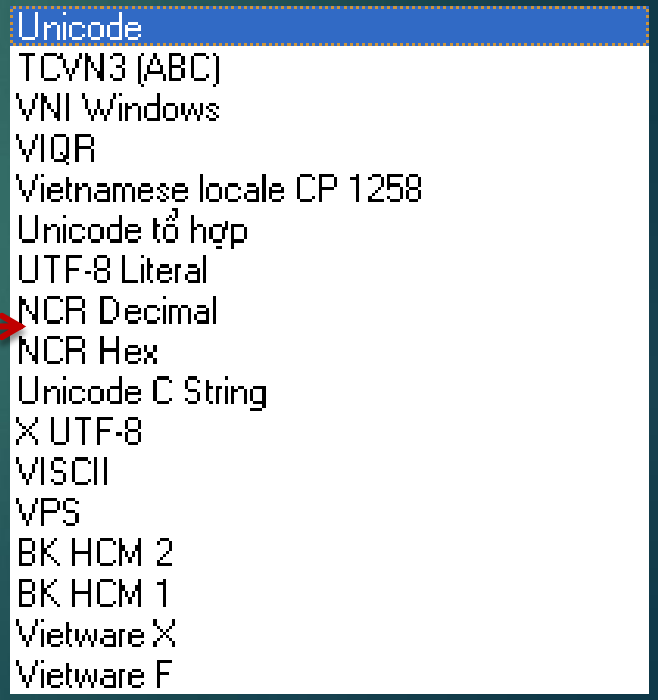
D. Biểu diễn ký tự theo hệ nhị phân

I. Bộ mã ASCII
(American Standard Code for Information Interchange)

II. Bộ mã ANSI
(American National Standard Institute)

III. Bộ mã Unicode





I. Bộ mã ASCII

- Bộ mã 8 bit, có thể mã hóa được 2^8(256)
- Được sử dụng ở trên hệ điều hành Windows/DOS và Unix

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

128 ký tự chuẩn:

$00_{(16)}$ – $1F_{(16)}$: Ký tự điều khiển

$20_{(16)}$ – $7F_{(16)}$: Ký số, ký tự tiếng Anh, ký tự đặc biệt và thông dụng (+,-,*,/, %, …)

- 128 ký tự mở rộng ($80_{(16)}$ – $FF_{(16)}$)
  - Bộ mã ký tự mở rộng của IBM: IBM-PC
  - Bộ mã ký tự mở rộng của Apple: Macintosh
  - Bộ mã tiếng việt TCVN3, VNI, …

II. Bộ mã ANSI

- Tên khác: ISO-8859-1, LATIN-1
- Mã hóa 8 bit
- Là bảng mã mở rộng của ASCII
  - 128 ký tự đầu giống như bảng mã ASCII
  - Có thể mã hóa các ngôn ngữ khác bên cạnh tiếng Anh
  - Do chỉ có tối đa 256 mã nên chưa mã hóa được các ký tự ngôn ngữ của Trung Quốc, Ả Rập, Do Thái, …

III. Bộ mã Unicode

- Được thiết kế để dùng làm bộ mã duy nhất cho tất cả các ngôn ngữ khác nhau trên thế giới
- Các bảng mã UTF (Unicode Transformation Format)
  - UTF-8 : mã hóa 1 đến 4 byte
  - UTF-16: mã hóa 2 đến 4 byte
  - UTF-32: mã hóa 4 byte

Bảng mã UTF-8

- Được thiết kế tương thích với chuẩn ASCII
- Được ưu tiên sử dụng mã hóa cho email, web,… (được các trình duyệt web như Netscape, Mozilla, Internet Explorer, Opera và Safari hỗ trợ)
- Là bảng mã mặc định cho XML

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<meta http-equiv='Content-Type' content='text/html; charset=UTF-8' />
```

Phụ lục: Bảng mã ACSII

00	NUL	10	DLE	20	SP	30	0	40	@	50	P	60	`	70	p
01	SOH	11	DC1	21	!	31	1	41	A	51	Q	61	a	71	q
02	STX	12	DC2	22	"	32	2	42	B	52	R	62	b	72	r
03	ETX	13	DC3	23	#	33	3	43	C	53	S	63	c	73	s
04	EOT	14	DC4	24	$	34	4	44	D	54	T	64	d	74	t
05	ENQ	15	NAK	25	%	35	5	45	E	55	U	65	e	75	u
06	ACK	16	SYN	26	&	36	6	46	F	56	V	66	f	76	v
07	BEL	17	ETB	27	'	37	7	47	G	57	W	67	g	77	w
08	BS	18	CAN	28	(	38	8	48	H	58	X	68	h	78	x
09	HT	19	EM	29	)	39	9	49	I	59	Y	69	i	79	y
0A	LF	1A	SUB	2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
0B	VT	1B	ESC	2B	+	3B	;	4B	K	5B	[	6B	k	7B	{
0C	FF	1C	FS	2C	´	3C	<	4C	L	5C	\	6C	l	7C	|
0D	CR	1D	GS	2D	-	3D	=	4D	M	5D	]	6D	m	7D	}
0E	SO	1E	RS	2E	.	3E	>	4E	N	5E	^	6E	n	7E	~
0F	SI	1F	US	2F	/	3F	?	4F	O	5F	_	6F	o	7F	DEL

NUL	Null	FF	Form feed	CAN	Cancel
SOH	Start of heading	CR	Carriage return	EM	End of medium
STX	Start of text	SO	Shift out	SUB	Substitute
ETX	End of text	SI	Shift in	ESC	Escape
EOT	End of transmission	DLE	Data link escape	FS	File separator
ENQ	Enquiry	DC1	Device control 1	GS	Group separator
ACK	Acknowledge	DC2	Device control 2	RS	Record separator
BEL	Bell	DC3	Device control 3	US	Unit separator
BS	Backspace	DC4	Device control 4	SP	Space
HT	Horizontal tab	NAK	Negative acknowledge	DEL	Delete
LF	Line feed	SYN	Synchronous idle		
VT	Vertical tab	ETB	End of transmission block		