Codes

Number codes

Digital circuits (like processors) represent information with two-valued “binary” codes. Why?

Every signal wire in a digital system can transport one binary digit, or bit.

A signal at or near system ground is said to be transporting a ‘0’ bit. A signal at or near Vdd, a ‘1’ bit. All information is represented using one or more signals that can transport a ‘0’ or ‘1’.

A collection of related signals that are logically considered as a single data element is called a bus. An “n-bit” bus can transport n bits representing an n-bit binary number. An n-bit number can have one of 2^n different values.

A 4-bit number is typically called a ***nibble***; b bits, a ***byte***; 16 bits, a ***word***, and 32 bits, a ***double word***. Since 32-bit processors are now the most typical, a 32-bit value is called a word as well. (The term “word” is contextual, and typically refers to the bus size of the system).

Memory system size is most often referred to in terms of bytes.

1 byte has 8 bits, and can represent 2^8 or 256 different numbers (0 – 255). Larger numbers need more bits.

Number codes

Common number bases include base 2 (binary numbers), base 10 (decimal numbers), and base 16 (hexadecimal numbers).

0 0 0 0	0
0 0 0 1	1
0 0 1 0	2
0 0 1 1	3
0 1 0 0	4
0 1 0 1	5
0 1 1 0	6
0 1 1 1	7
1 0 0 0	8
1 0 0 1	9
1 0 1 0	A
1 0 1 1	B
1 1 0 0	C
1 1 0 1	D
1 1 1 0	E
1 1 1 1	F

Base 2

Base 10

Base 16

Base 2 and base 16 numbers both directly show signal/bit values. Base 16 numbers are one-to-one replacements for 4-bit binary numbers. They are more compact and convenient than binary.

Number codes

Common number bases include base 2 (binary numbers), base 10 (decimal numbers), and base 16 (hexadecimal numbers).

0 0 0 0	0
0 0 0 1	1
0 0 1 0	2
0 0 1 1	3
0 1 0 0	4
0 1 0 1	5
0 1 1 0	6
0 1 1 1	7
1 0 0 0	8
1 0 0 1	9
1 0 1 0	A
1 0 1 1	B
1 1 0 0	C
1 1 0 1	D
1 1 1 0	E
1 1 1 1	F

Base 2

Base 10

Base 16

Base 2 and base 16 numbers both directly show signal/bit values. Base 16 numbers are one-to-one replacements for 4-bit binary numbers. They are more compact and convenient than binary.

Consider this number – which form is easier to use?

10101101010001100101011

Number codes

Common number bases include base 2 (binary numbers), base 10 (decimal numbers), and base 16 (hexadecimal numbers).

0 0 0 0	0
0 0 0 1	1
0 0 1 0	2
0 0 1 1	3
0 1 0 0	4
0 1 0 1	5
0 1 1 0	6
0 1 1 1	7
1 0 0 0	8
1 0 0 1	9
1 0 1 0	A
1 0 1 1	B
1 1 0 0	C
1 1 0 1	D
1 1 1 0	E
1 1 1 1	F

Base 2

Base 10

Base 16

Base 2 (binary) and base 16 (hex) numbers both directly show signal/bit values. Base 16 numbers are one-to-one replacements for 4-bit binary numbers. They are more compact and convenient than binary.

Consider this number – which form is easier to use?

1010 1101 0100 0110 0101 1011

A D 4 6 5 B

AD465B

Number systems use positional weighting. The value of a multi-digit number is the sum of the digit multiplied by the base raised to the power of the digit’s position. In the examples below, “d” and “h” are added to the numbers to make it clear which radix (base) the number us using. This is the case for any base.

Digit Base Position

$$342d = 3 \times 10^{2} + 4 \times 10^{1} + 2 \times 10^{0}$$

$$\begin{aligned} A2Fh &= A \times 16^{2} + 2 \times 16^{1} + F \times 16^{0} \\ &= 10 \times 256 + 2 \times 16 + 15 \times 1 \\ &= 2607d \end{aligned}$$

$$\begin{aligned} 10101b &= 1 \times 2^{4} + 0 \times 2^{3} + 1 \times 2^{2} + 0 \times 2^{1} + 1 \times 2^{0} \\ &= 1 \times 16 + 0 + 1 \times 4 + 0 + 1 \times 1 \\ &= 21d \end{aligned}$$

Converting from any base to base 10 is straight-forward as illustrated above. Converting from base 10 to any other base can follow a simple formula.

Modulo-divide the number to be converted by the desired base, and record the quotient and remainder. Then continue dividing the most recent quotient by the base and recording the remainders until the quotient is zero. The remainders are the digits of the number in the new base, with the last-produced remainder being the most significant digit. This works for any base.

	Quotients ↓		Remainders ↓		
2607	= 162	x 16	+ 15	(F)	
162	= 10	x 16	+ 2	(2)	= A2Fh
10	= 0	x 16	+ 10	(A)	

21	= 10	x 2	+ 1	
10	= 5	x 2	+ 0	
5	= 2	x 2	+ 1	=10101b
2	= 1	x 2	+ 0	
1	= 0	x 1	+ 1	

Other Number Codes

“Binary Coded Decimal” is a four-bit binary number code used to represent decimal numbers. Only the first 10 4-bit binary numbers are used (0000 through 1001); the other six binary numbers (1010 through 1111) are ignored.

Packed BCD places two BCD digits in one byte; unpacked BCD places one BCD digit in one byte.

A “Gray” code refers to a counting sequence where only one bit changes between consecutive numbers.

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

Physical digital circuits have fixed bus widths, and therefore only a finite set of numbers can be represented.

An 8-bit system has only 2^8 (256) codes available to represent numbers, and a 32 bits has 2^{32} (4 billion) codes available to represent numbers.

Larger numbers can use double-precision, or floating point.

Digital systems often represent negative numbers as well as positive numbers. How?

Two’s compliment

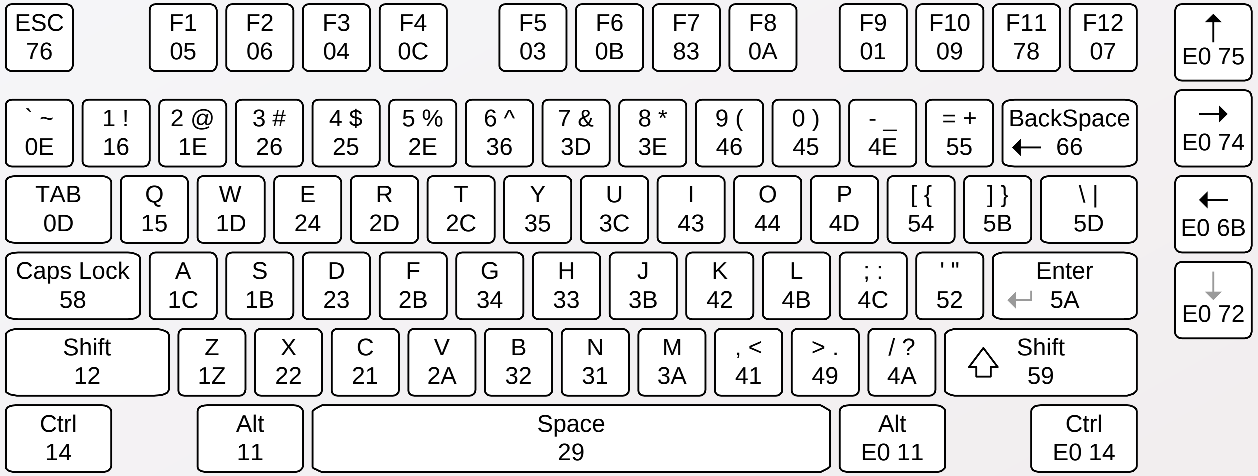
Overflow and Underflow

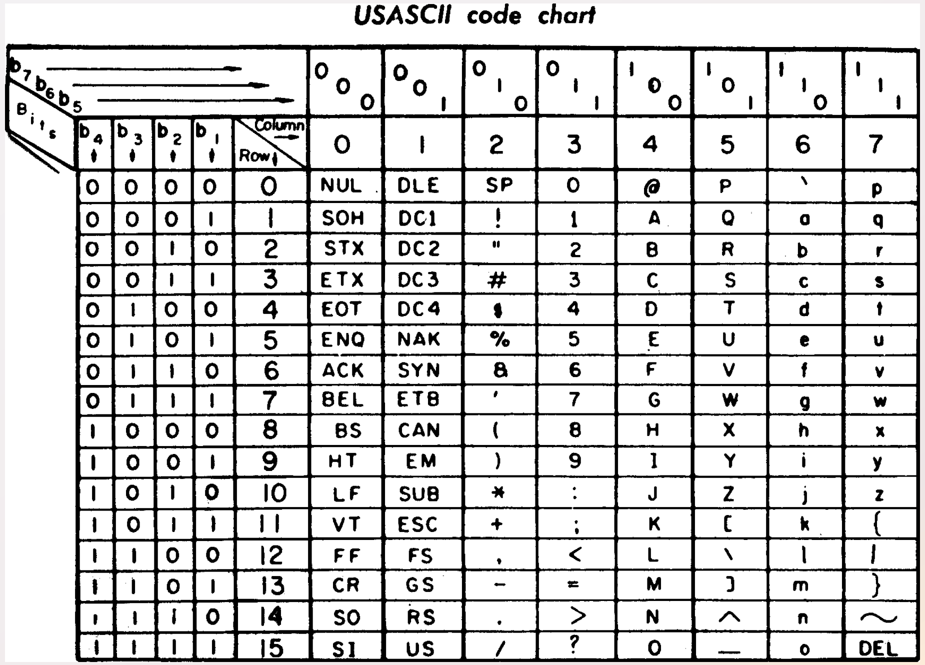
Character codes

Characters must also be represented. ASCII (American Standard Code for Information Interchange) is a 7-bit code that was developed in 1962 to define 128 alphanumeric characters used by teleprinters using American English. Later versions added an 8th bit and more characters, but these never enjoyed widespread use.

Unicode is an updated code that incorporates ASCII into its first 128 characters. Unicode currently defines more than 136,000 characters. UTF-8 is a subset used by many websites; it is an 8-bit code with ASCII as the first 128 characters.

Keyboards have their own unique “scan codes” that were developed to make the keyboard scanning circuits more efficient.





USASCII code chart

b_7 b_6 b_5 →					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b_4	b_3	b_2	b_1	Row ↓ / Column →	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	|
1	1	0	1	13	CR	GS	-	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	—	o	DEL