

Table E-1 (continued)
What a bit can represent

Context	Representing	0 =	1 =
Serial transfer	Data	0 value	1 value
Serial transfer	Parity	SPACE	MARK
Serial transfer	End		Stop bit(s)
Serial transfer	Communication state	BREAK	Carrier
P reg. bit N	Neg. result?	No	Yes
P reg. bit V	Overflow?	No	Yes
P reg. bit B	BRK command?	No	Yes
P reg. bit D	Decimal mode?	No	Yes
P reg. bit I	IRQ interrupts	Enabled	Disabled (masked out)
P reg. bit Z	Zero result?	No	Yes
P reg. bit C	Carry required?	No	Yes

* Sometimes ambiguously termed *reset*

Table E-2
Values represented by a nibble

Binary	Hex	Dec
0000	\$00	0
0001	\$01	1
0010	\$02	2
0011	\$03	3
0100	\$04	4
0101	\$05	5
0110	\$06	6
0111	\$07	7
1000	\$08	8
1001	\$09	9
1010	\$0A	10
1011	\$0B	11
1100	\$0C	12
1101	\$0D	13
1110	\$0E	14
1111	\$0F	15

- ☐ Bits can also be combined in groups of any size to represent numbers. Most of the commonly used sizes are multiples of four bits.
- ☐ Four bits are a **nibble** (sometimes spelled *nybble*).
- ☐ One nibble can represent any of 16 values. Each of these values is assigned a number from 0 through 9 and (because our decimal system has only 10 of the 16 digits we need) A through F.
- ☐ Eight bits (two nibbles) make a **byte** (Figure E-1 and Table E-2).
- ☐ One byte can represent any of 16 x 16 (or 256) values. The value can be specified by exactly two hexadecimal digits.

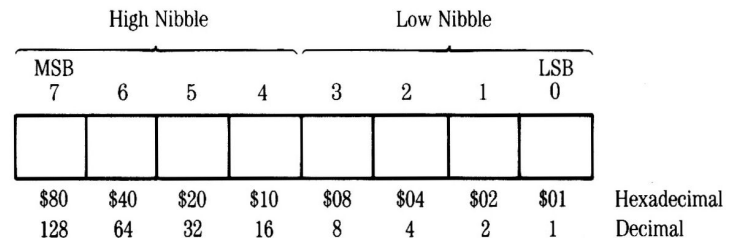


Figure E-1
Bits, nibbles, and bytes