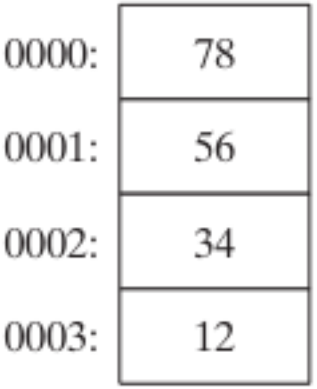
Little Endian Order

x86 processors use little-endian byte order to store and retrieve data from memory.

This means that the least significant byte (LSB) of a data item is stored at the lowest memory address, and the most significant byte (MSB) is stored at the highest memory address.



0000:	78
0001:	56
0002:	34
0003:	12

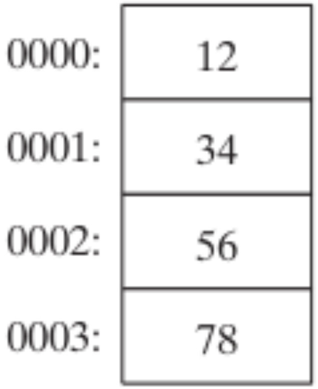
This is because the LSB is the least important byte in a doubleword, and the MSB is the most important byte. By storing the LSB first, x86 processors can access data more efficiently.

Little-endian byte order is also used by many other popular computing platforms, such as ARM and MIPS. This makes it easier to port software between different platforms.

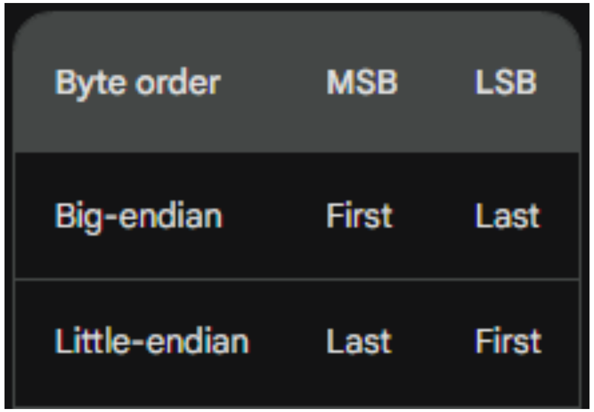
In big-endian byte order, the most significant byte (MSB) is stored at the lowest memory address, and the least significant byte (LSB) is stored at the highest memory address.

This is the opposite of little-endian byte order, which is used by x86 processors.

Figure 3-15 shows an example of 12345678h stored in big-endian order at offset 0:



0000:	12
0001:	34
0002:	56
0003:	78



Byte order	MSB	LSB
Big-endian	First	Last
Little-endian	Last	First