530128-16-1E AID:83578 | 03/17/2019

**Program Plan:**

The byte order is a characteristic of the processor architecture, dictating how bytes are ordered within larger data types, such as integers. If the processor architecture supports big-endian byte order, then the highest byte address occurs in the least significant byte (LSB). Little-endian byte order is the opposite the least significant byte contains the lowest byte address.

For example if we were to assign a 32-bit integer the value 0x04030201 and then to cast a character pointer (byte) to the address of the integer, we would see a difference from the byte ordering. On a little-endian processor, byte[0] contain 1; byte[3] contain 4 and a big-endian processor, where byte[0] contain 4; byte[3] contain 1.

**Program:**

/\*

\* This C program used to determine system's byte ordering.

\*/

**#include** <stdio.h>

**int** **main**()

{

**unsigned** **int** integer = 0x04030201;

**char** \*byte = (**char**\*) &integer;

**if** (1 == byte[0] && 4 == byte[3])

**printf**("Little-endian byte order");

**else**

**printf**("Big-endian byte order");

**return** 0;

}

**Sample Output:**

Little-endian byte order