|  |
| --- |
|  |

**DT081/4**

**Computer Architecture 3 Laboratory**

**Lab 1: Testing Endian-ness**

The endian-ness of an architecture refers to the order in which the individual bytes of a word are stored in memory. "Little endian" systems store bytes with the least-significant byte occurring first in the memory allotted to the word; "Big endian" systems do the reverse, storing the most-significant byte first.

For example, given a 32-bit word, represented in hexadecimal as ABCD1234, and stored in location 4030 onwards, the byte order in memory might look like:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | *BIG ENDIAN* | *LITTLE ENDIAN* | | |  |  | | --- | --- | | *Address* | *Byte* | | 4030 | AB | | 4031 | CD | | 4032 | 12 | | 4033 | 34 | | |  |  | | --- | --- | | *Address* | *Byte* | | 4030 | 34 | | 4031 | 12 | | 4032 | CD | | 4033 | AB | | |

Write a simple program in C or C++ to test the endian-ness of various computer architectures.

**Dr. R. Lynch**