Endianness

Prof. James L. Frankel
Harvard University
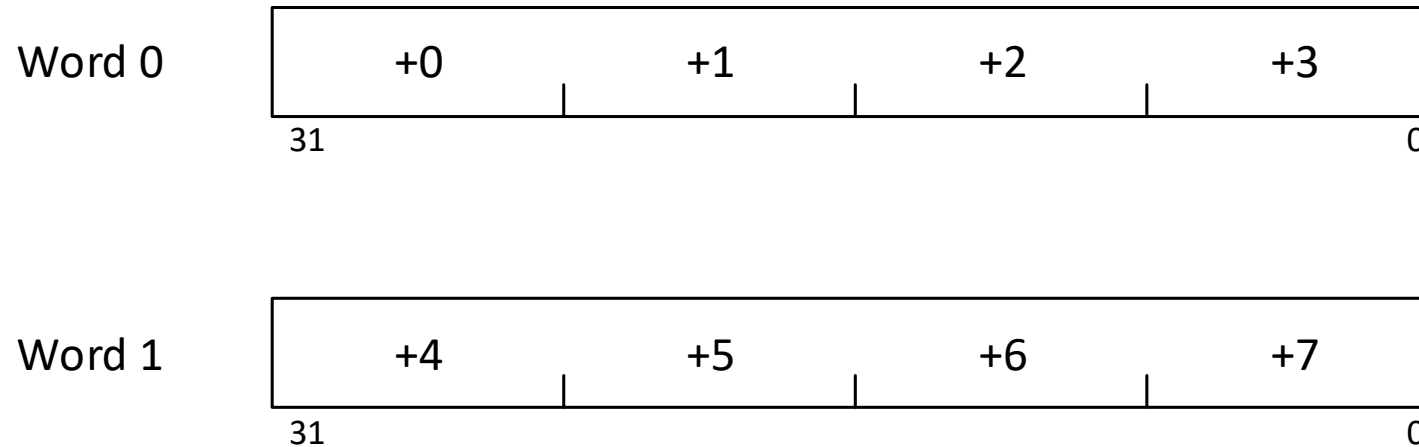
Endianness

- Order of bytes in word

- Big-endian
  - The MSB (Most Significant Byte) of a word is stored at the lowest memory address for that word
  - Subsequent bytes from MSB to LSB are stored in sequential addresses

- Little-endian
  - The LSB of a word is stored at the lowest memory address for that word
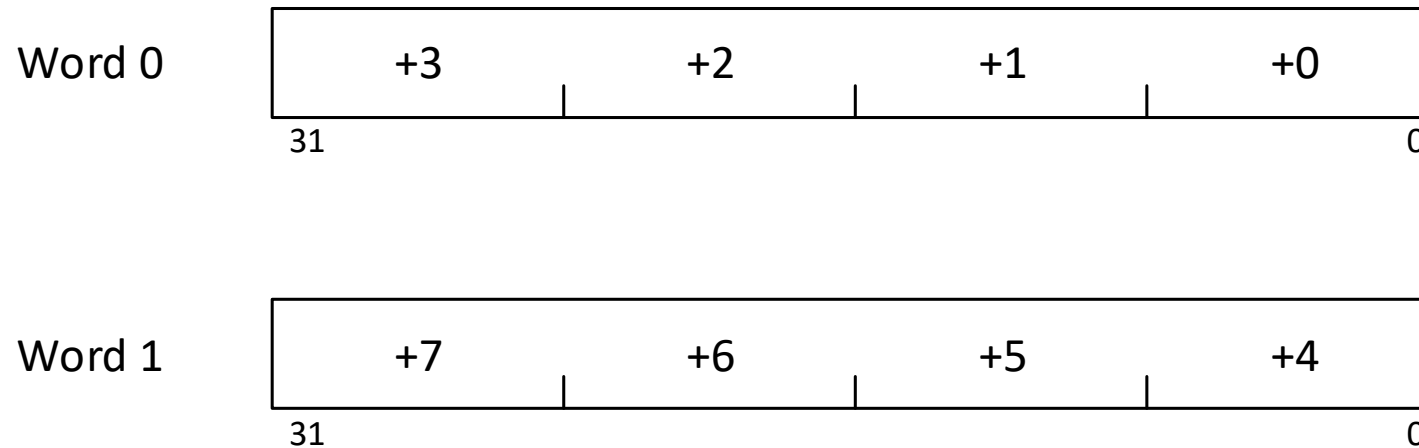 - Subsequent bytes from LSB to MSB are stored in sequential addresses

Big-Endian – 32-bit Word Example

Byte number offsets are memory address offsets

Word 0	+0	+1	+2	+3

31   0

Word 1	+4	+5	+6	+7

31   0

Little-Endian – 32-bit Word Example

Byte number offsets are memory address offsets

Word 0

	+3	+2	+1	+0	
31					0

Word 1

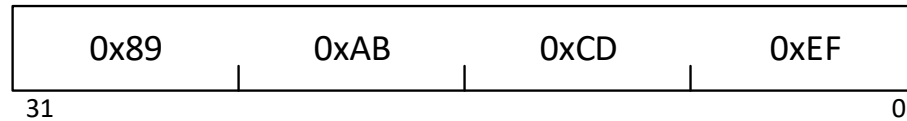|	+7	+6	+5	+4	
31					0

When Endianness is Visible

- Endianness becomes apparent when accessing memory as different sizes
  - For a 32-bit word size: as words, halfwords, and bytes

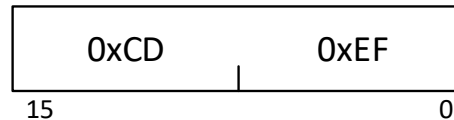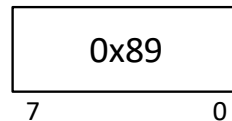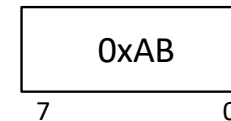# Big-Endian – 32-bit Word Different Accesses

Word at
Address 0x1000

| 0x89 | 0xAB | 0xCD | 0xEF |

31   0

Halfword at
Address 0x1000

| 0x89 | 0xAB |

15   0

Halfword at
Address 0x1002

| 0xCD | 0xEF |

15   0

Byte at
Address 0x1000

| 0x89 |

7   0

Byte at
Address 0x1001

| 0xAB |

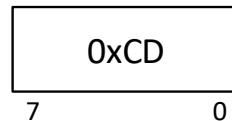7   0
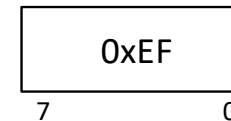
Byte at
Address 0x1002

| 0xCD |

7   0

Byte at
Address 0x1003

| 0xEF |

7   0

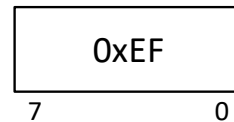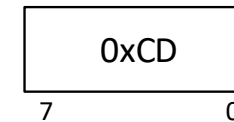# Little-Endian – 32-bit Word Different Accesses

Word at
Address 0x1000

0x89	0xAB	0xCD	0xEF

31   0

Halfword at
Address 0x1000

0xCD	0xEF

15   0

Halfword at
Address 0x1002

0x89	0xAB

15   0

Byte at
Address 0x1000

0xEF

7   0

Byte at
Address 0x1001

0xCD

7   0

Byte at
Address 0x1002

0xAB

7   0

Byte at
Address 0x1003

0x89

7   0
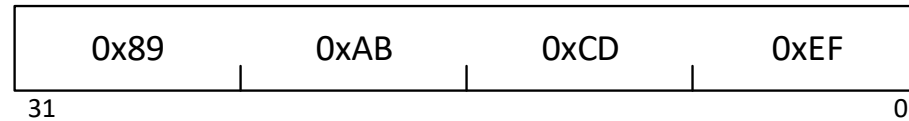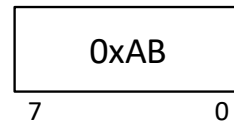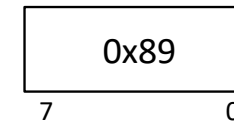
Examples of Endianness

- Big-endian computers
  - DEC PDP-10; IBM System/360, System/370, ESA/390, Series/1, z/Architecture; Intel 8051; Motorola 6800, 68k

- Little-endian computers
  - MOS Tech 6502; Intel 8008, MCS-48, x86; Zilog Z80; DEC PDP-11, VAX, Alpha; Altera Nios II

- Bi-endian computers
  - Sun SPARC (version 9 & later); ARM (version 3 & later); Apple/IBM/Motorola PowerPC; MIPS, HP PA-RISC, HP/Intel IA-64 (Itanium)

Advantages of Big- or Little-Endian

- Little-Endian
  - Bits, Bytes, Halfwords, and Words are all numbered from least to most significant
  - In a string with bytes packed into words, the first character is in the least significant address – might seem to be in the wrong ordering
  - For extended-precision byte arithmetic, carry bits would propagate from lower to greater significant bytes and also from lower to greater addresses
    - May allow a pipelined architecture to access memory in the order a pipeline requires the data
- Big-Endian
  - In a string with bytes packed into words, the first character is in the most significant address
    - Thus, when words are fully-packed with bytes, they can be compared more quickly using word comparisons
  - Printing bytes in ascending address order from left to right, makes them appear in the correct word order as well (from most to least significant)

Transferring Data and Network Ordering

- Beware when transferring binary numeric data between computers
  - Faster to transfer in binary format
  - More portable to transfer in a printable format such as ASCII
- Big- vs. Little-Endian matters a great deal during transfers across machines

- Big-Endian ordering is used in IPv4, IPv6, TCP, UDP
  - Referred to as Network Byte Ordering