Chapter 2
Data and Registers

Chapter 2 describes the organization of data in registers and in memory and gives a summary description of all the LR333x0 CPU registers. This chapter contains the following sections:

- Data Formats and Addressing
- LR333x0 CPU General Registers

The system control coprocessor (CP0) has a number of special purpose registers that are used in conjunction with each other during exception processing. The CP0's exception-handling registers are discussed in Chapter 4, "Exception Processing."

2.1 Data Formats and Addressing

The LR333x0 defines a 32-bit word, a 16-bit halfword, and an 8-bit byte. The byte ordering is configurable into either *big-endian* or *little-endian* byte ordering (this configuration is controlled by the BENDN signal as described in Chapter 8, "Signal Definitions").

- When configured as a *big-endian* system, byte 0 is always the most-significant (leftmost) byte, thereby providing compatibility with MC 68000 and IBM 370 conventions.
- When configured as a *little-endian* system, byte 0 is always the least-significant (rightmost) byte, thereby providing compatibility with iAPX x86, NS 32000, and DEC VAX conventions.

Bit 0 is always the least-significant (rightmost) bit. Bit designations are always little-endian (no instructions explicitly designate bit positions within words).

Figure 2.1 shows the ordering of bytes within words for the big-endian convention. In a big-endian configuration, the most-significant byte of a multiple-byte access is located at the lowest byte address. A word is addressed by the address of the most-significant byte.

Figure 2.1 Addresses of Bytes within Words: Big-Endian

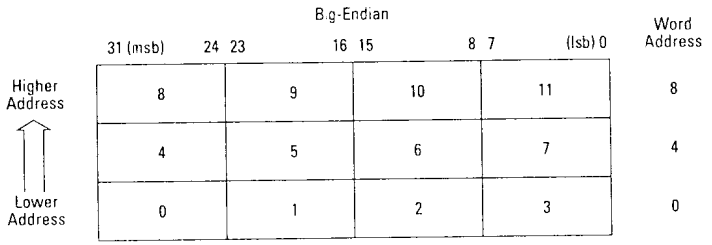
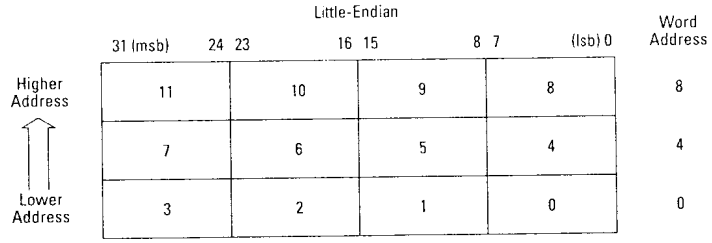


Figure 2.2 shows the ordering of bytes within words for the little-endian convention. In a little-endian configuration, the least-significant byte of a multiple-byte access is located at the lowest byte address. A word is addressed by the address of the least-significant byte.

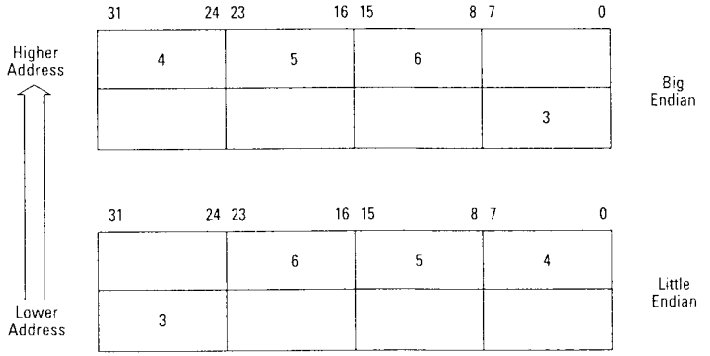
Figure 2.2 Addresses of Bytes within Words: Little-Endian



The LR333x0 uses byte addressing for halfword and word accesses with the following alignment constraints. Halfword accesses must be aligned on an even byte boundary. Word accesses must be aligned on a byte boundary divisible by four.

Special instructions are provided for addressing words that are not aligned on four-byte boundaries. The special instructions are Load Word Left (`LWL`), Load Word Right (`LWR`), Store Word Left (`SWL`), and Store Word Right (`SWR`). These instructions are used in pairs to provide addressing of misaligned words. Figure 2.3 shows the bytes accessed when addressing a misaligned word with a byte address of 3 for both big-endian and little-endian conventions.

Figure 2.3 Misaligned Word: Byte Addresses



2.2 LR333x0 CPU General Registers

Figure 2.4 shows the LR333x0 CPU registers. There are 32 general registers, each consisting of a single word (32 bits). The 32 general registers are treated symmetrically with two exceptions: *r0* is hardwired to a zero value and *r31* is defined as the link register for jump and link instructions.

Register *r0* may be specified as a target register for any instruction when the result of the operation is discarded. The register maintains a value of zero under most conditions when used as a source register.

The two Multiply/Divide registers (HI, LO) store the doubleword, 64-bit result of multiply and divide operations.

Figure 2.4 LR333x0 CPU Registers