Load/Store Byte Instructions

instruction	***example***	***meaning***
load byte	lb $1, 1002($2)	$1 = memory[$2+1002] in least sig. byte
load byte unsigned	lbu $1, 1002($2)	$1 = memory[$2+1002] in least sig. byte
store byte	sb $1, 1002($2)	memory[$2+1002] = $1 (byte modified only)

Load byte unsigned (`lbu`) extracts a specified byte from the appropriate word and places it in the least significant byte position of the target register (little endian). The remaining three upper bytes are cleared. Load Byte (`lb`) is similar except the loaded byte is signed extended so as to support an eight bit two's complement representation. Store Byte (`sb`) replaces a single byte in a word of memory. The other bytes in the word are not affected. Since all memory transactions are word-wide, store byte requires first loading the word before it is written back with the modified byte. This makes this operation more costly than a store word instruction (`sw`). To avoid initialized memory read warnings, it is necessary to initialize memory (e.g., using the `.word` directive) before storing bytes there.