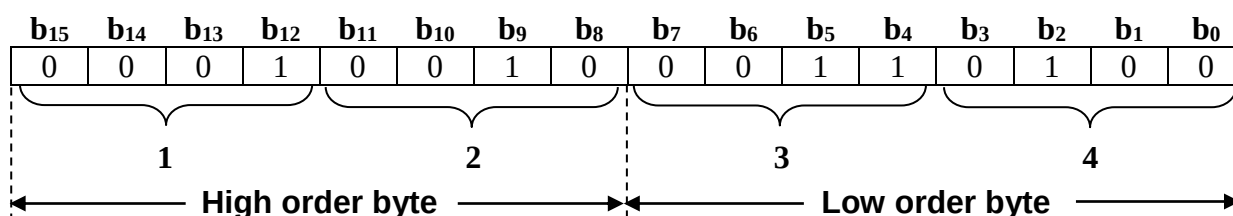
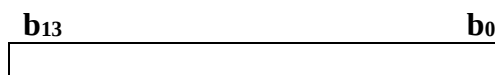


The **operand** defines the data or the address relating to the data which is to be used by the operation.

An **address** is the binary number allocated to a specific memory location so that it may be identified.



However, a typical PIC microcontroller instruction is a 14 bit word and contains the op. code and operand.



A **mnemonic** is an aid to memory. Typical microprocessor instructions are represented by mnemonics consisting of a group of letters such as MOVLW, (move literal to w or load w with a value). The instruction set of a microprocessor can contain hundreds of different instructions. Without mnemonics it would be an impossible task for the programmer to remember all the codes.

Preview page 2

hex. address	memory contents								in hex
	in binary								
	b7	b6	b5	b4	b3	b2	b1	b0	
0000	1	0	0	1	0	0	0	0	90
0001	0	0	0	1	0	0	1	0	12
0002	0	0	1	1	0	1	0	0	34
0003	etc								etc

The patterns of 0's and 1's representing the instruction are known as **machine code**. A **machine code program** is, strictly speaking, a series of binary instructions though for convenience we generally call a hex. code program machine code. The microprocessor can only respond to machine code instructions.

An **assembler program** converts each assembly language instruction into an equivalent machine code instruction on the basis of one machine code instruction to one assembly language instruction.