Appendix 1 Number Systems & Data Structure

1) Expression of number (data)

In PLC CPU, all information is saved and processed in the states of On & Off, or “1” & “0”. Numeric operation is also processed in 1 and 0, which are called Binary numbers (BIN).

Since decimal is easy and most widely used, numeric information to write or read through PLC needs to be converted from decimal to other number systems and vice-versa. This appendix shows how to better understand decimal (DEC), binary (BIN), hexadecimal (HEX), and binary coded decimal (BCD).

(1) Decimal

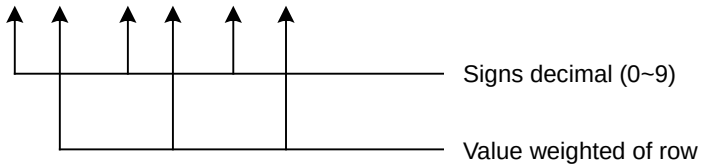
Decimal is a Base 10 number system meaning that it uses 10 digits before needing to repeat them: 0, 1, 2, 3, 4, …..9. “10” will be 2 figures.

For example, decimal 153 will be expressed as shown below in the aspect of row and “value weighted of row”

$$135=100+50+3$$

$$=1*100+5*10+3*1$$

$$=1*10^{2}+5*10^{1}+3*10^{0}$$



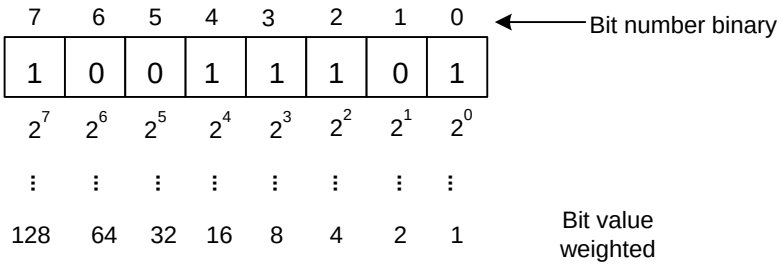
(2) Binary (Bin)

Binary is a Base 2 number system, meaning that it uses 2 digits before needing to repeat them: 0 and 1. In this system, the value of 2 will be two figures. Therefore in binary, “2” is represented as “10”. A single digit in Binary is known as a “Bit”.

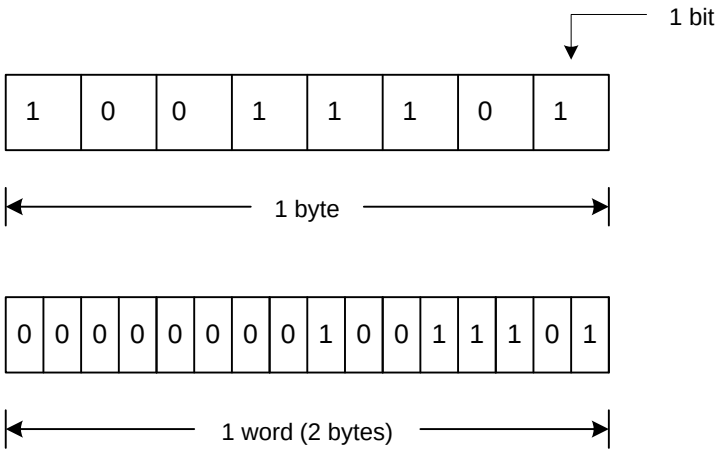
Binary	Decimal
0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
……	……

For example, the binary number “10011101” can be converted to decimal as follows;

As the row number and the value weighted of row have been considered in decimal, bit number and bit value weighted will be added from the right.



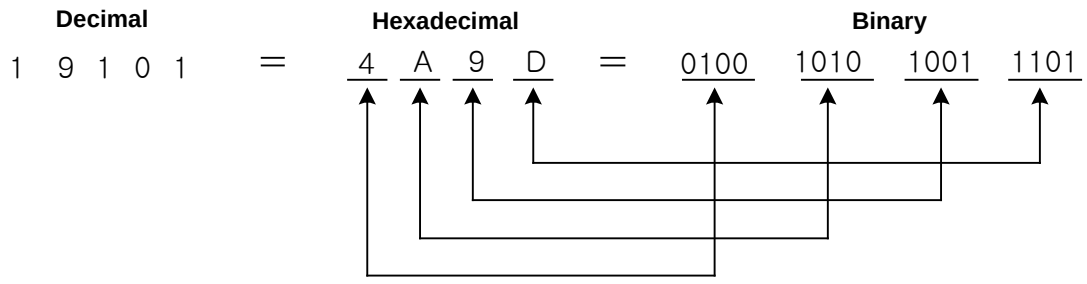
8 bits is 1 byte, and 16 bits (2 bytes) is 1 word.

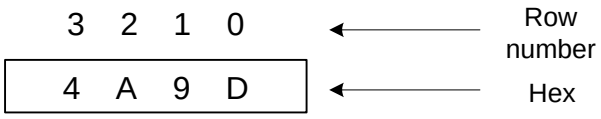


(3) Hexadecimal (HEX)

Hexadecimal is a Base 16 number system, meaning that it uses 16 digits before needing to repeat them: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. The value of 16 will be two figures. Therefore in hexadecimal, “16” is represented as “10”. Hexadecimal is used in programming because 4 bits in binary allows a range of values from 0 to 15 and it is much easier to express a number as “E” rather than “1110” (both of those equal 14).

Decimal	Hexadecimal	Binary
0	0	0
1	1	1
2	2	10
3	3	11
4	4	100
5	5	101
6	6	110
7	7	111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111
16	10	10000
17	11	10001
18	12	10010





= (4) × 163 + (A) × 162 + (9) × 161 + (D) × 160

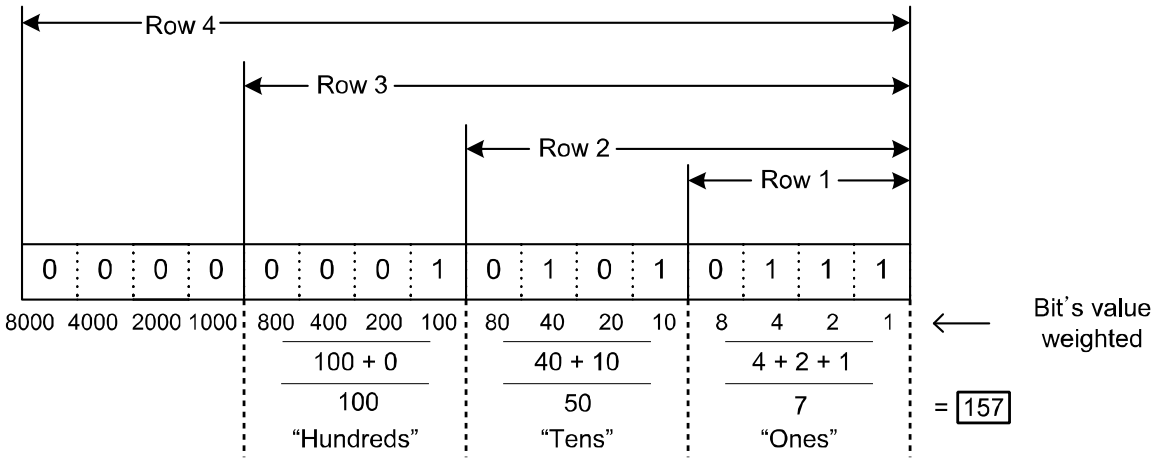
= 4 × 4096 + 10 × 2568 + 9 × 16 + 13 × 1

= 19101

(4) Binary Coded Decimal (BCD)

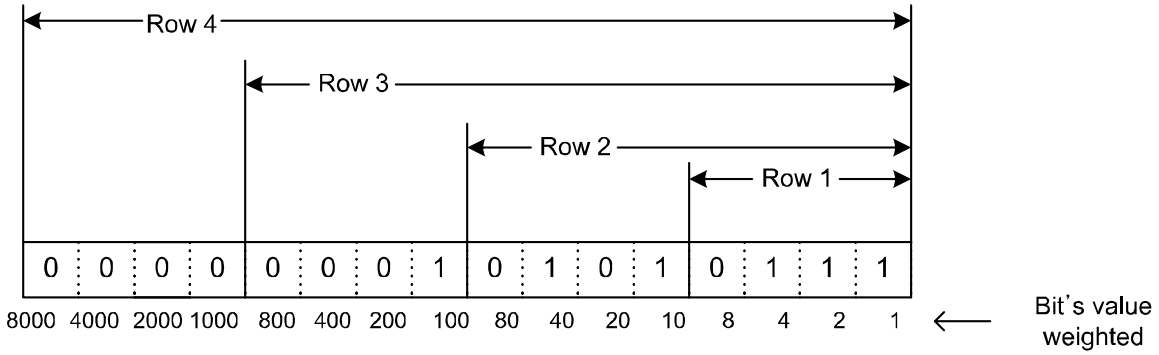
Binary coded decimal is “Decimal number of each row displayed in binary”. Each digit is represented in one byte, 0 through 9 per byte.

For example, decimal 157 can be expressed as below:



Thus, binary coded decimal can display decimal 0 ~ 9999 in 16 bits (a maximum of 4 rows).

Each bit’s value weighted is as follows:



(5) Numeric System Table

Binary coded Decimal (BCD)	Binary (BIN)	Decimal	Hexadecimal (H)
00000000 00000000	00000000 00000000	0	0000
00000000 00000001	00000000 00000001	1	0001
00000000 00000010	00000000 00000010	2	0002
00000000 00000011	00000000 00000011	3	0003
00000000 00000100	00000000 00000100	4	0004
00000000 00000101	00000000 00000101	5	0005
00000000 00000100	00000000 00000100	6	0006
00000000 00000111	00000000 00000111	7	0007
00000000 00001000	00000000 00001000	8	0008
00000000 00001001	00000000 00001001	9	0009
00000000 00010000	00000000 00001010	10	000A
00000000 00010001	00000000 00001011	11	000B
00000000 00010010	00000000 00001100	12	000C
00000000 00010011	00000000 00001101	13	000D
00000000 00010100	00000000 00001110	14	000E
00000000 00010101	00000000 00001111	15	000F
00000000 00000110	00000000 00010000	16	0010
00000000 00000111	00000000 00010001	17	0011
00000000 00001000	00000000 00010010	18	0012
00000000 00001001	00000000 00010011	19	0013
00000000 00100000	00000000 00010100	20	0014
00000000 00100001	00000000 00010101	21	0015
00000000 00100010	00000000 00010110	22	0016
00000000 00100011	00000000 00010111	23	0017
00000001 00000000	00000000 01100100	100	0064
00000001 00100111	00000000 01111111	127	007F
00000010 01010101	00000000 11111111	255	00FF
00010000 00000000	00000000 11100000	1000	03E8
00100000 01000111	00000000 11111111	2047	07FF
01000000 10010101	00000000 11111111	4095	0FFF
10011001 10011001	00000111 00001111	9999	270F
	00100111 00010000	10000	2710
	01111111 11111111	32767	7FFF

2) Expression of integer numbers

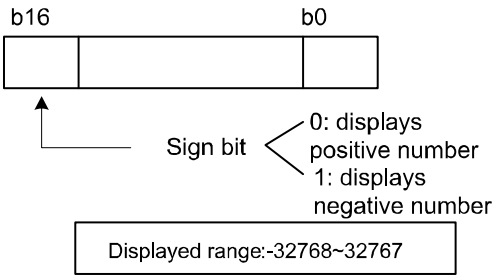
XGB instructions are based on negative operation system (Signed).

If the highest bit (MSB) is 0, it is a positive number; if it is 1, it is a negative number.

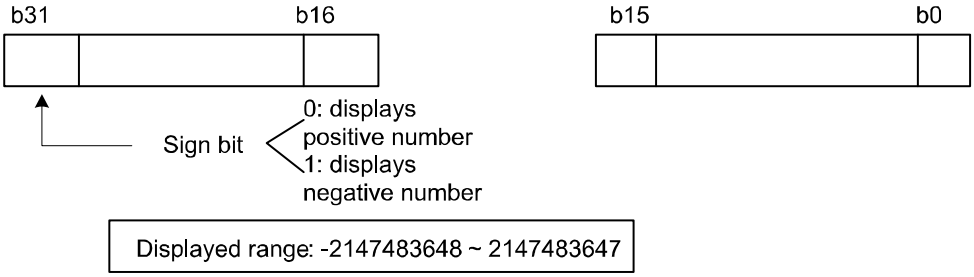
The highest bit expressing a negative or a positive number is called the Sign bit.

Since the position of MSB is different in 16 bits and 32 bits, pay attention to the position of Sign bit.

✸ In case of 16 bits



✸ In case of 32 bits



3) Expression of negative number

Ex.) How to mark – 0001

(1) Take out the negative sign and mark 0001. (b15=1)

b15				b0
1	0	~	0	1

(2) Reverse the result of (1). (b15 = excepted)

b15				b0
1	1	~	1	0

(3) Add +1 to the result of (2).

b15							b0
1	1				~	0	1

-0001 = hFFFF