Number System and Codes

Learning Outcome

- Define the following numbering systems
  - Decimal
  - Binary
  - Hexadecimal
- Capable of converting from one numbering or coding system to another
- Define Bit, Byte, Word, Least Significant Bit (LSB) and Most Significant Bit (MSB)
- Differentiate data types such as:
  - Boolean
  - Signed, unsigned Integer
  - Word, Double Word
  - Float, Real

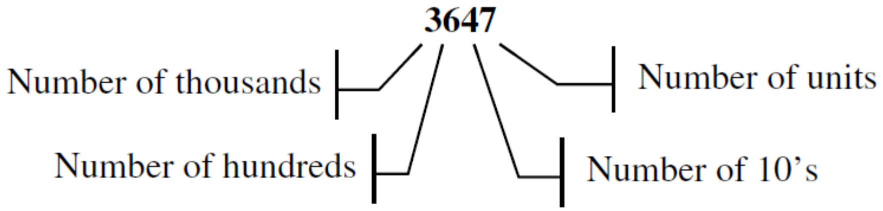
Numbering System Decimal

Basic knowledge of numbering system is necessary for programming PLC

Decimal: Base-10 numbering system or module 10 representation

Example:

$3647 = 3x10^3 + 6x10^2 + 4x10^1 + 7x10^0 = 3000 + 600 + 40 +7$



Numbering System Binary

Computers and digital systems including PLC uses binary system. Base-2 or module 2 representation.

- Each numeral has 2 states: 0 or 1.
- Binary could represent any value represented by Decimal or any other number systems by using more digits

Example:

$1011_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

2^3	2^2	2^1	2^0
1	0	1	1
MSB			LSB

Binary digit is also abbreviated to the term bit

- Most significant bit (MSB) has most weight
- Least significant bit (LSB) has least weight

Numbering System Hexadecimal

Hexadecimal number system is commonly used in computer systems

- Base-16 or module 16 representation
- 0 to 9 and A to F for the 16 digits symbols
- It is divided into groups of 4 bits

You could easily convert the numeric systems using calculator.

CODE - BCD

- What is coding?
- When number, letters or words are represented by a special group of symbols is encoding.
- Code = group of symbols

Binary-Coded-Decimal (BCD) Code

- Each decimal digit is up to 9
- 4 binary bits are required to code one decimal digit

8	7	4	Decimal
1000	0111	0100	BCD

Hexadecimal
Decimal
Binary
BCD

Hexadecimal	Decimal	Binary	BCD
0	0	0000	0000
1	1	0001	0001
2	2	0010	0010
3	3	0011	0011
4	4	0100	0100
5	5	0101	0101
6	6	0110	0110
7	7	0111	0111
8	8	1000	1000
9	9	1001	1001
A	10	1010	0001 0000
B	11	1011	0001 0001
C	12	1100	0001 0010
D	13	1101	0001 0011
E	14	1110	0001 0100
F	15	1111	0001 0101

Bits, Bytes & Words

- PLC and computer systems stores information in memory locations or registers as a Word.
- Word is capable of storing data in the form of binary digits or bits.
- 8 Bits are grouped to form a byte
- 2 Bytes are grouped to form a 16-bit Word

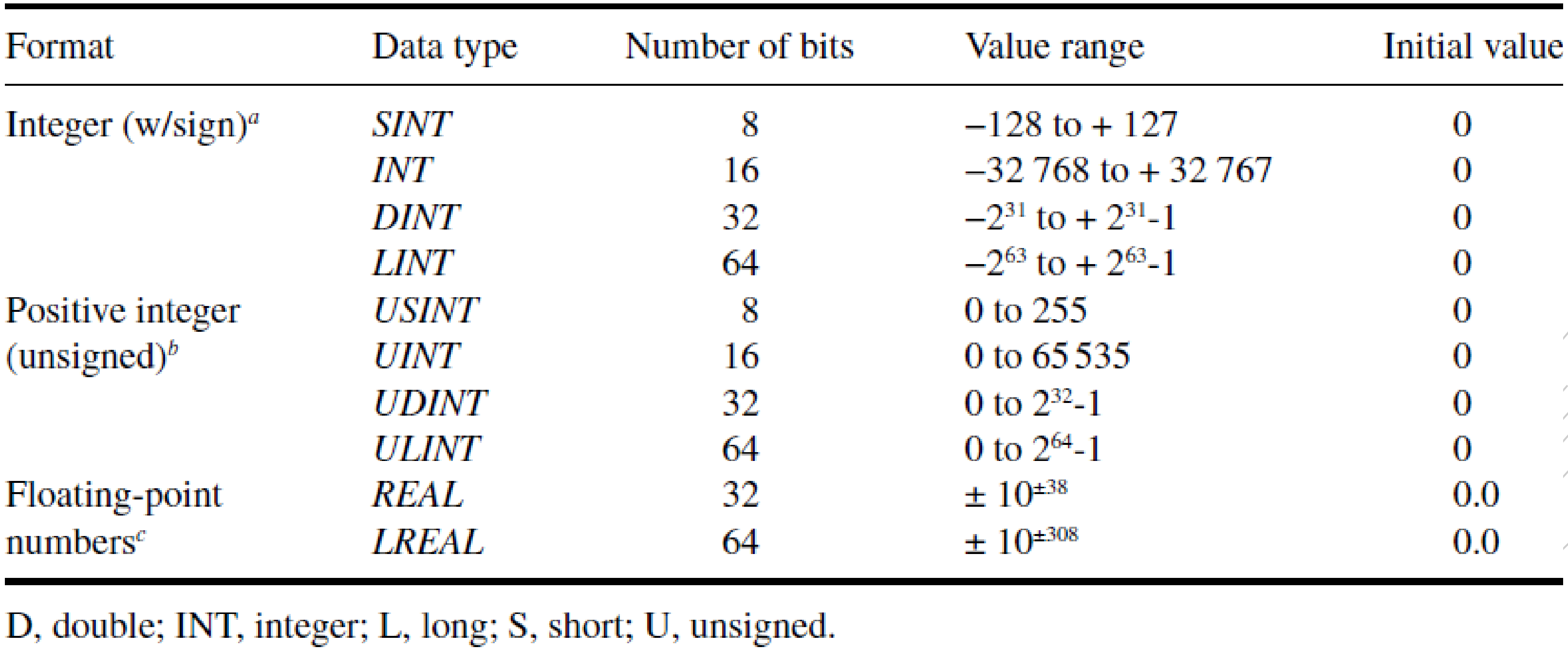
MSB							Bit 8	Bit 7							LSB
0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1
Upper Byte								Lower Byte							
16 – Bit Word															

Double Word

- The range of the numerical system depends on the number of bits
- 16-bit **Word (W)** would allow a maximum range of:
  - 2^{16} = 0 to 2^{16} (0 to 65535)
- In order to accommodate larger data, 2 words are grouped together to form a 32-bit **Double Word (DW)**
  - 2^{32} = 0 to 2^{32} -1

Data Types

- Data Type indicates the type of values or content the data could be assigned.
- ON/OFF digital signals are classified as Boolean data type (BOOL)
- Numerical and Binary data type has a great variety as below



Format	Data type	Number of bits	Value range	Initial value
Integer (w/sign)[a]	*SINT*	8	–128 to + 127	0
	INT	16	–32 768 to + 32 767	0
	DINT	32	-2^{31} to + 2^{31}-1	0
	LINT	64	-2^{63} to + 2^{63}-1	0
Positive integer (unsigned)[b]	*USINT*	8	0 to 255	0
	UINT	16	0 to 65 535	0
	UDINT	32	0 to 2^{32}-1	0
	ULINT	64	0 to 2^{64}-1	0
Floating-point numbers[c]	*REAL*	32	± $10^{\pm 38}$	0.0
	LREAL	64	± $10^{\pm 308}$	0.0

D, double; INT, integer; L, long; S, short; U, unsigned.

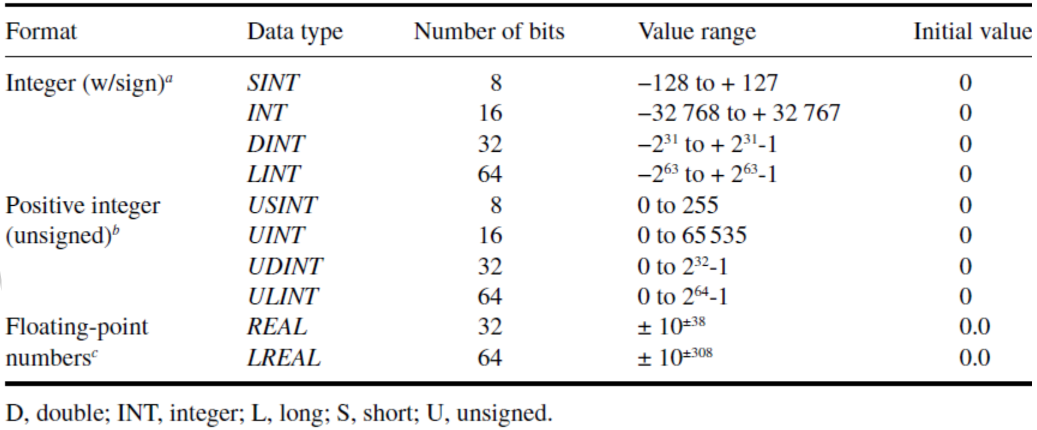
Positive & Negative Numbers

- Example for Short Integer (SINT)
- Unsigned SINT would have the full 8 bits for magnitude
  - Unsigned Short Integer (USINT) = 0 to 256 (2^8 – 1)

8 bits constituting to Magnitude							
1	0	1	0	1	0	1	0

- Did you notice that the signed data type range is half of the unsigned data type?
- Signed SINT, MSB is used as sign bit, hence only 7 bits available for magnitude
  - Signed Short Integer (SINT) = - 128 (2^7) to 127 (2^7 – 1)

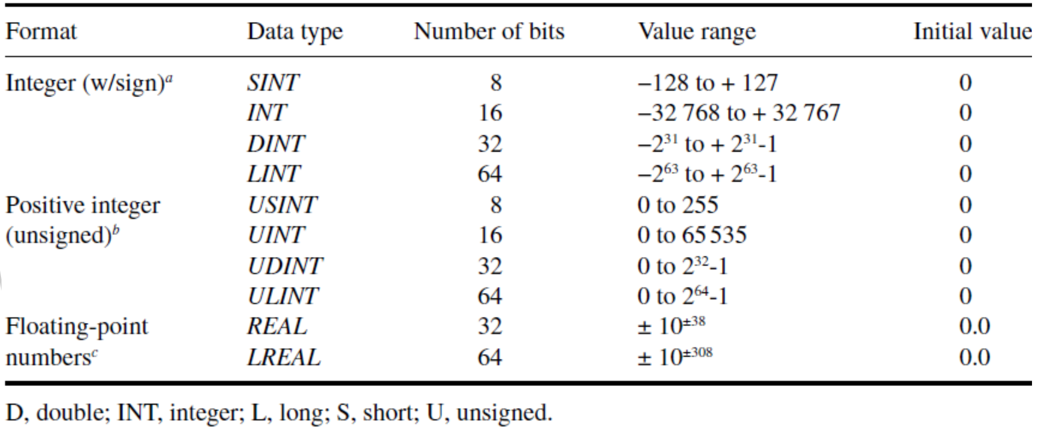
Sign Bit	7 bits constituting to Magnitude						
1	0	1	0	1	0	1	0



Format	Data type	Number of bits	Value range	Initial value
Integer (w/sign)[a]	*SINT*	8	−128 to + 127	0
	INT	16	−32 768 to + 32 767	0
	DINT	32	-2^{31} to $+ 2^{31}$-1	0
	LINT	64	-2^{63} to $+ 2^{63}$-1	0
Positive integer (unsigned)[b]	*USINT*	8	0 to 255	0
	UINT	16	0 to 65 535	0
	UDINT	32	0 to 2^{32}-1	0
	ULINT	64	0 to 2^{64}-1	0
Floating-point numbers[c]	*REAL*	32	± $10^{\pm 38}$	0.0
	LREAL	64	± $10^{\pm 308}$	0.0

D, double; INT, integer; L, long; S, short; U, unsigned.

Floating Point REAL



Format	Data type	Number of bits	Value range	Initial value
Integer (w/sign)[a]	*SINT*	8	−128 to + 127	0
	INT	16	−32 768 to + 32 767	0
	DINT	32	-2^{31} to + 2^{31}-1	0
	LINT	64	-2^{63} to + 2^{63}-1	0
Positive integer (unsigned)[b]	*USINT*	8	0 to 255	0
	UINT	16	0 to 65 535	0
	UDINT	32	0 to 2^{32}-1	0
	ULINT	64	0 to 2^{64}-1	0
Floating-point numbers[c]	*REAL*	32	$\pm 10^{\pm 38}$	0.0
	LREAL	64	$\pm 10^{\pm 308}$	0.0

D, double; INT, integer; L, long; S, short; U, unsigned.

- Floating Point is the representation of numbers with decimal point.
- This type of data require twice the register size to store the information that is split into 2 parts.
- Floating-point number = Mantissa · 10^{Exponent}
- 12532 can be represented by 1.2532E+4
- 0.00001234 can be represented by 1.234E-5
- REAL is the data type for floating point number
- 32-bit REAL would have a range of -10^{38} to 10^{38}
- REAL data type is commonly used for intermediate storage and results of arithmetic