7 BASIC INSTRUCTIONS

7.1 Comparison Operation Instructions

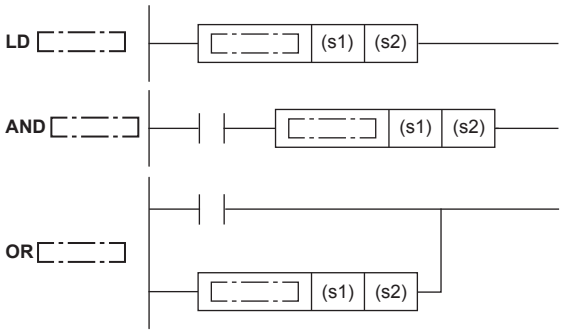
Comparing 16-bit binary data

LD□(_U), AND□(_U), OR□(_U)

FX5S FX5UJ FX5U FX5UC

These instructions perform a comparison operation between the 16-bit binary data in the device specified by (s1) and the 16-bit binary data in the device specified by (s2). (Devices are used as NO contacts.)

Ladder diagram



("=(_U)", "<>(_U)", ">(_U)", "<=(_U)", "<(_U)", ">=(_U)" enters □.)

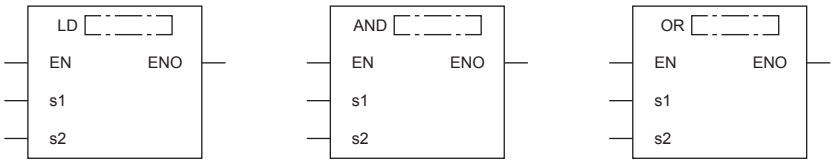
Structured text[*1]

ENO:=LD_□(EN,s1,s2);
ENO:=AND_□(EN,s1,s2);
ENO:=OR_□(EN,s1,s2);

ENO:=LD_□_U(EN,s1,s2);
ENO:=AND_□_U(EN,s1,s2);
ENO:=OR_□_U(EN,s1,s2);

("EQ", "NE", "GT", "LE", "LT", "GE" enters □.)[*2]

FBD/LD



("_EQ(_U)", "_NE(_U)", "_GT(_U)", "_LE(_U)", "_LT(_U)", "_GE(_U)" enters □.)[*2]

*1 Supported by engineering tool version "1.035M" and later.
*2 EQ is =, NE is <>, GT is >, LE is <=, LT is <, and GE is >=.

Setting data

■Descriptions, ranges, and data types

Operand		Description	Range	Data type	Data type (label)
(s1)	LD□, AND□, OR□	Comparison data or device where the comparison data is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	LD□_U, AND□_U, OR□_U		0 to 65535	16-bit unsigned binary	ANY16_U
(s2)	LD□, AND□, OR□	Comparison data or device where the comparison data is stored	-32768 to +32767	16-bit signed binary	ANY16_S
	LD□_U, AND□_U, OR□_U		0 to 65535	16-bit unsigned binary	ANY16_U
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

■Applicable devices

Operand	Bit	Word			Double word		Indirect specification	Constant			Others
	X, Y, M, L, SM, F, B, SB, S	T, ST, C, D, W, SD, SW, R	U□\G□	Z	LC	LZ		K, H	E	$	
(s1)	○	○	○	○	—	—	○	○	—	—	—
(s2)	○	○	○	○	—	—	○	○	—	—	—

Processing details

- These instructions perform a comparison operation between the 16-bit binary data in the device specified by (s1) and the 16-bit binary data in the device specified by (s2). (Devices are used as NO contacts.)
- The following table lists the comparison operation results of each instruction.

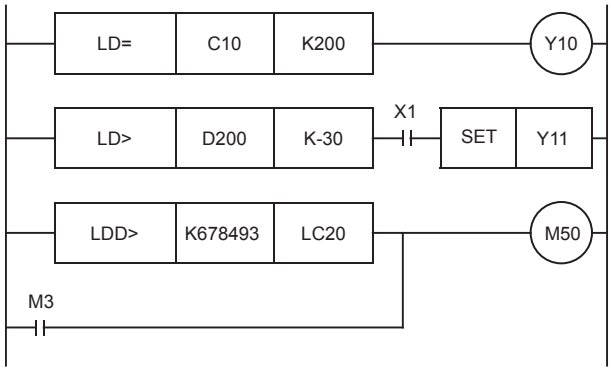
Instruction symbol	Condition	Result
=(_U)	(s1)=(s2)	Conductive state
<>(_U)	(s1)≠(s2)	
>(_U)	(s1)>(s2)	
<=(_U)	(s1)≤(s2)	
<(_U)	(s1)<(s2)	
>=(_U)	(s1)≥(s2)	
=(_U)	(s1)≠(s2)	Non-conductive state
<>(_U)	(s1)=(s2)	
>(_U)	(s1)≤(s2)	
<=(_U)	(s1)>(s2)	
<(_U)	(s1)≥(s2)	
>=(_U)	(s1)<(s2)	

Precautions

- When the most significant bit is "1" in the data stored in (s1) or (s2), it is regarded as a negative binary value for comparison. (Excluding unsigned operation)

Program example

- LD□(_U)

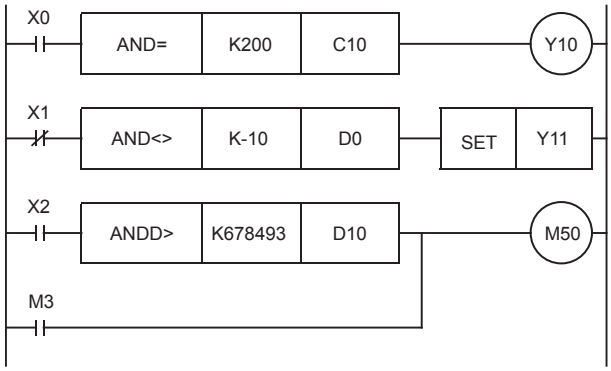


When the current value of C10 is "200", Y10 turns ON.

When the content value of D200 is "-29" or higher and X1 is ON, Y11 is set.

When the content value of LC20 is less than "K678493" or when M3 turns ON, M50 turns ON.

- AND□(_U)

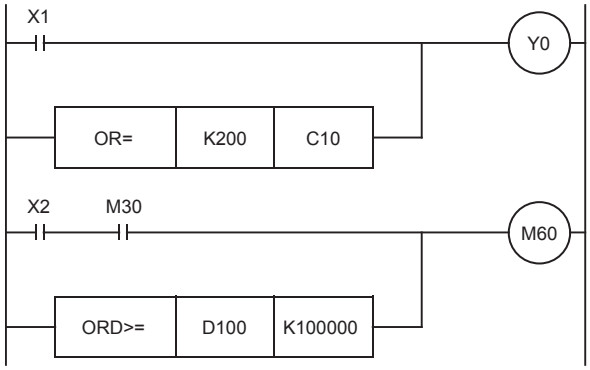


When X0 is ON and the current value of C10 is "200", Y10 turns ON.

When X1 is OFF and the content value of D0 is not "-10", Y11 is set.

When X2 is ON, and the contents value of D11 and D10 are less than "K678493", or when M3 turns ON, M50 turns ON.

- OR□(_U)



When X1 turns ON or when the current value of C10 is "200", Y0 turns ON.

When X2 and M30 turn ON or when the contents value of D101 and D100 are more than "K100000", M60 turns ON.

Operation error

There is no operation error.