

## 6.0 Logical Instructions

Use logical instructions to perform logic operations on input parameters.

Choose from these logical instructions:

- Logical AND (AND)
- Logical NOT (NOT)
- Logical OR (OR)
- Logical Exclusive OR (XOR)

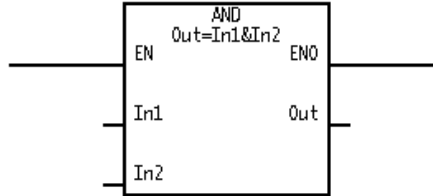
The supported parameters are:

- simple integers and double integers
- integer and double integer constants
- elements of integer and double integer arrays
- Timer variables (*name*.TPreset and *name*.Elapsed)
- Counter variables (*name*.CPreset and *name*.Current)

See each input and output parameter description for each instruction for specific information.

With the logical instructions, you can mix the data types for the inputs and outputs. When mixing integers and double integers within an instruction, the values are first converted to unsigned double integers. The result is then converted to the unsigned data type specified by the output.

## 6.1 Logical And (AND)



Use the Logical And instruction to perform a bit-wise logical AND between two variables. While EN is true, the instruction performs a logical AND operation on In1 and In2. The result is stored in Out.

The truth table for a bit-wise logical AND operation is as follows:

In1	In2	Out
0	0	0
1	0	0
0	1	0
1	1	1

### 6.1.1 Input Parameters for the Logical AND Instruction

This table lists the inputs for the AND instruction and the variable type and data type/range that each input supports.

Parameter	Description	Variable Type	Data Type/Range
EN	When this input is true, the instruction executes. When this input is false, the instruction is not executed and ENO is false.	Connect a Boolean input or output.	
In1 In2	Enter two variables or constants on which you want to perform the logical AND. <i>Note: Any constants entered are displayed in hexadecimal.</i>	<ul style="list-style-type: none"><li>• simple</li><li>• constant</li><li>• element of an array</li></ul>	integer (0-FFFF) double integer (0-FFFFFFFF)

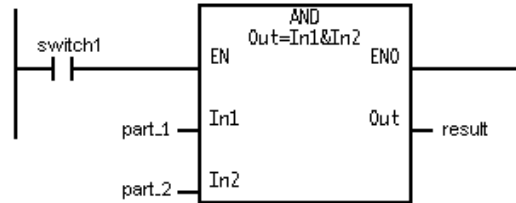
### 6.1.2 Output Parameters for the Logical AND Instruction

This table lists the outputs for the AND instruction and the variable type and data type/range that each output supports.

Parameter	Description	Variable Type	Data/Type Range
ENO	Use this output as the input to another instruction for easily chaining multiple instructions. This output follows the state of EN unless and error occurs.	Connect a contact, coil, or Boolean input to another instruction.	
Out	This output contains the result of the AND operation on In1 and In2.	<ul style="list-style-type: none"><li>• simple</li><li>• element of an array</li></ul>	integer (0-FFFF) double integer (0-FFFFFFFF)

### 6.1.3 Example of a Logical AND Instruction

When *switch1* is true, the instruction performs a logical AND operation on the variables *part\_1* and *part\_2*, and Out contains the result in the variable *result*.

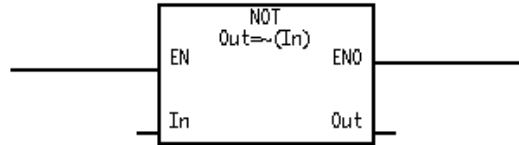


part\_1 1 0 1 1 1 0 0 1 1 0 1 0 0 0 1 1

part\_2 0 0 0 1 0 1 1 0 0 1 1 1 0 0 1 1

result 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1

## 6.2 Logical Not (NOT)



Use the Logical Not instruction to change each bit of a single variable to the opposite value. While EN is true, the instruction performs a bit-wise logical NOT operation on In. The result is stored in Out.

The truth table for a bit-wise logical NOT operation is as follows:

In	Out
1	0
0	1

## 6.2.1 Input Parameters for the Logical NOT Instruction

This table lists the inputs for the NOT instruction and the variable type and data type/range that each input supports.

Parameter	Description	Variable Type	Data Type/Range
EN	When this input is true, the instruction executes. When this input is false, the instruction is not executed and ENO is false.	Connect a Boolean input or output.	
In	Enter the variable or constant on which you want to perform the logical NOT. <i>Note: Any constants entered are displayed in hexadecimal.</i>	<ul style="list-style-type: none"><li>• simple</li><li>• constant</li><li>• element of an array</li></ul>	integer (0-FFFF) double integer (0-FFFFFFFF)

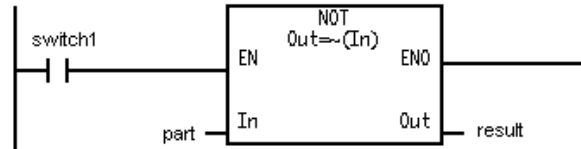
## 6.2.2 Output Parameters for the Logical NOT Instruction

This table lists the outputs for the NOT instruction and the variable type and data type/range that each output supports.

Parameter	Description	Variable Type	Data Type/Range
ENO	Use this output as the input to another instruction for easily chaining multiple instructions. This output follows the state of EN unless an error occurs.	Connect a contact, coil, or Boolean input to another instruction.	
Out	This output contains the result of the NOT operation on In.	<ul style="list-style-type: none"><li>• simple</li><li>• element of an array</li></ul>	integer (0-FFFF) double integer (0-FFFFFFFF)

### 6.2.3 Example of a Logical NOT Instruction

While *switch1* is true, the instruction performs a logical NOT on the variable *part*, and Out contains the result in the variable *result*.



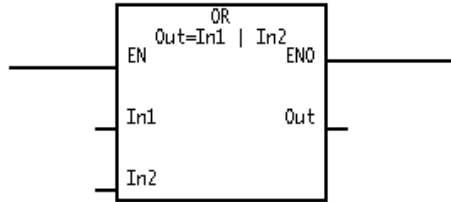
part 

0	0	1	1	0	0	1	1	1	1	1	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

result 

1	1	0	0	1	1	0	0	0	0	0	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

## 6.3 Logical Or (OR)



Use the Logical Or instruction to perform a bit-wise logical OR operation between two variables. While EN is true, the instruction performs a logical OR operation between In1 and In2. The result is stored in Out.

The truth table for a bit-wise logical OR operation is as follows:

In1	In2	Out
0	0	0
1	0	1
0	1	1
1	1	1

### 6.3.1 Input Parameters for the Logical OR Instruction

This table lists the inputs for the OR instruction and the variable type and data type/range that each input supports.

Parameter	Description	Variable Type	Data Type/Range
EN	When this input is true, the instruction executes. When this input is false, the instruction is not executed and ENO is false.	Connect a Boolean input or output.	
In1 In2	Enter two variables or constants on which you want to perform the logical OR. <i>Note: Any constants entered are displayed in hexadecimal.</i>	<ul style="list-style-type: none"><li>• simple</li><li>• constant</li><li>• element of an array</li></ul>	integer (0-FFFF) double integer (0-FFFFFFFF)

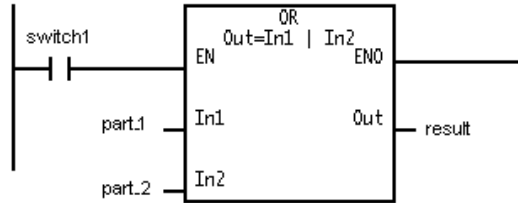
### 6.3.2 Output Parameters for the Logical OR Instruction

This table lists the outputs for the OR instruction and the variable type and data type/range that each output supports.

Parameter	Description	Variable Type	Data Type/Range
ENO	Use this output as the input to another instruction for easily chaining multiple instructions. This output follows the state of EN unless an error occurs.	Connect a contact, coil, or Boolean input to another instruction.	
Out	This output contains the result of the OR operation on In1 and In2.	<ul style="list-style-type: none"><li>• simple</li><li>• element of an array</li></ul>	integer (0-FFFF) double integer (0-FFFFFFFF)

### 6.3.3 Example of a Logical OR Instruction

While the variable *switch1* is true, the instruction performs a logical OR on the variables *part\_1* and *part\_2*, and Out contains the result in the variable *result*.



part\_1 

0	0	1	1	0	0	1	1	0	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

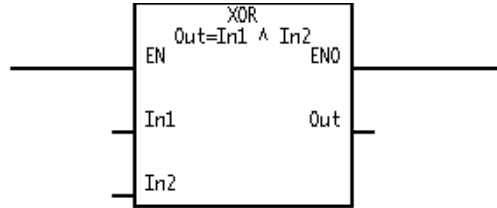
part\_2 

1	1	0	0	0	1	1	0	1	1	0	0	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

result 

1	1	1	1	0	1	1	1	1	1	1	1	1	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

## 6.4 Logical Exclusive Or (XOR)



Use the Logical Exclusive Or instruction to perform a bit-wise logical exclusive OR operation between two variables. While EN is true, the instruction performs a logical exclusive OR operation between In1 and In2. The result is stored in Out.

The truth table for a bit-wise logical XOR operation is as follows:

In1	In2	Out
0	0	0
1	0	1
0	1	1
1	1	0

### 6.4.1 Input Parameters for the Logical Exclusive OR Instruction

This table lists the inputs for the XOR instruction and the variable type and data type/range that each input supports.

Parameter	Description	Variable Type	Data Type/Range
EN	When this input is true, the instruction executes. When this input is false, the instruction is not executed and ENO is false.		Connect a Boolean input or output.
In1 In2	Enter two variables or constants on which you want to perform the logical XOR. <i>Note: Any constants entered are displayed in hexadecimal.</i>	<ul style="list-style-type: none"><li>• simple</li><li>• constant</li><li>• element of an array</li></ul>	integer (0-FFFF) double integer (0-FFFFFFFF)

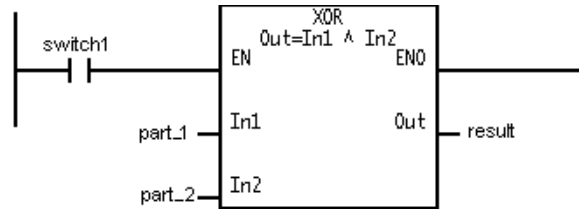
### 6.4.2 Output Parameters for the Logical Exclusive OR Instruction

This table lists the outputs for the XOR instruction and the variable type and data type/range that each output supports.

Parameter	Description	Variable Type	Data Type/Range
ENO	Use this output as the input to another instruction for easily chaining multiple instructions. This output follows the state of EN unless an error occurs.		Connect a contact, coil, or Boolean input to another instruction.
Out	This output contains the result of the XOR operation on In1 and In2.	<ul style="list-style-type: none"><li>• simple</li><li>• element of an array</li></ul>	integer (0-FFFF) double integer (0-FFFFFFFF)

### 6.4.3 Example of a Logical Exclusive OR Instruction

When the variable *switch1* is true, the instruction performs a logical exclusive OR operation on the variables *part\_1* and *part\_2*, and *Out* contains the result in the variable *result*.



part\_1 

0	0	1	1	0	0	1	1	1	1	0	0	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

part\_2 

0	1	0	1	0	1	0	0	0	0	1	0	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

result 

0	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

## 6.5 Errors Caused by Logical Instructions

These errors can occur when you are using the logical instructions in a program. They are logged in the error log.

<b>If this error occurs:</b>	<b>Then:</b>	<b>Do the following:</b>
The array index is negative.	ENO is set to ERROR_ENO, and element zero of the array is used for the instruction's operation.	Specify a valid array element.
The array index is too large.	ENO is set to ERROR_ENO, and the last element of the array is used for the instruction's operation.	Specify a valid array element.