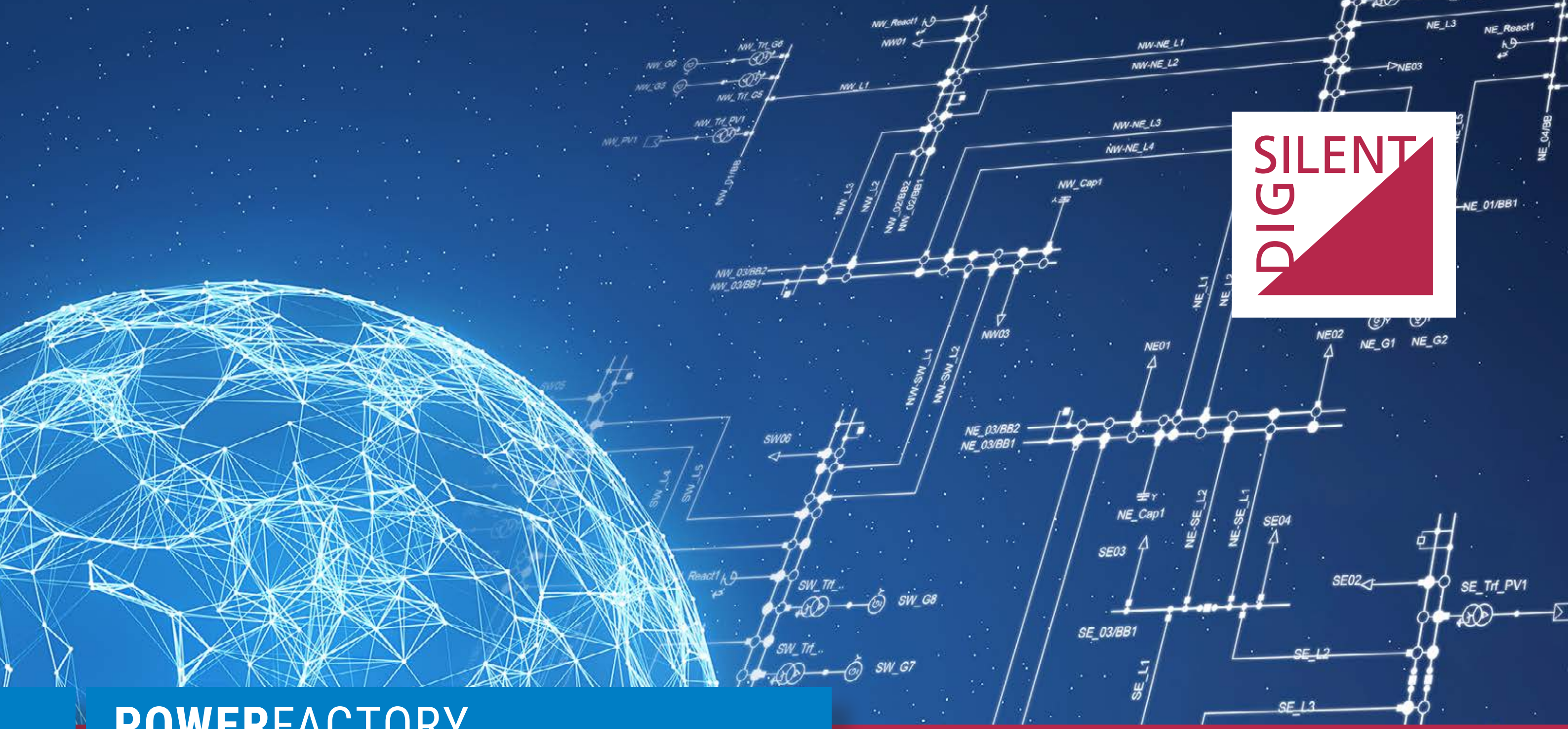


DIGSILENT

POWERFACTORY

PowerFactory 2021

Technical Reference

Logic Block

RelLogic, TypLogic

PF2021

POWER SYSTEM SOLUTIONS
MADE IN GERMANY

Publisher:
DIgSILENT GmbH
Heinrich-Hertz-Straße 9
72810 Gomaringen / Germany
Tel.: +49 (0) 7072-9168-0
Fax: +49 (0) 7072-9168-88
info@digsilent.de

Please visit our homepage at:
https://www.digsilent.de

December 1, 2020
PowerFactory 2021
Revision 1

Contents

1 General Description 1

2 Functional Parameter Descriptions 2

2.1 *TypLogic: No. of inputs - inputs* 2

2.2 *TypLogic: No. of inputs per block - iblkf* 2

2.3 *TypLogic: Logic - imat* 2

2.4 *RelLogic: Circuit breaker - pswitch* 4

2.5 *TypLogic: Breaker Event - itrip* 4

3 Function of the logic block during simulations 5

3.1 Function of the logic block during static simulations 5

3.2 Function of the logic block during dynamic simulations 6

A Logic Configuration Examples 7

B Input parameters 12

C Signals 13

List of Figures 14

List of Tables 15

1 General Description

The logic block, described by the type class TypLogic and the element class RelLogic, performs the following functions:

- Converts a specified number of input quantities into a single output quantity.
- Assigns a switching device, and a switching command to the relay with which the logic block is associated.

Figure 1.1 illustrates a typical protective relaying application. Note that in this example the logic block receives two inputs. One from a time overcurrent block and one from an instantaneous overcurrent block.

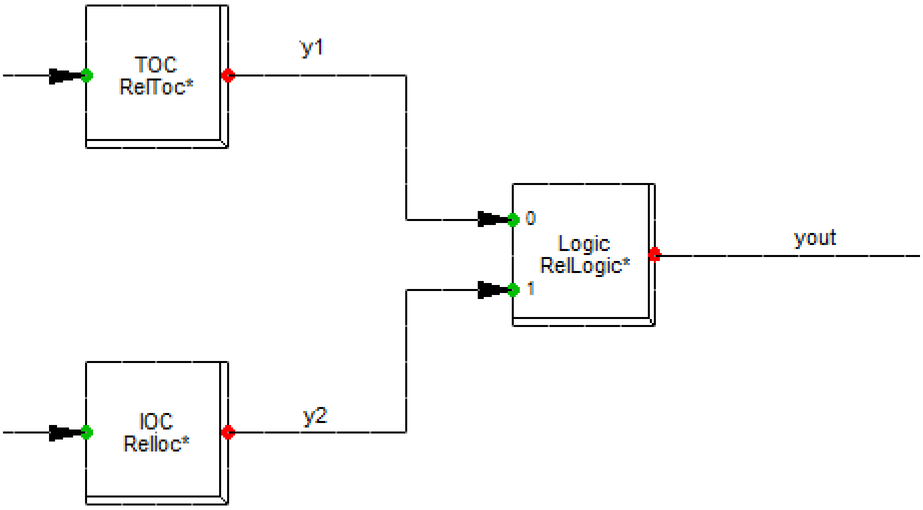


Figure 1.1: Typical application of a logic block

The function performed by a logic block depends on whether a static or a dynamic simulation is being carried out. The output from a logic block will depend upon the status of the inputs as well as the operation that the user defines for the block. The output of the logic block is used to initiate operation of a switching device. The switching device which is to be operated and the operation to be carried out are configured within the logic block. The parameters used to configure the function of the logic block are described in the following section.

2 Functional Parameter Descriptions

Parameters are referenced within this document using the following convention. *A: B - C*, where,

- *A* is the level of extraction of the block model at which the parameter can be edited.
- *B* is the description of the parameter used within the dialogue associated with A.
- *C* is the parameter name used within the software.

2.1 *TypLogic: No. of inputs - inputs*

This input parameter defines the total number of input signals the logic block can accept. The possible values are 2, 4 or 8. The number of input signals defined by this parameter, can exceed the actual number of input signals received by a logic block.

2.2 *TypLogic: No. of inputs per block - iblkf*

This input parameter represents the maximum number of inputs to any operator used in the logic block. The possible values are 2,4 or 8. The number of inputs per block should not be specified to be higher than the total number of inputs. If the number of block inputs is lower than the total number of inputs then more than one operator and more than one operation level is required to process all the inputs. The operation levels are defined as OP1, OP2 and OP3. The number of inputs per block defined by this parameter, can exceed the actual number of input signals received by a logic operator.

2.3 *TypLogic: Logic - imat*

Figure 2.1 illustrates an equivalent logic circuit for a logic block configured with 8 inputs and 2 inputs per block. In this example all gates are shown as OR gates. However, the user is able to specify the operation carried out by each gate. The operations available are AND, OR, NAND and NOR. The user should be aware that the NAND and NOR functions, although selectable, do not function and should not be selected. Furthermore, for static simulations, the operation carried out by a gate specified as an OR or an AND gate is not strictly a logical operation. This is discussed further in section 3.1.

The operations of the logic block are configured in a table such as the one illustrated in figure 2.2. The table defines the same conversion operations as the equivalent circuit illustrated by figure 2.1. With regards to the column headed OP1, each function defined in this column acts on the signals defined in the far left column at the same level. The upper function in the column headed OP2 relates to the operation which is carried out on signals $y12$ and $y34$ (as illustrated in figure 2.1), while the lower function relates to the operation which is carried out on signals $y56$ and $y78$. The function in the column headed OP3 relates to the operation which is carried out on the signals $y1234$ and $y5678$. For other combinations of input signals the operations are defined in a similar manner. See appendix A for further examples.

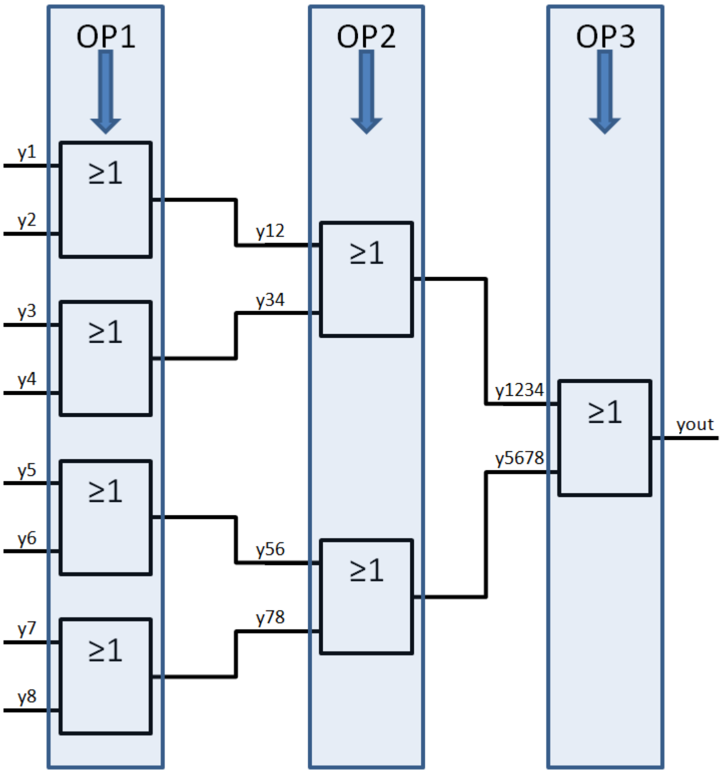
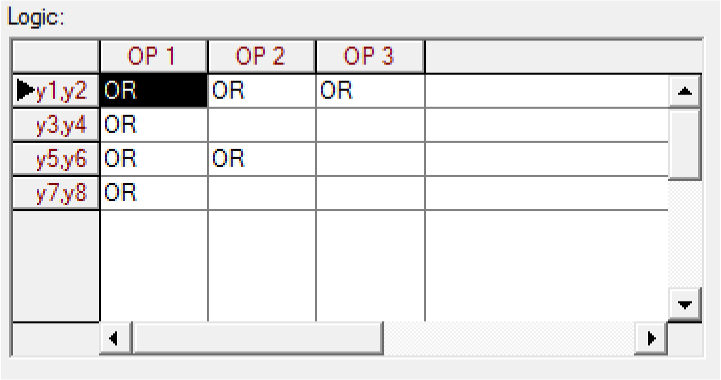


Figure 2.1: Equivalent logic circuit for a logic block configured with 8 inputs and 2 inputs per block


Logic:

	OP 1	OP 2	OP 3
y1,y2	OR	OR	OR
y3,y4	OR		
y5,y6	OR	OR	
y7,y8	OR		

Figure 2.2: *TypLogic: Logic - imat* configuration for a logic block configured with 8 inputs and 2 inputs per block

2.4 *RelLogic: Circuit breaker - pswitch*

This parameter can be used to assign a switching device to the relay with which the logic block is associated. If no value is entered for this parameter then the relay will by default be assigned to the switching device occupying the location specified by the parameter *ElmRelay: Location\Reference - pReference*. If there is also no value entered in this location, then the relay will be assigned to one of two switching devices. If the relay was created in a cubicle, then the relay will be assigned to the switching device existing within the same cubicle. If the relay was created outside of a cubicle then the relay will be assigned to the switching device sharing the measurement location of the current transformer assigned to the relay.

2.5 *TypLogic: Breaker Event - itrip*

This parameter is used to define the type of switching operation to be carried out by the relay upon the switching device to which it is assigned. A logic block can be selected to perform an open or a close operation on the switching device. Alternatively it can be selected to perform no operation at all.

3 Function of the logic block during simulations

Simulations can be classed as static or dynamic. A static simulation is defined as a load flow or short circuit analysis while a dynamic simulation is defined as a balanced RMS, three phase RMS or EMT simulation. The function performed by a logic block depends on whether the simulation being carried out is static or dynamic. The function performed differs because the types of signals handled under each of the simulations are of a different nature. Sections 3.1 and 3.2 will describe these differences and the functions carried out.

3.1 Function of the logic block during static simulations

During static simulations a logic block is used to convert 2,4 or 8 input quantities with dimensional units of time to a single output quantity also with dimensional units of time. For a typical protective relaying application the input quantities might relate to the operating time of different protective elements.

Since the inputs to a logic block in a static simulation are not Boolean values it is not possible to use the logic block to carry out a Boolean function. Instead, the logic block is used to determine the maximum or the minimum value of the input quantities. The maximum value is defined to be an AND function and the minimum value is defined to be an OR function. Using Figure 1.1 as an example, the time overcurrent and instantaneous overcurrent blocks will both output time values equal to their respective operating time. The operating times of these protective elements will depend upon their own inputs and the protective characteristics with which the blocks have been configured. Table 3.1 summarises the operation of the logic block in Figure 1.1 based on assumed, illustrative input values. The output of the logic block is provided both for an AND and an OR configuration.

Table 3.1: Operation of a 2 input logic block during static simulations with illustrative input values

Input y1 (TOC) ms	Input y2 (IOC) ms	AND(MAX) output ms	OR(MIN) output ms
500	100	500	100
500	9999999	9999999	500
20000	9999999	9999999	20000
9999999	9999999	9999999	9999999

It should be clear from the table that for the block arrangement given, the logic block should be configured with an OR function in order to generate a single output which reflects the operating time of the relay. If an AND function is used then the logic block output will be very high unless both protective elements are generating low output time values. A relay which operates only when both its time overcurrent and instantaneous elements have operated, and with an operating time equal to the longer of the two values would not be desirable for conventional protective relaying applications!

In some circumstances a logic block may be provided with fewer signals than it has been configured to accept using the parameters described in sections 2.1 and 2.2. For instance, if the logic block is used to convert the output signals from 3 separate overcurrent blocks each representing one phase, into a single output signal, there is no option for configuring the block to accept 3 inputs. In these circumstances the logic block should be configured as though it were receiving a larger number of signals than it is actually receiving. For static simulations, any signals that are not present are then treated by the logic block as a 9999999ms input signal for OR(MIN) functions and a 0ms input signal for AND(MAX) functions.

3.2 Function of the logic block during dynamic simulations

During dynamic simulations a logic block is used to convert 2,4 or 8 logical input signals to a single logical output signal. For a typical protective relaying application the input quantities might represent the trip status of various protective elements used within a relay model. Using Figure 1.1 as an example, the time overcurrent and instantaneous overcurrent blocks might both provide output signals indicating their status. ie. logical output status indicating whether the element has operated. The status of these protective elements will depend upon their own inputs and the protective characteristics with which the blocks have been configured. Table 3.2 summarises the operation of the logic block in Figure 1.1 based on assumed, illustrative input status values. The output of the logic block is provided for an AND and OR configuration.

Table 3.2: Operation of a 2 input logic block during static simulations with illustrative input status

Input y1 (TOC)	Input y2 (IOC)	AND output	OR output
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

If it assumed that the logic block should provide an output whenever one or both of the protective elements operate, it should be clear from the table that the logic block should be configured with an OR function.

As described for static simulations, in some circumstances a logic block may be provided with fewer signals than it has been configured to accept using the parameters described in sections 2.1 and 2.2. For dynamic simulations, any input signals which have been configured but are not present, are treated by the logic block as a 0 input for OR functions and a 1 input for AND functions.

A Logic Configuration Examples

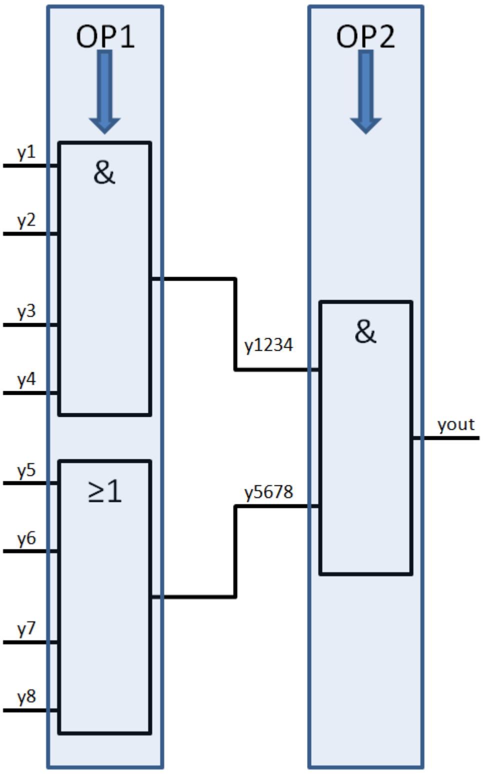
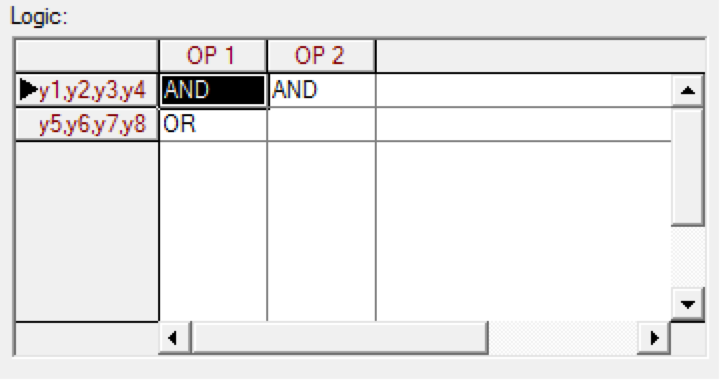


Figure A.1: Equivalent logic circuit for a logic block configured with 8 inputs and 4 inputs per block



Logic:

	OP 1	OP 2
y1,y2,y3,y4	AND	AND
y5,y6,y7,y8	OR	

Figure A.2: *TypLogic: Logic - imat* configuration for a logic block configured with 8 inputs and 4 inputs per block

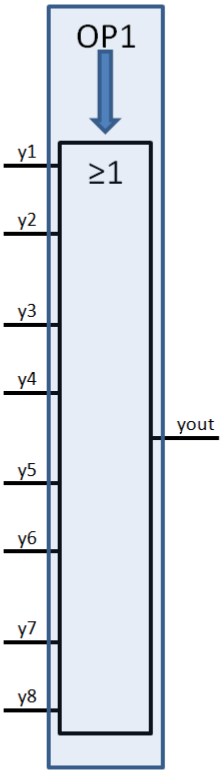
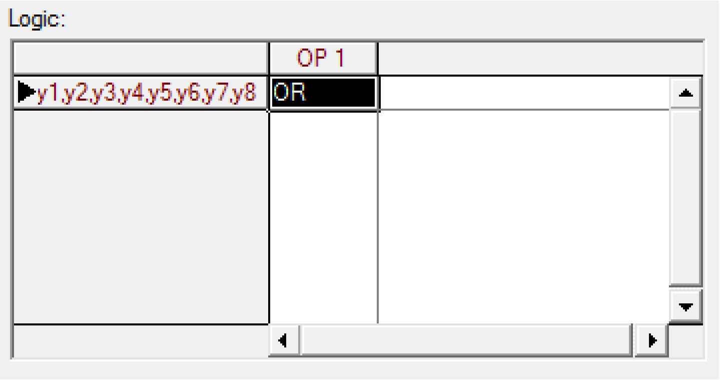


Figure A.3: Equivalent logic circuit for a logic block configured with 8 inputs and 8 inputs per block


Logic:

	OP 1
▶y1,y2,y3,y4,y5,y6,y7,y8	OR

Figure A.4: *TypLogic: Logic - imat* configuration for a logic block configured with 8 inputs and 8 inputs per block

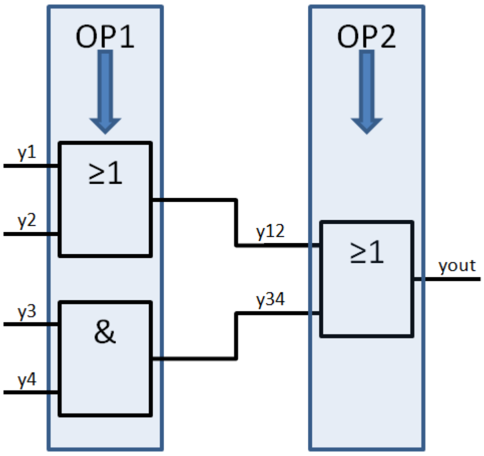
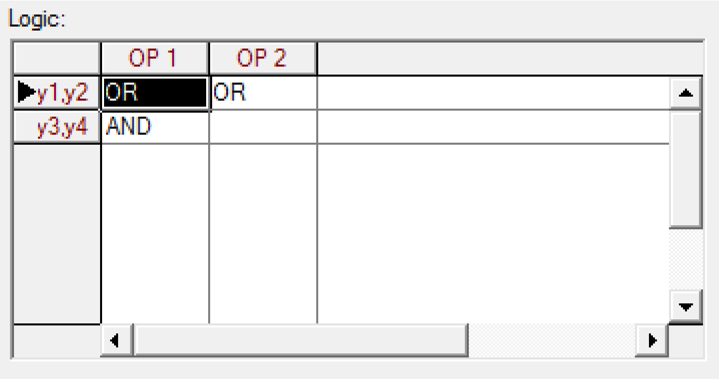


Figure A.5: Equivalent logic circuit for a logic block configured with 4 inputs and 2 inputs per block


Logic:

	OP 1	OP 2
▶y1,y2	OR	OR
y3,y4	AND	

Figure A.6: *TypLogic: Logic - imat* configuration for a logic block configured with 4 inputs and 2 inputs per block

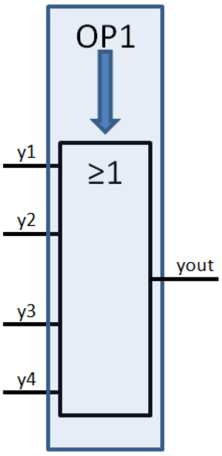
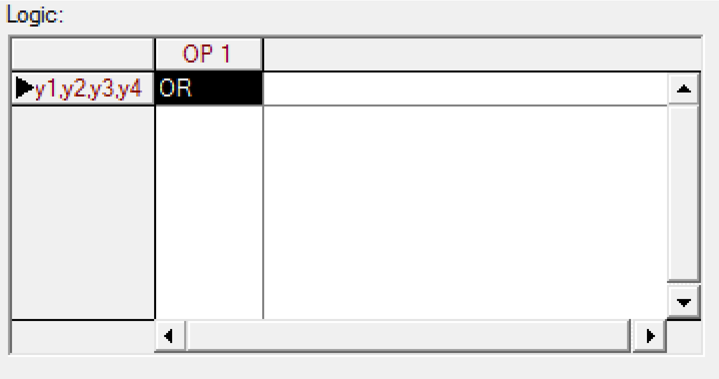


Figure A.7: Equivalent logic circuit for a logic block configured with 4 inputs and 4 inputs per block



Logic:

	OP 1
▶y1,y2,y3,y4	OR

Figure A.8: *TypLogic: Logic - imat* configuration for a logic block configured with 4 inputs and 4 inputs per block

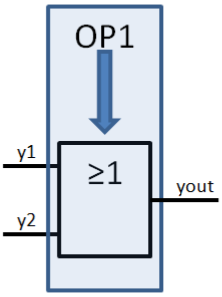
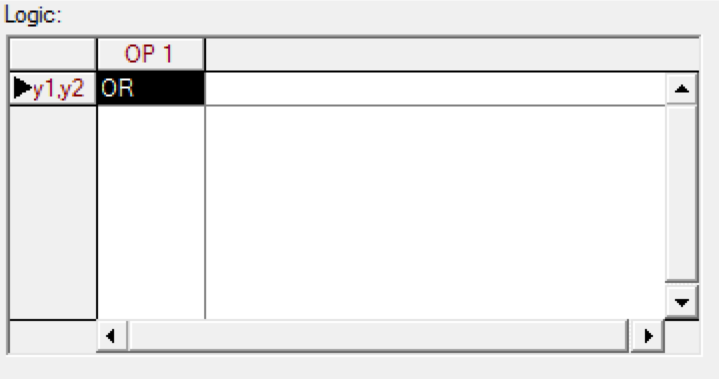


Figure A.9: Equivalent logic circuit for a logic block configured with 2 inputs and 2 inputs per block



Logic:

	OP 1
▶y1,y2	OR

Figure A.10: *TypLogic: Logic - imat* configuration for a logic block configured with 2 inputs and 2 inputs per block

B Input parameters

Table B.1: Type Parameters (Class: TypLogic)

Parameter name	Reference Object type	Default Value	Description	Selectable Values	Unit
gnrl_modif		01.01.1970 01:00:00	Object modified		int
gnrl_modby			Object modified by		char[40]
loc_name			Name		char[40]
itrip		1	Breaker Event	None:Open:-Close	int
inputs		2	No. of Inputs	2:4:8	int
iblkf		2	No. of Block Inputs	2:4:8	int
imat		0,	Logic		dMat*

Table B.2: Element Parameters (Class: RelLogic)

Parameter Name	Reference Object Type	Default Value	Description	Selectable Values	Unit
loc_name			Name		char[40]
typ_id			Type (TypLogic*)		DBObject*
outserv		0	Out of Service	$x \geq 0$ and $x \leq 1$	int
pSwitch	StaCubic*, StaSwitch*, ElmCoup*		Open		pVec*
iout		0,	Out of Service		dVec*
gnrl_modby			Object modified by		char[40]
chr_name			Characteristic Name		char[20]
dat_src		MAN	Data source		char[3]
for_name			Foreign Key		char[20]

C Signals

Table C.1: Logic block signals

Signal	Input / Output	Name
y1	Input	Input y1
y2	Input	Input y2
y3	Input	Input y3
y4	Input	Input y4
y5	Input	Input y5
y6	Input	Input y6
y7	Input	Input y7
y8	Input	Input y8
yout	Output	Tripping

List of Figures

1.1 Typical application of a logic block 1

2.1 Equivalent logic circuit for a logic block configured with 8 inputs and 2 inputs per block 3

2.2 *TypLogic: Logic - imat* configuration for a logic block configured with 8 inputs and 2 inputs per block 3

A.1 Equivalent logic circuit for a logic block configured with 8 inputs and 4 inputs per block 7

A.2 *TypLogic: Logic - imat* configuration for a logic block configured with 8 inputs and 4 inputs per block 7

A.3 Equivalent logic circuit for a logic block configured with 8 inputs and 8 inputs per block 8

A.4 *TypLogic: Logic - imat* configuration for a logic block configured with 8 inputs and 8 inputs per block 8

A.5 Equivalent logic circuit for a logic block configured with 4 inputs and 2 inputs per block 9

A.6 *TypLogic: Logic - imat* configuration for a logic block configured with 4 inputs and 2 inputs per block 9

A.7 Equivalent logic circuit for a logic block configured with 4 inputs and 4 inputs per block 10

A.8 *TypLogic: Logic - imat* configuration for a logic block configured with 4 inputs and 4 inputs per block 10

A.9 Equivalent logic circuit for a logic block configured with 2 inputs and 2 inputs per block 11

A.10 *TypLogic: Logic - imat* configuration for a logic block configured with 2 inputs and 2 inputs per block 11

List of Tables

3.1 Operation of a 2 input logic block during static simulations with illustrative input values 5

3.2 Operation of a 2 input logic block during static simulations with illustrative input status 6

B.1 Type Parameters (Class: TypLogic) 12

B.2 Element Parameters (Class: RelLogic) 12

C.1 Logic block signals 13