

My Project

Generated by Doxygen 1.8.17

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 FileManager Class Reference	7
4.1.1 Detailed Description	7
4.1.2 Member Function Documentation	7
4.1.2.1 loadGates()	7
4.1.2.2 loadStates()	8
4.1.2.3 save()	8
4.2 Gate Class Reference	8
4.2.1 Detailed Description	9
4.2.2 Member Function Documentation	10
4.2.2.1 getIn1()	10
4.2.2.2 getIn2()	10
4.2.2.3 getOut()	10
4.2.2.4 operate()	10
4.2.2.5 printGate()	11
4.2.2.6 setInOutPuts()	11
4.3 GateAND Class Reference	12
4.3.1 Detailed Description	12
4.3.2 Member Function Documentation	12
4.3.2.1 getIn1()	12
4.3.2.2 getIn2()	13
4.3.2.3 getOut()	13
4.3.2.4 operate()	13
4.3.2.5 printGate()	13
4.3.2.6 setInOutPuts()	14
4.4 GateNAND Class Reference	14
4.4.1 Detailed Description	15
4.4.2 Member Function Documentation	15
4.4.2.1 getIn1()	15
4.4.2.2 getIn2()	15
4.4.2.3 getOut()	15
4.4.2.4 operate()	15
4.4.2.5 printGate()	16
4.4.2.6 setInOutPuts()	16

4.5 GateNEG Class Reference	17
4.5.1 Detailed Description	17
4.5.2 Member Function Documentation	17
4.5.2.1 getIn1()	17
4.5.2.2 getIn2()	18
4.5.2.3 getOut()	18
4.5.2.4 operate()	18
4.5.2.5 printGate()	18
4.5.2.6 setInOutPuts()	19
4.6 GateNOR Class Reference	19
4.6.1 Detailed Description	20
4.6.2 Member Function Documentation	20
4.6.2.1 getIn1()	20
4.6.2.2 getIn2()	20
4.6.2.3 getOut()	20
4.6.2.4 operate()	20
4.6.2.5 printGate()	21
4.6.2.6 setInOutPuts()	21
4.7 GateOR Class Reference	22
4.7.1 Detailed Description	22
4.7.2 Member Function Documentation	22
4.7.2.1 getIn1()	22
4.7.2.2 getIn2()	23
4.7.2.3 getOut()	23
4.7.2.4 operate()	23
4.7.2.5 printGate()	23
4.7.2.6 setInOutPuts()	24
4.8 GateXNOR Class Reference	24
4.8.1 Detailed Description	25
4.8.2 Member Function Documentation	25
4.8.2.1 getIn1()	25
4.8.2.2 getIn2()	25
4.8.2.3 getOut()	25
4.8.2.4 operate()	25
4.8.2.5 printGate()	26
4.8.2.6 setInOutPuts()	26
4.9 GateXOR Class Reference	27
4.9.1 Detailed Description	27
4.9.2 Member Function Documentation	27
4.9.2.1 getIn1()	27
4.9.2.2 getIn2()	28
4.9.2.3 getOut()	28

4.9.2.4 operate()	28
4.9.2.5 printGate()	28
4.9.2.6 setInOutPuts()	29
4.10 InOut Class Reference	29
4.10.1 Detailed Description	30
4.10.2 Member Function Documentation	30
4.10.2.1 getIn1()	30
4.10.2.2 getIn2()	30
4.10.2.3 getOut()	30
4.10.2.4 operate()	30
4.10.2.5 printGate()	31
4.10.2.6 setInOutPuts()	31
4.11 ValueState Class Reference	32
4.11.1 Detailed Description	32
4.11.2 Member Function Documentation	32
4.11.2.1 getIn()	32
4.11.2.2 getState()	32
4.11.2.3 printFinalState()	33
4.11.2.4 printState()	33
4.11.2.5 setState()	33
5 File Documentation	35
5.1 fileManager.h File Reference	35
5.2 gate.h File Reference	35
5.3 gateAND.h File Reference	35
5.3.1 Macro Definition Documentation	36
5.3.1.1 gate_AND	36
5.4 gateNAND.h File Reference	36
5.4.1 Macro Definition Documentation	36
5.4.1.1 gate_NAND	36
5.5 gateNEG.h File Reference	37
5.5.1 Macro Definition Documentation	37
5.5.1.1 gate_NEG	37
5.6 gateNOR.h File Reference	37
5.6.1 Macro Definition Documentation	37
5.6.1.1 gate_NOR	38
5.7 gateOR.h File Reference	38
5.7.1 Macro Definition Documentation	38
5.7.1.1 gate_OR	38
5.8 gateXNOR.h File Reference	38
5.8.1 Macro Definition Documentation	39
5.8.1.1 gate_XNOR	39

5.9 gateXOR.h File Reference	39
5.9.1 Macro Definition Documentation	39
5.9.1.1 gate_XOR	39
5.10 inOut.h File Reference	39
5.11 logicFunctions.h File Reference	40
5.11.1 Function Documentation	40
5.11.1.1 readParams()	40
5.11.1.2 realize()	41
5.12 printFunctions.h File Reference	41
5.12.1 Function Documentation	41
5.12.1.1 print()	41
5.12.1.2 printErrorGates()	42
5.12.1.3 printHelp()	42
5.13 state.h File Reference	42
Index	43

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

FileManager	7
Gate	8
GateAND	12
GateNAND	14
GateNEG	17
GateNOR	19
GateOR	22
GateXNOR	24
GateXOR	27
InOut	29
ValueState	32

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

FileManager	7
Gate	8
GateAND	12
GateNAND	14
GateNEG	17
GateNOR	19
GateOR	22
GateXNOR	24
GateXOR	27
InOut	29
ValueState	32

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

fileManager.h	35
gate.h	35
gateAND.h	35
gateNAND.h	36
gateNEG.h	37
gateNOR.h	37
gateOR.h	38
gateXNOR.h	38
gateXOR.h	39
inOut.h	39
logicFunctions.h	40
printFunctions.h	41
state.h	42

Chapter 4

Class Documentation

4.1 FileManager Class Reference

```
#include <fileManager.h>
```

Public Member Functions

- int [loadGates](#) (gatesContainer &circuit, const std::string &fileName)
- int [loadStates](#) (std::vector< [ValueState](#) > &inputs, const std::string &fileName)
- void [save](#) (std::vector< [ValueState](#) > initialInput, std::vector< [ValueState](#) > finalInput, const std::string &fileName)

4.1.1 Detailed Description

Klasa reprezentująca plik zewnętrzny.

4.1.2 Member Function Documentation

4.1.2.1 loadGates()

```
int FileManager::loadGates (
    gatesContainer & circuit,
    const std::string & fileName )
```

Metoda odczytująca plik z układem bramek logicznych.

Parameters

<i>circuit</i>	obiekt przechowujący dane o odczytanym układzie.
<i>fileName</i>	zmienna tekstowa przechowująca nazwę pliku

Returns

zwraca wartości od -1 do -4 w zależności od wykrytego błędu, a w przypadku poprawnej realizacji wartość 1

4.1.2.2 loadStates()

```
int FileManager::loadStates (
    std::vector< ValueState > & inputs,
    const std::string & fileName )
```

Metoda odczytująca plik ze stanami początkowymi.

Parameters

<i>inputs</i>	obiekt przechowujący dane o odczytanych stanach logicznych
<i>fileName</i>	zmienna tekstowa przechowująca nazwę pliku

Returns

zwraca wartości -5 oraz -6 w zależności od wykrytego błędu, a w przypadku poprawnej realizacji wartość 1

4.1.2.3 save()

```
void FileManager::save (
    std::vector< ValueState > initialInput,
    std::vector< ValueState > finalInput,
    const std::string & fileName )
```

Metoda zapisująca wynik działania do pliku.

Parameters

<i>initialInput</i>	obiekt zawierający stany pierwotne
<i>finalInput</i>	obiekt zawierający stany wynikowe
<i>fileName</i>	zmienna tekstowa przechowująca nazwę pliku

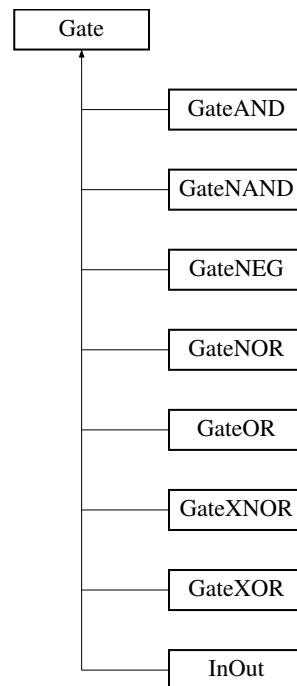
The documentation for this class was generated from the following files:

- [fileManager.h](#)
- fileManager.cpp

4.2 Gate Class Reference

```
#include <gate.h>
```

Inheritance diagram for Gate:



Public Member Functions

- virtual void [setInOutPuts](#) (std::string name, int in1, int out, int in2)=0
- virtual void [printGate](#) (std::ostream &stream)=0
- virtual void [operate](#) (std::vector< [ValueState](#) > &input)=0
- virtual int [getIn1](#) ()=0
- virtual int [getIn2](#) ()=0
- virtual int [getOut](#) ()=0

Protected Attributes

- std::string **Name**
- int **Out**
- int **In1**
- int **In2**

4.2.1 Detailed Description

Klasa abstrakcyjna opisująca bramkę logiczną, jest dziedziczona przez klasy reprezentujące konkretne bramki logiczne.

Parameters

<i>Name</i>	pole tekstowe przechowujące nazwę bramki
<i>Out</i>	numer wyjścia stanu logicznego
<i>In1</i>	pierwszy numer wejścia stanu logicznego
<i>In2</i>	drugi numer wejścia stanu logicznego

4.2.2 Member Function Documentation

4.2.2.1 getIn1()

```
virtual int Gate::getIn1 ( ) [pure virtual]
```

Metoda czysto wirtualna zwracająca numer pierwszego wejścia bramki.

Returns

zwraca numer pierwszego wejścia

Implemented in [GateAND](#), [GateNAND](#), [GateNEG](#), [GateNOR](#), [GateOR](#), [GateXNOR](#), [GateXOR](#), and [InOut](#).

4.2.2.2 getIn2()

```
virtual int Gate::getIn2 ( ) [pure virtual]
```

Metoda czysto wirtualna zwracająca numer drugiego wejścia bramki.

Returns

zwraca numer drugiego wejścia

Implemented in [GateAND](#), [GateNAND](#), [GateNEG](#), [GateNOR](#), [GateOR](#), [GateXNOR](#), [GateXOR](#), and [InOut](#).

4.2.2.3 getOut()

```
virtual int Gate::getOut ( ) [pure virtual]
```

Metoda czysto wirtualna zwracająca numer wyjścia bramki.

Returns

zwraca numer wyjścia

Implemented in [GateAND](#), [GateNAND](#), [GateNEG](#), [GateNOR](#), [GateOR](#), [GateXNOR](#), [GateXOR](#), and [InOut](#).

4.2.2.4 operate()

```
virtual void Gate::operate (
    std::vector< ValueState > & input ) [pure virtual]
```

Metoda czysto wirtualna symulująca działanie bramki logicznej.

Parameters

<i>input</i>	kontener zawierający wejściowe wartości stanów logicznych
--------------	---

Implemented in [GateAND](#), [GateNAND](#), [GateNEG](#), [GateNOR](#), [GateOR](#), [GateXNOR](#), [GateXOR](#), and [InOut](#).

4.2.2.5 printGate()

```
virtual void Gate::printGate (
    std::ostream & stream ) [pure virtual]
```

Metoda czysto wirtualna wypisująca bramkę logiczną.

Parameters

<i>stream</i>	strumień określający wyjście
---------------	------------------------------

Implemented in [GateAND](#), [GateNAND](#), [GateNEG](#), [GateNOR](#), [GateOR](#), [GateXNOR](#), [GateXOR](#), and [InOut](#).

4.2.2.6 setInOutPuts()

```
virtual void Gate::setInOutPuts (
    std::string name,
    int in1,
    int out,
    int in2 ) [pure virtual]
```

Czysto wirtualna metoda inicjalizująca wartości obiektu.

Parameters

<i>name</i>	nazwa bramki logicznej
<i>out</i>	numer wyjścia stanu logicznego
<i>in1</i>	pierwszy numer wejścia stanu logicznego
<i>in2</i>	drugi numer wejścia stanu logicznego

Implemented in [GateNEG](#), [GateAND](#), [GateNAND](#), [GateNOR](#), [GateOR](#), [GateXNOR](#), [GateXOR](#), and [InOut](#).

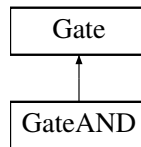
The documentation for this class was generated from the following file:

- [gate.h](#)

4.3 GateAND Class Reference

```
#include <gateAND.h>
```

Inheritance diagram for GateAND:



Public Member Functions

- virtual void [setInOutPuts](#) (std::string name, int in1, int out, int in2)
- virtual void [printGate](#) (std::ostream &stream)
- virtual void [operate](#) (std::vector< [ValueState](#) > &input)
- virtual int [getIn1](#) ()
- virtual int [getIn2](#) ()
- virtual int [getOut](#) ()

Additional Inherited Members

4.3.1 Detailed Description

Klasa pochodna, dziedzicząca po klasie abstrakcyjnej [Gate](#). Reprezentuje bramkę logiczną AND.

4.3.2 Member Function Documentation

4.3.2.1 [getIn1\(\)](#)

```
int GateAND::getIn1 ( ) [virtual]
```

Metoda wirtualna zwracająca numer pierwszego wejścia bramki AND.

Returns

zwraca numer pierwszego wejścia

Implements [Gate](#).

4.3.2.2 getIn2()

```
int GateAND::getIn2 ( ) [virtual]
```

Metoda wirtualna zwracająca numer drugiego wejścia bramki AND.

Returns

zwraca numer drugiego wejścia

Implements [Gate](#).

4.3.2.3 getOut()

```
int GateAND::getOut ( ) [virtual]
```

Metoda wirtualna zwracająca numer wyjścia bramki AND.

Returns

zwraca numer wyjścia

Implements [Gate](#).

4.3.2.4 operate()

```
void GateAND::operate (
    std::vector< ValueState > & input ) [virtual]
```

Metoda wirtualna symulująca działanie bramki logicznej AND.

Parameters

<i>input</i>	kontener zawierający wejściowe wartości stanów logicznych
--------------	---

Implements [Gate](#).

4.3.2.5 printGate()

```
void GateAND::printGate (
    std::ostream & stream ) [virtual]
```

Metoda wirtualna wypisująca bramkę logiczną AND.

Parameters

<i>stream</i>	strumień określający wyjście
---------------	------------------------------

Implements [Gate](#).

4.3.2.6 setInOutPuts()

```
void GateAND::setInOutPuts (
    std::string name,
    int in1,
    int out,
    int in2 ) [virtual]
```

Metoda wirtualna inicjalizująca wartości obiektu.

Parameters

<i>name</i>	nazwa bramki logicznej
<i>out</i>	numer wyjścia stanu logicznego
<i>in1</i>	pierwszy numer wejścia stanu logicznego
<i>in2</i>	drugi numer wejścia stanu logicznego

Implements [Gate](#).

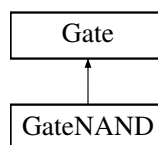
The documentation for this class was generated from the following files:

- [gateAND.h](#)
- [gateAND.cpp](#)

4.4 GateNAND Class Reference

```
#include <gateNAND.h>
```

Inheritance diagram for GateNAND:



Public Member Functions

- virtual void [setInOutPuts](#) (std::string name, int in1, int out, int in2)
- virtual void [printGate](#) (std::ostream &stream)
- virtual void [operate](#) (std::vector< [ValueState](#) > &input)
- virtual int [getIn1](#) ()
- virtual int [getIn2](#) ()
- virtual int [getOut](#) ()

Additional Inherited Members

4.4.1 Detailed Description

Klasa pochodna, dziedzicząca po klasie abstrakcyjnej [Gate](#). Reprezentuje bramkę logiczną NAND.

4.4.2 Member Function Documentation

4.4.2.1 getIn1()

```
int GateNAND::getIn1 ( ) [virtual]
```

Metoda wirtualna zwracająca numer pierwszego wejścia bramki NAND.

Returns

zwraca numer pierwszego wejścia

Implements [Gate](#).

4.4.2.2 getIn2()

```
int GateNAND::getIn2 ( ) [virtual]
```

Metoda wirtualna zwracająca numer drugiego wejścia bramki NAND.

Returns

zwraca numer drugiego wejścia

Implements [Gate](#).

4.4.2.3 getOut()

```
int GateNAND::getOut ( ) [virtual]
```

Metoda wirtualna zwracająca numer wyjścia bramki NAND.

Returns

zwraca numer wyjścia

Implements [Gate](#).

4.4.2.4 operate()

```
void GateNAND::operate (
    std::vector< ValueState > & input ) [virtual]
```

Metoda wirtualna symulująca działanie bramki logicznej NAND.

Parameters

<i>input</i>	kontener zawierający wejściowe wartości stanów logicznych
--------------	---

Implements [Gate](#).

4.4.2.5 printGate()

```
void GateNAND::printGate (
    std::ostream & stream ) [virtual]
```

Metoda wirtualna wypisująca bramkę logiczną NAND.

Parameters

<i>stream</i>	strumień określający wyjście
---------------	------------------------------

Implements [Gate](#).

4.4.2.6 setInOutPuts()

```
void GateNAND::setInOutPuts (
    std::string name,
    int in1,
    int out,
    int in2 ) [virtual]
```

Metoda wirtualna inicjalizująca wartości obiektu.

Parameters

<i>name</i>	nazwa bramki logicznej
<i>out</i>	numer wyjścia stanu logicznego
<i>in1</i>	pierwszy numer wejścia stanu logicznego
<i>in2</i>	drugi numer wejścia stanu logicznego

Implements [Gate](#).

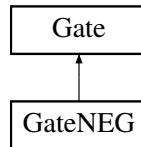
The documentation for this class was generated from the following files:

- [gateNAND.h](#)
- [gateNAND.cpp](#)

4.5 GateNEG Class Reference

```
#include <gateNEG.h>
```

Inheritance diagram for GateNEG:



Public Member Functions

- virtual void [setInOutPuts](#) (std::string name, int in1, int out, int in2=0)
- virtual void [printGate](#) (std::ostream &stream)
- virtual void [operate](#) (std::vector< [ValueState](#) > &input)
- virtual int [getIn1](#) ()
- virtual int [getIn2](#) ()
- virtual int [getOut](#) ()

Additional Inherited Members

4.5.1 Detailed Description

Klasa pochodna, dziedzicząca po klasie abstrakcyjnej [Gate](#). Reprezentuje bramkę logiczną NEG.

4.5.2 Member Function Documentation

4.5.2.1 [getIn1\(\)](#)

```
int GateNEG::getIn1 ( ) [virtual]
```

Metoda wirtualna zwracająca numer pierwszego wejścia bramki NEG.

Returns

zwraca numer pierwszego wejścia

Implements [Gate](#).

4.5.2.2 `getIn2()`

```
int GateNEG::getIn2 ( ) [virtual]
```

Metoda wirtualna zwracająca numer drugiego wejścia bramki NEG.

Returns

zwraca numer drugiego wejścia

Implements [Gate](#).

4.5.2.3 `getOut()`

```
int GateNEG::getOut ( ) [virtual]
```

Metoda wirtualna zwracająca numer wyjścia bramki NEG.

Returns

zwraca numer wyjścia

Implements [Gate](#).

4.5.2.4 `operate()`

```
void GateNEG::operate (
    std::vector< ValueState > & input ) [virtual]
```

Metoda wirtualna symulująca działanie bramki logicznej NEG.

Parameters

<i>input</i>	kontener zawierający wejściowe wartości stanów logicznych
--------------	---

Implements [Gate](#).

4.5.2.5 `printGate()`

```
void GateNEG::printGate (
    std::ostream & stream ) [virtual]
```

Metoda wirtualna wypisująca bramkę logiczną NEG.

Parameters

<i>stream</i>	strumień określający wyjście
---------------	------------------------------

Implements [Gate](#).

4.5.2.6 setInOutPuts()

```
void GateNEG::setInOutPuts (
    std::string name,
    int in1,
    int out,
    int in2 = 0 ) [virtual]
```

Metoda wirtualna inicjalizująca wartości obiektu.

Parameters

<i>name</i>	nazwa bramki logicznej
<i>out</i>	numer wyjścia stanu logicznego
<i>in1</i>	pierwszy numer wejścia stanu logicznego
<i>in2</i>	drugi numer wejścia stanu logicznego

Implements [Gate](#).

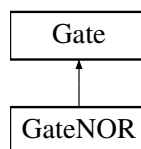
The documentation for this class was generated from the following files:

- [gateNEG.h](#)
- [gateNEG.cpp](#)

4.6 GateNOR Class Reference

```
#include <gateNOR.h>
```

Inheritance diagram for GateNOR:



Public Member Functions

- virtual void [setInOutPuts](#) (std::string name, int in1, int out, int in2)
- virtual void [printGate](#) (std::ostream &stream)
- virtual void [operate](#) (std::vector< [ValueState](#) > &input)
- virtual int [getIn1](#) ()
- virtual int [getIn2](#) ()
- virtual int [getOut](#) ()

Additional Inherited Members

4.6.1 Detailed Description

Klasa pochodna, dziedzicząca po klasie abstrakcyjnej [Gate](#). Reprezentuje bramkę logiczną NOR.

4.6.2 Member Function Documentation

4.6.2.1 `getIn1()`

```
int GateNOR::getIn1 ( ) [virtual]
```

Metoda wirtualna zwracająca numer pierwszego wejścia bramki NOR.

Returns

zwraca numer pierwszego wejścia

Implements [Gate](#).

4.6.2.2 `getIn2()`

```
int GateNOR::getIn2 ( ) [virtual]
```

Metoda wirtualna zwracająca numer drugiego wejścia bramki NOR.

Returns

zwraca numer drugiego wejścia

Implements [Gate](#).

4.6.2.3 `getOut()`

```
int GateNOR::getOut ( ) [virtual]
```

Metoda wirtualna zwracająca numer wyjścia bramki NOR.

Returns

zwraca numer wyjścia

Implements [Gate](#).

4.6.2.4 `operate()`

```
void GateNOR::operate (
    std::vector< ValueState > & input ) [virtual]
```

Metoda wirtualna symulująca działanie bramki logicznej NOR.

Parameters

<i>input</i>	kontener zawierający wejściowe wartości stanów logicznych
--------------	---

Implements [Gate](#).

4.6.2.5 printGate()

```
void GateNOR::printGate (
    std::ostream & stream ) [virtual]
```

Metoda wirtualna wypisująca bramkę logiczną NOR.

Parameters

<i>stream</i>	strumień określający wyjście
---------------	------------------------------

Implements [Gate](#).

4.6.2.6 setInOutPuts()

```
void GateNOR::setInOutPuts (
    std::string name,
    int in1,
    int out,
    int in2 ) [virtual]
```

Metoda wirtualna inicjalizująca wartości obiektu.

Parameters

<i>name</i>	nazwa bramki logicznej
<i>out</i>	numer wyjścia stanu logicznego
<i>in1</i>	pierwszy numer wejścia stanu logicznego
<i>in2</i>	drugi numer wejścia stanu logicznego

Implements [Gate](#).

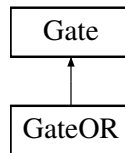
The documentation for this class was generated from the following files:

- [gateNOR.h](#)
- [gateNOR.cpp](#)

4.7 GateOR Class Reference

```
#include <gateOR.h>
```

Inheritance diagram for GateOR:



Public Member Functions

- virtual void [setInOutPuts](#) (std::string name, int in1, int out, int in2)
- virtual void [printGate](#) (std::ostream &stream)
- virtual void [operate](#) (std::vector< [ValueState](#) > &input)
- virtual int [getIn1](#) ()
- virtual int [getIn2](#) ()
- virtual int [getOut](#) ()

Additional Inherited Members

4.7.1 Detailed Description

Klasa pochodna, dziedzicząca po klasie abstrakcyjnej [Gate](#). Reprezentuje bramkę logiczną OR.

4.7.2 Member Function Documentation

4.7.2.1 [getIn1\(\)](#)

```
int GateOR::getIn1 ( ) [virtual]
```

Metoda wirtualna zwracająca numer pierwszego wejścia bramki OR.

Returns

zwraca numer pierwszego wejścia

Implements [Gate](#).

4.7.2.2 getIn2()

```
int GateOR::getIn2 ( ) [virtual]
```

Metoda wirtualna zwracająca numer drugiego wejścia bramki OR.

Returns

zwraca numer drugiego wejścia

Implements [Gate](#).

4.7.2.3 getOut()

```
int GateOR::getOut ( ) [virtual]
```

Metoda wirtualna zwracająca numer wyjścia bramki OR.

Returns

zwraca numer wyjścia

Implements [Gate](#).

4.7.2.4 operate()

```
void GateOR::operate (
    std::vector< ValueState > & input ) [virtual]
```

Metoda wirtualna symulująca działanie bramki logicznej OR.

Parameters

<i>input</i>	kontener zawierający wejściowe wartości stanów logicznych
--------------	---

Implements [Gate](#).

4.7.2.5 printGate()

```
void GateOR::printGate (
    std::ostream & stream ) [virtual]
```

Metoda wirtualna wypisująca bramkę logiczną OR.

Parameters

<i>stream</i>	strumień określający wyjście
---------------	------------------------------

Implements [Gate](#).

4.7.2.6 setInOutPuts()

```
void GateOR::setInOutPuts (
    std::string name,
    int in1,
    int out,
    int in2 ) [virtual]
```

Metoda wirtualna inicjalizująca wartości obiektu.

Parameters

<i>name</i>	nazwa bramki logicznej
<i>out</i>	numer wyjścia stanu logicznego
<i>in1</i>	pierwszy numer wejścia stanu logicznego
<i>in2</i>	drugi numer wejścia stanu logicznego

Implements [Gate](#).

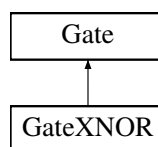
The documentation for this class was generated from the following files:

- [gateOR.h](#)
- [gateOR.cpp](#)

4.8 GateXNOR Class Reference

```
#include <gateXNOR.h>
```

Inheritance diagram for GateXNOR:



Public Member Functions

- virtual void [setInOutPuts](#) (std::string name, int in1, int out, int in2)
- virtual void [printGate](#) (std::ostream &stream)
- virtual void [operate](#) (std::vector< [ValueState](#) > &input)
- virtual int [getIn1](#) ()
- virtual int [getIn2](#) ()
- virtual int [getOut](#) ()

Additional Inherited Members

4.8.1 Detailed Description

Klasa pochodna, dziedzicząca po klasie abstrakcyjnej [Gate](#). Reprezentuje bramkę logiczną XNOR.

4.8.2 Member Function Documentation

4.8.2.1 `getIn1()`

```
int GateXNOR::getIn1 ( ) [virtual]
```

Metoda wirtualna zwracająca numer pierwszego wejścia bramki XNOR.

Returns

zwraca numer pierwszego wejścia

Implements [Gate](#).

4.8.2.2 `getIn2()`

```
int GateXNOR::getIn2 ( ) [virtual]
```

Metoda wirtualna zwracająca numer drugiego wejścia bramki XNOR.

Returns

zwraca numer drugiego wejścia

Implements [Gate](#).

4.8.2.3 `getOut()`

```
int GateXNOR::getOut ( ) [virtual]
```

Metoda wirtualna zwracająca numer wyjścia bramki XNOR.

Returns

zwraca numer wyjścia

Implements [Gate](#).

4.8.2.4 `operate()`

```
void GateXNOR::operate (
    std::vector< ValueState > & input ) [virtual]
```

Metoda wirtualna symulująca działanie bramki logicznej XNOR.

Parameters

<i>input</i>	kontener zawierający wejściowe wartości stanów logicznych
--------------	---

Implements [Gate](#).

4.8.2.5 printGate()

```
void GateXNOR::printGate (
    std::ostream & stream ) [virtual]
```

Metoda wirtualna wypisująca bramkę logiczną XNOR.

Parameters

<i>stream</i>	strumień określający wyjście
---------------	------------------------------

Implements [Gate](#).

4.8.2.6 setInOutPuts()

```
void GateXNOR::setInOutPuts (
    std::string name,
    int in1,
    int out,
    int in2 ) [virtual]
```

Metoda wirtualna inicjalizująca wartości obiektu.

Parameters

<i>name</i>	nazwa bramki logicznej
<i>out</i>	numer wyjścia stanu logicznego
<i>in1</i>	pierwszy numer wejścia stanu logicznego
<i>in2</i>	drugi numer wejścia stanu logicznego

Implements [Gate](#).

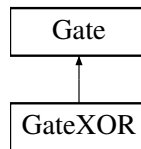
The documentation for this class was generated from the following files:

- [gateXNOR.h](#)
- [gateXNOR.cpp](#)

4.9 GateXOR Class Reference

```
#include <gateXOR.h>
```

Inheritance diagram for GateXOR:



Public Member Functions

- virtual void [setInOutPuts](#) (std::string name, int in1, int out, int in2)
- virtual void [printGate](#) (std::ostream &stream)
- virtual void [operate](#) (std::vector< [ValueState](#) > &input)
- virtual int [getIn1](#) ()
- virtual int [getIn2](#) ()
- virtual int [getOut](#) ()

Additional Inherited Members

4.9.1 Detailed Description

Klasa pochodna, dziedzicząca po klasie abstrakcyjnej [Gate](#). Reprezentuje bramkę logiczną XOR.

4.9.2 Member Function Documentation

4.9.2.1 [getIn1\(\)](#)

```
int GateXOR::getIn1 ( ) [virtual]
```

Metoda wirtualna zwracająca numer pierwszego wejścia bramki XOR.

Returns

zwraca numer pierwszego wejścia

Implements [Gate](#).

4.9.2.2 `getIn2()`

```
int GateXOR::getIn2 ( ) [virtual]
```

Metoda wirtualna zwracająca numer drugiego wejścia bramki XOR.

Returns

zwraca numer drugiego wejścia

Implements [Gate](#).

4.9.2.3 `getOut()`

```
int GateXOR::getOut ( ) [virtual]
```

Metoda wirtualna zwracająca numer wyjścia bramki XOR.

Returns

zwraca numer wyjścia

Implements [Gate](#).

4.9.2.4 `operate()`

```
void GateXOR::operate (
    std::vector< ValueState > & input ) [virtual]
```

Metoda wirtualna symulująca działanie bramki logicznej XOR.

Parameters

<i>input</i>	kontener zawierający wejściowe wartości stanów logicznych
--------------	---

Implements [Gate](#).

4.9.2.5 `printGate()`

```
void GateXOR::printGate (
    std::ostream & stream ) [virtual]
```

Metoda wirtualna wypisująca bramkę logiczną XOR.

Parameters

<i>stream</i>	strumień określający wyjście
---------------	------------------------------

Implements [Gate](#).

4.9.2.6 setInOutPuts()

```
void GateXOR::setInOutPuts (
    std::string name,
    int in1,
    int out,
    int in2 ) [virtual]
```

Metoda wirtualna inicjalizująca wartości obiektu.

Parameters

<i>name</i>	nazwa bramki logicznej
<i>out</i>	numer wyjścia stanu logicznego
<i>in1</i>	pierwszy numer wejścia stanu logicznego
<i>in2</i>	drugi numer wejścia stanu logicznego

Implements [Gate](#).

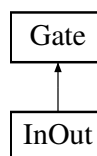
The documentation for this class was generated from the following files:

- [gateXOR.h](#)
- [gateXOR.cpp](#)

4.10 InOut Class Reference

```
#include <InOut.h>
```

Inheritance diagram for InOut:



Public Member Functions

- virtual void [setInOutPuts](#) (std::string name, int in1, int out, int in2)
- virtual void [printGate](#) (std::ostream &stream)
- virtual void [operate](#) (std::vector< [ValueState](#) > &input)
- virtual int [getIn1](#) ()
- virtual int [getIn2](#) ()
- virtual int [getOut](#) ()

Additional Inherited Members

4.10.1 Detailed Description

Klasa pochodna, dziedzicząca po klasie abstrakcyjnej [Gate](#). Reprezentuje stany pierwotne oraz stan wynikowy.

4.10.2 Member Function Documentation

4.10.2.1 `getIn1()`

```
int InOut::getIn1 ( ) [virtual]
```

Metoda wirtualna zwracająca numer pierwszego wejścia.

Returns

zwraca numer pierwszego wejścia

Implements [Gate](#).

4.10.2.2 `getIn2()`

```
int InOut::getIn2 ( ) [virtual]
```

Metoda wirtualna zwracająca numer drugiego wejścia.

Returns

zwraca numer drugiego wejścia

Implements [Gate](#).

4.10.2.3 `getOut()`

```
int InOut::getOut ( ) [virtual]
```

Metoda wirtualna zwracająca numer wyjścia.

Returns

zwraca numer wyjścia

Implements [Gate](#).

4.10.2.4 `operate()`

```
void InOut::operate (
    std::vector< ValueState > & input ) [virtual]
```

Metoda wirtualna symulująca działanie.

Parameters

<i>input</i>	kontener zawierający wejściowe wartości stanów logicznych
--------------	---

Implements [Gate](#).

4.10.2.5 printGate()

```
void InOut::printGate (
    std::ostream & stream ) [virtual]
```

Metoda wirtualna wypisująca stany wejściowe oraz wynikowy.

Parameters

<i>stream</i>	strumień określający wyjście
---------------	------------------------------

Implements [Gate](#).

4.10.2.6 setInOutPuts()

```
void InOut::setInOutPuts (
    std::string name,
    int in1,
    int out,
    int in2 ) [virtual]
```

Metoda wirtualna inicjalizująca wartości obiektu.

Parameters

<i>name</i>	nazwa wejścia lub wyjścia logicznego
<i>out</i>	numer wyjścia stanu logicznego
<i>in1</i>	pierwszy numer wejścia stanu logicznego
<i>in2</i>	drugi numer wejścia stanu logicznego

Implements [Gate](#).

The documentation for this class was generated from the following files:

- [inOut.h](#)
- [inOut.cpp](#)

4.11 ValueState Class Reference

```
#include <state.h>
```

Public Member Functions

- void [setState](#) (int in, bool state)
- void [printState](#) (std::ostream &stream)
- void [printFinalState](#) (std::ostream &stream)
- int [getIn](#) ()
- bool [getState](#) ()

4.11.1 Detailed Description

Klasa reprezentująca pojedynczy stan logiczny.

Parameters

<i>In</i>	numer stanu
<i>state</i>	wartość stanu

4.11.2 Member Function Documentation

4.11.2.1 [getIn\(\)](#)

```
int ValueState::getIn ( )
```

Metoda zwracająca numer stanu logicznego.

Returns

zwraca numer stanu

4.11.2.2 [getState\(\)](#)

```
bool ValueState::getState ( )
```

Metoda zwracająca wartość stanu logicznego.

Returns

zwraca wartość stanu

4.11.2.3 printFinalState()

```
void ValueState::printFinalState (
    std::ostream & stream )
```

Metoda wypisująca wynikowy stan logiczny na podane wyjście.

Parameters

<i>stream</i>	strumień określający wyjście
---------------	------------------------------

4.11.2.4 printState()

```
void ValueState::printState (
    std::ostream & stream )
```

Metoda wypisująca początkowy stan logiczny na podane wyjście.

Parameters

<i>stream</i>	strumień określający wyjście
---------------	------------------------------

4.11.2.5 setState()

```
void ValueState::setState (
    int in,
    bool state )
```

Metoda ustawiająca wartości pól obiektu.

Parameters

<i>in</i>	numer stanu
<i>state</i>	wartość stanu

The documentation for this class was generated from the following files:

- [state.h](#)
- [state.cpp](#)

Chapter 5

File Documentation

5.1 fileManager.h File Reference

```
#include "logicFunctions.h"  
#include "printFunctions.h"
```

Classes

- class [FileManager](#)

5.2 gate.h File Reference

```
#include <iostream>  
#include <vector>  
#include <fstream>  
#include <string>  
#include <sstream>  
#include <memory>  
#include "state.h"
```

Classes

- class [Gate](#)

5.3 gateAND.h File Reference

```
#include "gate.h"
```

Classes

- class [GateAND](#)

Macros

- `#define gate_AND "AND"`

5.3.1 Macro Definition Documentation

5.3.1.1 `gate_AND`

```
#define gate_AND "AND"
```

Łańcuch znakowy określający nazwę bramki AND.

5.4 `gateNAND.h` File Reference

```
#include "gate.h"
```

Classes

- class [GateNAND](#)

Macros

- `#define gate_NAND "NAND"`

5.4.1 Macro Definition Documentation

5.4.1.1 `gate_NAND`

```
#define gate_NAND "NAND"
```

Łańcuch znakowy określający nazwę bramki NAND.

5.5 gateNEG.h File Reference

```
#include "gate.h"
```

Classes

- class [GateNEG](#)

Macros

- #define [gate_NEG](#) "NEG"

5.5.1 Macro Definition Documentation

5.5.1.1 gate_NEG

```
#define gate_NEG "NEG"
```

Łańcuch znakowy określający nazwę bramki NEG.

5.6 gateNOR.h File Reference

```
#include "gate.h"
```

Classes

- class [GateNOR](#)

Macros

- #define [gate_NOR](#) "NOR"

5.6.1 Macro Definition Documentation

5.6.1.1 gate_NOR

```
#define gate_NOR "NOR"
```

Łańcuch znakowy określający nazwę bramki NOR.

5.7 gateOR.h File Reference

```
#include "gate.h"
```

Classes

- class [GateOR](#)

Macros

- #define [gate_OR](#) "OR"

5.7.1 Macro Definition Documentation

5.7.1.1 gate_OR

```
#define gate_OR "OR"
```

Łańcuch znakowy określający nazwę bramki OR.

5.8 gateXNOR.h File Reference

```
#include "gate.h"
```

Classes

- class [GateXNOR](#)

Macros

- #define [gate_XNOR](#) "XNOR"

5.8.1 Macro Definition Documentation

5.8.1.1 gate_XNOR

```
#define gate_XNOR "XNOR"
```

Łańcuch znakowy określający nazwę bramki XNOR.

5.9 gateXOR.h File Reference

```
#include "gate.h"
```

Classes

- class [GateXOR](#)

Macros

- #define [gate_XOR](#) "XOR"

5.9.1 Macro Definition Documentation

5.9.1.1 gate_XOR

```
#define gate_XOR "XOR"
```

Łańcuch znakowy określający nazwę bramki XOR.

5.10 inOut.h File Reference

```
#include "gate.h"
```

Classes

- class [InOut](#)

5.11 logicFunctions.h File Reference

```
#include "gateAND.h"
#include "gateNAND.h"
#include "gateNEG.h"
#include "gateNOR.h"
#include "gateOR.h"
#include "gateXNOR.h"
#include "gateXOR.h"
#include "inOut.h"
```

Typedefs

- typedef std::vector< std::shared_ptr< std::string > > **wordChain**
- typedef std::vector< std::shared_ptr< [Gate](#) > > **gatesContainer**

Functions

- int [readParams](#) (int argc, char *argv[], wordChain &fileNames, int &first, int &second, int &third)
- void [realize](#) (gatesContainer &circuit, std::vector< [ValueState](#) > copyInput, std::vector< [ValueState](#) > &finalInput)

5.11.1 Function Documentation

5.11.1.1 readParams()

```
int readParams (
    int argc,
    char * argv[],
    wordChain & fileNames,
    int & first,
    int & second,
    int & third )
```

Funkcja odczytująca parametry podane podczas uruchamiania programu.

Parameters

<i>argc</i>	liczba parametrów z lini poleceń
<i>argv</i>	tablica słów z lini poleceń
<i>fileNames</i>	obiekt przechowujący nazwy plików wydzielone z argv
<i>first</i>	określa miejsce przechowywania nazwy pliku zawierającego układ bramek w obiekcie fileNames
<i>second</i>	określa miejsce przechowywania nazwy pliku zawierającego stany wejściowe w obiekcie fileNames
<i>third</i>	określa miejsce przechowywania, w obiekcie fileNames, nazwy jaką ma zostać nazwany plik wynikowy

Returns

zwraca wartość -7 w przypadku wykrycia błędu, a gdy funkcja została zrealizowana poprawnie wartość 1

5.11.1.2 realize()

```
void realize (
    gatesContainer & circuit,
    std::vector< ValueState > copyInput,
    std::vector< ValueState > & finalInput )
```

Funkcja realizująca symulację bramek logicznych.

Parameters

<i>circuit</i>	obiekt przechowujący dane o układzie bramek
<i>copyInput</i>	obiekt przechowujący dane o stanach pierwotnych
<i>finalInput</i>	obiekt, do którego zostaje zapisane wyniki stanów wyjść

5.12 printFunctions.h File Reference

```
#include "logicFunctions.h"
```

Functions

- bool [printErrorGates](#) (int bug)
- void [print](#) (gatesContainer copyCircuit, std::vector< [ValueState](#) > copyInput, std::vector< [ValueState](#) > copyFinalInput)
- void [printHelp](#) ()

5.12.1 Function Documentation**5.12.1.1 print()**

```
void print (
    gatesContainer copyCircuit,
    std::vector< ValueState > copyInput,
    std::vector< ValueState > copyFinalInput )
```

Funkcja wypisuje na ekran zawartość odczytanych plików oraz zawartość utworzonego pliku wynikowego.

Parameters

<i>copyCircuit</i>	obiekt zawiera dane o odczytanym układzie bramek
<i>copyInput</i>	obiekt zawiera dane o odczytanych stanach początkowych
<i>copyFinalInput</i>	obiekt zawiera dane o stanach wynikowych

5.12.1.2 printErrorGates()

```
bool printErrorGates (
    int bug )
```

Funkcja wypisuje na ekran informacje o błędzie w zależności od przyjętej wartości.

Parameters

<i>bug</i>	przyjmuje wartość o błędzie
------------	-----------------------------

Returns

zwraca false w momencie znalezienia dopasowania wartości do błędu w przeciwnym wypadku zwraca true

5.12.1.3 printHelp()

```
void printHelp ( )
```

Funkcja wypisuje na ekran instrukcje obsługi przy błędnie uruchomionym programie

5.13 state.h File Reference

```
#include <iostream>
```

Classes

- class [ValueState](#)

Index

FileManager, 7
 loadGates, 7
 loadStates, 8
 save, 8
fileManager.h, 35

Gate, 8
 getIn1, 10
 getIn2, 10
 getOut, 10
 operate, 10
 printGate, 11
 setInOutPuts, 11
gate.h, 35
gate_AND
 gateAND.h, 36
gate_NAND
 gateNAND.h, 36
gate_NEG
 gateNEG.h, 37
gate_NOR
 gateNOR.h, 37
gate_OR
 gateOR.h, 38
gate_XNOR
 gateXNOR.h, 39
gate_XOR
 gateXOR.h, 39
GateAND, 12
 getIn1, 12
 getIn2, 12
 getOut, 13
 operate, 13
 printGate, 13
 setInOutPuts, 14
gateAND.h, 35
 gate_AND, 36
GateNAND, 14
 getIn1, 15
 getIn2, 15
 getOut, 15
 operate, 15
 printGate, 16
 setInOutPuts, 16
gateNAND.h, 36
 gate_NAND, 36
GateNEG, 17
 getIn1, 17
 getIn2, 17
 getOut, 18
 operate, 18
 printGate, 18
 setInOutPuts, 19
gateNEG.h, 37
 gate_NEG, 37
GateNOR, 19
 getIn1, 20
 getIn2, 20
 getOut, 20
 operate, 20
 printGate, 21
 setInOutPuts, 21
gateNOR.h, 37
 gate_NOR, 37
GateOR, 22
 getIn1, 22
 getIn2, 22
 getOut, 23
 operate, 23
 printGate, 23
 setInOutPuts, 24
gateOR.h, 38
 gate_OR, 38
GateXNOR, 24
 getIn1, 25
 getIn2, 25
 getOut, 25
 operate, 25
 printGate, 26
 setInOutPuts, 26
gateXNOR.h, 38
 gate_XNOR, 39
GateXOR, 27
 getIn1, 27
 getIn2, 27
 getOut, 28
 operate, 28
 printGate, 28
 setInOutPuts, 29
gateXOR.h, 39
 gate_XOR, 39
getIn
 ValueState, 32
getIn1
 Gate, 10
 GateAND, 12
 GateNAND, 15
 GateNEG, 17
 GateNOR, 20

- GateOR, [22](#)
- GateXNOR, [25](#)
- GateXOR, [27](#)
- InOut, [30](#)
- getIn2
 - Gate, [10](#)
 - GateAND, [12](#)
 - GateNAND, [15](#)
 - GateNEG, [17](#)
 - GateNOR, [20](#)
 - GateOR, [22](#)
 - GateXNOR, [25](#)
 - GateXOR, [27](#)
 - InOut, [30](#)
- getOut
 - Gate, [10](#)
 - GateAND, [13](#)
 - GateNAND, [15](#)
 - GateNEG, [18](#)
 - GateNOR, [20](#)
 - GateOR, [23](#)
 - GateXNOR, [25](#)
 - GateXOR, [28](#)
 - InOut, [30](#)
- getState
 - ValueState, [32](#)
- InOut, [29](#)
 - getIn1, [30](#)
 - getIn2, [30](#)
 - getOut, [30](#)
 - operate, [30](#)
 - printGate, [31](#)
 - setInOutPuts, [31](#)
- inOut.h, [39](#)
- loadGates
 - FileManager, [7](#)
- loadStates
 - FileManager, [8](#)
- logicFunctions.h, [40](#)
 - readParams, [40](#)
 - realize, [41](#)
- operate
 - Gate, [10](#)
 - GateAND, [13](#)
 - GateNAND, [15](#)
 - GateNEG, [18](#)
 - GateNOR, [20](#)
 - GateOR, [23](#)
 - GateXNOR, [25](#)
 - GateXOR, [28](#)
 - InOut, [30](#)
- print
 - printFunctions.h, [41](#)
- printErrorGates
 - printFunctions.h, [42](#)
- printFinalState
 - ValueState, [32](#)
- printFunctions.h, [41](#)
 - print, [41](#)
 - printErrorGates, [42](#)
 - printHelp, [42](#)
- printGate
 - Gate, [11](#)
 - GateAND, [13](#)
 - GateNAND, [16](#)
 - GateNEG, [18](#)
 - GateNOR, [21](#)
 - GateOR, [23](#)
 - GateXNOR, [26](#)
 - GateXOR, [28](#)
 - InOut, [31](#)
- printHelp
 - printFunctions.h, [42](#)
- printState
 - ValueState, [33](#)
- readParams
 - logicFunctions.h, [40](#)
- realize
 - logicFunctions.h, [41](#)
- save
 - FileManager, [8](#)
- setInOutPuts
 - Gate, [11](#)
 - GateAND, [14](#)
 - GateNAND, [16](#)
 - GateNEG, [19](#)
 - GateNOR, [21](#)
 - GateOR, [24](#)
 - GateXNOR, [26](#)
 - GateXOR, [29](#)
 - InOut, [31](#)
- setState
 - ValueState, [33](#)
- state.h, [42](#)
- ValueState, [32](#)
 - getIn, [32](#)
 - getState, [32](#)
 - printFinalState, [32](#)
 - printState, [33](#)
 - setState, [33](#)