

Digit Simulator

0.9.0

Generated by Doxygen 1.10.0

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 AND Class Reference	7
4.1.1 Member Function Documentation	9
4.1.1.1 performLogicCalculation()	9
4.2 Circuit Class Reference	10
4.3 Component Class Reference	10
4.3.1 Constructor & Destructor Documentation	11
4.3.1.1 Component()	11
4.3.2 Member Function Documentation	11
4.3.2.1 executeFunction()	11
4.4 Gate Class Reference	12
4.4.1 Member Function Documentation	14
4.4.1.1 executeFunction()	14
4.5 InPin Class Reference	14
4.5.1 Constructor & Destructor Documentation	15
4.5.1.1 InPin()	15
4.5.2 Member Function Documentation	15
4.5.2.1 connectToComponent()	15
4.5.2.2 SignalReady()	16
4.6 InPin_Component Class Reference	16
4.6.1 Constructor & Destructor Documentation	17
4.6.1.1 InPin_Component()	17
4.6.2 Member Function Documentation	17
4.6.2.1 executeFunction()	17
4.7 IOPin_Component Class Reference	18
4.7.1 Constructor & Destructor Documentation	20
4.7.1.1 IOPin_Component()	20
4.7.2 Member Function Documentation	20
4.7.2.1 executeFunction()	20
4.8 Lamp Class Reference	20
4.8.1 Member Function Documentation	22
4.8.1.1 executeFunction()	22
4.8.1.2 getState()	22
4.9 NAND Class Reference	22

4.10 Node Class Reference	25
4.10.1 Member Function Documentation	26
4.10.1.1 executeFunction()	26
4.11 NOR Class Reference	26
4.12 NOT Class Reference	29
4.13 OR Class Reference	31
4.13.1 Member Function Documentation	33
4.13.1.1 performLogicCalculation()	33
4.14 OutPin Class Reference	33
4.14.1 Constructor & Destructor Documentation	34
4.14.1.1 OutPin()	34
4.14.2 Member Function Documentation	34
4.14.2.1 connectToPin()	34
4.14.2.2 sendSignal()	34
4.15 OutPin_Component Class Reference	35
4.15.1 Constructor & Destructor Documentation	36
4.15.1.1 OutPin_Component()	36
4.15.2 Member Function Documentation	36
4.15.2.1 connectTo()	36
4.15.2.2 executeFunction()	36
4.16 Pin Class Reference	37
4.16.1 Constructor & Destructor Documentation	37
4.16.1.1 Pin()	37
4.16.2 Member Function Documentation	38
4.16.2.1 getSignal()	38
4.16.2.2 setSignal()	38
4.17 Queue< T > Class Template Reference	38
4.17.1 Constructor & Destructor Documentation	39
4.17.1.1 Queue() [1/2]	39
4.17.1.2 Queue() [2/2]	39
4.17.2 Member Function Documentation	39
4.17.2.1 get()	39
4.17.2.2 isEmpty()	40
4.17.2.3 put()	40
4.17.2.4 size()	40
4.18 QueueMember< T > Struct Template Reference	40
4.19 Signal Class Reference	41
4.19.1 Constructor & Destructor Documentation	41
4.19.1.1 Signal()	41
4.19.2 Member Function Documentation	41
4.19.2.1 getValue()	41
4.19.2.2 operator==()	41

4.19.2.3 setValue()	42
4.20 Source Class Reference	42
4.20.1 Member Function Documentation	43
4.20.1.1 executeFunction()	43
4.20.1.2 getOutput()	44
4.20.1.3 setOutput()	44
4.21 Switch Class Reference	44
4.21.1 Member Function Documentation	46
4.21.1.1 executeFunction()	46
4.21.1.2 getState()	46
4.22 XNOR Class Reference	47
4.23 XOR Class Reference	49
4.23.1 Member Function Documentation	51
4.23.1.1 performLogicCalculation()	51
5 File Documentation	53
5.1 Circuit.h	53
5.2 Component.h	54
5.3 Gate.h	55
5.4 Lamp.h	56
5.5 Node.h	57
5.6 Pin.h	57
5.7 Queue.h	57
5.8 Signal.h	59
5.9 Source.h	59
5.10 Switch.h	60
Index	61

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Circuit	10
Component	10
InPin_Component	16
IOPin_Component	18
Gate	12
AND	7
NAND	22
NOT	29
OR	31
NOR	26
XOR	49
XNOR	47
Switch	44
Lamp	20
Node	25
OutPin_Component	35
IOPin_Component	18
Source	42
Pin	37
InPin	14
OutPin	33
Queue< T >	38
Queue< Component >	38
Queue< InPin_Component >	38
Queue< Lamp >	38
Queue< Node >	38
Queue< OutPin >	38
Queue< Source >	38
Queue< Switch >	38
QueueMember< T >	40
QueueMember< Component >	40
QueueMember< InPin_Component >	40
QueueMember< Lamp >	40
QueueMember< Node >	40
QueueMember< OutPin >	40
QueueMember< Source >	40
QueueMember< Switch >	40
Signal	41

Chapter 2

Class Index

2.1 Class List

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

AND	7
Circuit	10
Component	10
Gate	12
InPin	14
InPin_Component	16
IOPin_Component	18
Lamp	20
NAND	22
Node	25
NOR	26
NOT	29
OR	31
OutPin	33
OutPin_Component	35
Pin	37
Queue< T >	38
QueueMember< T >	40
Signal	41
Source	42
Switch	44
XNOR	47
XOR	49

Chapter 3

File Index

3.1 File List

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

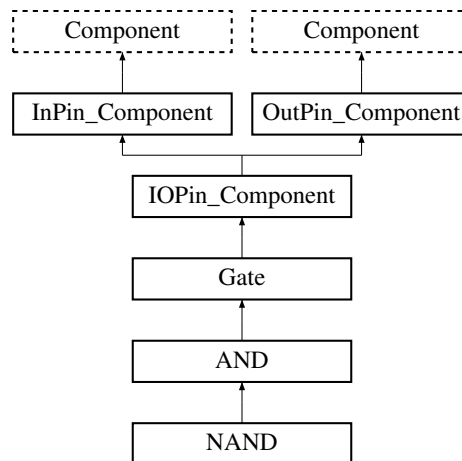
C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/ Circuit.h	53
C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/ Component.h	54
C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/ Gate.h	55
C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/ Lamp.h	56
C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/ Node.h	57
C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/ Pin.h	57
C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/ Queue.h	57
C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/ Signal.h	59
C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/ Source.h	59
C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/ Switch.h	60

Chapter 4

Class Documentation

4.1 AND Class Reference

Inheritance diagram for AND:



Public Member Functions

- **AND** (size_t inCount)

Public Member Functions inherited from **Gate**

- **Gate** (size_t inCount, size_t outCount)
- virtual void **executeFunction** ()

Végrehajtja a funkcióját az áramköri elemnek. Leszármazottban konkretizálva.

Public Member Functions inherited from **IOPin_Component**

- **IOPin_Component** (size_t inCount, size_t outCount)
Létrehozza a lábakat.
- bool **connectedToNodes** (size_t connectedNode1, size_t connectedNode2)
- void **printConnectedNodes** (std::ostream &os) const
- virtual **~IOPin_Component** ()

Virtuális destruktor, mert öröklés.

Public Member Functions inherited from [InPin_Component](#)

- [InPin_Component](#) (size_t inCount)
Létrehozza a bemeneti pin-ek tömbjét.
- void **resetActiveCount** ()
Reseteli az aktív pin-ek számát 0-ra.
- void **setInNodeID** (size_t at, size_t id)
- bool **connectedToNodeIn** (size_t id)
- void **printlnConnectedNodes** (std::ostream &os) const
- void **resetForSimulation** ()
- void **tickCounter** ()
Növeli az aktív lábak számát, és aktivizálja az áramköri elemet, ha eléri a maximumot.
- [InPin](#) * **getInPinsBaseAdress** () const
Visszaadja a bemeneti pin-ek tömbjének címét.
- virtual **~InPin_Component** ()
Törli a bemeneti pin-ek tömbjét.

Public Member Functions inherited from [Component](#)

- [Component](#) ([Queue](#)< [Component](#) > *newActiveQueue=nullptr)
Felparaméterezi az aktív sort.
- void **setActiveQueue** ([Queue](#)< [Component](#) > *newActiveQueue)
Beállítja az aktív fifo-t.
- void **addToActiveQueue** ()
Hozzáadja az aktív sorhoz az áramköri elemet.
- virtual **~Component** ()
Virtuális destruktork öröklés miatt.

Public Member Functions inherited from [OutPin_Component](#)

- [OutPin_Component](#) (size_t outCount)
Létrehozza a kimeneti pin-ek tömbjét.
- void **connectTo** (size_t outPinIndex, [InPin_Component](#) *component, size_t inPinIndex)
Összeköti a megadott kimeneti pin-jét egy másik áramköri elem bemeneti pin-jével.
- void **sendOutSignals** ()
Kiküldi minden kimeneti lábán a lábakban tárolt jeleket.
- [OutPin](#) * **getOutPinBaseAdress** ()
- void **setOutNodeID** (size_t at, size_t id)
- bool **connectedToNodeOut** (size_t id)
- void **printOutConnectedNodes** (std::ostream &os) const
- virtual **~OutPin_Component** ()
Törli a kimeneti pin-ek tömbjét.

Protected Member Functions

- virtual void **performLogicCalculation** ()

Additional Inherited Members

Protected Attributes inherited from [InPin_Component](#)

- `size_t inPinCount`
A bemeneti pin-ek száma.
- `size_t activeInPins`
Az aktív, valid adattal rendelkező pin-ek száma.
- `InPin * inPins`
A bemeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- `size_t * inNodeIDs`

Protected Attributes inherited from [Component](#)

- `Queue< Component > * activeQueue`
Az aktív sor, amihez kell hozzáadni, ha ki kell értékelni az elemet, azaz végrehajtani a funkcióját.

Protected Attributes inherited from [OutPin_Component](#)

- `size_t outPinCount`
A kimeneti pin-ek száma.
- `OutPin * outPins`
A kimeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- `size_t * outPinIDs`

4.1.1 Member Function Documentation

4.1.1.1 `performLogicCalculation()`

```
void AND::performLogicCalculation ( ) [protected], [virtual]
```

Implements [Gate](#).

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

- `C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Gate.h`
- `C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Gate.cpp`

4.2 Circuit Class Reference

Public Member Functions

- **Circuit** (const [Circuit](#) &source)
- **Circuit** & **operator=** (const [Circuit](#) &source)
- void **setSourceFile** (const std::string &fileName)
- const std::string & **getSourceFileName** () const
- void **simulate** (std::ostream &os)
- void **setSource** (size_t connectedNode, [Signal](#) newSignal)
- void **setAllSources** (int *connectedNodes, [Signal](#) *newSignals)
- void **flipSource** (size_t connectedNode)
- void **flipAllSources** ()
- [Signal](#) **getSourceSignal** (size_t connectedNode) const
- void **printSourceState** (std::ostream &os) const
- void **printAllSourceStates** (std::ostream &os) const
- void **setSwitch** (size_t connectedNode1, size_t connectedNode2, bool closed)
- void **setAllSwitches** (size_t *connectedNodes1, size_t *connectedNodes2, bool closedStates[])
- void **flipSwitch** (size_t connectedNode1, size_t connectedNode2)
- void **flipAllSwitches** ()
- bool **isSwitchClosed** (size_t connectedNode1, size_t connectedNode2)
- void **printSwitchState** (size_t connectedNode1, size_t connectedNode2, std::ostream &os) const
- void **printAllSwitchStates** (std::ostream &os) const
- [Signal](#) **getLampSignal** (int connectedNode) const
- bool **isLampGlowing** (int connectedNode) const
- void **printLampState** (int connectedNode, std::ostream &os)
- void **printAllLampStates** (std::ostream &os) const

Static Public Member Functions

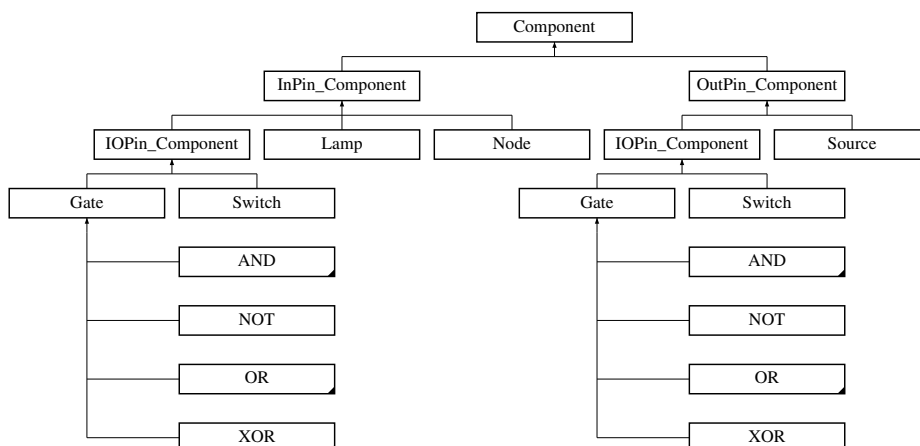
- static void **setErrorStream** (std::ostream *os)

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

- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Circuit.h
- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Circuit.cpp

4.3 Component Class Reference

Inheritance diagram for Component:



Public Member Functions

- [Component](#) ([Queue](#)< [Component](#) > *newActiveQueue=nullptr)
Felparaméterezi az aktív sort.
- void **setActiveQueue** ([Queue](#)< [Component](#) > *newActiveQueue)
Beállítja az aktív fifo-t.
- void **addToActiveQueue** ()
Hozzáadja az aktív sorhoz az áramköri elemet.
- virtual void **executeFunction** ()=0
Végrehajtja a funkcióját az áramköri elemnek. Leszármazottban konkretizálva.
- virtual ~**Component** ()
Virtuális destruktork öröklés miatt.

Protected Attributes

- [Queue](#)< [Component](#) > * **activeQueue**
Az aktív sor, amihez kell hozzáadni, ha ki kell értékelni az elemet, azaz végrehajtani a funkcióját.

4.3.1 Constructor & Destructor Documentation

4.3.1.1 Component()

```
Component::Component (
    Queue< Component > * newActiveQueue = nullptr ) [inline]
```

Felparaméterezi az aktív sort.

Parameters

<i>newActiveQueue</i>	A kívánt aktív sor.
-----------------------	---------------------

4.3.2 Member Function Documentation

4.3.2.1 executeFunction()

```
virtual void Component::executeFunction ( ) [pure virtual]
```

Végrehajtja a funkcióját az áramköri elemnek. Leszármazottban konkretizálva.

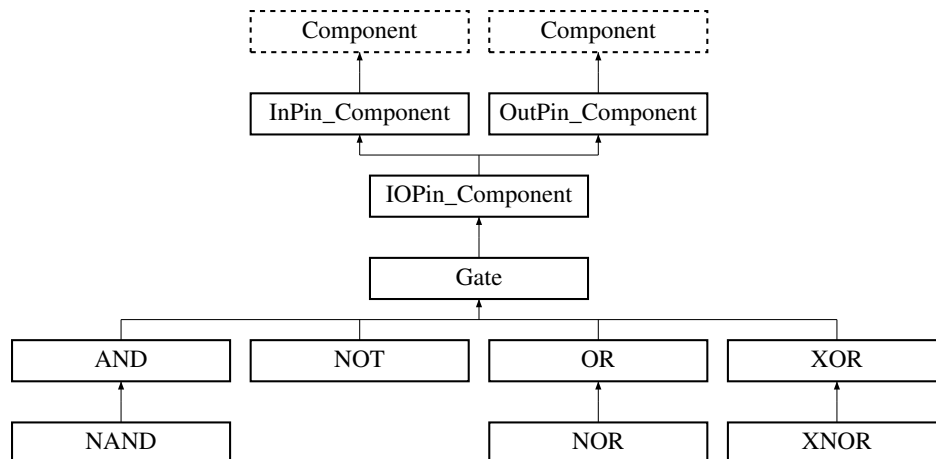
Implemented in [Gate](#), [Lamp](#), [Node](#), [Source](#), [Switch](#), [InPin_Component](#), [OutPin_Component](#), and [IOPin_Component](#).

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

- C:/Users/palin/VSCodeProjects/Egyetem/2.Felev/Prog2NHF/src/Component.h
- C:/Users/palin/VSCodeProjects/Egyetem/2.Felev/Prog2NHF/src/Component.cpp

4.4 Gate Class Reference

Inheritance diagram for Gate:



Public Member Functions

- **Gate** (size_t inCount, size_t outCount)
- virtual void **executeFunction** ()

Végrehajtja a funkcióját az áramköri elemnek. Leszármazottban konkretizálva.

Public Member Functions inherited from IOPin_Component

- **IOPin_Component** (size_t inCount, size_t outCount)
- *Létrehozza a lábakat.*
- bool **connectedToNodes** (size_t connectedNode1, size_t connectedNode2)
- void **printConnectedNodes** (std::ostream &os) const
- virtual ~**IOPin_Component** ()

Virtuális destruktork, mert öröklés.

Public Member Functions inherited from InPin_Component

- **InPin_Component** (size_t inCount)
- *Létrehozza a bemeneti pin-ek tömbjét.*
- void **resetActiveCount** ()
- *Reseteli az aktív pin-ek számát 0-ra.*
- void **setInNodeID** (size_t at, size_t id)
- bool **connectedToNodeIn** (size_t id)
- void **printlnConnectedNodes** (std::ostream &os) const
- void **resetForSimulation** ()
- void **tickCounter** ()
- *Növeli az aktív lábak számát, és aktivizálja az áramköri elemet, ha eléri a maximumot.*
- **InPin * getInPinsBaseAdress** () const
- *Visszaadja a bemeneti pin-ek tömbjének címét.*
- virtual ~**InPin_Component** ()
- *Törli a bemeneti pin-ek tömbjét.*

Public Member Functions inherited from [Component](#)

- [Component](#) ([Queue](#)< [Component](#) > *newActiveQueue=nullptr)
Felparaméterezi az aktív sort.
- void **setActiveQueue** ([Queue](#)< [Component](#) > *newActiveQueue)
Beállítja az aktív fifo-t.
- void **addToActiveQueue** ()
Hozzáadja az aktív sorhoz az áramköri elemet.
- virtual ~**Component** ()
Virtuális destruktork öröklés miatt.

Public Member Functions inherited from [OutPin_Component](#)

- [OutPin_Component](#) (size_t outCount)
Létrehozza a kimeneti pin-ek tömbjét.
- void **connectTo** (size_t outPinIndex, [InPin_Component](#) *component, size_t inPinIndex)
Összeköti a megadott kimeneti pin-jét egy másik áramköri elem bemeneti pin-jével.
- void **sendOutSignals** ()
Kiküldi minden kimeneti lábán a lábakban tárolt jeleket.
- [OutPin](#) * **getOutPinBaseAddress** ()
- void **setOutNodeID** (size_t at, size_t id)
- bool **connectedToNodeOut** (size_t id)
- void **printOutConnectedNodes** (std::ostream &os) const
- virtual ~[OutPin_Component](#) ()
Törli a kimeneti pin-ek tömbjét.

Protected Member Functions

- virtual void **performLogicCalculation** ()=0

Additional Inherited Members

Protected Attributes inherited from [InPin_Component](#)

- size_t **inPinCount**
A bemeneti pin-ek száma.
- size_t **activeInPins**
Az aktív, valid adattal rendelkező pin-ek száma.
- [InPin](#) * **inPins**
A bemeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- size_t * **inNodeIDs**

Protected Attributes inherited from [Component](#)

- [Queue](#)< [Component](#) > * **activeQueue**
Az aktív sor, amihez kell hozzáadni, ha ki kell értékelni az elemet, azaz végrehajtani a funkcióját.

Protected Attributes inherited from [OutPin_Component](#)

- `size_t outPinCount`
A kimeneti pin-ek száma.
- `OutPin * outPins`
A kimeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- `size_t * outPinIDs`

4.4.1 Member Function Documentation

4.4.1.1 `executeFunction()`

```
void Gate::executeFunction ( ) [virtual]
```

Végrehajtja a funkcióját az áramköri elemnek. Leszármazottban konkretizálva.

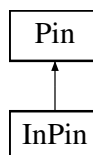
Implements [IOPin_Component](#).

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

- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Gate.h
- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Gate.cpp

4.5 InPin Class Reference

Inheritance diagram for InPin:



Public Member Functions

- `InPin (Signal baseSignal=Signal(false))`
Létrehoz egy bemeneti pin-t, komponensét NULL-ra állítva.
- `void connectToComponent (InPin_Component *component)`
Egy áramköri elemhez köti a bemeneti pin-t, így tud majd neki üzenni.
- `InPin_Component * getComponent () const`
- `void SignalReady () const`
Jelzi az áramköri elem felé, hogy ezen a lábán rendelkezésre áll a jel.

Public Member Functions inherited from [Pin](#)

- [Pin](#) ([Signal](#) baseSignal=[Signal](#)(false))
Létrehoz egy pin-t kezdő jelértékkal.
- void [setSignal](#) (const [Signal](#) &newSignal)
Beállítja a pin jelét.
- [Signal](#) [getSignal](#) () const
Visszaadja a pin jelét. Mivel kicsi a [Signal](#) osztály, ezért nem kell pointer/referencia.
- void [flipSignal](#) ()
Megfordítja a pin jelét.
- virtual ~[Pin](#) ()
Virtuál destruktor, mert öröklés.

Additional Inherited Members

Protected Attributes inherited from [Pin](#)

- [Signal](#) [ownedSignal](#)
A pin által birtokolt jel.

4.5.1 Constructor & Destructor Documentation

4.5.1.1 InPin()

```
InPin::InPin (  
    Signal baseSignal = Signal(false) ) [inline]
```

Létrehoz egy bemeneti pin-t, komponensét NULL-ra állítva.

Parameters

<i>baseSignal</i>	Az alapjel, ha van megadva.
-------------------	-----------------------------

4.5.2 Member Function Documentation

4.5.2.1 connectToComponent()

```
void InPin::connectToComponent (  
    InPin\_Component * component ) [inline]
```

Egy áramköri elemhez köti a bemeneti pin-t, így tud majd neki üzenni.

Parameters

<i>component</i>	Az áramköri elem, amihez kötjük.
------------------	----------------------------------

4.5.2.2 SignalReady()

```
void InPin::SignalReady ( ) const
```

Jelzi az áramköri elem felé, hogy ezen a lábán rendelkezésre áll a jel.

Exceptions

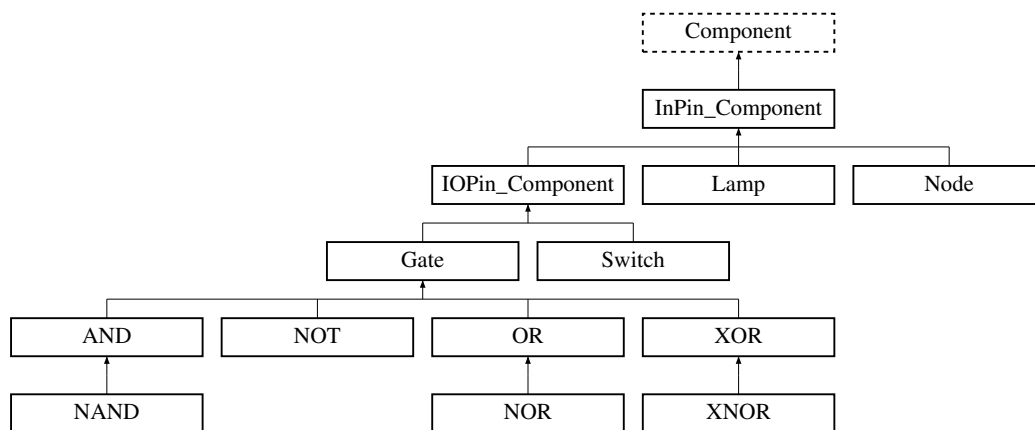
<i>const</i>	char* = amennyiben nincs hozzá kötve áramköri elem.
--------------	---

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

- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Pin.h
- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Pin.cpp

4.6 InPin_Component Class Reference

Inheritance diagram for InPin_Component:



Public Member Functions

- [InPin_Component](#) (size_t inCount)
Létrehozza a bemeneti pin-ek tömbjét.
- void **resetActiveCount** ()
Reseteli az aktív pin-ek számát 0-ra.
- void **setInNodeID** (size_t at, size_t id)
- bool **connectedToNodeIn** (size_t id)
- void **printlnConnectedNodes** (std::ostream &os) const
- void **resetForSimulation** ()
- void **tickCounter** ()
Növeli az aktív lábak számát, és aktivizálja az áramköri elemet, ha eléri a maximumot.
- [InPin *](#) **getInPinsBaseAddress** () const
Visszaadja a bemeneti pin-ek tömbjének címét.
- virtual void [executeFunction](#) ()=0
Végrehajtja a funkcióját az áramköri elemnek. Leszármazottban konkretizálva.
- virtual **~InPin_Component** ()
Törli a bemeneti pin-ek tömbjét.

Public Member Functions inherited from [Component](#)

- [Component](#) ([Queue](#)< [Component](#) > *newActiveQueue=nullptr)
Felparaméterezi az aktív sort.
- void **setActiveQueue** ([Queue](#)< [Component](#) > *newActiveQueue)
Beállítja az aktív fifo-t.
- void **addToActiveQueue** ()
Hozzáadja az aktív sorhoz az áramkörü elemet.
- virtual ~**Component** ()
Virtuális destruktork öröklés miatt.

Protected Attributes

- size_t **inPinCount**
A bemeneti pin-ek száma.
- size_t **activeInPins**
Az aktív, valid adattal rendelkező pin-ek száma.
- [InPin](#) * **inPins**
A bemeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- size_t * **inNodeIDs**

Protected Attributes inherited from [Component](#)

- [Queue](#)< [Component](#) > * **activeQueue**
Az aktív sor, amihez kell hozzáadni, ha ki kell értékelni az elemet, azaz végrehajtani a funkcióját.

4.6.1 Constructor & Destructor Documentation

4.6.1.1 InPin_Component()

```
InPin_Component::InPin_Component (
    size_t inCount )
```

Létrehozza a bemeneti pin-ek tömbjét.

Parameters

<i>inCount</i>	A bemeneti pin-ek száma.
----------------	--------------------------

4.6.2 Member Function Documentation

4.6.2.1 executeFunction()

```
virtual void InPin_Component::executeFunction ( ) [pure virtual]
```

Végrehajtja a funkcióját az áramkörü elemnek. Leszármazottban konkretizálva.

Implements [Component](#).

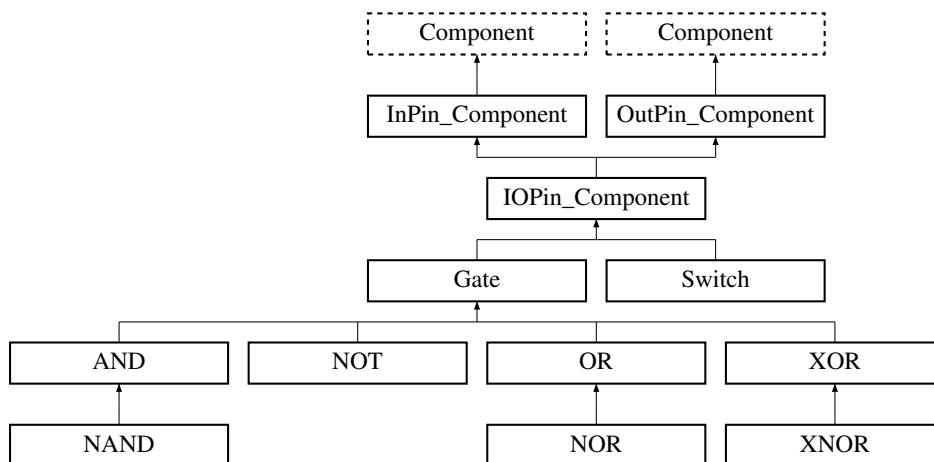
Implemented in [Gate](#), [Lamp](#), [Node](#), [Switch](#), and [IOPin_Component](#).

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

- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Component.h
- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Component.cpp

4.7 IOPin_Component Class Reference

Inheritance diagram for IOPin_Component:



Public Member Functions

- [IOPin_Component](#) (size_t inCount, size_t outCount)
Létrehozza a lábakat.
- virtual void [executeFunction](#) ()=0
Végrehajtja a funkcióját az áramköri elemnek. Leszármazottban konkretizálva.
- bool **connectedToNodes** (size_t connectedNode1, size_t connectedNode2)
- void **printConnectedNodes** (std::ostream &os) const
- virtual **~IOPin_Component** ()
Virtuális destruktork, mert öröklés.

Public Member Functions inherited from [InPin_Component](#)

- [InPin_Component](#) (size_t inCount)
Létrehozza a bemeneti pin-ek tömbjét.
- void **resetActiveCount** ()
Reseteli az aktív pin-ek számát 0-ra.
- void **setInNodeID** (size_t at, size_t id)
- bool **connectedToNodeIn** (size_t id)
- void **printlnConnectedNodes** (std::ostream &os) const
- void **resetForSimulation** ()
- void **tickCounter** ()
Növeli az aktív lábak számát, és aktivizálja az áramköri elemet, ha eléri a maximumot.
- [InPin](#) * **getInPinsBaseAdress** () const
Visszaadja a bemeneti pin-ek tömbjének címét.
- virtual **~InPin_Component** ()
Törli a bemeneti pin-ek tömbjét.

Public Member Functions inherited from Component

- **Component** (Queue< **Component** > *newActiveQueue=nullptr)
Felparaméterezi az aktív sort.
- void **setActiveQueue** (Queue< **Component** > *newActiveQueue)
Beállítja az aktív fifo-t.
- void **addToActiveQueue** ()
Hozzáadja az aktív sorhoz az áramköri elemet.
- virtual ~**Component** ()
Virtuális destruktorköröklés miatt.

Public Member Functions inherited from OutPin_Component

- **OutPin_Component** (size_t outCount)
Létrehozza a kimeneti pin-ek tömbjét.
- void **connectTo** (size_t outPinIndex, **InPin_Component** *component, size_t inPinIndex)
Összeköti a megadott kimeneti pin-jét egy másik áramköri elem bemeneti pin-jével.
- void **sendOutSignals** ()
Kiküldi minden kimeneti lábán a lábakban tárolt jeleket.
- **OutPin** * **getOutPinBaseAddress** ()
- void **setOutNodeID** (size_t at, size_t id)
- bool **connectedToNodeOut** (size_t id)
- void **printOutConnectedNodes** (std::ostream &os) const
- virtual ~**OutPin_Component** ()
Törli a kimeneti pin-ek tömbjét.

Additional Inherited Members

Protected Attributes inherited from InPin_Component

- size_t **inPinCount**
A bemeneti pin-ek száma.
- size_t **activeInPins**
Az aktív, valid adattal rendelkező pin-ek száma.
- **InPin** * **inPins**
A bemeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- size_t * **inNodeIDs**

Protected Attributes inherited from Component

- Queue< **Component** > * **activeQueue**
Az aktív sor, amihez kell hozzáadni, ha ki kell értékelni az elemet, azaz végrehajtani a funkcióját.

Protected Attributes inherited from OutPin_Component

- size_t **outPinCount**
A kimeneti pin-ek száma.
- **OutPin** * **outPins**
A kimeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- size_t * **outPinIDs**

4.7.1 Constructor & Destructor Documentation

4.7.1.1 IOPin_Component()

```
IOPin_Component::IOPin_Component (
    size_t inCount,
    size_t outCount ) [inline]
```

Lértéhozza a lábakat.

Parameters

<i>inCount</i>	Bemeneti lábak száma.
<i>outCount</i>	Kimeneti lábak száma.

4.7.2 Member Function Documentation

4.7.2.1 executeFunction()

```
virtual void IOPin_Component::executeFunction ( ) [pure virtual]
```

Végrehajtja a funkcióját az áramköri elemnek. Leszármazottban konkretizálva.

Implements [InPin_Component](#).

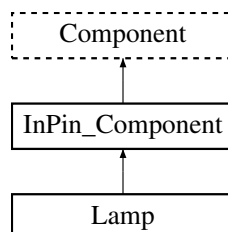
Implemented in [Gate](#), and [Switch](#).

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

- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Component.h
- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Component.cpp

4.8 Lamp Class Reference

Inheritance diagram for Lamp:



Public Member Functions

- **Lamp** ()
Létrehoz egy lámpát.
- **Signal** **getState** () const
Visszaadja a lámpa jelét, azaz, hogy világít-e.
- virtual void **executeFunction** ()
Itt igazából haszontalan, lámpának nincs végezni valója.

Public Member Functions inherited from **InPin_Component**

- **InPin_Component** (size_t inCount)
Létrehozza a bemeneti pin-ek tömbjét.
- void **resetActiveCount** ()
Reseteli az aktív pin-ek számát 0-ra.
- void **setInNodeID** (size_t at, size_t id)
- bool **connectedToNodeIn** (size_t id)
- void **printlnConnectedNodes** (std::ostream &os) const
- void **resetForSimulation** ()
- void **tickCounter** ()
Növeli az aktív lábak számát, és aktivizálja az áramköri elemet, ha eléri a maximumot.
- **InPin** * **getInPinsBaseAddress** () const
Visszaadja a bemeneti pin-ek tömbjének címét.
- virtual ~**InPin_Component** ()
Törli a bemeneti pin-ek tömbjét.

Public Member Functions inherited from **Component**

- **Component** (Queue< **Component** > *newActiveQueue=nullptr)
Felparaméterezi az aktív sort.
- void **setActiveQueue** (Queue< **Component** > *newActiveQueue)
Beállítja az aktív fifo-t.
- void **addToActiveQueue** ()
Hozzáadja az aktív sorhoz az áramköri elemet.
- virtual ~**Component** ()
Virtuális destruktork öröklés miatt.

Additional Inherited Members

Protected Attributes inherited from **InPin_Component**

- size_t **inPinCount**
A bemeneti pin-ek száma.
- size_t **activeInPins**
Az aktív, valid adattal rendelkező pin-ek száma.
- **InPin** * **inPins**
A bemeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- size_t * **inNodeIDs**

Protected Attributes inherited from [Component](#)

- [Queue](#)< [Component](#) > * **activeQueue**

Az aktív sor, amihez kell hozzáadni, ha ki kell értékelni az elemet, azaz végrehajtani a funkcióját.

4.8.1 Member Function Documentation

4.8.1.1 executeFunction()

```
virtual void Lamp::executeFunction ( ) [inline], [virtual]
```

Itt igazából haszontalan, lámpának nincs végezni valója.

Implements [InPin_Component](#).

4.8.1.2 getState()

```
Signal Lamp::getState ( ) const [inline]
```

Visszaadja a lámpa jelét, azaz, hogy világít-e.

Returns

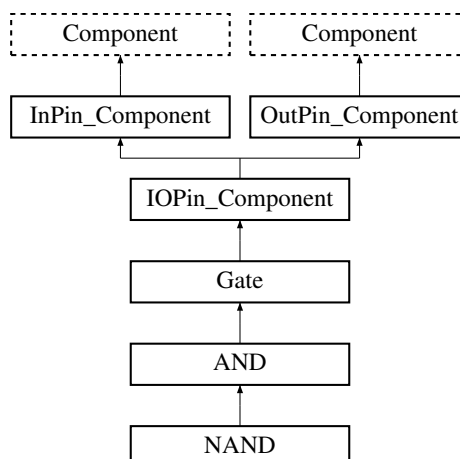
A jelérték.

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

- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Lamp.h
- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Lamp.cpp

4.9 NAND Class Reference

Inheritance diagram for NAND:



Public Member Functions

- **NAND** (size_t inCount)

Public Member Functions inherited from AND

- **AND** (size_t inCount)

Public Member Functions inherited from Gate

- **Gate** (size_t inCount, size_t outCount)
- virtual void **executeFunction** ()

Végrehajtja a funkcióját az áramköri elemnek. Leszármazottban konkretizálva.

Public Member Functions inherited from IOPin_Component

- **IOPin_Component** (size_t inCount, size_t outCount)
Létrehozza a lábakat.
- bool **connectedToNodes** (size_t connectedNode1, size_t connectedNode2)
- void **printConnectedNodes** (std::ostream &os) const
- virtual **~IOPin_Component** ()

Virtuális destruktork, mert öröklés.

Public Member Functions inherited from InPin_Component

- **InPin_Component** (size_t inCount)
Létrehozza a bemeneti pin-ek tömbjét.
- void **resetActiveCount** ()
Reseteli az aktív pin-ek számát 0-ra.
- void **setInNodeID** (size_t at, size_t id)
- bool **connectedToNodeIn** (size_t id)
- void **printlnConnectedNodes** (std::ostream &os) const
- void **resetForSimulation** ()
- void **tickCounter** ()
Növeli az aktív lábak számát, és aktivizálja az áramköri elemet, ha eléri a maximumot.
- **InPin * getInPinsBaseAdress** () const
Visszaadja a bemeneti pin-ek tömbjének címét.
- virtual **~InPin_Component** ()
Törli a bemeneti pin-ek tömbjét.

Public Member Functions inherited from Component

- **Component** (Queue< **Component** > *newActiveQueue=nullptr)
Felparaméterezi az aktív sort.
- void **setActiveQueue** (Queue< **Component** > *newActiveQueue)
Beállítja az aktív fifo-t.
- void **addToActiveQueue** ()
Hozzáadja az aktív sorhoz az áramköri elemet.
- virtual **~Component** ()
Virtuális destruktork öröklés miatt.

Public Member Functions inherited from OutPin_Component

- [OutPin_Component](#) (size_t outCount)
Létrehozza a kimeneti pin-ek tömbjét.
- void [connectTo](#) (size_t outPinIndex, [InPin_Component](#) *component, size_t inPinIndex)
Összeköti a megadott kimeneti pin-jét egy másik áramköri elem bemeneti pin-jével.
- void [sendOutSignals](#) ()
Kiküldi minden kimeneti lábán a lábakban tárolt jeleket.
- [OutPin](#) * [getOutPinBaseAddress](#) ()
- void [setOutNodeID](#) (size_t at, size_t id)
- bool [connectedToNodeOut](#) (size_t id)
- void [printOutConnectedNodes](#) (std::ostream &os) const
- virtual ~[OutPin_Component](#) ()
Törli a kimeneti pin-ek tömbjét.

Additional Inherited Members

Protected Attributes inherited from InPin_Component

- size_t [inPinCount](#)
A bemeneti pin-ek száma.
- size_t [activeInPins](#)
Az aktív, valid adattal rendelkező pin-ek száma.
- [InPin](#) * [inPins](#)
A bemeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- size_t * [inNodeIDs](#)

Protected Attributes inherited from Component

- [Queue](#)< [Component](#) > * [activeQueue](#)
Az aktív sor, amihez kell hozzáadni, ha ki kell értékelni az elemet, azaz végrehajtani a funkcióját.

Protected Attributes inherited from OutPin_Component

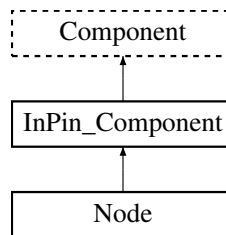
- size_t [outPinCount](#)
A kimeneti pin-ek száma.
- [OutPin](#) * [outPins](#)
A kimeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- size_t * [outPinIDs](#)

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

- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Gate.h
- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Gate.cpp

4.10 Node Class Reference

Inheritance diagram for Node:



Public Member Functions

- **Node** (size_t nodeID)
- **InPin** * **getInPin** ()
- size_t **getID** () const
- void **addOutPin** (**InPin** *endPoint)
- virtual void **executeFunction** ()

Végrehajtja a funkcióját az áramköri elemnek. Leszármazottban konkretizálva.

Public Member Functions inherited from **InPin_Component**

- **InPin_Component** (size_t inCount)
Létrehozza a bemeneti pin-ek tömbjét.
- void **resetActiveCount** ()
Reseteli az aktív pin-ek számát 0-ra.
- void **setInNodeID** (size_t at, size_t id)
- bool **connectedToNodeIn** (size_t id)
- void **printlnConnectedNodes** (std::ostream &os) const
- void **resetForSimulation** ()
- void **tickCounter** ()
Növeli az aktív lábak számát, és aktivizálja az áramköri elemet, ha eléri a maximumot.
- **InPin** * **getInPinsBaseAdress** () const
Visszaadja a bemeneti pin-ek tömbjének címét.
- virtual ~**InPin_Component** ()
Törli a bemeneti pin-ek tömbjét.

Public Member Functions inherited from **Component**

- **Component** (Queue< **Component** > *newActiveQueue=nullptr)
Felparaméterezi az aktív sort.
- void **setActiveQueue** (Queue< **Component** > *newActiveQueue)
Beállítja az aktív fifo-t.
- void **addToActiveQueue** ()
Hozzáadja az aktív sorhoz az áramköri elemet.
- virtual ~**Component** ()
Virtuális destruktork öröklés miatt.

Additional Inherited Members

Protected Attributes inherited from [InPin_Component](#)

- `size_t inPinCount`
A bemeneti pin-ek száma.
- `size_t activeInPins`
Az aktív, valid adattal rendelkező pin-ek száma.
- `InPin * inPins`
A bemeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- `size_t * inNodeIDs`

Protected Attributes inherited from [Component](#)

- `Queue< Component > * activeQueue`
Az aktív sor, amihez kell hozzáadni, ha ki kell értékelni az elemet, azaz végrehajtani a funkcióját.

4.10.1 Member Function Documentation

4.10.1.1 executeFunction()

```
void Node::executeFunction ( ) [virtual]
```

Végrehajtja a funkcióját az áramköri elemnek. Leszármazottban konkretizálva.

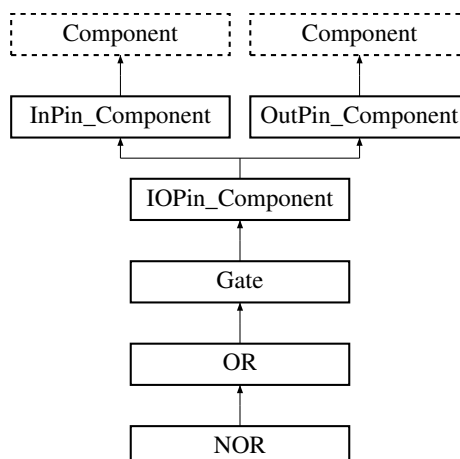
Implements [InPin_Component](#).

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

- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Node.h
- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Node.cpp

4.11 NOR Class Reference

Inheritance diagram for NOR:



Public Member Functions

- **NOR** (size_t inCount)

Public Member Functions inherited from OR

- **OR** (size_t inCount)

Public Member Functions inherited from Gate

- **Gate** (size_t inCount, size_t outCount)
- virtual void **executeFunction** ()

Végrehajtja a funkcióját az áramköri elemnek. Leszármazottban konkretizálva.

Public Member Functions inherited from IOPin_Component

- **IOPin_Component** (size_t inCount, size_t outCount)
Létrehozza a lábakat.
- bool **connectedToNodes** (size_t connectedNode1, size_t connectedNode2)
- void **printConnectedNodes** (std::ostream &os) const
- virtual **~IOPin_Component** ()

Virtuális destruktork, mert öröklés.

Public Member Functions inherited from InPin_Component

- **InPin_Component** (size_t inCount)
Létrehozza a bemeneti pin-ek tömbjét.
- void **resetActiveCount** ()
Reseteli az aktív pin-ek számát 0-ra.
- void **setInNodeID** (size_t at, size_t id)
- bool **connectedToNodeIn** (size_t id)
- void **printlnConnectedNodes** (std::ostream &os) const
- void **resetForSimulation** ()
- void **tickCounter** ()
Növeli az aktív lábak számát, és aktivizálja az áramköri elemet, ha eléri a maximumot.
- **InPin * getInPinsBaseAdress** () const
Visszaadja a bemeneti pin-ek tömbjének címét.
- virtual **~InPin_Component** ()
Törli a bemeneti pin-ek tömbjét.

Public Member Functions inherited from Component

- **Component** (Queue< **Component** > *newActiveQueue=nullptr)
Felparaméterezi az aktív sort.
- void **setActiveQueue** (Queue< **Component** > *newActiveQueue)
Beállítja az aktív fifo-t.
- void **addToActiveQueue** ()
Hozzáadja az aktív sorhoz az áramköri elemet.
- virtual **~Component** ()
Virtuális destruktork öröklés miatt.

Public Member Functions inherited from [OutPin_Component](#)

- [OutPin_Component](#) (size_t outCount)
Létrehozza a kimeneti pin-ek tömbjét.
- void [connectTo](#) (size_t outPinIndex, [InPin_Component](#) *component, size_t inPinIndex)
Összeköti a megadott kimeneti pin-jét egy másik áramköri elem bemeneti pin-jével.
- void [sendOutSignals](#) ()
Kiküldi minden kimeneti lábán a lábakban tárolt jeleket.
- [OutPin](#) * [getOutPinBaseAddress](#) ()
- void [setOutNodeID](#) (size_t at, size_t id)
- bool [connectedToNodeOut](#) (size_t id)
- void [printOutConnectedNodes](#) (std::ostream &os) const
- virtual ~[OutPin_Component](#) ()
Törli a kimeneti pin-ek tömbjét.

Additional Inherited Members

Protected Attributes inherited from [InPin_Component](#)

- size_t [inPinCount](#)
A bemeneti pin-ek száma.
- size_t [activeInPins](#)
Az aktív, valid adattal rendelkező pin-ek száma.
- [InPin](#) * [inPins](#)
A bemeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- size_t * [inNodeIDs](#)

Protected Attributes inherited from [Component](#)

- [Queue](#)< [Component](#) > * [activeQueue](#)
Az aktív sor, amihez kell hozzáadni, ha ki kell értékelni az elemet, azaz végrehajtani a funkcióját.

Protected Attributes inherited from [OutPin_Component](#)

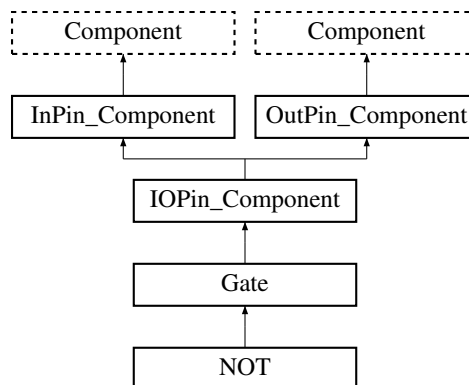
- size_t [outPinCount](#)
A kimeneti pin-ek száma.
- [OutPin](#) * [outPins](#)
A kimeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- size_t * [outPinIDs](#)

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

- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Gate.h
- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Gate.cpp

4.12 NOT Class Reference

Inheritance diagram for NOT:



Additional Inherited Members

Public Member Functions inherited from [Gate](#)

- **Gate** (size_t inCount, size_t outCount)
- virtual void [executeFunction](#) ()

Végrehajtja a funkcióját az áramköri elemnek. Leszármazottban konkretizálva.

Public Member Functions inherited from [IOPin_Component](#)

- [IOPin_Component](#) (size_t inCount, size_t outCount)
- *Létrehozza a lábakat.*
- bool **connectedToNodes** (size_t connectedNode1, size_t connectedNode2)
- void **printConnectedNodes** (std::ostream &os) const
- virtual ~**IOPin_Component** ()

Virtuális destruktork, mert öröklés.

Public Member Functions inherited from [InPin_Component](#)

- [InPin_Component](#) (size_t inCount)
- *Létrehozza a bemeneti pin-ek tömbjét.*
- void **resetActiveCount** ()
- *Reseteli az aktív pin-ek számát 0-ra.*
- void **setInNodeID** (size_t at, size_t id)
- bool **connectedToNodeIn** (size_t id)
- void **printlnConnectedNodes** (std::ostream &os) const
- void **resetForSimulation** ()
- void **tickCounter** ()
- *Növeli az aktív lábak számát, és aktivizálja az áramköri elemet, ha eléri a maximumot.*
- [InPin](#) * **getInPinsBaseAdress** () const
- *Visszaadja a bemeneti pin-ek tömbjének címét.*
- virtual ~**InPin_Component** ()
- *Törli a bemeneti pin-ek tömbjét.*

Public Member Functions inherited from [Component](#)

- [Component](#) ([Queue](#)< [Component](#) > *newActiveQueue=nullptr)
Felparaméterezi az aktív sort.
- void **setActiveQueue** ([Queue](#)< [Component](#) > *newActiveQueue)
Beállítja az aktív fifo-t.
- void **addToActiveQueue** ()
Hozzáadja az aktív sorhoz az áramköri elemet.
- virtual ~[Component](#) ()
Virtuális destruktork öröklés miatt.

Public Member Functions inherited from [OutPin_Component](#)

- [OutPin_Component](#) (size_t outCount)
Létrehozza a kimeneti pin-ek tömbjét.
- void **connectTo** (size_t outPinIndex, [InPin_Component](#) *component, size_t inPinIndex)
Összeköti a megadott kimeneti pin-jét egy másik áramköri elem bemeneti pin-jével.
- void **sendOutSignals** ()
Kiküldi minden kimeneti lábán a lábakban tárolt jeleket.
- [OutPin](#) * **getOutPinBaseAddress** ()
- void **setOutNodeID** (size_t at, size_t id)
- bool **connectedToNodeOut** (size_t id)
- void **printOutConnectedNodes** (std::ostream &os) const
- virtual ~[OutPin_Component](#) ()
Törli a kimeneti pin-ek tömbjét.

Protected Attributes inherited from [InPin_Component](#)

- size_t **inPinCount**
A bemeneti pin-ek száma.
- size_t **activeInPins**
Az aktív, valid adattal rendelkező pin-ek száma.
- [InPin](#) * **inPins**
A bemeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- size_t * **inNodeIDs**

Protected Attributes inherited from [Component](#)

- [Queue](#)< [Component](#) > * **activeQueue**
Az aktív sor, amihez kell hozzáadni, ha ki kell értékelni az elemet, azaz végrehajtani a funkcióját.

Protected Attributes inherited from [OutPin_Component](#)

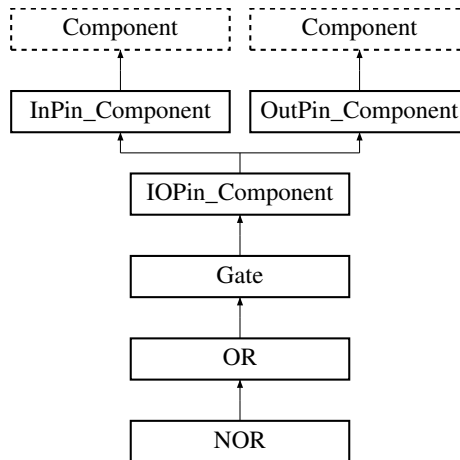
- size_t **outPinCount**
A kimeneti pin-ek száma.
- [OutPin](#) * **outPins**
A kimeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- size_t * **outPinIDs**

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

- C:/Users/palin/VsCodeProjects/Egyetem/2.Felev/Prog2NHF/src/Gate.h
- C:/Users/palin/VsCodeProjects/Egyetem/2.Felev/Prog2NHF/src/Gate.cpp

4.13 OR Class Reference

Inheritance diagram for OR:



Public Member Functions

- **OR** (size_t inCount)

Public Member Functions inherited from [Gate](#)

- **Gate** (size_t inCount, size_t outCount)
- virtual void [executeFunction](#) ()

Végrehajtja a funkcióját az áramköri elemnek. Leszármazottban konkretizálva.

Public Member Functions inherited from [IOPin_Component](#)

- [IOPin_Component](#) (size_t inCount, size_t outCount)
Létrehozza a lábakat.
- bool **connectedToNodes** (size_t connectedNode1, size_t connectedNode2)
- void **printConnectedNodes** (std::ostream &os) const
- virtual ~[IOPin_Component](#) ()

Virtuális destruktork, mert öröklés.

Public Member Functions inherited from [InPin_Component](#)

- [InPin_Component](#) (size_t inCount)
Létrehozza a bemeneti pin-ek tömbjét.
- void **resetActiveCount** ()
Reseteli az aktív pin-ek számát 0-ra.
- void **setInNodeID** (size_t at, size_t id)
- bool **connectedToNodeIn** (size_t id)
- void **printlnConnectedNodes** (std::ostream &os) const
- void **resetForSimulation** ()
- void **tickCounter** ()
Növeli az aktív lábak számát, és aktivizálja az áramköri elemet, ha eléri a maximumot.
- [InPin](#) * **getInPinsBaseAdress** () const
Visszaadja a bemeneti pin-ek tömbjének címét.
- virtual ~[InPin_Component](#) ()
Törli a bemeneti pin-ek tömbjét.

Public Member Functions inherited from [Component](#)

- [Component](#) ([Queue](#)< [Component](#) > *newActiveQueue=nullptr)
Felparaméterezi az aktív sort.
- void **setActiveQueue** ([Queue](#)< [Component](#) > *newActiveQueue)
Beállítja az aktív fifo-t.
- void **addToActiveQueue** ()
Hozzáadja az aktív sorhoz az áramköri elemet.
- virtual ~**Component** ()
Virtuális destruktorköröklés miatt.

Public Member Functions inherited from [OutPin_Component](#)

- [OutPin_Component](#) (size_t outCount)
Létrehozza a kimeneti pin-ek tömbjét.
- void **connectTo** (size_t outPinIndex, [InPin_Component](#) *component, size_t inPinIndex)
Összeköti a megadott kimeneti pin-jét egy másik áramköri elem bemeneti pin-jével.
- void **sendOutSignals** ()
Kiküldi minden kimeneti lábán a lábakban tárolt jeleket.
- [OutPin](#) * **getOutPinBaseAddress** ()
- void **setOutNodeID** (size_t at, size_t id)
- bool **connectedToNodeOut** (size_t id)
- void **printOutConnectedNodes** (std::ostream &os) const
- virtual ~[OutPin_Component](#) ()
Törli a kimeneti pin-ek tömbjét.

Protected Member Functions

- virtual void [performLogicCalculation](#) ()

Additional Inherited Members

Protected Attributes inherited from [InPin_Component](#)

- size_t **inPinCount**
A bemeneti pin-ek száma.
- size_t **activeInPins**
Az aktív, valid adattal rendelkező pin-ek száma.
- [InPin](#) * **inPins**
A bemeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- size_t * **inNodeIDs**

Protected Attributes inherited from [Component](#)

- [Queue](#)< [Component](#) > * **activeQueue**
Az aktív sor, amihez kell hozzáadni, ha ki kell értékelni az elemet, azaz végrehajtani a funkcióját.

Protected Attributes inherited from [OutPin_Component](#)

- `size_t outPinCount`
A kimeneti pin-ek száma.
- `OutPin * outPins`
A kimeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- `size_t * outPinIDs`

4.13.1 Member Function Documentation

4.13.1.1 performLogicCalculation()

```
void OR::performLogicCalculation ( ) [protected], [virtual]
```

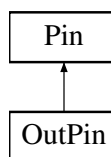
Implements [Gate](#).

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

- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Gate.h
- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Gate.cpp

4.14 OutPin Class Reference

Inheritance diagram for OutPin:



Public Member Functions

- `OutPin (Signal baseSignal=Signal(false))`
Létrhoz egy kimeneti pin-t, kapcsolt bemeneti pin-jét NULL-ra állítva.
- `void connectToPin (InPin *pin)`
Összekapcsolja egy bemeneti pin-nel.
- `void sendSignal () const`
Jelet küld a kapcsolt bemeneti pin-nek.

Public Member Functions inherited from [Pin](#)

- `Pin (Signal baseSignal=Signal(false))`
Létrehoz egy pin-t kezdő jelértékkel.
- `void setSignal (const Signal &newSignal)`
Beállítja a pin jelét.
- `Signal getSignal () const`
Visszaadja a pin jelét. Mivel kicsi a Signal osztály, ezért nem kell pointer/referencia.
- `void flipSignal ()`
Megfordítja a pin jelét.
- `virtual ~Pin ()`
Virtuál destruktor, mert öröklés.

Additional Inherited Members

Protected Attributes inherited from [Pin](#)

- [Signal](#) **ownedSignal**
A pin által birtokolt jel.

4.14.1 Constructor & Destructor Documentation

4.14.1.1 OutPin()

```
OutPin::OutPin (
    Signal baseSignal = Signal(false) ) [inline]
```

Létrhoz egy kimeneti pin-t, kapcsolt bemeneti pin-jét NULL-ra állítva.

Parameters

<i>baseSignal</i>	Az alapjel, ha van megadva.
-------------------	-----------------------------

4.14.2 Member Function Documentation

4.14.2.1 connectToPin()

```
void OutPin::connectToPin (
    InPin * pin ) [inline]
```

Összekapcsolja egy bemeneti pin-nel.

Parameters

<i>pin</i>	A kapcsolni kívánt pin.
------------	-------------------------

4.14.2.2 sendSignal()

```
void OutPin::sendSignal ( ) const
```

Jelet küld a kapcsolt bemeneti pin-nek.

Exceptions

<i>const</i>	char* = amennyiben nincs hozzá kötve bemeneti pin.
--------------	--

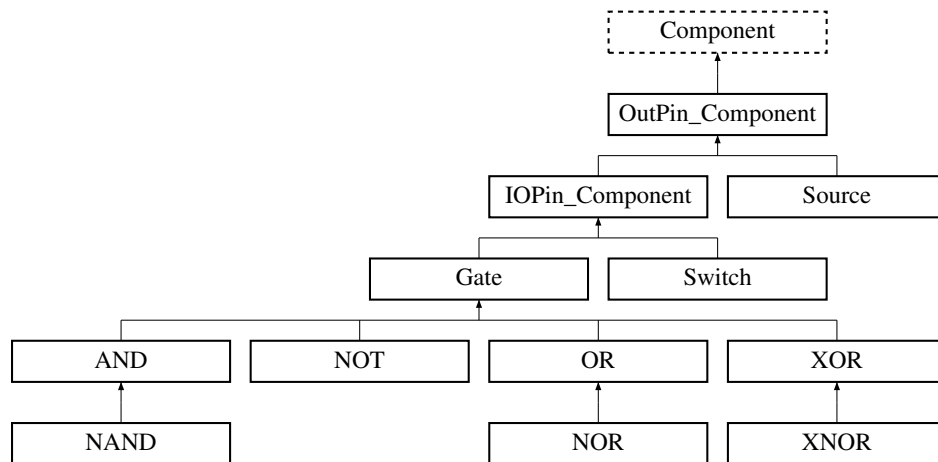
The documentation for this class was generated from the following files:

- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Pin.h

- C:/Users/palin/VsCodeProjects/Egyetem/2.Felev/Prog2NHF/src/Pin.cpp

4.15 OutPin_Component Class Reference

Inheritance diagram for OutPin_Component:



Public Member Functions

- [OutPin_Component](#) (size_t outCount)
Létrehozza a kimeneti pin-ek tömbjét.
- void [connectTo](#) (size_t outPinIndex, [InPin_Component](#) *component, size_t inPinIndex)
Összeköti a megadott kimeneti pin-jét egy másik áramköri elem bemeneti pin-jével.
- void [sendOutSignals](#) ()
Kiküldi minden kimeneti lábán a lábakban tárolt jeleket.
- [OutPin](#) * [getOutPinBaseAddress](#) ()
- void [setOutNodeID](#) (size_t at, size_t id)
- bool [connectedToNodeOut](#) (size_t id)
- void [printOutConnectedNodes](#) (std::ostream &os) const
- virtual void [executeFunction](#) ()=0
Végrehajtja a funkcióját az áramköri elemnek. Leszármazottban konkretizálva.
- virtual ~[OutPin_Component](#) ()
Törli a kimeneti pin-ek tömbjét.

Public Member Functions inherited from [Component](#)

- [Component](#) (Queue< [Component](#) > *newActiveQueue=nullptr)
Felparaméterezi az aktív sort.
- void [setActiveQueue](#) (Queue< [Component](#) > *newActiveQueue)
Beállítja az aktív fifo-t.
- void [addToActiveQueue](#) ()
Hozzáadja az aktív sorhoz az áramköri elemet.
- virtual ~[Component](#) ()
Virtuális destruktorköröklés miatt.

Protected Attributes

- `size_t outPinCount`
A kimeneti pin-ek száma.
- `OutPin * outPins`
A kimeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- `size_t * outPinIDs`

Protected Attributes inherited from `Component`

- `Queue< Component > * activeQueue`
Az aktív sor, amihez kell hozzáadni, ha ki kell értékelni az elemet, azaz végrehajtani a funkcióját.

4.15.1 Constructor & Destructor Documentation

4.15.1.1 `OutPin_Component()`

```
OutPin_Component::OutPin_Component (
    size_t outCount )
```

Létrehozza a kimeneti pin-ek tömbjét.

Parameters

<i>outCount</i>	A kívánt kimeneti pin-ek száma.
-----------------	---------------------------------

4.15.2 Member Function Documentation

4.15.2.1 `connectTo()`

```
void OutPin_Component::connectTo (
    size_t outPinIndex,
    InPin_Component * component,
    size_t inPinIndex )
```

Összeköti a megadott kimeneti pin-jét egy másik áramköri elem bemeneti pin-jével.

Parameters

<i>outPinIndex</i>	A kimeneti pin indexe.
<i>component</i>	Amivel össze szeretnénk kötni.
<i>inPinIndex</i>	A bemeneti pin indexe.

4.15.2.2 `executeFunction()`

```
virtual void OutPin_Component::executeFunction ( ) [pure virtual]
```

Végrehajtja a funkcióját az áramköri elemnek. Leszármazottban konkretizálva.

Implements [Component](#).

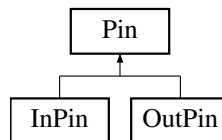
Implemented in [Gate](#), [Source](#), [Switch](#), and [IOPin_Component](#).

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

- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Component.h
- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Component.cpp

4.16 Pin Class Reference

Inheritance diagram for Pin:



Public Member Functions

- [Pin](#) ([Signal](#) baseSignal=[Signal](#)(false))
Létrehoz egy pin-t kezdő jelértékkel.
- void [setSignal](#) (const [Signal](#) &newSignal)
Beállítja a pin jelét.
- [Signal](#) [getSignal](#) () const
Visszaadja a pin jelét. Mivel kicsi a [Signal](#) osztály, ezért nem kell pointer/referencia.
- void [flipSignal](#) ()
Megfordítja a pin jelét.
- virtual ~[Pin](#) ()
Virtuál destruktor, mert öröklés.

Protected Attributes

- [Signal](#) [ownedSignal](#)
A pin által birtokolt jel.

4.16.1 Constructor & Destructor Documentation

4.16.1.1 Pin()

```
Pin::Pin (
    Signal baseSignal = Signal(false) )
```

Létrehoz egy pin-t kezdő jelértékkel.

Parameters

<i>baseSignal</i>	Az alap jelérték, default-ként LOW (0) jelszinttel.
-------------------	---

4.16.2 Member Function Documentation

4.16.2.1 `getSignal()`

```
Signal Pin::getSignal ( ) const
```

Visszaadja a pin jelét. Mivel kicsi a [Signal](#) osztály, ezért nem kell pointer/referencia.

Returns

A visszaadott jel.

4.16.2.2 `setSignal()`

```
void Pin::setSignal (
    const Signal & newSignal )
```

Beállítja a pin jelét.

Parameters

<i>newSignal</i>	Az új jel.
------------------	------------

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

- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Pin.h
- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Pin.cpp

4.17 `Queue< T >` Class Template Reference

Public Member Functions

- [Queue](#) (bool owner=false)
Beállítja az üres fifo állapotát.
- [Queue](#) (const [Queue](#)< T > &source)
Másolást teszi lehetővé, hogy ideiglenesen valamit tudjunk futtatni a fifo-n, a tagokat nem birtokolja.
- void [put](#) (T *added)
Berak egy tagot a sor végére.
- T * [get](#) ()
Kivesz a sor elejéről egy tagot.
- bool [isEmpty](#) ()

Megmondja, hogy üres-e a fifo.

- void **clear** ()

Kiüríti a fifo-t, ha tulaj, akkor törli a memóriát is.

- size_t **size** () const

Visszaadja a fifo elemszámát.

- ~**Queue** ()

Detruktor, mely felszabadítja a tárolt elemeket, attól függően, hogy birtokolja-e őket.

4.17.1 Constructor & Destructor Documentation

4.17.1.1 Queue() [1/2]

```
template<class T >
Queue< T >::Queue (
    bool owner = false ) [inline]
```

Beállítja az üres fifo állapotát.

Parameters

<i>owner</i>	Tulaja-e az elemeknek, azaz fel kell majd szabadítani a mutatott objektumokat-e.
--------------	--

4.17.1.2 Queue() [2/2]

```
template<class T >
Queue< T >::Queue (
    const Queue< T > & source )
```

Másolást teszi lehetővé, hogy ideiglenesen valamit tudjunk futtatni a fifo-n, a tagokat nem birtokolja.

Parameters

<i>source</i>	A másolás forrása.
---------------	--------------------

4.17.2 Member Function Documentation

4.17.2.1 get()

```
template<class T >
T * Queue< T >::get ( )
```

Kivesz a sor elejéről egy tagot.

Returns

T* A soron következő tag.

4.17.2.2 isEmpty()

```
template<class T >
bool Queue< T >::isEmpty ( ) [inline]
```

Megmondja, hogy üres-e a fifo.

Returns

true = üres a fifo

false = nem üres a fifo

4.17.2.3 put()

```
template<class T >
void Queue< T >::put (
    T * added )
```

Berak egy tagot a sor végére.

Parameters

<i>added</i>	A hozzáadott tag.
--------------	-------------------

4.17.2.4 size()

```
template<class T >
size_t Queue< T >::size ( ) const [inline]
```

Visszaadja a fifo elemszámát.

Returns

size_t A fifo mérete.

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

- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Queue.h

4.18 QueueMember< T > Struct Template Reference

Public Attributes

- [QueueMember< T > * nextMember](#)
A következő elem címe a fifo-ban.
- [T * pointingTo](#)
A mutatott elem.

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

- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Queue.h

4.19 Signal Class Reference

Public Member Functions

- [Signal](#) (bool baseValue=false)
Létrehozza a jel objektumot, adott értékkel, ha van.
- void [setValue](#) (bool newValue)
Beállítja egy új értékre a jelet.
- bool [getValue](#) () const
Visszaadja a jel értékét.
- void [flip](#) ()
Megfordítja a jelértéket.
- bool [operator==](#) (const [Signal](#) &other)
Egyenlőséget vizsgálja két jelszint között.

4.19.1 Constructor & Destructor Documentation

4.19.1.1 Signal()

```
Signal::Signal (
    bool baseValue = false ) [inline]
```

Létrehozza a jel objektumot, adott értékkel, ha van.

Parameters

<i>baseValue</i>	Az alapérték.
------------------	---------------

4.19.2 Member Function Documentation

4.19.2.1 getValue()

```
bool Signal::getValue ( ) const [inline]
```

Visszaadja a jel értékét.

Returns

true = 1 a jelérték.
false = 0 a jelérték.

4.19.2.2 operator==()

```
bool Signal::operator==(
    const Signal & other ) [inline]
```

Egyenlőséget vizsgálja két jelszint között.

Parameters

<i>other</i>	A másik jel.
--------------	--------------

Returns

true, ha gyeznek.

false, ha nem egyeznek

4.19.2.3 setValue()

```
void Signal::setValue (
    bool newValue ) [inline]
```

Beállítja egy új értékre a jelet.

Parameters

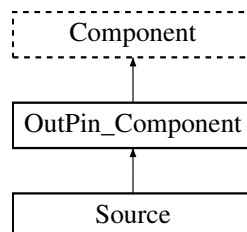
<i>newValue</i>	Az új jel értéke.
-----------------	-------------------

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

- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Signal.h

4.20 Source Class Reference

Inheritance diagram for Source:



Public Member Functions

- **Source** ()
Létrehoz egy forrást.
- virtual void **executeFunction** ()
Végrehajtja a forrás funkcióját, azaz kiküldi a jelet a kimenetén.
- void **setOutput** (const **Signal** &signal)
Beállítja a forrás kimeneti jelét.
- **Signal** **getOutput** () const
Visszaadja a forrás kimeneti jelét.
- void **flipOutput** ()
Megfordítja a kimeneti jel értékét.

Public Member Functions inherited from [OutPin_Component](#)

- [OutPin_Component](#) (size_t outCount)
Létrehozza a kimeneti pin-ek tömbjét.
- void [connectTo](#) (size_t outPinIndex, [InPin_Component](#) *component, size_t inPinIndex)
Összeköti a megadott kimeneti pin-jét egy másik áramkörü elem bemeneti pin-jével.
- void [sendOutSignals](#) ()
Kiküldi minden kimeneti lábán a lábakban tárolt jeleket.
- [OutPin](#) * [getOutPinBaseAddress](#) ()
- void [setOutNodeID](#) (size_t at, size_t id)
- bool [connectedToNodeOut](#) (size_t id)
- void [printOutConnectedNodes](#) (std::ostream &os) const
- virtual ~[OutPin_Component](#) ()
Törli a kimeneti pin-ek tömbjét.

Public Member Functions inherited from [Component](#)

- [Component](#) ([Queue](#)< [Component](#) > *newActiveQueue=nullptr)
Felparaméterezi az aktív sort.
- void [setActiveQueue](#) ([Queue](#)< [Component](#) > *newActiveQueue)
Beállítja az aktív fifo-t.
- void [addToActiveQueue](#) ()
Hozzáadja az aktív sorhoz az áramkörü elemet.
- virtual ~[Component](#) ()
Virtuális destruktork öröklés miatt.

Additional Inherited Members

Protected Attributes inherited from [OutPin_Component](#)

- size_t [outPinCount](#)
A kimeneti pin-ek száma.
- [OutPin](#) * [outPins](#)
A kimeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- size_t * [outPinIDs](#)

Protected Attributes inherited from [Component](#)

- [Queue](#)< [Component](#) > * [activeQueue](#)
Az aktív sor, amihez kell hozzáadni, ha ki kell értékelni az elemet, azaz végrehajtani a funkcióját.

4.20.1 Member Function Documentation

4.20.1.1 executeFunction()

```
virtual void Source::executeFunction ( ) [inline], [virtual]
```

Végrehajtja a forrás funkcióját, azaz kiküldi a jelet a kimenetén.

Implements [OutPin_Component](#).

4.20.1.2 `getOutput()`

```
Signal Source::getOutput ( ) const [inline]
```

Visszaadja a forrás kimeneti jelét.

Returns

A forrás jele.

4.20.1.3 `setOutput()`

```
void Source::setOutput (
    const Signal & signal ) [inline]
```

Beállítja a forrás kimeneti jelét.

Parameters

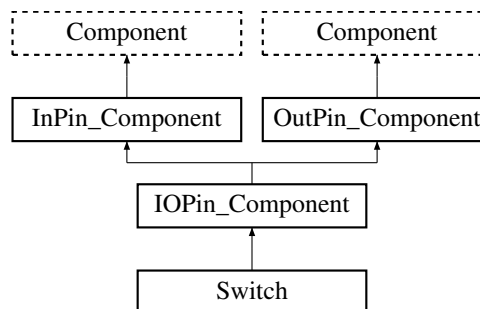
<i>signal</i>	Az új jel.
---------------	------------

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

- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Source.h
- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Source.cpp

4.21 Switch Class Reference

Inheritance diagram for Switch:



Public Member Functions

- **Switch** ()
Létrehozza a kapcsolót.
- virtual void **executeFunction** ()
Megvalósítja a kapcsolót, azaz ha zárt akkor a bemeneti jelet továbbítja, egyébként meg LOW (0) jelet ad.
- bool **getState** () const
Vissza adja hogy zárt-e a kapcsoló.
- void **setState** (bool newState)
- void **flipState** ()
Átbillenti másik állapotba a kapcsolót.

Public Member Functions inherited from [IOPin_Component](#)

- [IOPin_Component](#) (size_t inCount, size_t outCount)
Létrehozza a lábakat.
- bool **connectedToNodes** (size_t connectedNode1, size_t connectedNode2)
- void **printConnectedNodes** (std::ostream &os) const
- virtual **~IOPin_Component** ()
Virtuális destruktork, mert öröklés.

Public Member Functions inherited from [InPin_Component](#)

- [InPin_Component](#) (size_t inCount)
Létrehozza a bemeneti pin-ek tömbjét.
- void **resetActiveCount** ()
Reseteli az aktív pin-ek számát 0-ra.
- void **setInNodeID** (size_t at, size_t id)
- bool **connectedToNodeIn** (size_t id)
- void **printlnConnectedNodes** (std::ostream &os) const
- void **resetForSimulation** ()
- void **tickCounter** ()
Növeli az aktív lábak számát, és aktivizálja az áramköri elemet, ha eléri a maximumot.
- [InPin](#) * **getInPinsBaseAddress** () const
Visszaadja a bemeneti pin-ek tömbjének címét.
- virtual **~InPin_Component** ()
Törli a bemeneti pin-ek tömbjét.

Public Member Functions inherited from [Component](#)

- [Component](#) (Queue< [Component](#) > *newActiveQueue=nullptr)
Felparaméterezi az aktív sort.
- void **setActiveQueue** (Queue< [Component](#) > *newActiveQueue)
Beállítja az aktív fifo-t.
- void **addToActiveQueue** ()
Hozzáadja az aktív sorhoz az áramköri elemet.
- virtual **~Component** ()
Virtuális destruktork öröklés miatt.

Public Member Functions inherited from [OutPin_Component](#)

- [OutPin_Component](#) (size_t outCount)
Létrehozza a kimeneti pin-ek tömbjét.
- void **connectTo** (size_t outPinIndex, [InPin_Component](#) *component, size_t inPinIndex)
Összeköti a megadott kimeneti pin-jét egy másik áramköri elem bemeneti pin-jével.
- void **sendOutSignals** ()
Kiküldi minden kimeneti lábán a lábakban tárolt jeleket.
- [OutPin](#) * **getOutPinBaseAddress** ()
- void **setOutNodeID** (size_t at, size_t id)
- bool **connectedToNodeOut** (size_t id)
- void **printOutConnectedNodes** (std::ostream &os) const
- virtual **~OutPin_Component** ()
Törli a kimeneti pin-ek tömbjét.

Additional Inherited Members

Protected Attributes inherited from [InPin_Component](#)

- `size_t inPinCount`
A bemeneti pin-ek száma.
- `size_t activeInPins`
Az aktív, valid adattal rendelkező pin-ek száma.
- `InPin * inPins`
A bemeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- `size_t * inNodeIDs`

Protected Attributes inherited from [Component](#)

- `Queue< Component > * activeQueue`
Az aktív sor, amihez kell hozzáadni, ha ki kell értékelni az elemet, azaz végrehajtani a funkcióját.

Protected Attributes inherited from [OutPin_Component](#)

- `size_t outPinCount`
A kimeneti pin-ek száma.
- `OutPin * outPins`
A kimeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- `size_t * outPinIDs`

4.21.1 Member Function Documentation

4.21.1.1 executeFunction()

```
void Switch::executeFunction ( ) [virtual]
```

Megvalósítja a kapcsolót, azaz ha zárt akkor a bemeneti jelet továbbítja, egyébként meg LOW (0) jelet ad.

Implements [IOPin_Component](#).

4.21.1.2 getState()

```
bool Switch::getState ( ) const [inline]
```

Vissza adja hogy zárt-e a kapcsoló.

Returns

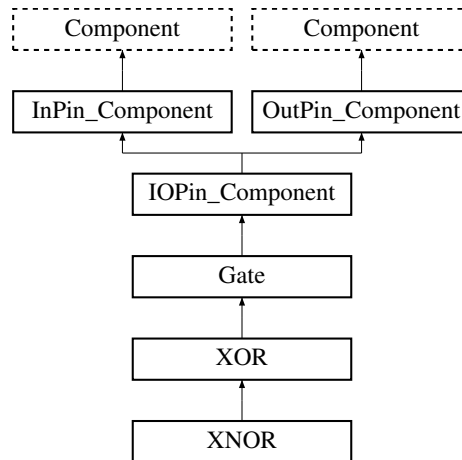
true = zárt,
false = nyitott

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

- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Switch.h
- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Switch.cpp

4.22 XNOR Class Reference

Inheritance diagram for XNOR:



Public Member Functions

- **XNOR** (size_t inCount)

Public Member Functions inherited from XOR

- **XOR** (size_t inCount)

Public Member Functions inherited from Gate

- **Gate** (size_t inCount, size_t outCount)
- virtual void **executeFunction** ()

Végrehajtja a funkcióját az áramköri elemnek. Leszármazottban konkretizálva.

Public Member Functions inherited from IOPin_Component

- **IOPin_Component** (size_t inCount, size_t outCount)
Létrehozza a lábakat.
- bool **connectedToNodes** (size_t connectedNode1, size_t connectedNode2)
- void **printConnectedNodes** (std::ostream &os) const
- virtual **~IOPin_Component** ()

Virtuális destruktork, mert öröklés.

Public Member Functions inherited from [InPin_Component](#)

- [InPin_Component](#) (size_t inCount)
Létrehozza a bemeneti pin-ek tömbjét.
- void **resetActiveCount** ()
Reseteli az aktív pin-ek számát 0-ra.
- void **setInNodeID** (size_t at, size_t id)
- bool **connectedToNodeIn** (size_t id)
- void **printlnConnectedNodes** (std::ostream &os) const
- void **resetForSimulation** ()
- void **tickCounter** ()
Növeli az aktív lábak számát, és aktivizálja az áramköri elemet, ha eléri a maximumot.
- [InPin](#) * **getInPinsBaseAddress** () const
Visszaadja a bemeneti pin-ek tömbjének címét.
- virtual ~[InPin_Component](#) ()
Törli a bemeneti pin-ek tömbjét.

Public Member Functions inherited from [Component](#)

- [Component](#) ([Queue](#)< [Component](#) > *newActiveQueue=nullptr)
Felparaméterezi az aktív sort.
- void **setActiveQueue** ([Queue](#)< [Component](#) > *newActiveQueue)
Beállítja az aktív fifo-t.
- void **addToActiveQueue** ()
Hozzáadja az aktív sorhoz az áramköri elemet.
- virtual ~[Component](#) ()
Virtuális destruktorköröklés miatt.

Public Member Functions inherited from [OutPin_Component](#)

- [OutPin_Component](#) (size_t outCount)
Létrehozza a kimeneti pin-ek tömbjét.
- void **connectTo** (size_t outPinIndex, [InPin_Component](#) *component, size_t inPinIndex)
Összeköti a megadott kimeneti pin-jét egy másik áramköri elem bemeneti pin-jével.
- void **sendOutSignals** ()
Kiküldi minden kimeneti lábán a lábakban tárolt jeleket.
- [OutPin](#) * **getOutPinBaseAddress** ()
- void **setOutNodeID** (size_t at, size_t id)
- bool **connectedToNodeOut** (size_t id)
- void **printOutConnectedNodes** (std::ostream &os) const
- virtual ~[OutPin_Component](#) ()
Törli a kimeneti pin-ek tömbjét.

Additional Inherited Members

Protected Attributes inherited from [InPin_Component](#)

- size_t **inPinCount**
A bemeneti pin-ek száma.
- size_t **activeInPins**
Az aktív, valid adattal rendelkező pin-ek száma.
- [InPin](#) * **inPins**
A bemeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- size_t * **inNodeIDs**

Protected Attributes inherited from [Component](#)

- [Queue](#)< [Component](#) > * **activeQueue**

Az aktív sor, amihez kell hozzáadni, ha ki kell értékelni az elemet, azaz végrehajtani a funkcióját.

Protected Attributes inherited from [OutPin_Component](#)

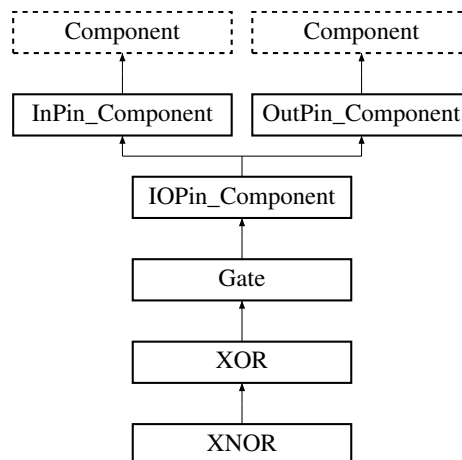
- size_t **outPinCount**
A kimeneti pin-ek száma.
- [OutPin](#) * **outPins**
A kimeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- size_t * **outPinIDs**

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

- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Gate.h
- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Gate.cpp

4.23 XOR Class Reference

Inheritance diagram for XOR:



Public Member Functions

- **XOR** (size_t inCount)

Public Member Functions inherited from [Gate](#)

- **Gate** (size_t inCount, size_t outCount)
- virtual void [executeFunction](#) ()
Végrehajtja a funkcióját az áramköri elemnek. Leszármazottban konkretizálva.

Public Member Functions inherited from [IOPin_Component](#)

- [IOPin_Component](#) (size_t inCount, size_t outCount)
Létrehozza a lábakat.
- bool **connectedToNodes** (size_t connectedNode1, size_t connectedNode2)
- void **printConnectedNodes** (std::ostream &os) const
- virtual **~IOPin_Component** ()
Virtuális destruktork, mert öröklés.

Public Member Functions inherited from [InPin_Component](#)

- [InPin_Component](#) (size_t inCount)
Létrehozza a bemeneti pin-ek tömbjét.
- void **resetActiveCount** ()
Reseteli az aktív pin-ek számát 0-ra.
- void **setInNodeID** (size_t at, size_t id)
- bool **connectedToNodeIn** (size_t id)
- void **printlnConnectedNodes** (std::ostream &os) const
- void **resetForSimulation** ()
- void **tickCounter** ()
Növeli az aktív lábak számát, és aktivizálja az áramköri elemet, ha eléri a maximumot.
- [InPin](#) * **getInPinsBaseAddress** () const
Visszaadja a bemeneti pin-ek tömbjének címét.
- virtual **~InPin_Component** ()
Törli a bemeneti pin-ek tömbjét.

Public Member Functions inherited from [Component](#)

- [Component](#) (Queue< [Component](#) > *newActiveQueue=nullptr)
Felparaméterezi az aktív sort.
- void **setActiveQueue** (Queue< [Component](#) > *newActiveQueue)
Beállítja az aktív fifo-t.
- void **addToActiveQueue** ()
Hozzáadja az aktív sorhoz az áramköri elemet.
- virtual **~Component** ()
Virtuális destruktork öröklés miatt.

Public Member Functions inherited from [OutPin_Component](#)

- [OutPin_Component](#) (size_t outCount)
Létrehozza a kimeneti pin-ek tömbjét.
- void **connectTo** (size_t outPinIndex, [InPin_Component](#) *component, size_t inPinIndex)
Összeköti a megadott kimeneti pin-jét egy másik áramköri elem bemeneti pin-jével.
- void **sendOutSignals** ()
Kiküldi minden kimeneti lábán a lábakban tárolt jeleket.
- [OutPin](#) * **getOutPinBaseAddress** ()
- void **setOutNodeID** (size_t at, size_t id)
- bool **connectedToNodeOut** (size_t id)
- void **printOutConnectedNodes** (std::ostream &os) const
- virtual **~OutPin_Component** ()
Törli a kimeneti pin-ek tömbjét.

Protected Member Functions

- virtual void [performLogicCalculation](#) ()

Additional Inherited Members

Protected Attributes inherited from [InPin_Component](#)

- `size_t inPinCount`
A bemeneti pin-ek száma.
- `size_t activeInPins`
Az aktív, valid adattal rendelkező pin-ek száma.
- `InPin * inPins`
A bemeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- `size_t * inNodeIDs`

Protected Attributes inherited from [Component](#)

- `Queue< Component > * activeQueue`
Az aktív sor, amihez kell hozzáadni, ha ki kell értékelni az elemet, azaz végrehajtani a funkcióját.

Protected Attributes inherited from [OutPin_Component](#)

- `size_t outPinCount`
A kimeneti pin-ek száma.
- `OutPin * outPins`
A kimeneti pin-ek tömbje. Az index jelentést a specifikált alkatrész adja meg.
- `size_t * outPinIDs`

4.23.1 Member Function Documentation

4.23.1.1 [performLogicCalculation\(\)](#)

```
void XOR::performLogicCalculation ( ) [protected], [virtual]
```

Implements [Gate](#).

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

- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Gate.h
- C:/Users/palin/VSCoDeProjects/Egyetem/2.Felev/Prog2NHF/src/Gate.cpp

Chapter 5

File Documentation

5.1 Circuit.h

```
00001 #ifndef _CIRCUIT_H
00002 #define _CIRCUIT_H
00003
00004 #include "Component.h"
00005 #include "Gate.h"
00006 #include "Lamp.h"
00007 #include "Node.h"
00008 #include "Pin.h"
00009 #include "Queue.h"
00010 #include "Signal.h"
00011 #include "Source.h"
00012 #include "Switch.h"
00013
00014 #include <iostream>
00015 #include <fstream>
00016 #include <cstring>
00017 #include <sstream>
00018 #include <string>
00019
00020 class Circuit {
00021     static std::ostream* errorStream;
00022
00023     bool configured;
00024     bool simulated;
00025
00026     std::string inputFileName;
00027     std::ifstream inputfile;
00028
00029     Queue<Component> componentList;
00030     Queue<InPin_Component> incomponents;
00031     Queue<Source> sourceList;
00032     Queue<Switch> switchList;
00033     Queue<Lamp> lampList;
00034     Queue<Component> activeList;
00035     Queue<Node> nodeList;
00036
00037     void reset();
00038     void configure();
00039
00040     void build();
00041     enum ComponentType {
00042         INVALID = 0, SOURCE = 1, LAMP = 2, SWITCH = 3, And = 4, Or = 5, Not = 6, Xor = 7, Nand = 8, Nor =
00043         9, Xnor = 10
00044     };
00045     struct ContentInfo {
00046         size_t lineCount;
00047         std::string line;
00048         size_t idx;
00049         ComponentType type;
00050         ContentInfo() : lineCount(0), line(""), idx(0), type(INVALID) {}
00051     };
00052     void buildLine(ContentInfo& info);
00053     void checkLineType(ContentInfo& info);
00054     void buildComponent(ContentInfo& info);
00055     void getNodeNumbers(ContentInfo& info, Queue<int>& nodeNumbers);
00056     void checkNodeCount(ContentInfo& info, size_t count);
00057     void createBasedOnType(ContentInfo& info, size_t count, Queue<int>& nodeNumbers);
00058 }
```

```

00058     template<typename T>
00059     void create(size_t count, Queue<int>& nodeNumbers);
00060
00061     template<typename T>
00062     void create(Queue<int>& nodeNumbers);
00063
00064     void connectInPinWithNode(InPin* pin, size_t id, size_t idx);
00065     void connectOutPinWithNode(OutPin_Component* component, OutPin* pin, size_t id, size_t idx);
00066
00067 public:
00068     Circuit();
00069     Circuit(const Circuit& source);
00070     Circuit& operator=(const Circuit& source);
00071
00072     static void setErrorStream(std::ostream* os);
00073
00074     void setSourceFile(const std::string& fileName);
00075     const std::string& getSourceFileName() const;
00076
00077     void simulate(std::ostream& os);
00078
00079     void setSource(size_t connectedNode, Signal newSignal);
00080     void setAllSources(int* connectedNodes, Signal* newSignals);
00081     void flipSource(size_t connectedNode);
00082     void flipAllSources();
00083     Signal getSourceSignal(size_t connectedNode) const;
00084     void printSourceState(std::ostream& os) const;
00085     void printAllSourceStates(std::ostream& os) const;
00086
00087     void setSwitch(size_t connectedNode1, size_t connectedNode2, bool closed);
00088     void setAllSwitches(size_t* connectedNodes1, size_t* connectedNodes2, bool closedStates[]);
00089     void flipSwitch(size_t connectedNode1, size_t connectedNode2);
00090     void flipAllSwitches();
00091     bool isSwitchClosed(size_t connectedNode1, size_t connectedNode2);
00092     void printSwitchState(size_t connectedNode1, size_t connectedNode2, std::ostream& os) const;
00093     void printAllSwitchStates(std::ostream& os) const;
00094
00095     Signal getLampSignal(int connectedNode) const;
00096     bool isLampGlowing(int connectedNode) const;
00097     void printLampState(int connectedNode, std::ostream& os);
00098     void printAllLampStates(std::ostream& os) const;
00099
00100     ~Circuit();
00101 };
00102
00103 std::ostream& operator<<(std::ostream& os, const Circuit& circuit);
00104 void printSeparatorLine(std::ostream& os, char c, int times);
00105
00106 template<typename T>
00107 void Circuit::create(size_t count, Queue<int>& nodeNumbers) {
00108     T* created = new T(count - 1);
00109     ((Component*)created)->setActiveQueue(&activeList);
00110     Queue<int> copy(nodeNumbers);
00111     for (size_t i = 0; i < count - 1; i++) {
00112         connectInPinWithNode(((InPin_Component*)created)->getInPinsBaseAdress() + i, *(copy.get()), i);
00113     }
00114     connectOutPinWithNode(created, ((OutPin_Component*)created)->getOutPinBaseAdress(), *(copy.get()),
00115 0);
00116     componentList.put(created);
00117     incomponents.put(created);
00118 }
00119 #endif

```

5.2 Component.h

```

00001 #ifndef _COMPONENT_H
00002 #define _COMPONENT_H
00003
00004 #include "Pin.h"
00005 #include "Queue.h"
00006
00007 typedef long long unsigned int size_t;
00008
00009 class Component {
00010 protected:
00011     Queue<Component>* activeQueue;
00012
00013 public:
00014     Component(Queue<Component>* newActiveQueue = nullptr) : activeQueue(newActiveQueue) {}
00015
00016     void setActiveQueue(Queue<Component>* newActiveQueue) { activeQueue = newActiveQueue; }
00017 }

```

```

00034 void addToActiveQueue();
00035
00040 virtual void executeFunction() = 0;
00041
00046 virtual ~Component() {}
00047
00048 };
00049 class InPin_Component : virtual public Component {
00050 protected:
00054 size_t inPinCount;
00058 size_t activeInPins;
00062 InPin* inPins;
00063 size_t* inNodeIDs;
00064
00065 public:
00071 InPin_Component(size_t inCount);
00072
00077 void resetActiveCount() { activeInPins = 0; }
00078
00079 void setInNodeID(size_t at, size_t id);
00080 bool connectedToNodeIn(size_t id);
00081 void printInConnectedNodes(std::ostream& os) const;
00082
00083 void resetForSimulation();
00084
00089 void tickCounter();
00090
00095 InPin* getInPinsBaseAddress() const { return inPins; }
00096
00101 virtual void executeFunction() = 0;
00102
00107 virtual ~InPin_Component();
00108
00109 };
00110 class OutPin_Component : virtual public Component {
00111 protected:
00115 size_t outPinCount;
00120 OutPin* outPins;
00121 size_t* outPinIDs;
00122
00123 public:
00129 OutPin_Component(size_t outCount);
00130
00138 void connectTo(size_t outPinIndex, InPin_Component* component, size_t inPinIndex);
00139
00144 void sendOutSignals();
00145
00150 OutPin* getOutPinBaseAddress() { return outPins; }
00151
00152 void setOutNodeID(size_t at, size_t id);
00153 bool connectedToNodeOut(size_t id);
00154 void printOutConnectedNodes(std::ostream& os) const;
00155
00160 virtual void executeFunction() = 0;
00161
00166 virtual ~OutPin_Component();
00167
00168 };
00169 class IOPin_Component : public InPin_Component, public OutPin_Component {
00170 public:
00177 IOPin_Component(size_t inCount, size_t outCount) : InPin_Component(inCount),
OutPin_Component(outCount) {}
00178
00183 virtual void executeFunction() = 0;
00184
00185 bool connectedToNodes(size_t connectedNode1, size_t connectedNode2);
00186 void printConnectedNodes(std::ostream& os) const;
00187
00192 virtual ~IOPin_Component() {}
00193 };
00194 #endif

```

5.3 Gate.h

```

00001 #ifndef _GATE_H
00002 #define _GATE_H
00003
00004
00005 #include "Component.h"
00006
00007 class Gate : public IOPin_Component {
00008 protected:
00009 virtual void performLogicCalculation() = 0;

```

```

00010 public:
00011     Gate(size_t inCount, size_t outCount) : IOPin_Component(inCount, outCount) {}
00012
00013     virtual void executeFunction();
00014
00015     virtual ~Gate() {}
00016 };
00017
00018 class AND : public Gate {
00019 protected:
00020     virtual void performLogicCalculation();
00021 public:
00022     AND(size_t inCount) : Gate(inCount, 1) {}
00023     virtual ~AND() {}
00024 };
00025
00026 class OR : public Gate {
00027 protected:
00028     virtual void performLogicCalculation();
00029 public:
00030     OR(size_t inCount) : Gate(inCount, 1) {}
00031     virtual ~OR() {}
00032 };
00033
00034 class NOT : public Gate {
00035     virtual void performLogicCalculation();
00036 public:
00037     NOT() : Gate(1, 1) {}
00038     virtual ~NOT() {}
00039 };
00040
00041 class XOR : public Gate {
00042 protected:
00043     virtual void performLogicCalculation();
00044 public:
00045     XOR(size_t inCount) : Gate(inCount, 1) {}
00046     virtual ~XOR() {}
00047 };
00048
00049 class NAND : public AND {
00050     virtual void performLogicCalculation();
00051 public:
00052     NAND(size_t inCount) : AND(inCount) {}
00053     virtual ~NAND() {}
00054 };
00055
00056 class NOR : public OR {
00057     virtual void performLogicCalculation();
00058 public:
00059     NOR(size_t inCount) : OR(inCount) {}
00060     virtual ~NOR() {}
00061 };
00062
00063 class XNOR : public XOR {
00064     virtual void performLogicCalculation();
00065 public:
00066     XNOR(size_t inCount) : XOR(inCount) {}
00067     virtual ~XNOR() {}
00068 };
00069 #endif

```

5.4 Lamp.h

```

00001 #ifndef _LAMP_H
00002 #define _LAMP_H
00003
00004 #include "Component.h"
00005
00006 #include <iostream>
00007
00008 class Lamp : public InPin_Component {
00009 public:
00014     Lamp();
00015
00021     Signal getState() const { return inPins[0].getSignal(); }
00022
00027     virtual void executeFunction() {}
00028
00029     virtual ~Lamp() {}
00030 };
00031
00039 std::ostream& operator<<(std::ostream& os, const Lamp& x);
00040 #endif

```

5.5 Node.h

```

00001 #ifndef _NODE_H
00002 #define _NODE_H
00003
00004 #include "Component.h"
00005
00006 #include <iostream>
00007
00008 class Node : public InPin_Component {
00009     size_t ID;
00010     Queue<OutPin> outPins;
00011 public:
00012     Node(size_t nodeID) : InPin_Component(1), ID(nodeID), outPins(true) {}
00013
00014     InPin* getInPin();
00015     size_t getID() const;
00016     void addOutPin(InPin* endPoint);
00017
00018     virtual void executeFunction();
00019
00020     virtual ~Node() {}
00021 };
00022 #endif

```

5.6 Pin.h

```

00001 #ifndef _PIN_H
00002 #define _PIN_H
00003
00004 #include "Signal.h"
00005
00006 typedef long long unsigned int size_t;
00007
00008 class InPin_Component;
00009 class OutPin_Component;
00010
00011 class Pin {
00012 protected:
00016     Signal ownedSignal;
00017 public:
00023     Pin(Signal baseSignal = Signal(false));
00024
00030     void setSignal(const Signal& newSignal);
00031
00037     Signal getSignal() const;
00038
00043     void flipSignal();
00044
00049     virtual ~Pin() {}
00050 };
00051
00052 class InPin : public Pin {
00056     InPin_Component* ofComponent;
00057 public:
00063     InPin(Signal baseSignal = Signal(false)) : Pin(baseSignal), ofComponent(nullptr) {}
00064
00070     void connectToComponent(InPin_Component* component) { ofComponent = component; }
00071
00072     InPin_Component* getComponent() const { return ofComponent; }
00073
00079     void SignalReady() const;
00080 };
00081
00082 class OutPin : public Pin {
00086     InPin* connectedTo;
00087 public:
00093     OutPin(Signal baseSignal = Signal(false)) : Pin(baseSignal), connectedTo(nullptr) {}
00094
00100     void connectToPin(InPin* pin) { connectedTo = pin; }
00106     void sendSignal() const;
00107 };
00108 #endif

```

5.7 Queue.h

```

00001 #ifndef _QUEUE_H
00002 #define _QUEUE_H
00003

```

```

00004 typedef long long unsigned int size_t;
00005
00006 #include <iostream>
00007
00008 template<class T>
00009 struct QueueMember {
00013     QueueMember<T>* nextMember;
00017     T* pointingTo;
00018 };
00019
00020 template<class T>
00021 class Queue {
00026     bool ownsMembers;
00027
00031     QueueMember<T>* begin;
00036     QueueMember<T>* end;
00037
00044     Queue<T>& operator=(const Queue<T>& source);
00045     size_t siz;
00046
00047 public:
00053     Queue(bool owner = false) : ownsMembers(owner), begin(nullptr), end(nullptr), siz(0) {}
00054
00060     Queue(const Queue<T>& source);
00061
00067     void put(T* added);
00068
00074     T* get();
00075
00082     bool isEmpty();
00083
00088     void clear();
00089
00095     size_t size() const;
00096
00101     ~Queue();
00102 };
00103
00104
00105 template<class T>
00106 Queue<T>& Queue<T>::operator=(const Queue<T>& source)
00107 {
00108     QueueMember<T>* moving = source.begin;
00109     while (moving != nullptr) {
00110         this->put(moving->pointingTo);
00111         moving = moving->nextMember;
00112     }
00113     return *this;
00114 }
00115
00116 template<class T>
00117 Queue<T>::Queue(const Queue<T>& source)
00118 : ownsMembers(false), begin(nullptr), end(nullptr), siz(0)
00119 {
00119     *this = source;
00120 }
00121
00122 template<class T>
00122 void Queue<T>::put(T* added)
00123 {
00124     if (begin == nullptr) {
00125         begin = new QueueMember<T>;
00126         begin->pointingTo = added;
00127         begin->nextMember = nullptr;
00128         end = begin;
00129     }
00130     else {
00131         QueueMember<T>* newMember = new QueueMember<T>;
00132         newMember->pointingTo = added;
00133         newMember->nextMember = nullptr;
00134         end->nextMember = newMember;
00135         end = newMember;
00136     }
00137     siz++;
00138 }
00139
00139 template<class T>
00140 T* Queue<T>::get()
00141 {
00142     if (siz > 0) {
00143         QueueMember<T>* returnedMember = begin;
00144         begin = begin->nextMember;
00145         T* returned = returnedMember->pointingTo;
00146         delete returnedMember;
00147         siz--;
00148         return returned;
00149     }
00150     return nullptr;
00151 }
00152

```



```

00153 template<class T>
00154 inline bool Queue<T>::isEmpty()
00155 {
00156     return siz == 0;
00157 }
00158
00159 template<class T>
00160 inline void Queue<T>::clear()
00161 {
00162     if (ownsMembers) {
00163         while (!isEmpty()) {
00164             delete get();
00165         }
00166     }
00167     else {
00168         while (!isEmpty()) {
00169             get();
00170         }
00171     }
00172 }
00173
00174 template<class T>
00175 inline size_t Queue<T>::size() const
00176 {
00177     return siz;
00178 }
00179
00180 template<class T>
00181 Queue<T>::~~Queue() {
00182     if (ownsMembers) {
00183         while (begin != nullptr) {
00184             QueueMember<T>* next = begin->nextMember;
00185             delete begin->pointingTo;
00186             delete begin;
00187             begin = next;
00188         }
00189     }
00190     else {
00191         while (begin != nullptr) {
00192             QueueMember<T>* next = begin->nextMember;
00193             delete begin;
00194             begin = next;
00195         }
00196     }
00197 }
00198 }
00199 #endif

```

5.8 Signal.h

```

00001 #ifndef _SIGNAL_H
00002 #define _SIGNAL_H
00003
00004 typedef long long unsigned int size_t;
00005
00006 class Signal {
00010     bool signal;
00011
00012 public:
00018     Signal(bool baseValue = false) : signal(baseValue) {}
00019
00025     void setValue(bool newValue) { signal = newValue; }
00026
00033     bool getValue() const { return signal; }
00034
00039     void flip() { signal = !signal; }
00040
00048     bool operator==(const Signal& other) { return signal == other.signal; }
00049 };
00050
00051 #endif

```

5.9 Source.h

```

00001 #ifndef _SOURCE_H
00002 #define _SOURCE_H
00003
00004 #include "Component.h"
00005

```

```
00006 #include <iostream>
00007
00008 class Source : public OutPin_Component {
00009 public:
00013     Source();
00014
00019     virtual void executeFunction() { sendOutSignals(); }
00020
00026     void setOutput(const Signal& signal) { outPins[0].setSignal(signal); }
00027
00033     Signal getOutput() const { return outPins[0].getSignal(); }
00034
00039     void flipOutput() { outPins[0].flipSignal(); }
00040
00041     virtual ~Source() {}
00042 };
00043
00051 std::ostream& operator<<(std::ostream& os, const Source& x);
00052 #endif
```

5.10 Switch.h

```
00001 #ifndef _SWITCH_H
00002 #define _SWITCH_H
00003
00004 #include "Component.h"
00005
00006 #include <iostream>
00007
00008 class Switch : public IOPin_Component {
00013     bool closed;
00014 public:
00019     Switch() : IOPin_Component(1, 1), closed(false) {}
00020
00025     virtual void executeFunction();
00026
00033     bool getState() const { return closed; }
00034
00035     void setState(bool newState) { closed = newState; }
00036
00041     void flipState() { closed = !closed; }
00042
00043     virtual ~Switch() {}
00044 };
00045
00053 std::ostream& operator<<(std::ostream& os, const Switch& x);
00054 #endif
```

Index

AND, [7](#)
 performLogicCalculation, [9](#)
C:/Users/palin/VSCodeProjects/Egyetem/2.Felev/Prog2NHF/src/Circuit.h, [53](#)
C:/Users/palin/VSCodeProjects/Egyetem/2.Felev/Prog2NHF/src/Component.h, [54](#)
C:/Users/palin/VSCodeProjects/Egyetem/2.Felev/Prog2NHF/src/Gate.h, [55](#)
C:/Users/palin/VSCodeProjects/Egyetem/2.Felev/Prog2NHF/src/Lamp.h, [56](#)
C:/Users/palin/VSCodeProjects/Egyetem/2.Felev/Prog2NHF/src/Node.h, [57](#)
C:/Users/palin/VSCodeProjects/Egyetem/2.Felev/Prog2NHF/src/OutPin_Component.h, [57](#)
C:/Users/palin/VSCodeProjects/Egyetem/2.Felev/Prog2NHF/src/Queue_Component.h, [57](#)
C:/Users/palin/VSCodeProjects/Egyetem/2.Felev/Prog2NHF/src/Signal.h, [59](#)
C:/Users/palin/VSCodeProjects/Egyetem/2.Felev/Prog2NHF/src/Source.h, [59](#)
C:/Users/palin/VSCodeProjects/Egyetem/2.Felev/Prog2NHF/src/Switch.h, [60](#)
Circuit, [10](#)
Component, [10](#)
 Component, [11](#)
 executeFunction, [11](#)
connectTo
 OutPin_Component, [36](#)
connectToPin
 OutPin, [34](#)
connectToComponent
 InPin, [15](#)
executeFunction
 Component, [11](#)
 Gate, [14](#)
 InPin_Component, [17](#)
 IOPin_Component, [20](#)
 Lamp, [22](#)
 Node, [26](#)
 OutPin_Component, [36](#)
 Source, [43](#)
 Switch, [46](#)
Gate, [12](#)
 executeFunction, [14](#)
get
 Queue< T >, [39](#)
getOutput
 Source, [43](#)
 getSignal
 Pin, [38](#)
 Lamp, [22](#)
 Switch, [46](#)
 getValue
 Signal, [41](#)
 InPin, [14](#)
 connectToComponent, [15](#)
 Node, [15](#)
 SignalReady, [15](#)
 InPin_Component, [16](#)
 executeFunction, [17](#)
 Queue_Component, [17](#)
 IOPin_Component, [18](#)
 executeFunction, [20](#)
 IOPin_Component, [20](#)
 Source.h, [59](#)
 Queue< T >, [39](#)
 Lamp, [20](#)
 executeFunction, [22](#)
 getState, [22](#)
NAND, [22](#)
Node, [25](#)
 executeFunction, [26](#)
NOR, [26](#)
NOT, [29](#)
operator==
 Signal, [41](#)
OR, [31](#)
 performLogicCalculation, [33](#)
OutPin, [33](#)
 connectToPin, [34](#)
 OutPin, [34](#)
 sendSignal, [34](#)
OutPin_Component, [35](#)
 connectTo, [36](#)
 executeFunction, [36](#)
 OutPin_Component, [36](#)
performLogicCalculation
 AND, [9](#)
 OR, [33](#)
 XOR, [51](#)
Pin, [37](#)
 getSignal, [38](#)

- Pin, [37](#)
 - setSignal, [38](#)
- put
 - Queue< T >, [40](#)
- Queue
 - Queue< T >, [39](#)
- Queue< T >, [38](#)
 - get, [39](#)
 - isEmpty, [39](#)
 - put, [40](#)
 - Queue, [39](#)
 - size, [40](#)
- QueueMember< T >, [40](#)
- sendSignal
 - OutPin, [34](#)
- setOutput
 - Source, [44](#)
- setSignal
 - Pin, [38](#)
- setValue
 - Signal, [42](#)
- Signal, [41](#)
 - getValue, [41](#)
 - operator==, [41](#)
 - setValue, [42](#)
 - Signal, [41](#)
- SignalReady
 - InPin, [15](#)
- size
 - Queue< T >, [40](#)
- Source, [42](#)
 - executeFunction, [43](#)
 - getOutput, [43](#)
 - setOutput, [44](#)
- Switch, [44](#)
 - executeFunction, [46](#)
 - getState, [46](#)
- XNOR, [47](#)
- XOR, [49](#)
 - performLogicCalculation, [51](#)