

My Project

Generated by Doxygen 1.13.2

1 Format plików wejściowych	1
1.1 Opis układu (opis_ukladu.txt)	1
1.2 Zestawy wejść (wejscie.txt)	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 BramkaPrototyp Class Reference	7
4.1.1 Detailed Description	7
4.1.2 Constructor & Destructor Documentation	8
4.1.2.1 BramkaPrototyp()	8
4.1.3 Member Data Documentation	8
4.1.3.1 czas_propagacji	8
4.1.3.2 czas_skumulowany	8
4.1.3.3 typ	8
4.1.3.4 value	8
4.1.3.5 wejscie1	9
4.1.3.6 wejscie2	9
4.1.3.7 wynik	9
5 File Documentation	11
5.1 BramkaPrototyp.cpp File Reference	11
5.1.1 Detailed Description	11
5.1.2 Function Documentation	11
5.1.2.1 obliczBramke()	11
5.2 BramkaPrototyp.h File Reference	12
5.2.1 Detailed Description	12
5.2.2 Function Documentation	12
5.2.2.1 obliczBramke()	12
5.3 BramkaPrototyp.h	13
5.4 main.cpp File Reference	13
5.4.1 Detailed Description	13
5.4.2 Function Documentation	14
5.4.2.1 main()	14

Chapter 1

Format plików wejściowych

1.1 Opis układu (opis_ukladu.txt)

Plik zawiera opis kolejnych bramek logicznych. Każda linia ma postać:

typ wejście1 wejście2 wyjście czas_propagacji

Przykład: NAND a a c 10

1.2 Zestawy wejść (wejscie.txt)

Pierwsza linia zawiera nagłówki sygnałów wejściowych. Kolejne linie zawierają wartości logiczne sygnałów.

Przykład: a b 0 1

Chapter 2

Class Index

2.1 Class List

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

BramkaPrototyp	Reprezentuje pojedynczą bramkę logiczną w symulowanym układzie	7
--------------------------------	--	-------------------

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

BramkaPrototyp.cpp		
Implementacja klasy BramkaPrototyp i funkcji obliczania bramek logicznych	11
BramkaPrototyp.h		
Deklaracja klasy BramkaPrototyp oraz funkcji obsługujących bramki logiczne	12
main.cpp		
Program symulujący działanie układu bramek logicznych	13

Chapter 4

Class Documentation

4.1 BramkaPrototyp Class Reference

Reprezentuje pojedynczą bramkę logiczną w symulowanym układzie.

```
#include <BramkaPrototyp.h>
```

Public Member Functions

- **BramkaPrototyp** (std::string typA, std::string in1, std::string in2, std::string wynikA, int czasA)
Konstruktor klasy BramkaPrototyp.

Public Attributes

- std::string **typ**
Typ bramki logicznej (np. AND, OR, NAND, XOR)
- std::string **wejscie1**
Nazwa pierwszego sygnału wejściowego.
- std::string **wejscie2**
Nazwa drugiego sygnału wejściowego.
- std::string **wynik**
Nazwa sygnału wyjściowego bramki.
- int **czas_propagacji**
Czas propagacji sygnału przez bramkę
- int **value**
Aktualna wartość logiczna wyjścia (0, 1 lub -1 gdy nieobliczona)
- int **czas_skumulowany**
Skumulowany czas propagacji sygnału do tej bramki.

4.1.1 Detailed Description

Reprezentuje pojedynczą bramkę logiczną w symulowanym układzie.

Klasa przechowuje informacje o typie bramki logicznej, nazwach sygnałów wejściowych i wyjściowych oraz danych potrzebnych do symulacji sygnału w czasie.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 BramkaPrototyp()

```
BramkaPrototyp::BramkaPrototyp (
    std::string typA,
    std::string in1,
    std::string in2,
    std::string wynikA,
    int czasA)
```

Konstruktor klasy [BramkaPrototyp](#).

Inicjalizuje bramkę logiczną na podstawie podanych danych

Parameters

<i>typA</i>	typ bramki logicznej
<i>in1</i>	nazwa pierwszego wejścia
<i>in2</i>	nazwa drugiego wejścia
<i>wynikA</i>	nazwa sygnału wyjściowego
<i>czasA</i>	czas propagacji bramki

Ustawia wszystkie pola obiektu bramki oraz inicjalizuje wartości (value = -1 oznacza brak obliczonego wyniku).

4.1.3 Member Data Documentation

4.1.3.1 czas_propagacji

```
int BramkaPrototyp::czas_propagacji
```

Czas propagacji sygnału przez bramkę

4.1.3.2 czas_skumulowany

```
int BramkaPrototyp::czas_skumulowany
```

Skumulowany czas propagacji sygnału do tej bramki.

4.1.3.3 typ

```
std::string BramkaPrototyp::typ
```

Typ bramki logicznej (np. AND, OR, NAND, XOR)

4.1.3.4 value

```
int BramkaPrototyp::value
```

Aktualna wartość logiczna wyjścia (0, 1 lub -1 gdy nieobliczona)

4.1.3.5 **wejscie1**

```
std::string BramkaPrototyp::wejscie1
```

Nazwa pierwszego sygnału wejściowego.

4.1.3.6 **wejscie2**

```
std::string BramkaPrototyp::wejscie2
```

Nazwa drugiego sygnału wejściowego.

4.1.3.7 **wynik**

```
std::string BramkaPrototyp::wynik
```

Nazwa sygnału wyjściowego bramki.

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

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

Chapter 5

File Documentation

5.1 BramkaPrototyp.cpp File Reference

Implementacja klasy [BramkaPrototyp](#) i funkcji obliczania bramek logicznych.

```
#include "BramkaPrototyp.h"
```

Functions

- int [obliczBramke](#) ([BramkaPrototyp](#) &br, int val1, int val2)
Implementacja obliczania wartości bramki logicznej.

5.1.1 Detailed Description

Implementacja klasy [BramkaPrototyp](#) i funkcji obliczania bramek logicznych.

5.1.2 Function Documentation

5.1.2.1 [obliczBramke\(\)](#)

```
int obliczBramke (
    BramkaPrototyp & br,
    int val1,
    int val2)
```

Implementacja obliczania wartości bramki logicznej.

Oblicza wartość logiczną wyjścia bramki.

Na podstawie typu bramki (np. AND, OR, NAND) wykonywana jest odpowiednia operacja logiczna na wejściach.

5.2 BramkaPrototyp.h File Reference

Deklaracja klasy [BramkaPrototyp](#) oraz funkcji obsługujących bramki logiczne.

```
#include <string>
```

Classes

- class [BramkaPrototyp](#)
Reprezentuje pojedynczą bramkę logiczną w symulowanym układzie.

Functions

- int [obliczBramke](#) ([BramkaPrototyp](#) &br, int val1, int val2)
Oblicza wartość logiczną wyjścia bramki.

5.2.1 Detailed Description

Deklaracja klasy [BramkaPrototyp](#) oraz funkcji obsługujących bramki logiczne.

Plik zawiera definicję klasy reprezentującej bramkę logiczną używaną w symulacji układu logicznego wraz z deklaracją funkcji obliczającej wartość wyjścia bramki.

5.2.2 Function Documentation

5.2.2.1 [obliczBramke\(\)](#)

```
int obliczBramke (
    BramkaPrototyp &br,
    int val1,
    int val2)
```

Oblicza wartość logiczną wyjścia bramki.

Funkcja na podstawie typu bramki zapisanej w obiekcie [BramkaPrototyp](#) oblicza wynik operacji logicznej dla podanych wartości wejściowych.

Parameters

<i>br</i>	referencja do obiektu bramki
<i>val1</i>	wartość pierwszego wejścia (0 lub 1)
<i>val2</i>	wartość drugiego wejścia (0 lub 1)

Returns

wynik operacji logicznej (0 lub 1)

Oblicza wartość logiczną wyjścia bramki.

Na podstawie typu bramki (np. AND, OR, NAND) wykonywana jest odpowiednia operacja logiczna na wejściach.

5.3 BramkaPrototyp.h

[Go to the documentation of this file.](#)

```
00001
00009
00010 #ifndef BRAMKAPROTOTYP_H
00011 #define BRAMKAPROTOTYP_H
00012
00013 #include <string>
00014
00023 class BramkaPrototyp {
00024 public:
00026     std::string typ;
00027
00029     std::string wejsciel;
00030
00032     std::string wejście2;
00033
00035     std::string wynik;
00036
00038     int czas_propagacji;
00039
00041     int value;
00042
00044     int czas_skumulowany;
00045
00057     BramkaPrototyp(std::string typA, std::string in1, std::string in2, std::string wynikA, int czasA);
00058 };
00059
00071 int obliczBramke(BramkaPrototyp &br, int val1, int val2);
00072
00073 #endif // BRAMKAPROTOTYP_H
00074
```

5.4 main.cpp File Reference

Program symulujący działanie układu bramek logicznych.

```
#include <iostream>
#include <fstream>
#include <vector>
#include <string>
#include "BramkaPrototyp.h"
```

Functions

- int `main ()`

Główna funkcja programu.

5.4.1 Detailed Description

Program symulujący działanie układu bramek logicznych.

Program wczytuje opis układu bramek oraz zestawy wejść z plików tekstowych, a następnie symuluje propagację sygnałów przez układ z uwzględnieniem czasów propagacji poszczególnych bramek.

5.4.2 Function Documentation

5.4.2.1 main()

```
int main ()
```

Główna funkcja programu.

Odpowiada za:

- wczytanie opisu układu bramek z pliku,
- wczytanie zestawów wejściowych,
- symulację propagacji sygnałów,
- zapis i wyświetlenie wyników.

Returns

0 w przypadku poprawnego zakończenia programu