

## Description

Il s'agit d'un projet écrit en *C++* pour simuler le fonctionnement de *circuits combinatoires*.

## Fonctionnalités

- Création d'un circuit combinatoire récursif, les deux entrées d'un circuit combinatoire peuvent être deux autres circuits, un circuit et un input ou bien deux inputs.
- Afficher un circuit combinatoire sous forme textuelle.
- Afficher un circuit combinatoire en console.
- Sauvegarder la forme textuelle d'un circuit dans un fichier.
- Lire un circuit à partir d'un fichier en utilisant sa forme textuelle.

## Installation

1. Ouvrir une ligne de commande dans le même emplacement de projet.
2. Compiler le projet :

`make`

3. Lancer le main :

`make test`

4. Pour supprimer les fichiers objets (file.o):

`make clean`

## Usage

### Dans le main

- Créer un *InputGate* : Le constructeur de *InputGate* accepte deux valeurs, un **nom** de type *char* et une **valeur** de type *booléen*.

```
InputGate * a = new InputGate('a', false);  
InputGate * b = new InputGate('b', true);
```

- Créer une *Gate* *And*, *Nand*, *Or*, *Nor*, *Xor*, *Nxor* : Pour Créer une de ces *Gates*, on doit intrduire deux valeurs. Ces deux valeurs peuvent être deux autres *Gate*, un *Gate* et un *InputGate* ou bien deux *InputGate*.

```
AndGate * andGate = new AndGate(a, b);  
OrGate * orGate = new OrGate(a, andGate);  
XorGate * xorGate = new XorGate(andGate, orGate);
```

- Créer une *NegationGate* : Pour Créer une *NegationGate*, on doit intrduire une seule valeur. Cette valeur peut être une autre *GATE* ou bien un *InputGate*.

```
NegationGate * negation_one = new NegationGate(a);
NegationGate * negation_two = new NegationGate(andGate);
```

- Créer une **OutputGate** : Pour créer une **OutputGate**, on doit fournir deux valeurs, la première valeur étant le **nom** dont le type est *char*. La deuxième valeur c'est le **output** dont le type est une **GATE** ou bien un **InputGate**.

```
OutputGate * A = new OutputGate('A', a);
OutputGate * B = new OutputGate('B', andGate);
```

- Afficher un circuit combinatoire sous forme textuelle :

```
A->displayTextualForm();
```

- Afficher un circuit combinatoire en **console** :

```
A->displayInConsole();
```

- Sauvegarder la forme textuelle d'un circuit dans un fichier : Un fichier du même nom que **OutputGate** sera créé, ce fichier contenant la forme textuelle du circuit.

```
A->saveInFile();
```

- Synthétiser le **OutputGate** à partir d'une forme textuelle écrit dans un fichier.

```
OutputGate * A = new OutputGate("A.txt");
```