

Prise en main de l'outil Logisim

I. Introduction à l'outil et objectif

Logisim est un outil d'édition et de simulation de circuits. L'utilisation de ce type d'outil est très courante dans le processus de conception de circuits intégrés, et il en existe d'ailleurs de nombreux exemples disponibles sur internet. Le choix de Logisim pour ce module s'explique par sa grande facilité d'utilisation, l'aspect pédagogique de son simulateur, sa gratuité ainsi que la possibilité d'utilisation sur toutes les plateformes.

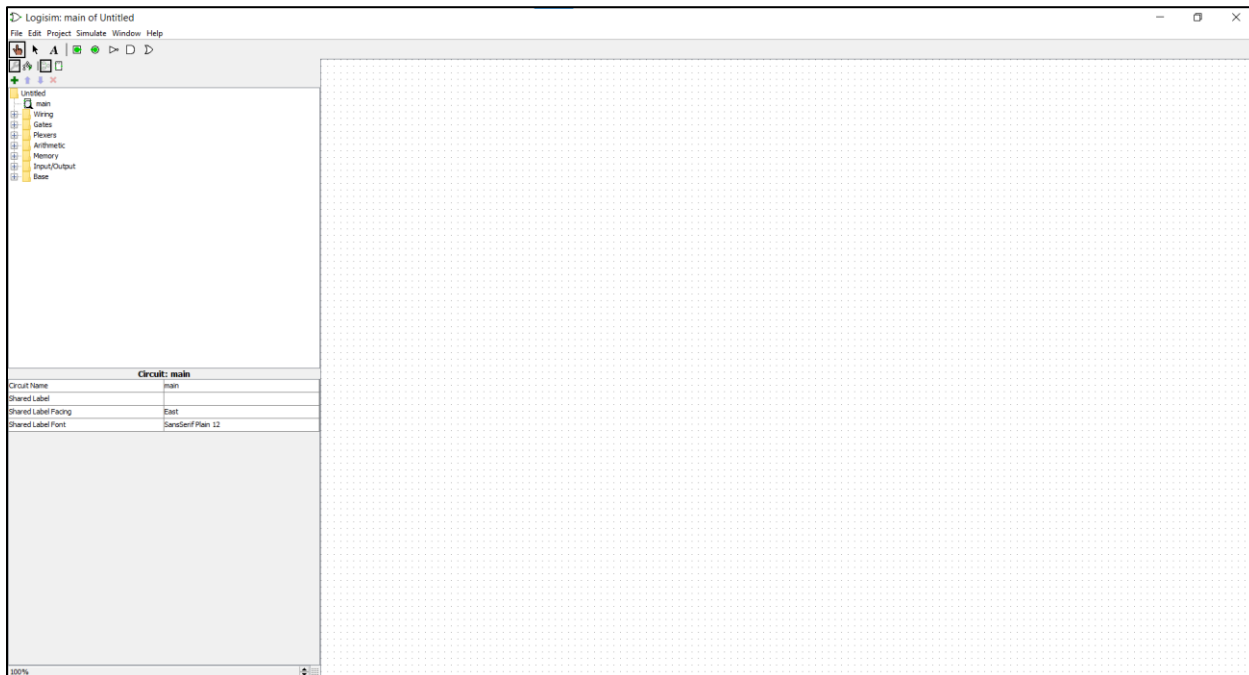


Figure 1 : Interface du Logiciel Logisim

1) Installation

L'outil Logisim est disponible sur toutes les plateformes (Linux, MacOS, Windows). Vous pouvez le télécharger à l'adresse suivante : <http://www.cburch.com/logisim/download.html>. Si vous êtes sous Linux, vous pouvez lancer le logiciel sans forcément l'installer. Une fois que vous avez téléchargé le fichier logisim-XX.jar, vous pouvez lancer le logiciel à l'aide de la commande : **java -jar logisim- XX.jar**

2) Premier circuit

Une fois le logiciel lancé, l'interface graphique de l'outil s'affiche. Alors, une feuille de travail vierge est automatiquement disposée devant vous. La réalisation de circuits avec cet outil se réalise en 3 étapes :

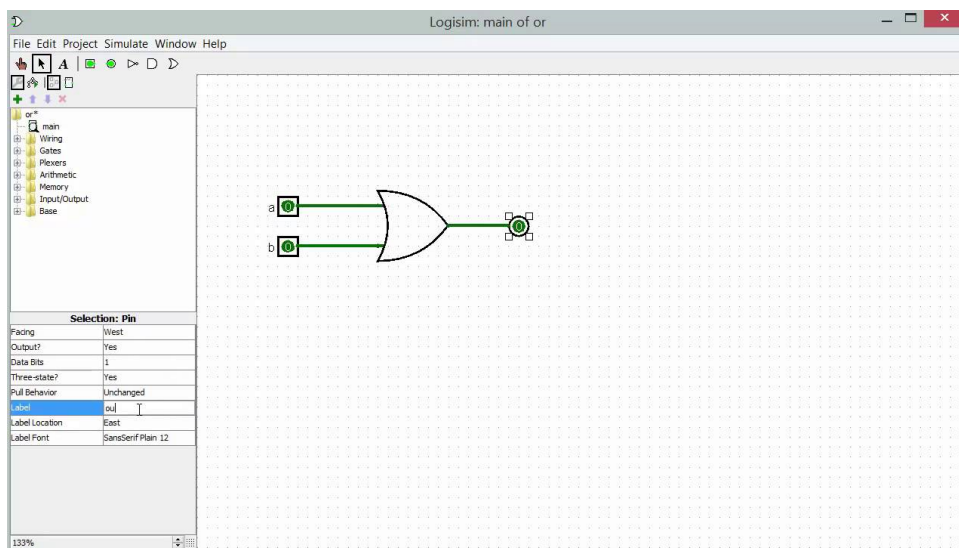
- i. Edition du modèle (réalisation de circuits basiques) ;
- ii. Instanciation du modèle (appel au(x) modèle(s) pour les utiliser dans des circuits plus complexes) ;
- iii. Simulation.

Pour toute information supplémentaire consulter l'aide de l'outil dans la barre des tâches. Vous allez réaliser une première édition de circuit. Pour cela, suivez les étapes suivantes :

a) Création de la porte logique OR

Nous allons à présent créer une porte logique **OU** (OR gate en anglais). Pour cela, on clique dans la barre d'outils sur l'élément logique correspondant, puis on clique dans le canvas à l'endroit où on souhaite le déposer. Cette porte réalise l'opération logique **OU** entre deux variables d'entrée **a** et **b** et nous donne son résultat **s** en sortie. Nous allons ajouter deux entrées, représentées par un carré vert dans la barre d'outils et une sortie, représentée par un rond vert.

Il est possible d'ajouter un label (une étiquette) pour chaque entrée et chaque sortie en modifiant le paramètre Label dans le tableau d'attributs. Enfin, nous allons relier les entrées et les sorties à la porte logique en créant des fils avec le bouton gauche de la souris.



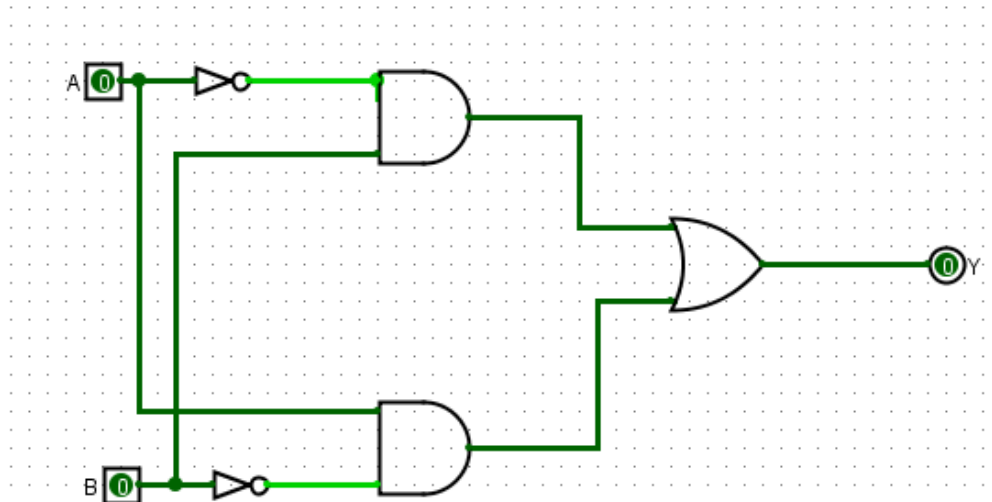
Reliez l'ensemble du circuit et simulez son fonctionnement. La simulation se fait simplement en passant en mode simulation en appuyant sur le bouton **(représenté par une main)** qui est tout en haut à gauche. Vous pouvez visualiser l'évolution de certaines entrées ou sorties en cliquant sur ces dernières. Vous pouvez durant la simulation interagir avec les activateurs binaires pour changer les valeurs injectées aux entrées...

IMPORTANT : Notez qu'en plus des "Pin" (Input/Output, **représenté par un carré et un cercle vert**), vous pouvez utiliser toute une batterie d'Input/Output, comme des LEDs, des Boutons poussoirs, des afficheurs 7-segments, etc. Vous les trouverez dans le sous répertoire "Input/Output".

b) Création de la porte logique XOR

Cliquez sur Project > Add circuit et ajouter « XOR » comme nom du circuit. Vous remarquerez la création d'un nouveau fichier nommé XOR.

Réalisez le circuit de la porte logique XOR schématisé dans la figure suivante et simulez son fonctionnement. Vous pouvez vérifier la table de vérité en cliquant sur « *Analyze circuit* » sous *Project*



c) Réalisation de la macro XOR et simulation

Si nous voulons utiliser le XOR, nous devons d'abord encapsuler le circuit réalisé précédemment dans une boîte, appelée Macro, qui résume ce circuit sous forme d'une simple boîte avec des entrées/sorties. Pour ce faire, cliquez sur "**Projet → Edit circuit appearance**" ou bien sur l'icône qui est au bout à droite de la deuxième ligne des icônes. Vous pourrez alors éditer la Macro du circuit et la redessiner comme vous le souhaitez. Si vous cliquez sur les "Pin" d'entrée/sortie de votre Macro, vous pourrez visualiser leurs positions sur le circuit (et aussi les changer !).

d) Intégration de la Macro XOR dans un autre circuit

Une fois la macro XOR validée, vous pouvez instancier votre circuit (faire appel au circuit) pour la réalisation d'autres fonctions logiques. Ajouter un autre circuit à votre projet que vous appellerez "Additionneur". Double cliquez sur "Additionneur" dans l'arborescence en dessous du "main". Vous aurez une page vide. Cliquez ensuite (une fois !!) sur le XOR et cliquez une fois dans la page vide. La macro du XOR que vous avez précédemment réalisée est alors instanciée pour la réalisation du "Additionneur". En vous aidant du concept d'additionneur vu lors du CM, réalisez un "Demi-Additionneur" et simulez son fonctionnement.

IMPORTANT : Vous remarquerez que vous instanciez les XOR de la même façon que les portes logiques présentes dans les bibliothèques de Logisim. Vous pouvez explorer ces bibliothèques pour chercher des circuits dont vous avez l'utilité.

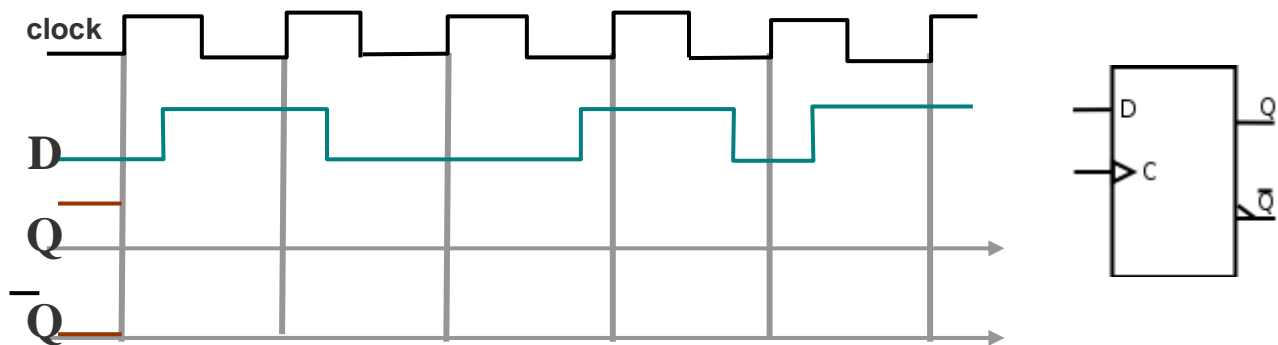
II. Circuits séquentiels

1) Bascule D :

Une bascule (Composant D Flip-flop dans le répertoire Memory) est une mémoire 1 bit. Une bascule reçoit un signal d'horloge, et présente une entrée et une sortie.

Remarque : En plus de pouvoir simuler en temps réel un schéma logique, Logisim peut représenter cette simulation sous la forme d'un chronogramme (voir section *Configuration du Chronogramme*).

Branchez une bascule D de Logisim et servez-vous de ce circuit pour compléter le chronogramme suivant :



2) Bascule JK :

JK est la bascule la plus largement utilisée. Il est considéré comme un circuit de bascule universel.

- a) **Rappeler** l'équation que vérifie la bascule JK après chaque top d'horloge (\uparrow) en vous aidant de la table de vérité.

Equation JK

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

- b) Branchez une bascule **JK** de Logisim et servez-vous de ce circuit pour compléter le chronogramme suivant :

