

TP1

Pour ces TP, on va utiliser le logiciel Logisim pour simuler facilement des circuits logiques : à défaut d'implanter un vrai processeur en Logisim, l'objectif sera d'implanter, dans les quelques séances qui nous sont imparties, une petite « calculatrice programmable. » Cela donnera déjà un certain nombre d'idées sur les circuits requis pour implanter l'architecture d'un processeur.

Le but de ce premier TP est de se familiariser avec Logisim, et de réaliser certains circuits combinatoires que l'on utilisera par la suite. On part du principe que les TP sont faits sous Linux, et que vous êtes déjà un peu autonome avec un terminal.

1.1. Avant de commencer

Créez un répertoire pour les TP de LIFASR3, puis, dans votre navigateur web, créez un favori pour la page de Logisim (Google « Logisim ») : <http://www.cburch.com/logisim/>.

Télécharger l'archive `logisim-generic-2.7.1.jar` de Logisim dans votre répertoire de TP : vous lancerez le logiciel en entrant la commande `java -jar logisim-generic-2.7.1.jar` dans un terminal.

1.2. Premiers exercices

Une des particularités de Logisim est de pouvoir éditer et simuler son circuit en même temps.

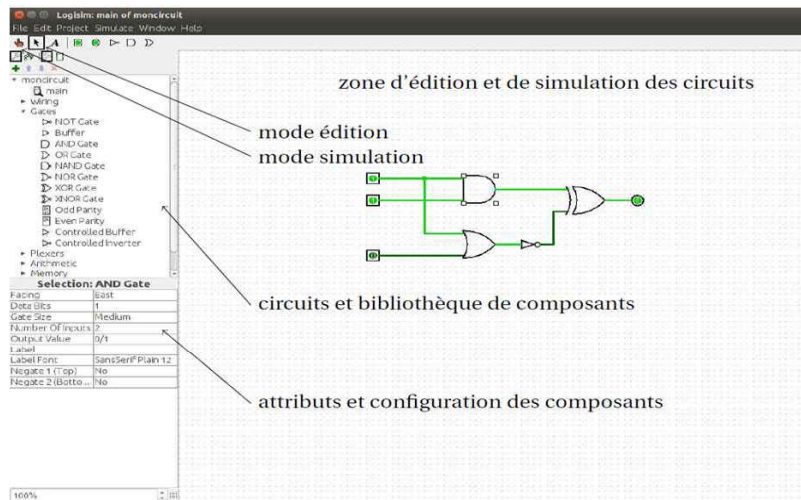


FIGURE 1.1 – Fenêtre de Logisim.

1.2.1 Mode édition

Pour utiliser le mode édition, il faut sélectionner la flèche comme indiqué en haut de la figure 1.1.

1. On peut alors choisir un composant dans la bibliothèque, sur la gauche. Pour l'ajouter dans son schéma, il suffit de cliquer sur le composant désiré, puis de le déposer dans le schéma.
2. Chaque composant que vous utiliserez aura des attributs modifiables dans la zone inférieur gauche de Logisim. Par exemple si l'on pose une porte AND, on peut modifier le nombre de signaux qu'elle prend en entrée (attribut `Number Of Inputs`), ou son orientation (attribut `Facing`).
3. Il est possible de faire des copier/coller d'un ou plusieurs composants (sélectionner, puis `Ctrl+C/V` ou bien `Ctrl+D`). Dans ce cas, les composants conserveront aussi tous les attributs préalablement définis.

Des éléments dont vous aurez besoin rapidement :

- Pour les entrées, l'élément Pin de Wiring dans la bibliothèque, avec l'attribut `output?=no` (voir aussi le « petit carré » dans la barre d'outils).
- Pour les sorties, l'élément Pin avec `output?=yes` ; (voir aussi le « petit rond » dans la barre d'outils).
- Les portes logiques sont présentes dans le répertoire Gates de la bibliothèque.

Notes pour plus tard :

- pour tous les composants qui ont cet attribut, vérifiez que **ThreeState=No**,
- méfiez vous « des fils qui se touchent » en particuliers entres les entrées rapprochées des portes logiques !

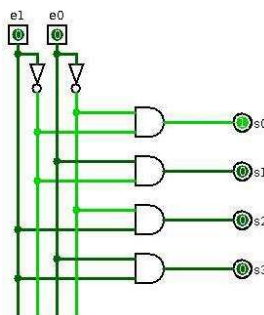
1.2.2 Mode simulation

Logisim est capable de simuler le circuit en affichant les valeurs des signaux directement sur le schéma : l'utilisateur peut définir les valeurs des bits en entrée et observer la réaction du circuit.

1. Pour utiliser le mode simulation, il faut sélectionner la main en haut à gauche de Logisim.
2. Il est possible de contrôler l'état des différentes entrées en cliquant directement dessus avec la « petite main » (forcément, uniquement en mode simulation).
3. En cliquant sur une entrée, la valeur doit alterner entre 0 en vert clair, et 1 en vert foncé.
4. Les sorties prennent les valeurs calculées par le circuit en fonction des entrées. Pour les fils qui ne transportent qu'un bit, la convention des couleurs est identique : 0 en vert clair, et 1 en vert foncé.

Exercice 1.2.1 : Un premier circuit pour apprendre à se servir de l'interface.

- 1) Ouvrez Logisim, et reproduisez le circuit ci-dessous aussi fidèlement que possible. Indications :
 - Pour orienter les entrées (Pin avec `output?=no`), utilisez leur attribut `Facing`. Pour nommer les entrées, utilisez l'attribut `Label`, et pour placer le nom au bon endroit, l'attribut `Label Location`.
 - Les mêmes remarquent valent pour les sorties (Pin avec `output?=yes`).
 - Les portes AND doivent présenter deux entrées, donc utilisez `Number Of Inputs=2`.
 - Faites des copiers-collers, quitte à modifier les attributs ensuite!
 - Pour effacer un composant ou un fil : placez vous dessus, puis faites un click-droit et `Delete`.



- 2) Enregistrez votre travail dans le fichier `exo11.circ` (`File`→`Save`).
- 3) En vous mettant en mode simulation, complétez la table de vérité suivante :

e_1	e_0	s_3	s_2	s_1	s_0
0	0				
0	1				
1	0				
1	1				

- 4) Comment s'appelle le circuit que l'on vient de réaliser ?

Exercice 1.2.2 : Un deuxième petit circuit

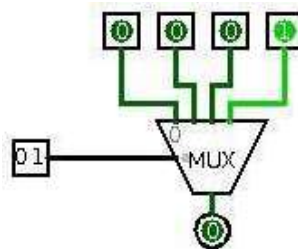
Commencez par créer un nouveau projet avec `File`→`New`. Vous enregistrerez votre travail dans le fichier `exo12.circ`. Dans cet exercice, on expérimente autour de multiplexeurs 4 bits vers 1 bit.

- 1) Notre multiplexeur doit comporter :
 - 4 bits e_0, e_1, e_2, e_3 en entrée,

- 1 bit s en sortie,
 - les bits de sélection c_1 et c_0 également en entrée.
- L'entrée dont l'indice est donné par $(c_1 c_0)_2$ doit être envoyée sur la sortie s . Indiquez quelle valeur doit prendre s en fonction des entrées du circuit dans le tableau suivant.

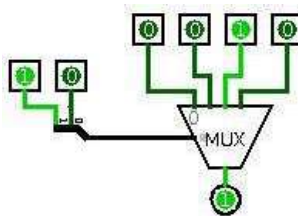
c_1	c_0	s
0	0	
0	1	
1	0	
1	1	

- 2) Donnez une formule pour s , en n'utilisant que les fonctions AND, OR et NOT.
.....
- 3) Implantez¹ votre multiplexeur dans Logisim, en n'utilisant que les fonctions AND, OR et NOT. Comment faire pour tester son bon fonctionnement?
.....
- 4) Maintenant, trichons : il y a déjà un composant multiplexeur dans la bibliothèque de Logisim ! Instanciez² un composant **Multiplexer** du répertoire **Plexers**, en mettant bien dans les attributs **Include Enable?=No** et **Disabled Output=Zero**. Implantez ensuite le circuit suivant, et testez le :



Indications :

- Pour le multiplexeur, il faut mettre **Include Enable?=No** et **Disabled Output=Zero**, puis choisir judicieusement les attributs **Data Bits** et **Select Bits**.
 - Notez que **l'entrée située sur la gauche dans le circuit regroupe 2 bits** : il s'agit d'une entrée **Pin** avec **Three-state?=No** (toujours vérifier !), **Output?=No** (logique) et **DataBits=2**.
- 5) On modifie le circuit précédent pour faire apparaître individuellement les deux entrées de sélection. Pour cela, on utilise un composant qui permet de séparer le paquet de deux fils de sélection du multiplexeur en deux fils : ce composant est le **Splitter** du répertoire **Wiring**. Implantez le circuit suivant, et testez le :



Un **Splitter** présente deux attributs entiers importants : **Bit Width In** et **Fan Out**. Un **Splitter** prend d'un côté un bus de (**Bit Width In**) bits, et les répartit en (**Fan Out**) petits paquets.

Exercice 1.2.3 : Multiplexeurs 'aide des portes logiques

Dans cet exercice, vous n'utiliserez que des portes AND ou OR à deux entrées, et des portes NOT.

- 1) Implantez dans Logisim un circuit **mux41table** qui réalise un multiplexeur 4 bits vers 1 bits, en utilisant la technique de la table de vérité (comme on l'a déjà vu précédemment). Combien de portes logiques

1. Le verbe est ici de la même façon que lorsqu'on dit « je vais planter mon algorithme de tri en C. »
 2. Verbe compliqué mais d'usage courant qui veut dire ici « placez dans votre circuit ».

- avez-vous utilisées?
-
- 2) Implantez un circuit `mux21` qui réalise un multiplexeur 2 bits vers 1 bits, en utilisant la technique de la table de vérité. Combien de portes logiques avez-vous utilisées?
-
- 3) Réalisez un circuit `mux41` en utilisant comme brique de base le circuit `mux21`. En tout, combien de portes logiques comporte votre circuit?
-
- 4) Que peut-on conclure?
-
-