

1 Introduction

Logisim est un logiciel open-source permettant de concevoir et de simuler des circuits logiques.

Ce document est un tutoriel qui décrit comment établir un système numérique à l'aide de cet éditeur de schéma. Nous expliquerons les démarches nécessaires afin de concevoir et simuler un demi-additionneur à 1 bit puis un additionneur complet à un bit. Enfin, nous simulerons un additionneur à 4 bits.

Il existe différentes façons de décrire formellement les systèmes numériques : des langages de description du matériel (HDL), des tables de vérité, des graphes d'états, ou des schémas. Logisim permet uniquement de travailler sur des schémas.

Vous pouvez voir l'interface de Logisim sur la Figure 1.

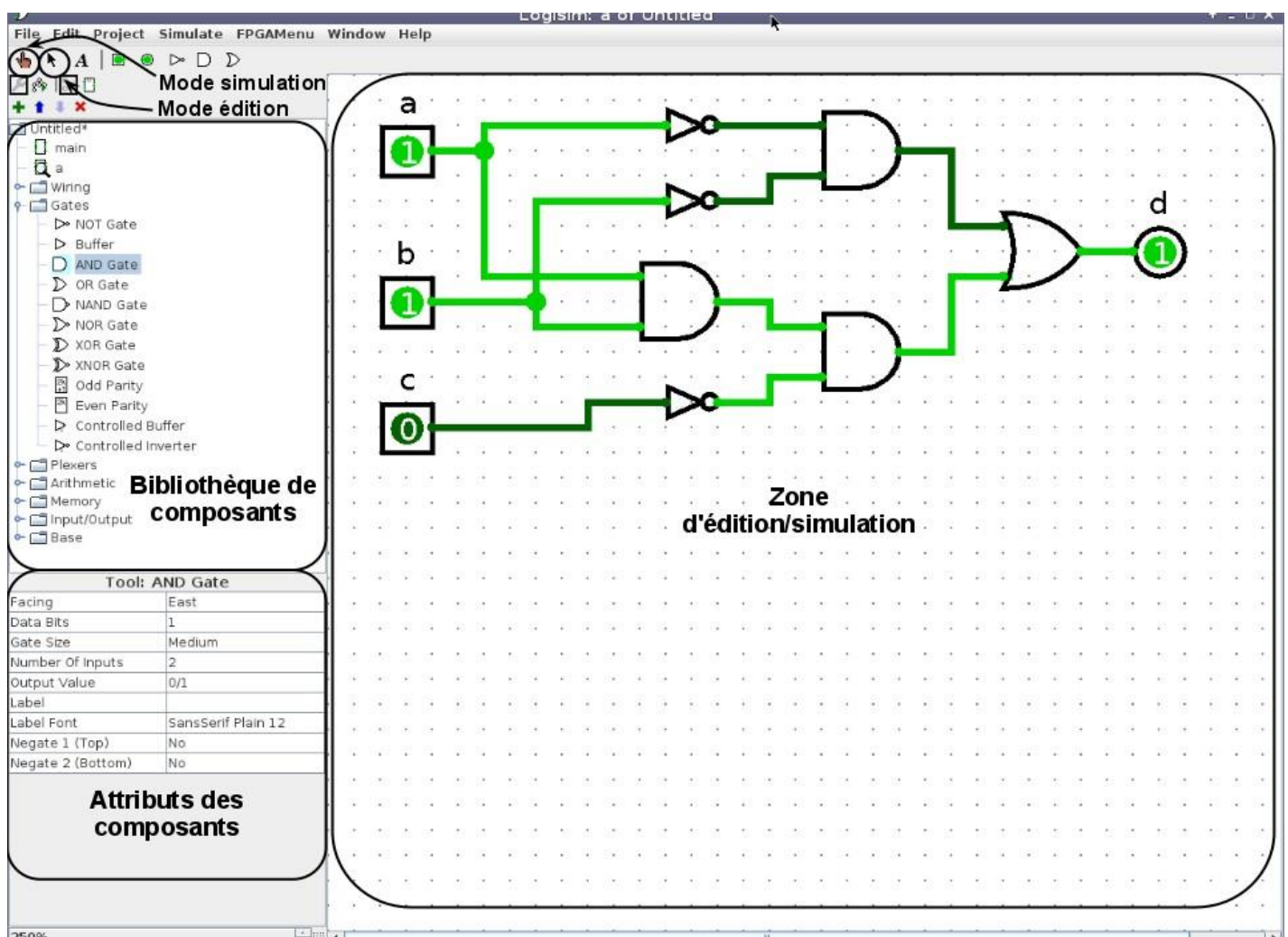


FIGURE 1. Interface de Logisim

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

2 Installation de Logisim sur votre Raspberry

- Vérifiez que vous êtes bien connecté à Internet : le symbole " double flèches bleues" doit être allumé en haut à droite de l'écran.

Sinon, dans le terminal, avec la ligne de commande `~ $ sudo raspi-config` allez dans 2. Network Options Wi-fi

Entrez le SSID et le mot de passe de votre connexion Wi-fi s'il y en a un.

Puis sélectionnez Finish appuyez sur Entrée

Le symbole "flèches vers le haut et le bas bleues" doit alors s'allumer en haut à droite de l'écran.

Une fois connecté à Internet :

- Saisissez dans le terminal les commandes suivantes :

Commande	Commentaire
<code>~ \$ sudo apt-get update</code>	<p>su pour se connecter en tant que super user autrement dit utilisateur qui a toutes les permissions.</p> <p>do signifie "faire".</p> <p>sudo indique que vous allez exécuter la commande qui suit en mode "super user".</p> <p>Raspbian va donc vous demander votre mot de passe et vérifier que votre compte d'utilisateur est bien un compte d'administrateur.</p> <p>apt-get est une commande qui permet de trouver, d'installer, de mettre à jour et de supprimer des "packages" (des ensembles de programmes) depuis des sources authentifiées, ce qui évite d'installer par exemple des programmes malveillants.</p> <p>apt est l'acronyme de <i>Advanced Packaging Tool</i>.</p> <p>apt-get update permet d'obtenir les dernières mises à jour de votre système d'exploitation Linux Raspbian.</p> <p>Il faut le faire avant toute nouvelle installation de programme.</p>

Fermez le terminal.

- Nous poursuivons l'installation à l'aide de l'outil d'installation intégré de Raspbian :
1. Cliquez sur l'icône de la framboise en haut à gauche de l'écran puis dans le menu qui s'ouvre, cliquez sur "Préférences". Dans le menu qui s'ouvre cliquez sur "Add / Remove Software".
 2. Dans la fenêtre qui s'ouvre, saisissez dans la zone de recherche en haut à gauche **Logisim** et appuyez sur la touche Entrée.
 3. Cochez la case devant le dossier jaune logisim-2.7.1 et cliquez sur le bouton OK en bas à droite.
 4. Saisissez votre mot de passe pour continuer l'installation de logisim.
 5. L'installation des paquets se poursuit (un paquet – package en anglais – est une archive (fichier compressé) comprenant les informations et procédures nécessaires à l'installation d'un logiciel sur un système d'exploitation, en particulier sur Linux).
 6. Une fois l'installation des paquets terminée, cliquez à nouveau sur la framboise en haut à gauche, allez dans le menu "Education" et cliquez sur Logisim.
 7. Agrandissez la fenêtre de Logisim à tout l'écran.

3 Mode édition

1. Pour utiliser le **mode édition**, il faut simplement **sélectionner la flèche** comme indiqué en haut de la figure 1.
2. 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 cliquer sur le schéma.
3. Chaque composant que vous utiliserez aura des attributs modifiables dans la zone inférieure gauche de Logisim. Par exemple, si on pose une porte AND, on peut modifier le nombre de signaux d'entrée.
4. Voici un descriptif des éléments dont vous allez avoir besoin pour ce tutoriel :
 - Pour les entrées, l'élément Pin de la bibliothèque *Wiring*.
 - Pour les sorties, l'élément Pin de la bibliothèque *Wiring* avec l'attribut `output?=yes`.
 - Les portes logiques sont présentes dans la bibliothèque *Gates*.
5. Une fois qu'on a posé tous les composants, il faut alors les connecter. Pour cela, il suffit de placer le curseur de la souris sur un des ports à connecter et, en gardant pressé le bouton gauche de la souris, le déplacer jusqu'au port de destination.
6. Faites attention à l'orientation des composants (le 'Facing'). Il correspond à l'endroit où le fil arrive.

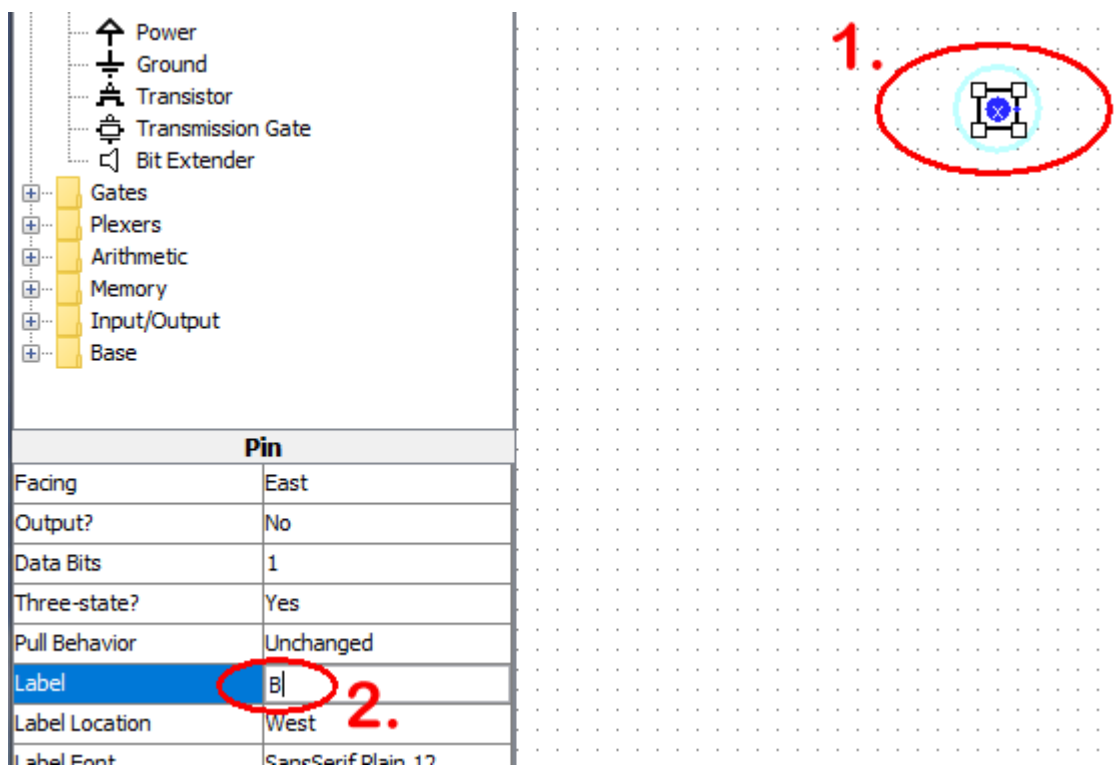
4 Création d'un premier circuit

Tous les circuits réalisés dans Logisim peuvent être réutilisés dans d'autres circuits. Afin de créer un nouveau circuit, il faut aller dans Project -> Add circuit... -> nommer le circuit. Le circuit créé devient un composant disponible dans la bibliothèque. En suivant les étapes décrites ci-après, on va réaliser le schéma de la Figure 4.

Allez dans *File / Save as* et enregistrez dans le dossier Documents en tant que projet logisim le fichier **half_adder**

1. Placez deux entrées en prenant des Pin dans *Wiring* et en les déposant sur le canevas du schéma.

Ecrivez leur label, A pour l'une et B pour l'autre (pour modifier un attribut -ici le label – il faut cliquer sur le composant puis cliquer dans la zone en bas à gauche et faire les modifications).



Vérifiez que les labels que vous avez saisis apparaissent bien à côté des pins sur le schéma. Dans la zone de modification des attributs, vous pouvez changer la taille des caractères, par exemple taille 20, les mettre en gras et les positionner à l'EST, au Sud, à l'Ouest ou au Nord de l'objet.

2. Placez deux sorties en prenant à nouveau des Pin dans *Wiring* et en les déposant sur le schéma. Ecrivez leur label S (la Somme des deux bits A et B) pour l'une et R (la Retenue -Carry en anglais -) pour l'autre. Modifiez aussi leur **attribut Output**. Sa valeur doit être placé sur **Yes**. Les pins prennent alors la forme d'un cercle.
 3. Pour chacune des Pins d'entrée et de sortie, il faut que l'**attribut Three-states** ait la valeur **No**.
 4. Placez les deux portes logiques du demi-additionneur (une AND et une XOR). Pour chacune, réduisez à deux le nombre d'entrées.
 5. Une fois que l'on a posé tous les composants, il faut alors les connecter. Pour cela, il suffit de placer le curseur de la souris au-dessus d'un des ports à connecter (un petit cercle vert apparaît alors) et, en gardant pressé le bouton gauche de la souris, le déplacer jusqu'au port de destination.
 6. Si on s'est trompé pour un fil on peut cliquer droit dessus et choisir Delete pour le supprimer.
 7. Si on s'est trompé dans une manipulation, on peut annuler les dernières manipulations faites en allant dans le menu *Edit* et en cliquant sur la première ligne Undo ...
- Ne pas prendre en compte la couleur des fils ni la valeur des Pin d'entrées (ces dernières sont un X bleu par défaut).
 - Il est possible de déplacer des Pins ou des Portes déjà placées. Il faut placer le curseur au centre du composant. Lorsqu'on *clique gauche* dessus apparaît alors une quadruple flèche et on fait glisser le composant.
 - Vous devez obtenir un schéma ressemblant à celui-ci :

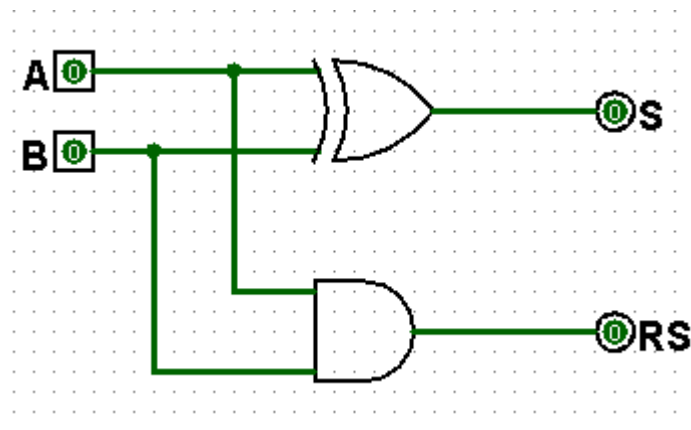


FIGURE 4. demi additionneur 1 bit



Pour voir les 0 ou les 1 en sortie, les fils doivent être connectés correctement sur les Pins, c'est-à-dire à l'aide des petits cercles verts qui apparaissent quand on place la souris au-dessus. On obtient des connexions comme sur la figure 4.

5 Mode simulation

Logisim est capable de simuler le circuit en affichant les valeurs des signaux directement sur le schéma. L'utilisateur peut alors définir les valeurs des bits en entrée et observer le comportement du design.

1. Pour utiliser le **mode simulation**, il faut **sélectionner la main** en haut à gauche de Logisim (cf Figure 1).
2. En cliquant sur une entrée, la valeur doit alterner entre '0' ou '1'.
3. Voici un descriptif des couleurs utilisées pour les signaux en mode simulation.

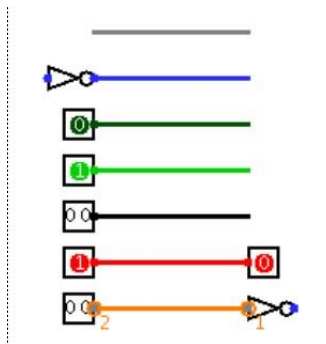


FIGURE 5. Couleurs des fils en simulation

- **Gris** : La taille du fil est inconnue. Le fil n'est relié à aucune entrée ou sortie.
- **Bleu** : Le fil comporte une valeur, cependant elle est inconnue.
- **Vert foncé** : Le fil comporte la valeur '0'.
- **Vert clair** : Le fil comporte la valeur '1'.
- **Noir** : Le fil comporte plusieurs bits (BUS).
- **Rouge** : Le fil comporte une erreur.
- **Orange** : Les composants reliés au fil n'ont pas la bonne taille.

4. Tester le bon fonctionnement de votre demi-additionneur 1 bit.

Rappel sur le fonctionnement du demi-additionneur :

Retenue (carry) RS		0
A		0
B	+	0
Somme S		0

Retenue (carry) RS		0
A		1
B	+	0
Somme S		1

Retenue RS		0
A		0
B	+	1
Somme S		1

Retenue (carry) RS		1
A		1
B	+	1
Somme S		0

On a un demi-additionneur car on seulement une retenue "sortante".

6 Additionneur complet

On va faire maintenant un additionneur complet (*full adder*) puisqu'il *accepte une retenue entrante* (Carry in).

La table de vérité de l'additionneur complet comporte 8 cas :

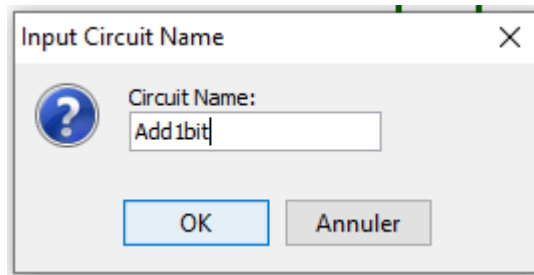
4 cas où la retenue de la colonne précédente, c'est-à-dire la "retenue entrante" vaut 0 :			4 cas où la retenue de la colonne précédente, c'est-à-dire la "retenue entrante" vaut 1 :		
Retenue sortante	0	0	Retenue sortante	0	1
A		0	A		0
B	+	0	B	+	0
Somme S	<hr/>		Somme S	<hr/>	
		0			1
Retenue sortante	0	0	Retenue sortante	1	1
A		0	A		0
B	+	1	B	+	1
Somme S	<hr/>		Somme S	<hr/>	
		1			0
Retenue sortante	0	0	Retenue sortante	1	1
A		1	A		1
B	+	0	B	+	0
Somme S	<hr/>		Somme S	<hr/>	
		1			0
Retenue sortante	1	0	Retenue sortante	1	1
A		1	A		1
B	+	1	B	+	1
Somme S	<hr/>		Somme S	<hr/>	
		0			1

Quittez Logisim (*File / Exit*)

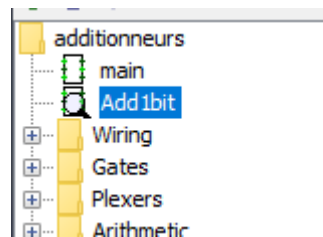
Ouvrez le à nouveau et enregistrez ce nouveau projet dans Documents sous le nom **additionneurs**

Allez dans *Project / Add Circuit*

Nommez le nouveau circuit Add1bit et appuyez sur OK.



Vous disposez maintenant de deux canevas : Main (le principal) et Add1bit où vous allez construire le circuit additionneur complet 1 bit.



Double-cliquez sur l'icône Add1bit (la loupe indique que c'est ce canevas qui est sélectionné). Construisez dans le canevas de Add1bit le circuit d'additionneur complet en suivant la même méthode que pour le demi additionneur, sauf qu'on va commencer par les portes puisque le circuit est plus complexe. Prenez comme exemple le schéma de la p. 182 du livre de première NSI de Serge BAYS.

1. Placez les cinq portes logiques de l'additionneur complet (deux AND, une OR et deux XOR). Pour chacune, réduisez à deux le nombre d'entrées.
2. Placez trois entrées en prenant des Pin dans *Wiring* et en les déposant sur le canevas.

Ecrivez leur label A, B et RE (les bits A et B et la retenue entrante).

3. Placez deux sorties S et RS (la somme et la retenue sortante) en prenant à nouveau des Pins dans *Wiring* et en les déposant sur le schéma. Modifiez aussi leur **attribut Output**. Leur valeur doit être **Yes**. Les pins prennent alors la forme d'un cercle.
 4. Pour chacune des Pins d'entrée et de sortie, il faut que l'**attribut Three-states** ait la valeur **No**.
 5. Une fois que l'on a posé tous les composants, il faut alors les connecter. Pour cela, il suffit de placer le curseur de la souris au-dessus d'un des ports à connecter (un petit cercle vert apparaît alors) et, en gardant pressé le bouton gauche de la souris, le déplacer jusqu'au port de destination.
 6. Si on s'est trompé dans un fil on peut cliquer droit dessus et choisir Delete pour le supprimer.
 7. Si on s'est trompé dans une manipulation, on peut annuler les dernières manipulations faites en allant dans le menu *Edit* et en cliquant sur la première ligne Undo ...
- Ne pas prendre en compte la couleur des fils ni la valeur des Pin d'entrées (ces dernières sont un X bleu par défaut).
 - Il est possible de déplacer des Pins ou des composants déjà placés. Il faut cliquer sur le label ou haut milieu des composants de façon à faire apparaître une quadruple flèche sur l'objet à déplacer.
 - Vous devez obtenir un schéma ressemblant à celui-ci :

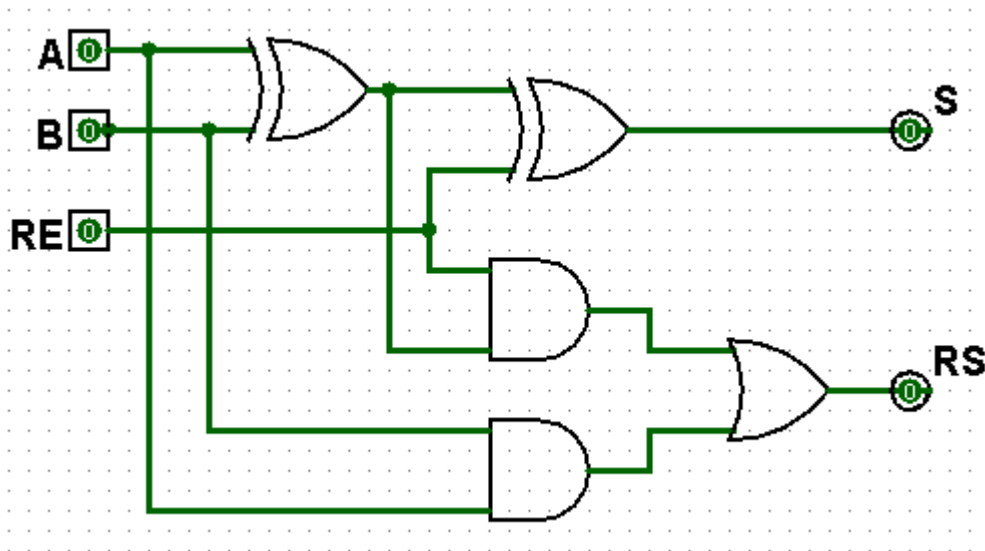


FIGURE 5. Additionneur complet 1 bit

Basculez Logisim en mode Simulation (cliquez sur la main).

En entrant des 0 ou des 1 sur A, B et RE, complétez la table de vérité de S ci-dessous :

A	B	RE	S
0	0	0	
0	1	0	
1	0	0	
1	1	0	
0	0	1	
0	1	1	
1	0	1	
1	1	1	

En entrant des 0 ou des 1 sur A, B et RE, complétez la table de vérité de RS ci-dessous :

A	B	RE	RS
0	0	0	
0	1	0	
1	0	0	
1	1	0	
0	0	1	
0	1	1	
1	0	1	
1	1	1	

Vérifiez que cela correspond aux tables de vérité de la p. 181. Si tout est correct, enregistrez le projet (*File / Save*)

7 Design hiérarchique

La méthodologie de design que l'on vient d'utiliser est valable pour la conception de systèmes numériques relativement simples, c'est-à-dire avec un nombre de portes logiques plutôt faible. Lorsqu'on vise des systèmes plus complexes, le nombre de portes et de connexions augmente rapidement. Dans ce cas, le risque d'erreurs devient très important.

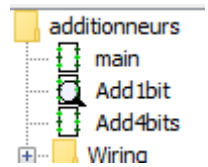
La clé pour gérer correctement une complexité plus grande est d'utiliser le design hiérarchique. Grâce au design hiérarchique, on peut travailler à différents niveaux d'abstraction. D'abord on décrit des blocs de base à l'aide des portes logiques, pour ensuite utiliser ces blocs de base comme parties d'un système plus large. Dans le cas de notre additionneur 4 bits, l'utilisation de quatre additionneurs 1 bit sera nécessaire. Ce nouveau bloc pourra aussi être utilisé dans d'autres systèmes plus grands.

Pour créer un design hiérarchique, en incluant l'additionneur 1 bit qu'on a déjà conçu, il faudra suivre les étapes suivantes :

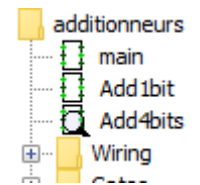
1. Allez dans *Project / Add circuit*

Entrez le nom **Add4bits** Remarquez que ce nom apparaît sous Addbit1 dans le menu de gauche.

Pour passer de l'édition d'un circuit à l'autre, il suffit de double-cliquer sur le nom de celui désiré dans le menu de gauche. La loupe indique quel circuit est actuellement visible dans le canevas.



2. Double cliquez sur l'icône Add4bits. Un canevas vide apparaît alors. C'est celui de Add4bits. Placez le curseur dessus. Il est alors possible d'ajouter un sous circuit add1bit de la même manière que l'utilisation d'un composant quelconque. On clique sur Add1bit dans le menu indiqué sur la Figure 6, puis on le place en cliquant sur le canevas.



3. Si le circuit Add1Bit a été créé correctement, alors il devrait être représenté par un petit bloc, avec sur sa gauche trois points bleus correspondant aux entrées A, B et RE et deux points rouges S et RS sur la droite correspondant aux sorties.

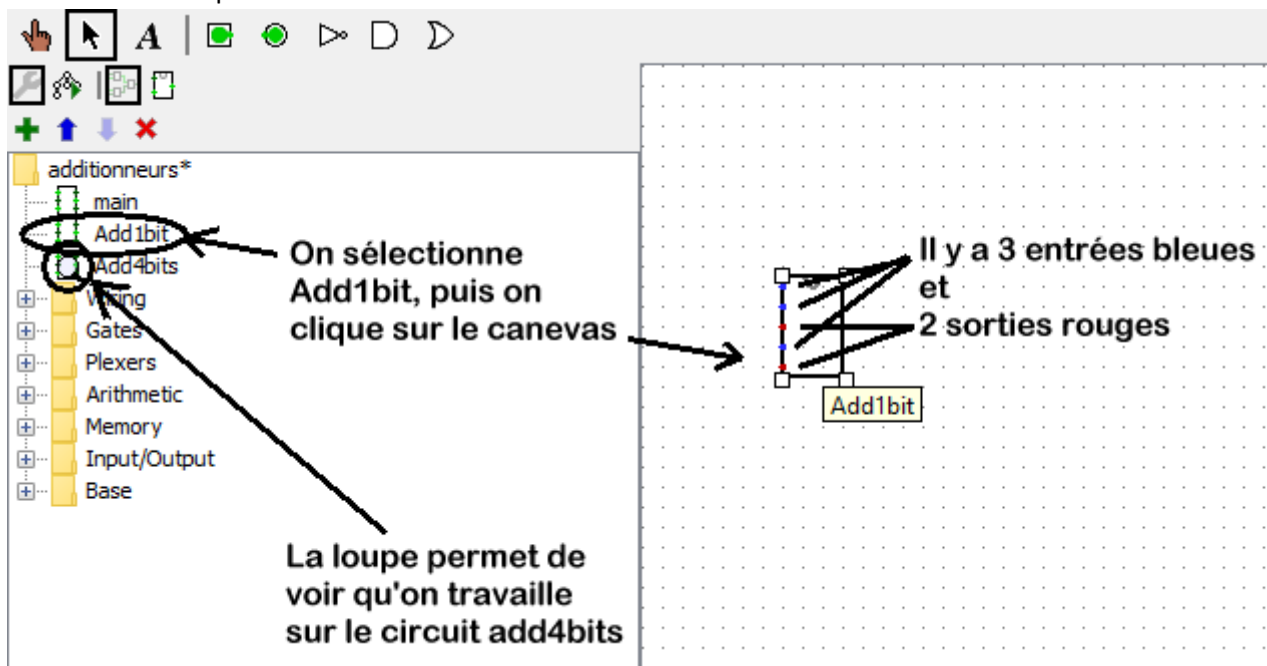


FIGURE 6. Sous circuit Add1bit orienté face à l'est (ses connecteurs sont à l'ouest)

4. Si les sorties apparaissent en bleu et non en rouge sur le schéma, vérifier que vous avez bien affecté l'attribut `output?=yes` dans les Pins de sorties.

5. Pour l'implémentation de l'additionneur 4 bits, il vous faut 4 additionneurs 1 bit, donc compléter le schéma en incluant les ports d'entrée et sortie. Les différences entre les circuits pour les additionneurs 1 et 4 bits, sont les entrées et les sorties et plus précisément leurs tailles, l'un avec des fils et l'autre avec des bus. Par exemple, pour définir l'entrée A comme un bus de 4 bits, il faut ajouter un **élément Pin** et définir sa taille via l'attribut **Data bits = 4**.
6. Lorsqu'on tire un fil de l'une de ces entrées, ce n'est plus un simple signal mais un bus de 4 bits. Pour pouvoir connecter les éléments de ce bus aux entrées des additionneurs 1 bits, on va devoir séparer les différents fils du bus afin de pouvoir les traiter un par un. L'élément *splitter* de la bibliothèque "wiring" permet d'effectuer ces conversions dans les deux sens : d'un bus de 4 bits vers 4 fils, et de 4 fils vers un bus de 4 bits voir Figure 7.

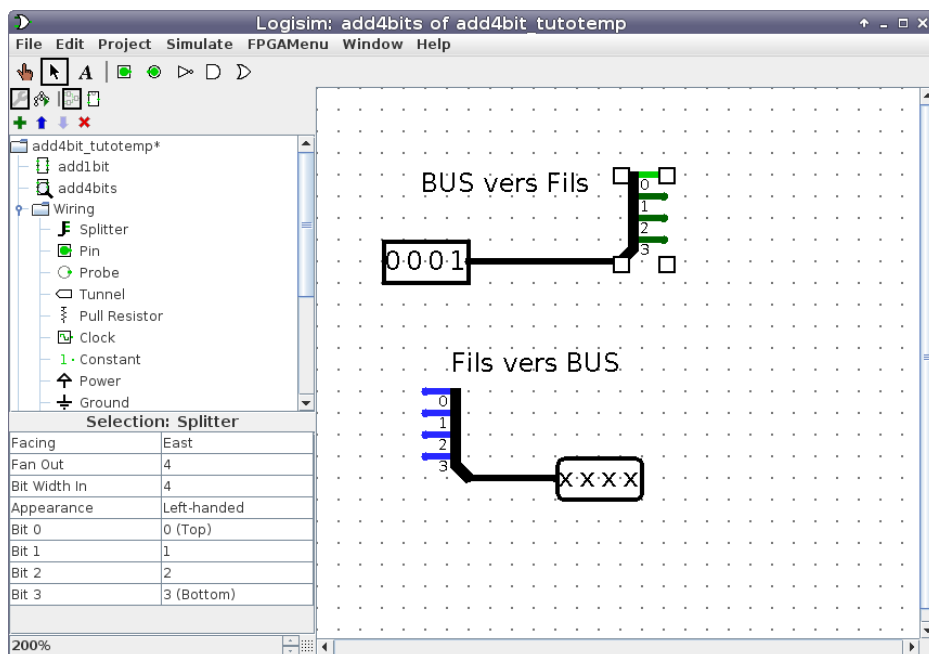


FIGURE 7. Exemples splitters

Il faut définir les tailles d'entrées et de sorties du *splitter* via les attributs Bit Width In (pour ici il faut 4 bits) et Fan Out (-sortie en éventail-) aussi avec 4 bits (normal parce qu'il y a 4 bits à l'entrée de ce bus). Vous pouvez aussi orienter vers l'Est le 'Facing'.

Note : Le bit de poids faible est indexé à 0 en sortie du *splitter*.

8 Additionneur 4 bits

Réalisez l'additionneur 4 bits de la Figure 8, puis vérifiez son bon fonctionnement en simulation.

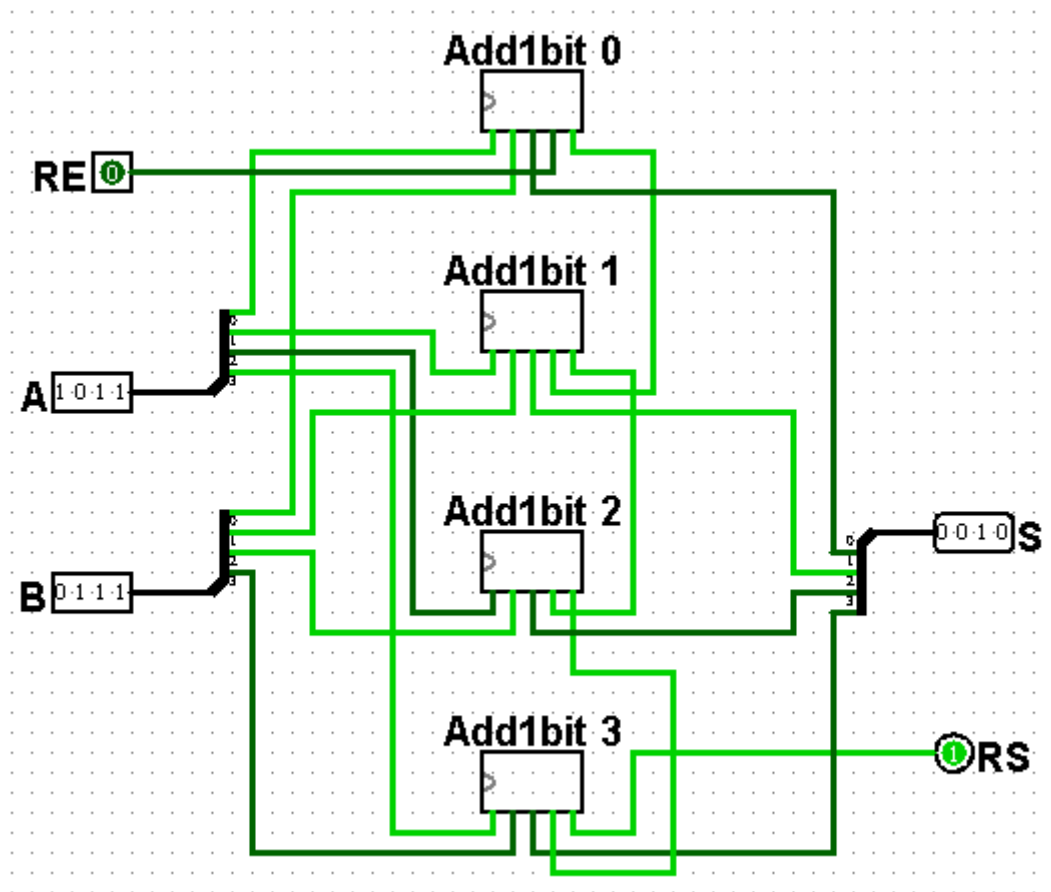


FIGURE 8. Additionneur 4 bits

9 Vérification

Visionnez la vidéo <https://www.youtube.com/watch?v=Nbm1XQEFBBQ&t=0s>

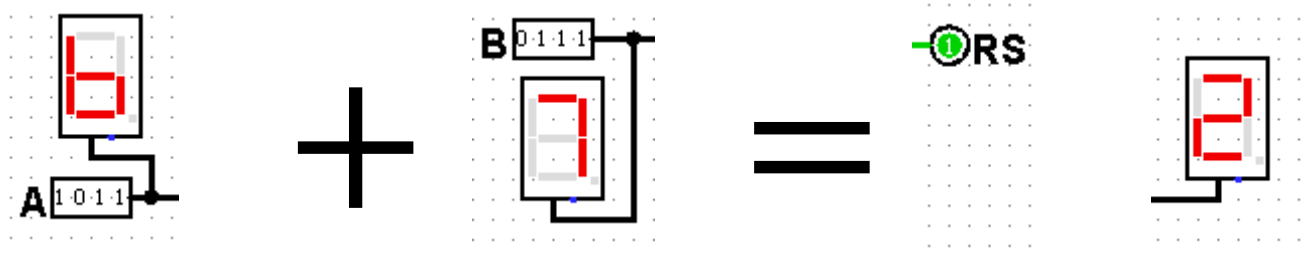
Réalisez alors sur votre additionneur 4 bits l'addition binaire qui est expliquée dans la vidéo.

10 Pour aller plus loin

Dans la bibliothèque Input / Output vous pouvez trouver l'élément "Hex Digit Display" qui prend en entrée (située en bas à gauche) un bus de 4 bits et qui affiche le chiffre hexadécimal correspondant.

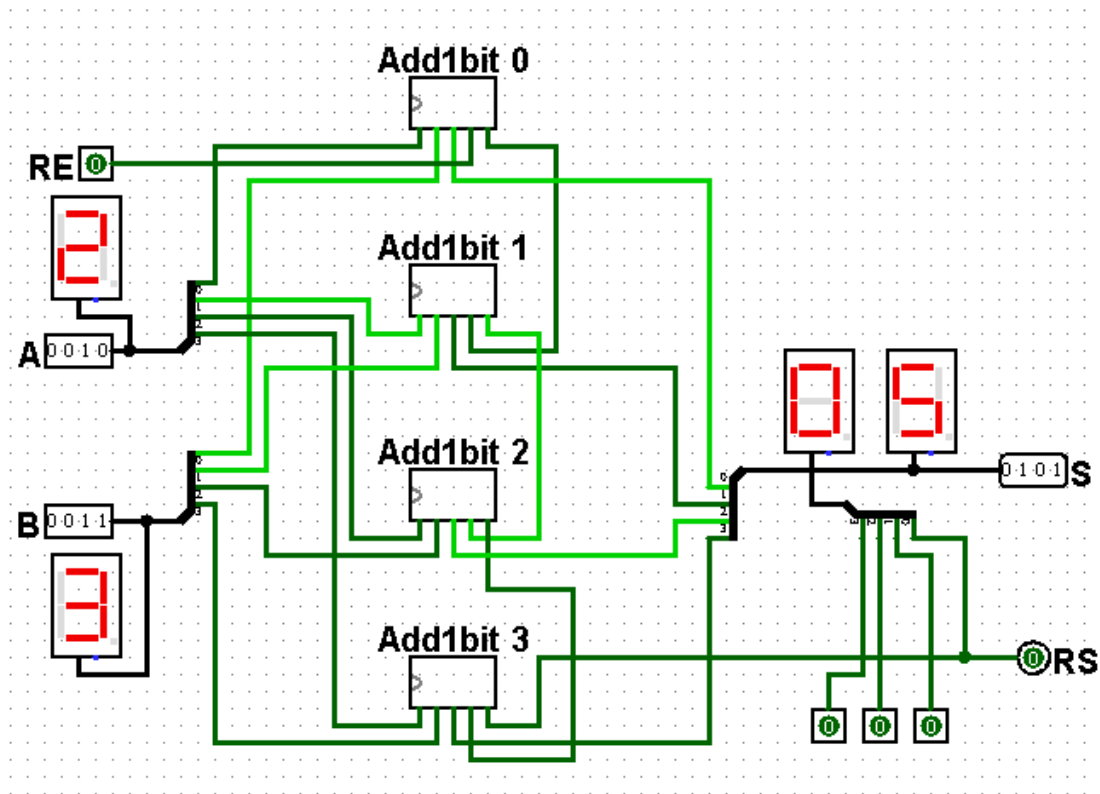
Ainsi, vous pouvez placez deux afficheurs hexadécimaux à gauche (un pour A et un pour B) et un à droite (pour S).

On obtient ainsi l'addition hexadécimale $b + 7 = 2$ avec sa retenue sortante de poids 16. C'est la même addition que celle qui est expliquée dans la vidéo du §9. Mais dans la vidéo, elle est expliquée en décimal : $11 + 7 = 18$



- On peut aussi ajouter en sortie un deuxième afficheur hexadécimal pour la retenue sortante. Il est plus convivial de voir un 0 ou un 1 sur un afficheur plutôt qu'un Pin avec 0 ou 1 écrit dessus.

Il faut seulement penser à compléter le bus de 4 bits qui alimente ce deuxième afficheur avec 3 bits de valeur 0 :



- Challenge :

Dans Projet / Add circuit, créez un circuit avec le nom Add8bits.

Y dessiner le schéma d'un additionneur de 2 mots A et B de 8 bits chacun.

Vous pourrez utiliser 6 afficheurs hexadécimaux pour visualiser de façon conviviale les octets A, B et S (somme).

Indication : Utilisez l'additionneur 4 bits comme "élément" pour faire l'addition des 4 bits de poids faible et des 4 bits de poids fort.

