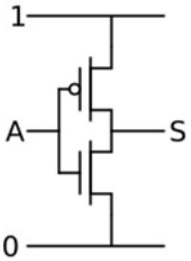
Architectures Matérielles et Systèmes d'Exploitation

Logisim simulateur logique

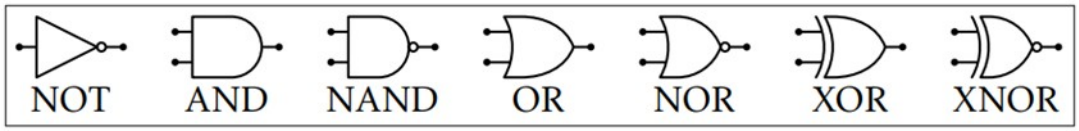
1 Rappel

Dans une activité précédente, on a vu qu'il est possible de fabriquer des composants à l'aide des transistors pour réaliser des opérations booléennes ou des opérations. On appelle ces circuits des **portes logiques**.

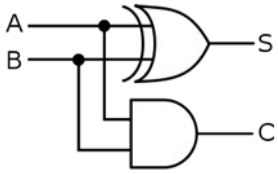


Porte logique NOT avec transistors nMOS et pMOS

Les principales portes logiques sont les suivantes :



Les portes NAND et NOR sont dites universelles, elles permettent chacune de fabriquer les autres portes logiques.



Demi-additionneur (1 bit) où A et B sont les entrées, S la somme A ⊕ B et C la retenue

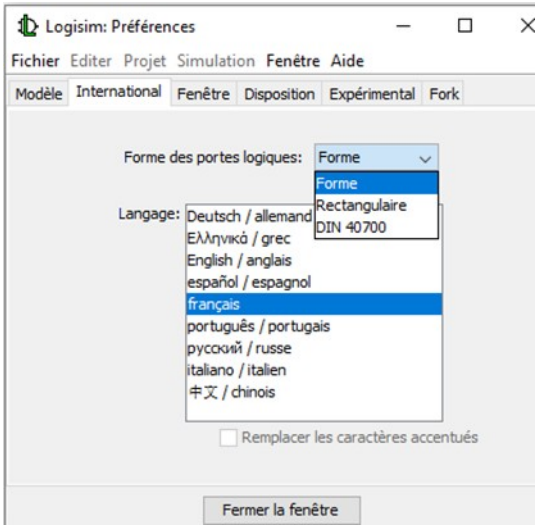
Lorsqu'on assemble des portes logiques entre elles on peut obtenir des circuits combinatoires : ce sont des circuits électroniques qui comportent plusieurs entrées booléennes (généralement notées A, B, C …) et une seule sortie booléenne (généralement notée S). La table de vérité que l'on obtient correspond au tableau de valeurs d'une fonction booléenne.

Pour clarifier les choses, le circuit combinatoire est le terme "concret / électronique" alors que le terme fonction booléenne est le terme "théorique / mathématique".

Présentation du logiciel Logisim[1]

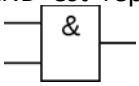
Logisim est un outil d'édition et de simulation de circuits mêlant portes logiques, boutons, voyants, afficheurs.... L'utilisation de ce type d'outil est très courante dans le processus de conception de circuits intégrés.

Lors de l'ouverture de Logiciel, il faut vérifier que le logiciel est en français et l'affichage des portes logiques est au format Forme avec le menu Fichier -> Préférences… -> International



Deux ensembles de symboles sont utilisés pour représenter les portes logiques :

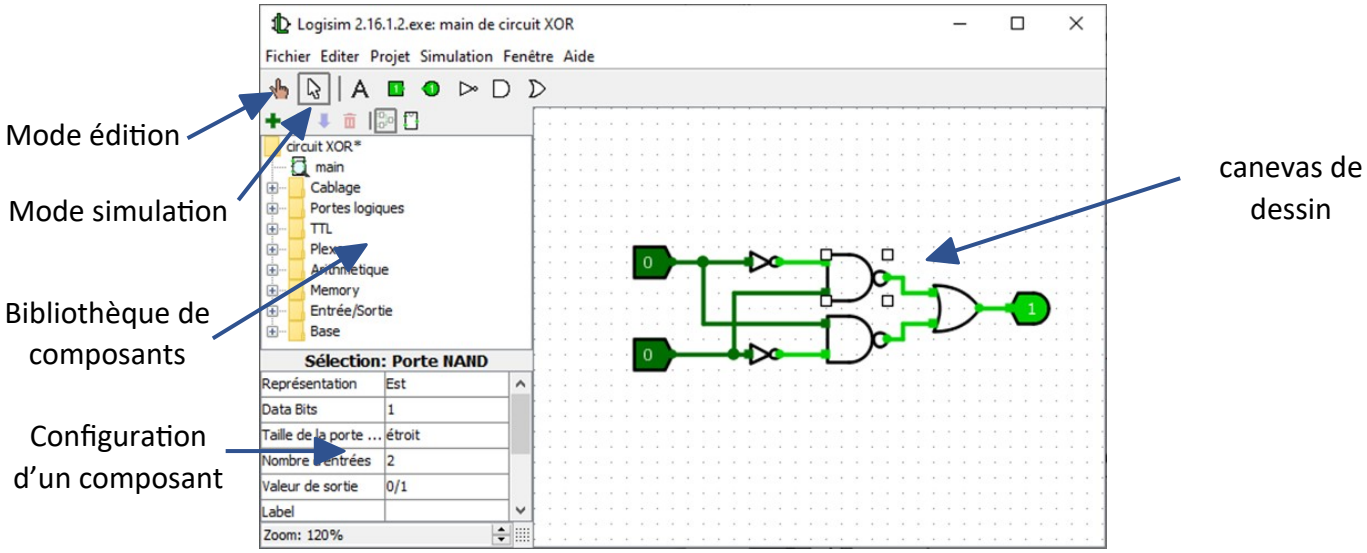
- En représentation classique, dite « américaine », les portes sont représentées par des formes distinctes.
- En représentation « rectangulaire », dite « européenne » les portes sont toutes représentées par des boites rectangulaires contenant le symbole de la porte.

Exemple : La porte logique AND est représentée par  en représentation classique ou par 

en rectangulaire.

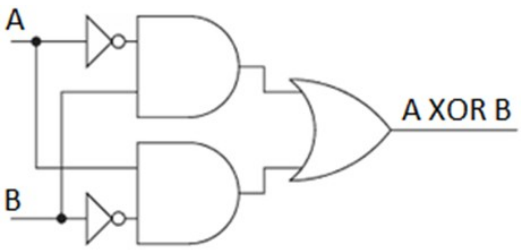
La fenêtre d'affichage se divise en plusieurs zones, à gauche une bibliothèque de composants et une zone de configuration de composants et à droite un canevas où dessiner les circuits. Au-dessus de ces parties se trouvent la barre de menus et la barre d'outils avec notamment les boutons pour basculer entre le mode édition et simulation.

1 Le logiciel est téléchargeable à cette adresse : http://logisim.altervista.org/



Un premier circuit simple

On veut vérifier que le circuit combinatoire suivant correspond bien à la porte logique XOR.

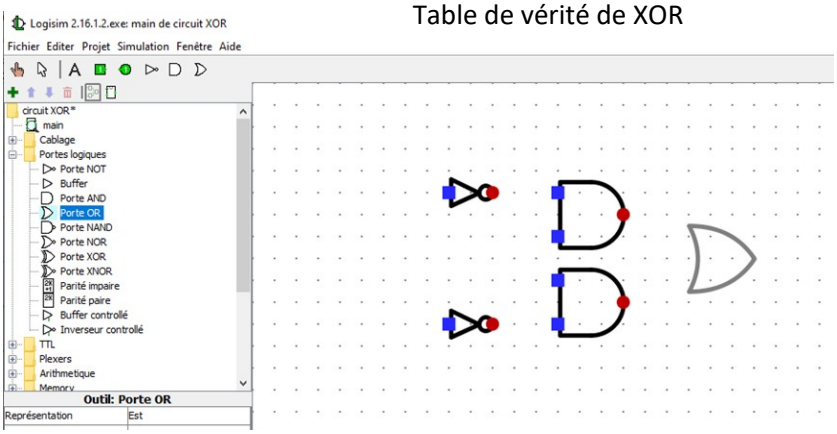


A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

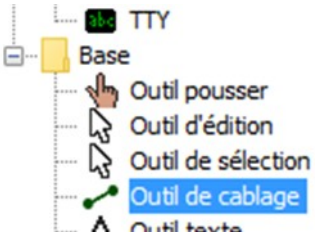
Table de vérité de XOR

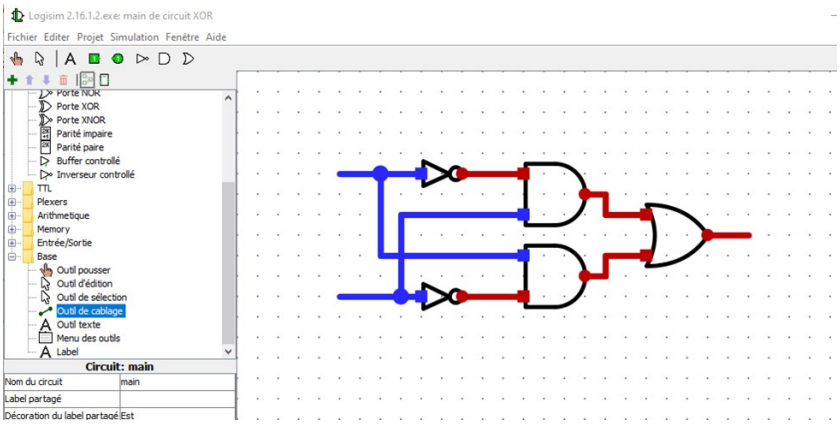
Pour réaliser le circuit dans Logisim, commencez par placer les cinq portes logiques du circuit (NOT, AND et OR) sur le canevas.

1. Ouvrez la liste du menu **Portes Logiques** de la bibliothèque pour sélectionner les portes à placer puis cliquez à l'emplacement voulu sur le canevas.

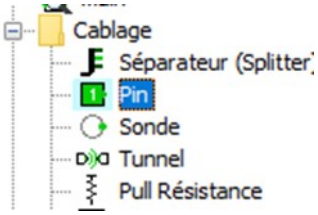


2. Puis connectez les portes logiques entre elles avec l'**Outil de cablage** dans le répertoire **Base** de la bibliothèque

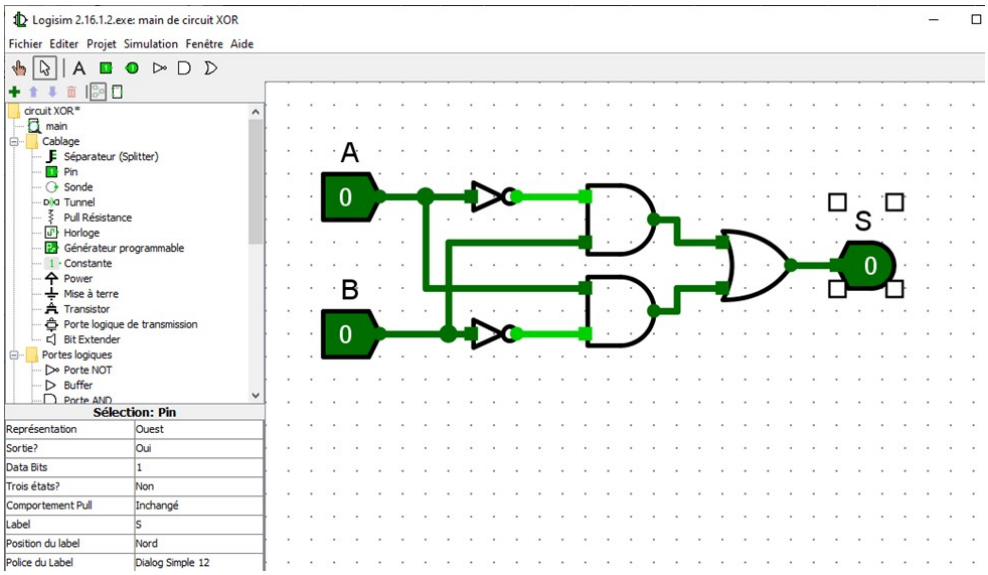




3. Ajoutez les entrées et la sortie avec **Pin** dans **Cablage**. Les configurer dans la fenêtre en bas à gauche :
   - Pour les entrées mettre **Trois états** ? à **Non**
   - Pour la sortie mettre **Représentation** à **Ouest**, **Sortie ?** à **Oui** et **Trois états** ? à **Non**



4. Affectez des noms A, B et S aux entrées et sorties par un "double clic" sur les éléments du schéma.



5. Le circuit est complet, enregistrez-le.

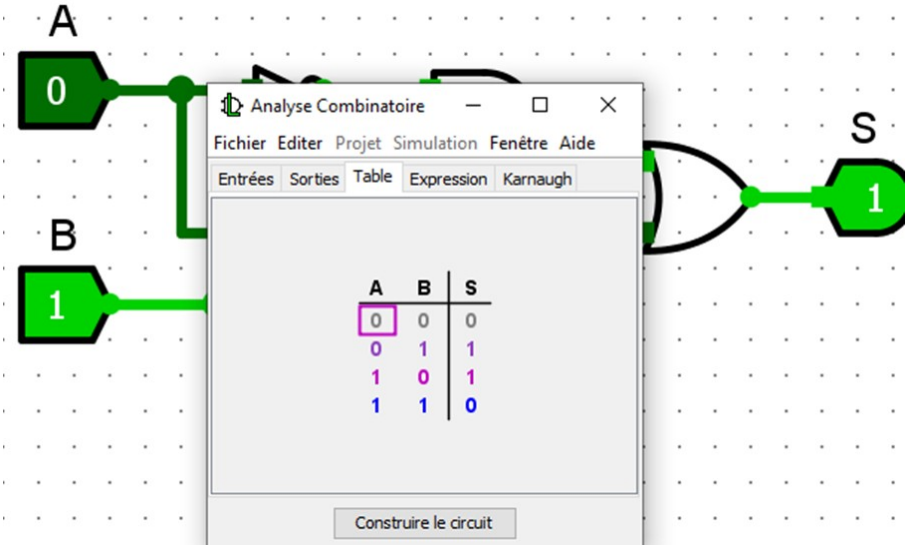
2 Mode simulation

Logisim est capable de simuler le circuit en affichant les valeurs des signaux directement sur le schéma. En mode simulation , un clic sur une entrée permet de changer la valeur de l'état logique à présenter.

En fonction des valeurs sur les entrées, les différents fils et sorties changeront de valeur.
· Gris : Le fil n'est relié à aucune entrée ou sortie.
· Bleu : Le fil a une valeur inconnue.
· Vert foncé : Le fil a valeur 0.
· Vert clair : Le fil a valeur 1.
· Noir : Le fil comporte plusieurs bits.
· Rouge : Le fil comporte une erreur.
· Orange : Les ports reliés au fils n'ont pas la bonne taille (par exemple un port a 8 bits et l'autre port a 2 bits).

6. Testez le bon fonctionnement du circuit XOR en cliquant sur les entrées pour les faire basculer entre 0 et 1.

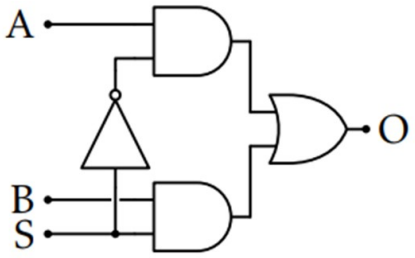
On peut aussi afficher directement la table de vérité avec le menu Projet/Analyser le circuit :



A noter : En plus d'analyser un circuit déjà saisi, Logisim permet également de créer un circuit à partir d'une expression logique ou d'une table de vérité.

Le « muxer »

On a vu dans une activité précédente un circuit combinatoire appelé « muxer », qui permet de sélectionner la valeur à envoyer en sortie, entre les deux entrées A et B, en fonction de la valeur de S.

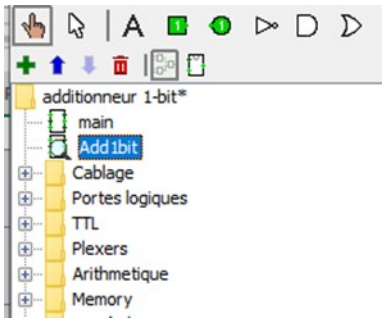


A	B	S	O
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	0	1

7. Réalisez ce circuit et complétez sa table de vérité.

Un-additionneur 4 bits

Lorsque l'on créé des circuits plus complexes, le nombre de portes et de connexions augmente rapidement. Le design hiérarchique permet de 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 comme parties d'un système plus large.



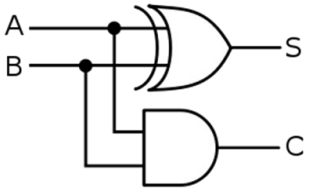
L'additionneur 4 bits utilise quatre blocs de base d'additionneur 1 bit. On commence donc par créer ce bloc de base.

8. Utilisez le menu **Projet/Ajouter Circuit…** pour créer le circuit Add1bit. Le circuit apparaît dans la bibliothèque d'éléments que l'on pourra plus tard utiliser pour construire de nouveaux circuits.

Demi-additionneur

Le demi-additionneur (ou semi-adonner) permet d'ajouter deux bits avec une retenue.

9. Construisez le demi-additionneur à 1 bit où A et B sont les entrées, S la somme A + B et C la retenue puis vérifiez sa table de de vérité.

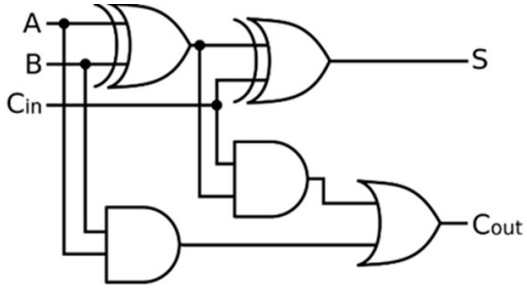


A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

Additionneur 1 bit

En combinant deux demi-additionneurs ensembles, on peut former un l'additionneur à 1 bit complet prenant une retenue entrée.

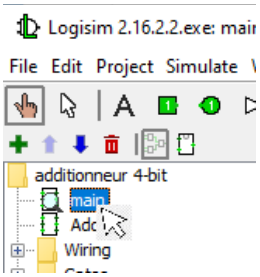
Complétez le circuit précédent pour créer un circuit additionneur 1 bit avec les retenues en entrée (Cin) et sortie (Cout) et complétez sa table de vérité.



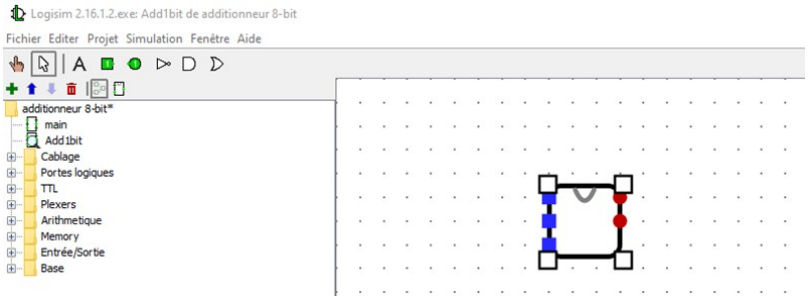
A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Additionneur 4 bit

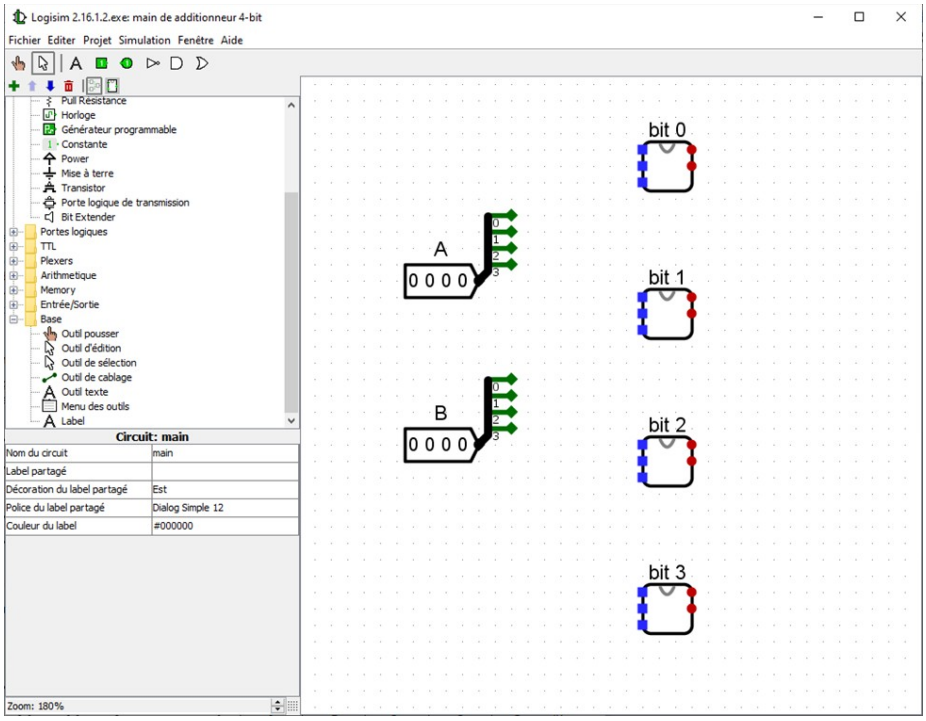
10. **Ouvrez maintenant le circuit « main »** et positionnez sur le canevas vide un circuit additionneur 1 bit.

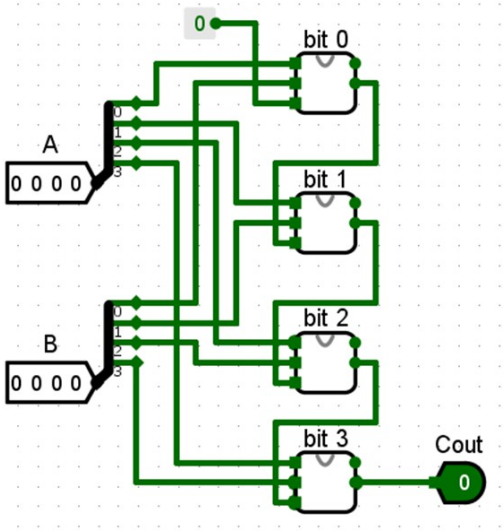


Si le circuit Add1Bit a été créé correctement, alors il devrait être représenté par un petit bloc,



avec sur sa gauche trois points correspondant aux entrées et deux points sur sa droite correspondant aux sorties (S et Cout).





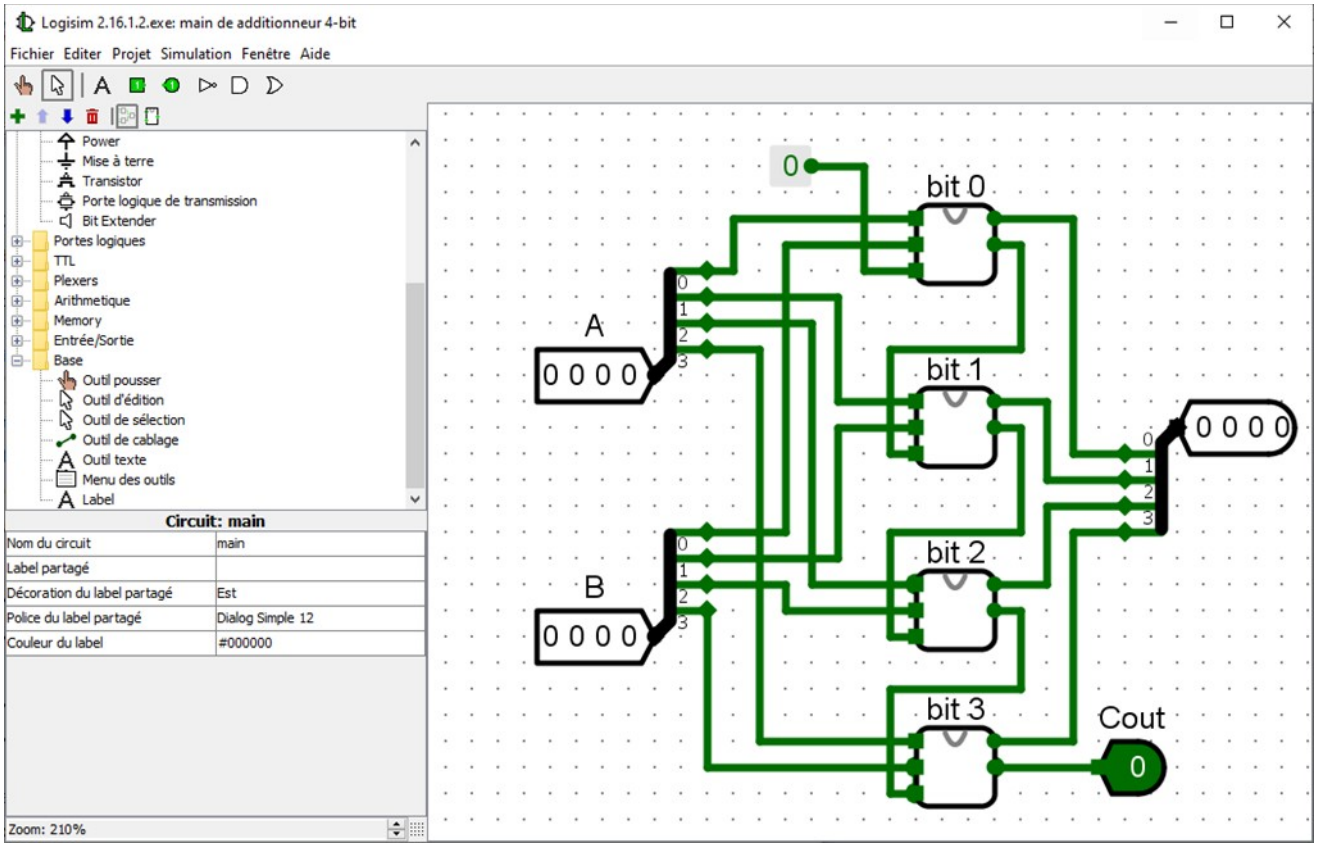
11. Positionnez maintenant 4 additionneurs 1-bit les uns sous les autres et ajoutez deux entrées A et B avec Data Bits à 4 et Trois états à Non et ajoutez pour chaque entrée un **Séparateur** (Splitter) de la bibliothèque **Cablage**.

12. Câblez les 4 bits de A à la première entrée des 4 Add1bits et les 4 bits de B à la deuxième entrée des 4 Add1bits.

    Attention de ne pas mélanger les fils !

13. Ajoutez une Constante à la valeur 0 en entrée du premier additionneur (Cin), et une retenue à 1 bit Cout en sortie du dernier additionneur, puis reliez toutes les retenues entre additionneurs entre-elles.

14. Pour finir, ajoutez une sortie sur 4 bits, avec un Séparateur et ajoutez les câbles sur les sorties des autres additionneurs.



15. Passez en mode simulation et changez les bits en entrée de A, B pour vérifier le résultat d'opérations binaires.
16. Le circuit est complet, enregistrez-le.

Soustracteur 4 bit

Pour calculer la différence A - B de deux nombres binaires A et B, on utilise un circuit qui calcule d'abord −B (l'opposé de B), puis effectue la somme de A avec −B grâce à un additionneur.

−B est calculé en prenant le complément à 2 de B, c'est-à-dire en prenant la négation de B bit à bit (0 devient 1 et 1 devient 0) puis en ajoutant 1 au résultat obtenu. Ce dernier 1 est en fait ajouté directement à la somme de A et -B en l'injectant comme retenue au premier additionneur 1 bit.

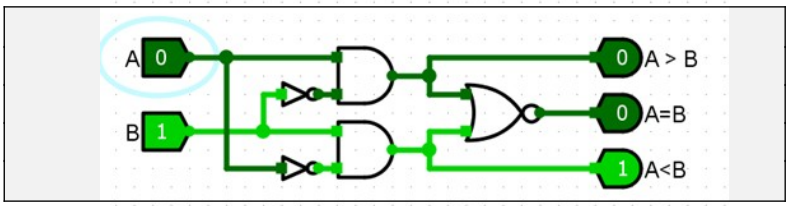
17. Modifiez le circuit précédent pour réaliser un soustracteur 4 bit. Comparez la valeur de Cout quand le résultat de la soustraction est positif ou négatif. Qu'en déduisez-vous ?

On observe que la retenue finale est un bit de signe : 1 quand le résultat est positif, 0 quand il est négatif.

Comparateur

Le comparateur est un circuit permettant de comparer deux nombres binaires A et B, ayant le même nombre de bits. On cherche à savoir si A > B, A<B ou A=B, le circuit répond donc à une question à trois choix.

A	B	A>B	A<B	A=B
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1



18. Entrez la table de vérité du comparateur 1-bit (menu Projet/Analyser le circuit) et construisez un circuit combinatoire.