IUT GRAND OUEST NORMANDIE

iNFO

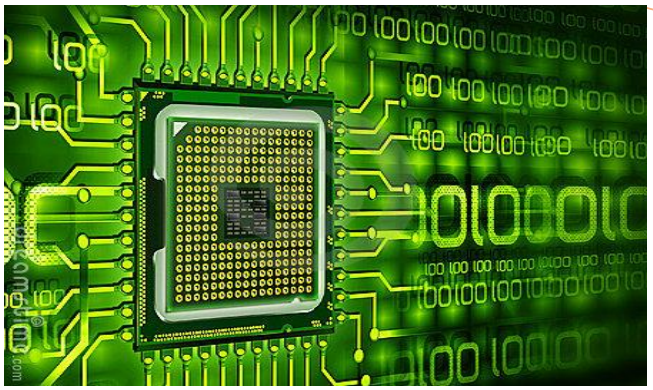
R 1.03

2023 - 2024

Introduction à l'architecture des ordinateurs

TP n° 3

Logique Binaire



ANNE Jean-François

D'après le TP 4TIN304U: Architecture des ordinateurs

Le but de ce TD est de se familiariser avec la logique binaire interne à nos ordinateurs.

A. Introduction :

L'outil que vous allez utiliser pour ce TP est le simulateur de circuits SimcirJS, dont une copie adaptée au cours est installée sur :

- http://dept-info.labri.fr/ENSEIGNEMENT/archi/circuits/blank.html.

Vous avez une vidéo de démonstration à cette adresse, si besoin :

- https://mediapod.u-bordeaux.fr/video/9500-utilisation-du-simulateur-de-circuits-simcirjs/

1°) Consignes d'utilisation :

- La boîte à outils des composants est située dans le bandeau de gauche, et le plan de travail est la fenêtre de droite. Pour placer un exemplaire d'un composant sur le plan de travail, effectuez une action de « glisser-déposer » : cliquez sur le composant et maintenez le bouton gauche de la souris enfoncé tant que vous déplacez le composant depuis la boîte à outils jusqu'à son emplacement de destination sur le plan de travail. Utilisez aussi le glisser-déposer pour déplacer les composants sur le plan de travail.
- Les entrées d'un composant sont en jaune, et ses sorties en blanc.
- Les connexions entre composants s'effectuent également par glisser-déposer, en cliquant sur une entrée jaune et en amenant le pointeur jusqu'à la sortie blanche destinée à l'alimenter.

Il est possible d'alimenter de multiples entrées avec la même sortie. En revanche, il est impossible de connecter une entrée à plusieurs sorties : cela créerait un court-circuit entre les sources de tension alimentant les différentes sorties. Une entrée ne peut donc être alimentée que par une unique sortie. Pour qu'une entrée dépende de plusieurs sorties, il faut combiner leurs valeurs au moyen de portes logiques.

- *Cliquez sur un connecteur d'entrée pour le déconnecter de la sortie qui l'alimentait.*
- *Déplacez un composant dans la boîte à outils si vous n'en avez plus besoin.*
- *Double-cliquez sur une étiquette pour éditer le nom d'un composant.*
- *Double-cliquez sur une bibliothèque pour accéder à son câblage interne.*
- *Effectuez un contrôle+clic pour changer de vue, entre la forme graphique des circuits et leur codage en JSON.*

2°) Exercice 1 : Premier pas avec le simulateur

Effectuez les câblages suivants sur le plan de travail.

- *Le composant DC n'a qu'une sortie, qui est toujours à 1 (tension de référence). Branchez une LED directement sur un DC, et constatez qu'elle reste allumée.*
- *Intercalez une porte AND entre DC et la LED. Constatez que les deux entrées de la porte doivent effectivement être branchées pour que la LED s'allume.*

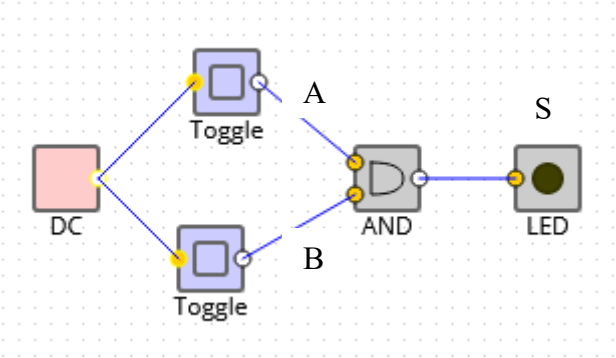
- *Ajoutez maintenant deux boutons Toggle, en les intercalant chacun entre DC et l'une des entrées de la porte AND. Étudiez le comportement de ce circuit selon que les boutons sont activés ou non.*
- *Le composant OSC est un oscillateur, qui fournit un signal alternativement à 0 et 1. Vous pouvez le tester en le reliant à une LED.*
- *Cliquez sur le bouton « Save » en bas à gauche de la page web. Cela modifiera l'URL de votre page, de sorte que le contenu de votre plan de travail sera codé dans cette URL. Vous pouvez ainsi, en sauvegardant cette URL, enregistrer votre travail et le recharger à tout moment.*

Afin de passer à la suite, vous pouvez maintenant nettoyer votre plan de travail. Pour cela, vous pouvez sélectionner l'ensemble des composants d'une zone par glisser-déposer avec le bouton de gauche de la souris, puis glisser-déposer l'un des composants sélectionnés pour tout remettre d'un coup dans la barre d'outils.

B. Découverte :

1°) Exercice 1 : Porte ET

Réaliser le circuit suivant :



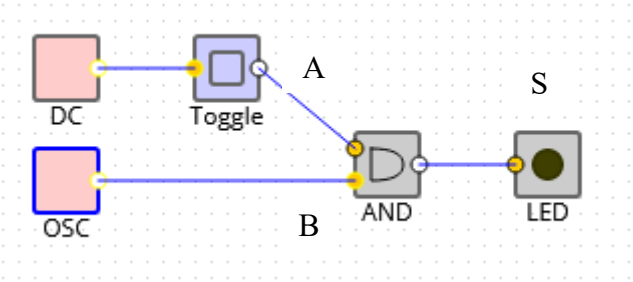
Remplir la table de vérité suivante :

A	B		S
0	0		
0	1		
1	0		
1	1		

Cette table correspond-t-elle à un circuit logique ET ?

2°) Exercice 2 : Interrupteur

Réaliser le circuit suivant :



Remplir la table de vérité suivante :

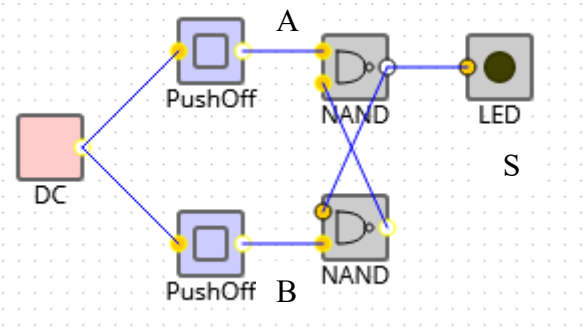
A	B		S
0	OSC		
1	OSC		

Le bouton A n'est pas actionné, quelle est la couleur de la LED ?
Le bouton A est actionné, quelle est la couleur de la LED ?

C. Circuits mémoire

1°) Exercice 1 : La bascule Set-Reset

Réaliser le circuit Set-Reset du TP

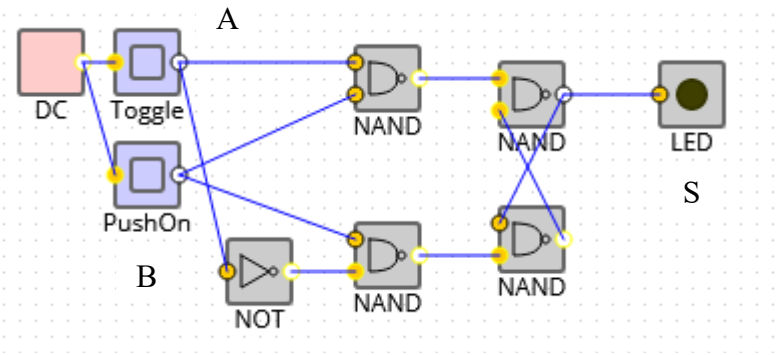


Appuyez sur le bouton A, quelle est l'état de la LED ?
Appuyez sur le bouton B, quelle est l'état de la LED ?
Appuyez sur le bouton A, quelle est l'état de la LED ?

Comment pourrait-on appeler ce circuit ?

2°) Exercice 2 : La bascule D

Réaliser le circuit suivant

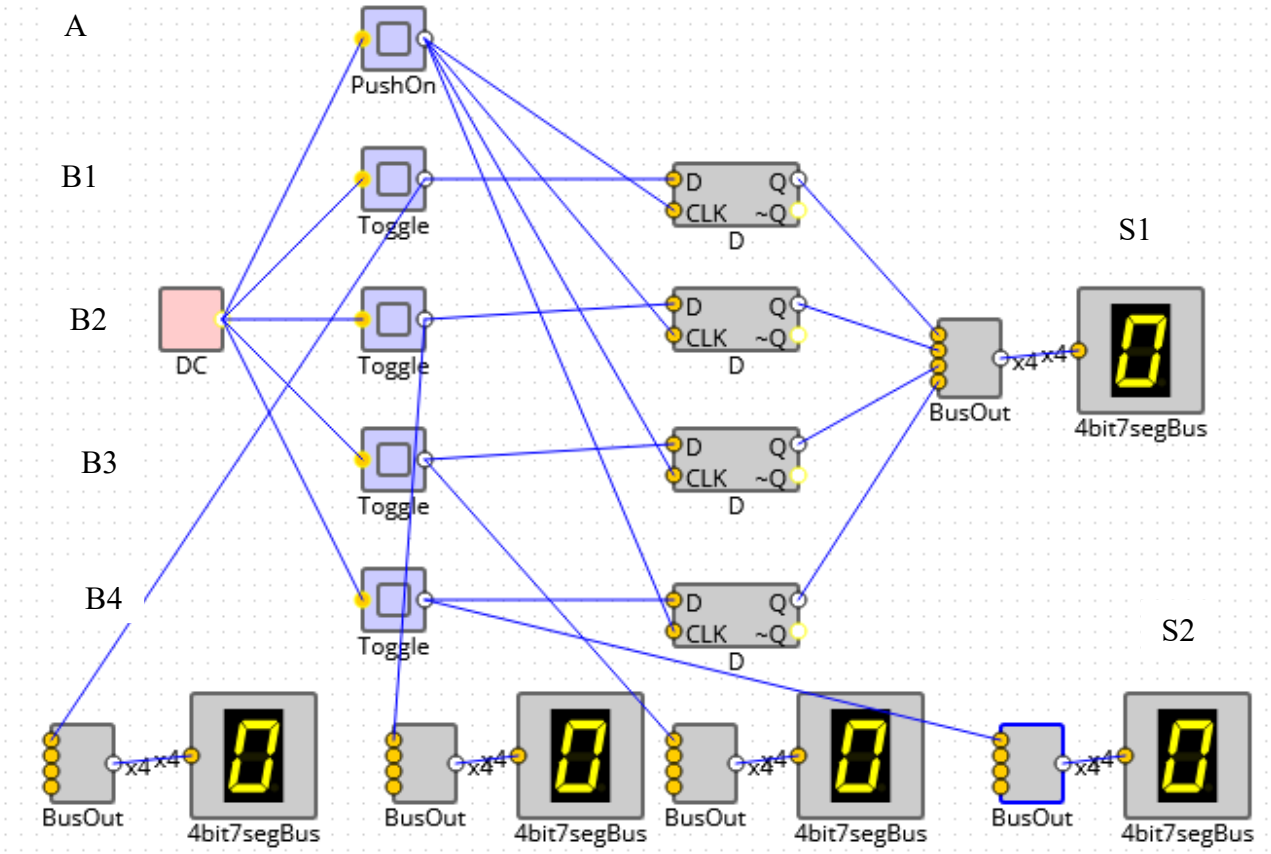


Appuyez sur le bouton A, quelle est l'état de la LED ?
Appuyez sur le bouton B, quelle est l'état de la LED ?
Appuyez sur le bouton A, quelle est l'état de la LED ?
Appuyez sur le bouton B, quelle est l'état de la LED ?

Comment pourrait-on appeler ce circuit ?

3°) Exercice 3 : Affichage

Réalisez le circuit suivant



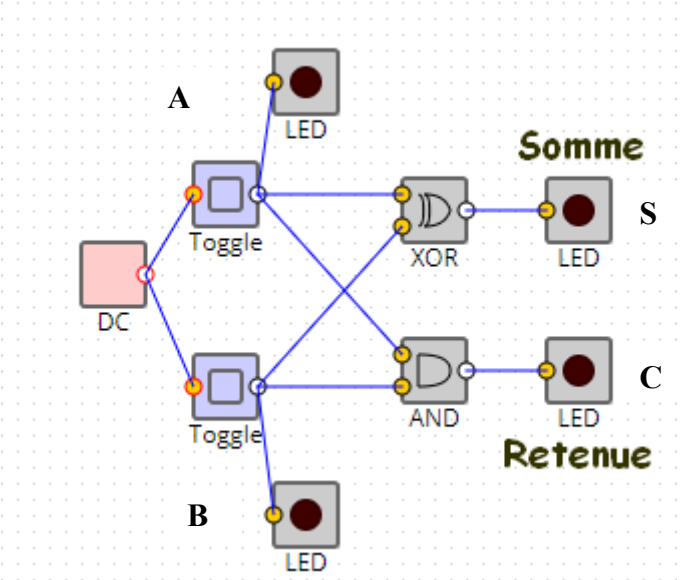
Sélectionnez un nombre binaire avec les boutons Bx, puis actionnez le bouton A.

Que représenta l'affichage S1 par rapport aux affichages S2 ?

D. Circuits Additionneur

1°) Exercice 1 : demi Additionneur

Réalisez le circuit ci-dessous d'un demi-additionneur à 1 bit, à l'aide de portes logiques AND, OR, XOR, etc. Ce circuit prend deux entrées A et B, et produit en sortie S, le résultat de A+B, sur 1 bit, ainsi qu'une retenue Cout sur 1 bit. Vérifiez à l'aide de boutons Toggle et de LED que votre circuit fonctionne correctement. Pour clarifier votre circuit, pensez à renommer les Toggles et les LED représentant vos entrées/sorties, et placez les LED de poids fort (retenue) et de poids faible (somme) d'une façon lisible.



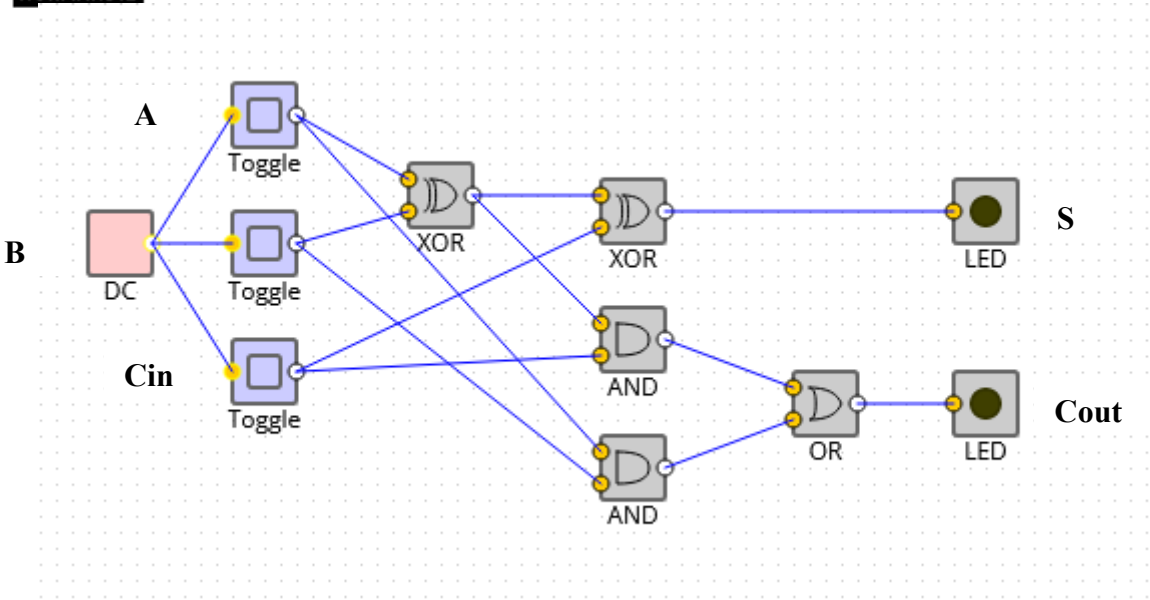
Remplir la table de vérité suivante :

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

Pourquoi l'appelle-ton demi additionneur ?

2°) Exercice 2 : Additionneur complet :

Réalisez maintenant un additionneur complet à 1 bit, toujours à l'aide de portes logiques. Ce circuit est comparable à un demi-additionneur, auquel on ajoute une entrée supplémentaire Cin représentant une retenue entrante. Vous pouvez construire cet additionneur complet en combinant deux demi-additionneurs.
Ce circuit est disponible dans la bibliothèque des composants sous le nom FullAdder. Ouvrez un nouvel onglet sur SimcirJS, et double-cliquez sur ce composant pour consulter son circuit, qui doit ressembler au vôtre.



Construire la table de vérité et vérifier que l'on a bien un additionneur complet

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

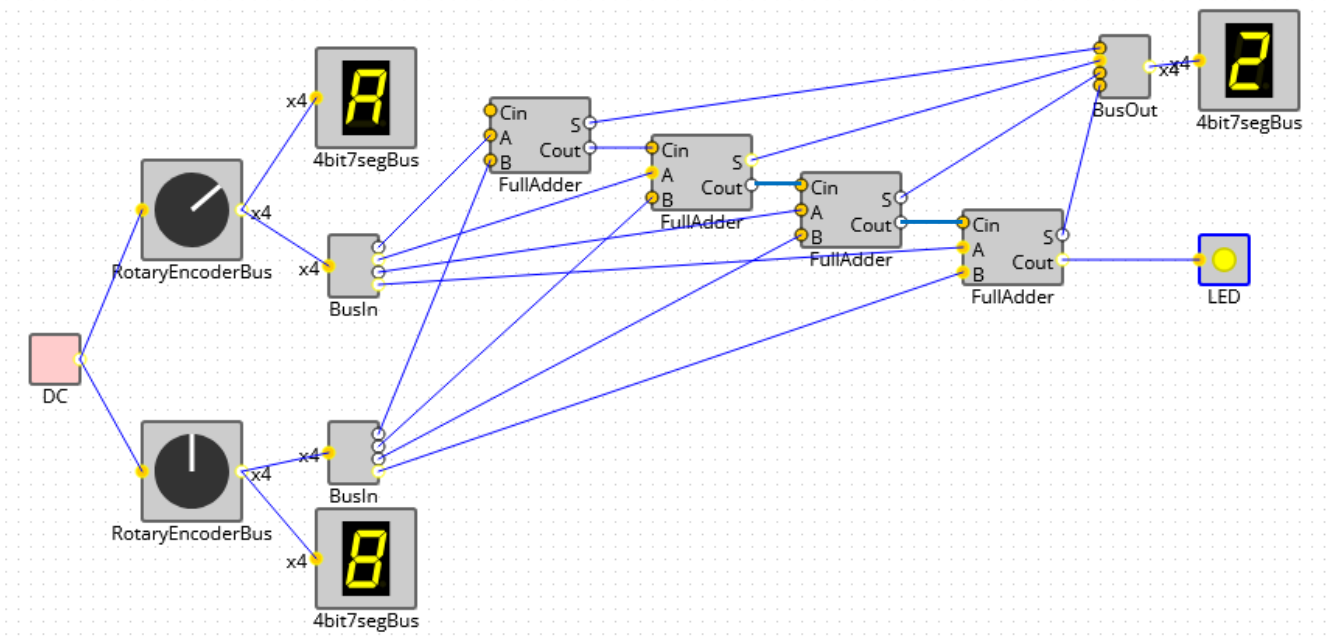
3°) Exercice 3 : Additionneur 4 bits :

En combinant plusieurs composants FullAdder, construisez maintenant un additionneur complet à 4 bits. En pratique, il faut brancher en cascade 4 additionneurs complets à 1 bit, afin de propager la retenue des poids faibles vers les poids forts.
Pour tester votre circuit, utilisez deux boutons RotaryEncoderBus en entrée et un afficheur à 7 segments 4bit7segBus en sortie. Ces composants sont fournis dans la bibliothèque.

N.B. :

1. Afin de réduire les câblages, ces composants sont interconnectés par des « bus », c'est-à-dire des ensembles parallèles de fils, ici sur 4 bits (d'où le « x4 »). Pour passer d'un bus « x4 » à 4 fils simples, et réciproquement, vous disposez des composants BusIn et BusOut permettant de considérer séparément chacun des fils du bus.

2. L'entrée des RotaryEncoderBus doit être connectée à un DC pour qu'ils produisent une valeur en sortie.



4°) ***Exercice 4 : Additionneur - Soustracteur :***

a) ***Multiplexeur :***

Observez, en double-cliquant dessus, le circuit du composant Mux2 de la bibliothèque, qui réalise le circuit logique d'un multiplexeur avec une entrée de sélection à 1 bit.
Le composant MuxBus2 étend ce circuit en prenant en entrée/sortie des bus de 4 fils (« x4 »), l'entrée de sélection restant sur un seul bit.
Testez le fonctionnement du composant MuxBus2 en lui fournissant en entrée deux nombres sur 4 bits issus de RotaryEncoderBus, en sélectionnant celui qui sera affiché au moyen d'un Toggle.

b) ***Additionneur - Soustracteur :***

En vous appuyant sur les propriétés de la notation en complément à deux (ici sur 4 bits), créez un circuit permettant, selon l'entrée donnée par un Toggle, d'additionner (« A+B ») ou bien de soustraire (« A + (-B) ») deux nombres sur 4 bits. Pour cela, utilisez un multiplexeur MuxBus2 pour choisir entre la valeur de B et son complément (calculé par un NotBus), et faites en sorte que la retenue d'entrée Cin du circuit AddBus que vous utilisez soit également pilotée par le Toggle. Sinon, vous pouvez utiliser un NegBus.

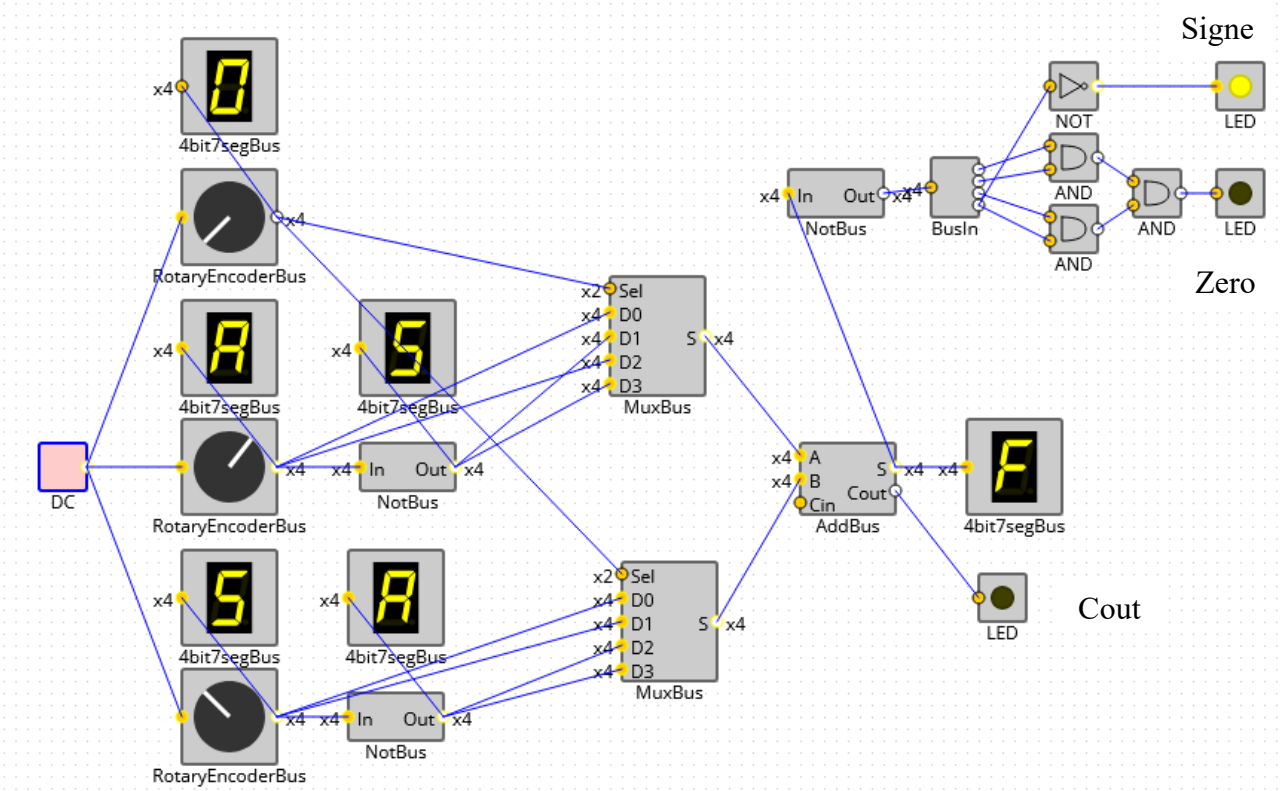
Note :

Un multiplexeur est un circuit à n fils de contrôle d'entrée, 2n fils de données d'entrée, et une unique sortie, et câblé de telle sorte que la valeur de la sortie soit égale à la valeur de l'entrée de numéro i, où i est la valeur binaire codée par les n fils de contrôle. Les fils de contrôle permettent donc de choisir laquelle des 2n entrées sera routée vers la sortie.

E. Circuits UAL

1°) Exercice 1 : UAL à quatre fonctions

En étendant le circuit précédent au moyen d'autres multiplexeurs et fonctions logiques, construisez une unité arithmétique et logique qui, selon la valeur à deux bits fournis par deux Toggle, calcule soit l'addition A + B, soit la soustraction A - B, soit AND (A ; B), soit NOT(B).
Ajoutez en sortie de cette UAL les bits S (signe) et Z (zéro), qui s'appliquent à la valeur calculée en sortie.



F. Circuits CPU

1°) Exercice 1 : CPU simple

- https://dept-info.labri.fr/ENSEIGNEMENT/archi/circuits/mycpu.html

- Analysez le programme,
- Faites la simulation pour vérifier son fonctionnement,
- Double cliquez sur la ROM pour voir à l'intérieur du circuit,
- Double cliquez sur le CPU pour voir à l'intérieur du circuit.

G. Webographie :

- https://kazuhikoarase.github.io/simcirjs/
- http://dept-info.labri.fr/ENSEIGNEMENT/archi/circuits/blank.html
- https://dept-info.labri.fr/ENSEIGNEMENT/archi/td-tm/tp_06.pdf
- https://sites.google.com/view/nsi-toulouse-centre/accueil/premiere-nsi/architecture-mat%C3%A9rielle/exercices-simcirjs#h.p_1K1SqZfYqyQ1
- https://sites.google.com/view/nsi-toulouse-centre/accueil/premiere-nsi/architecture-mat%C3%A9rielle/circuits-additionneurs
- https://isn.lycee-ozenne.fr/simcirjs-master/additionneurs.html
- https://www.isnbreizh.fr/nsi/activity/logicGate/index.html
- https://dept-info.labri.fr/ENSEIGNEMENT/archi/circuits/mycpu.html
-