

ARCHITECTURE DES ORDINATEURS

TP : 06

CONSTRUCTION D'UNE UNITÉ ARITHMÉTIQUE ET LOGIQUE (UAL)

L'outil que vous allez utiliser 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>
(ce lien est disponible depuis le site du cours).

Exercice 1 : Additionneur complet sur 4 bits

Avant de construire un circuit logique, pensez à écrire sur papier la table de vérité et la fonction logique recherchée.

Question 1

Réalisez le circuit 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 C_{out} 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 `Toggle` et les LED représentant vos entrées/sorties, et placez les LED de poids fort (retenue) et de poids faible (somme) d'une façon lisible.

Question 2

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 C_{in} 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`. Double-cliquez sur ce composant pour consulter son circuit, qui doit ressembler au vôtre.

Question 3

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.

Exercice 2 : Additionneur/soustracteur

Question 1

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`.

Question 2

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 inverse (calculée par un `NotBus`), et faites en sorte que la retenue d'entrée C_{in} du circuit `AddBus` que vous utiliserez soit également pilotée par le `Toggle`.

Exercice 3 : UAL à quatre fonctions

Question 1

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 fournie par deux `Toggle`, calcule soit l'addition $A + B$, soit la soustraction $A - B$, soit $AND(A, B)$, soit $NOT(B)$.

Question 2

Ajoutez en sortie de cette UAL les bits S (signe) et Z (zéro), qui s'appliquent à la valeur calculée en sortie.