

ARCHITECTURE DES ORDINATEURS

TP : 07

ÉTUDE DE CIRCUITS COMBINATOIRES ET SÉQUENTIELS

Nous allons utiliser le simulateur de circuits SimcirJS, dont une copie est installée sur <http://dept-info.labri.fr/ENSEIGNEMENT/archi/circuits/blank.html> (lien disponible depuis le site du cours).

Consignes d'utilisation :

- Prendre un composant dans la boîte à outils et le déplacer à droite.
- Les connecter en glissant entre les connecteur. Les entrées sont en jaune, les sorties sont en blanc. On peut connecter une sortie blanche sur de multiples entrées jaunes, mais on ne peut connecter qu'une sortie blanche à une entrée jaune.
- Cliquer sur un connecteur d'entrée pour le déconnecter.
- Déplacer un composant dans la boîte à outils si vous n'en avez plus besoin.
- Double-Cliquer sur une étiquette pour editer le nom d'un composant.
- Double-Cliquer sur une bibliothèque pour ouvrir le circuit à l'intérieur.
- Ctrl+Clic pour changer de vue (Circuit en live, ou données JSON).

Exercice 1 : Premier pas : portes logiques

Pour se familiariser avec l'interface et les outils de base, utilisons quelques portes logiques.

DC n'a qu'une sortie, qui est toujours à 1. Branchez une LED directement sur un DC, 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.

Intercalez des boutons Toggle entre DC et la porte AND. Constatez le comportement.

Exercice 2 : Un additionneur complet sur 4 bits

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

- Commençons par réaliser le circuit d'un *demi-additionneur* (1 bit) à l'aide de portes logiques (AND, OR, XOR, ...). 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 son circuit, pensez à renommer les Toggle et les LED représentant vos entrées/sorties.
- Réalisez maintenant un *additionneur complet* (1 bit) à l'aide de portes logiques (AND, OR, XOR, ...). Ce circuit est comparable à un demi-additionneur, auquel on ajoute une entrée supplémentaire *Cin*, représentant une retenue "entrante". (Indice : vous pouvez construire votre additionneur complet en combinant de deux demi-additionneurs.)
- Ce circuit est disponible dans la bibliothèque des composants sous le nom *FullAdder*. Double-cliquez sur ce composant pour regarder son circuit, qui doit ressembler au votre... En combinant plusieurs composants *FullAdder*, construisez maintenant un *additionneur complet à 4 bits*. En pratique, il faut brancher 4 additionneur complet (1 bit) en cascade en faisant suivre la retenue. Pour tester votre circuit, utiliser deux boutons *RotaryEncoderBus* en entrées et deux afficheurs

7 segments en sortie. Ces composants sont fournies dans la bibliothèque, mais attention ils manipulent des "fils 4 bits" (x4). Pour passer d'un fil x4 à 4 fils simples, et réciproquement, vous disposez des composants BusIn et BusOut. A vous de jouer.

Exercice 3 : Mise en oeuvre d'un 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 paquets de 4 fils (x4), l'entrée de sélection restant sur 1 seul bit. Illustrez le fonctionnement du composant *MuxBus2* pour piloter à l'aide d'un bouton Toggle (branché sur l'entrée de sélection *Sel*) l'addition (*AddBus*) ou la soustraction (*SubBus*) de deux entiers A et B (sur 4 bits) dont les valeurs sont définies par des boutons *RotaryEncoderBus*. La sortie du multiplexeur doit envoyer le résultat du calcul sur un afficheur 7 segments.

Exercice 4 : Bascule D

À la fin de la liste se trouve la bascule D. Branchez son entrée CLK sur un OSC. Branchez sa sortie \overline{Q} sur son entrée D. On constate qu'elle se met à clignoter. Pour que le dessin soit plus lisible, vous pouvez utiliser des petites épingles situées juste en-dessous de la porte XNOR, pour faire passer le fil à côté de la bascule D. Attention à ne pas confondre avec les épingles x4 plus bas, qui serviront plus tard. Attention au sens : on branche une sortie blanche sur une entrée jaune.

Remplacez la bascule D par une bascule D-E, qui possède en plus une entrée "enable". Branchez à l'entrée E un DC et un Toggle, on peut ainsi choisir quand on veut que le circuit clignote.

Double-cliquez sur la bascule D-E, pour voir comment elle est construite : on utilise un simple multiplexeur ! (Appelé Mux2 car il a 2 entrées.)

Exercice 5 : Deuxième pas : bus

Prenez un afficheur LED (appelé 4bit7segBus). Notez qu'il prend en entrée un bus à 4 fils. On ne peut donc pas le brancher directement sur DC, il faut intercaler un BusOut qui permet de "rassembler" 4 fils en un bus. En choisissant les fils que l'on branche à DC, on peut ainsi encoder un nombre sur 4bits. Remplacez BusOut par un RotaryEncoderBus, constatez que c'est plus agréable à utiliser. :)

Exercice 6 : Compteur

RegBus est un registre 4bit, simplement composé de 4 bascules D (double-cliquez dessus et sur Reg4 pour le constater). Faites une boucle composée d'un RegBus et un IncBus (qui incrémente juste la valeur codée en binaire), et branchez un OSC sur l'entrée CLK. Attention, si vous utilisez des épingles, à utiliser celles qui comportent 4 fils (notées x4). Branchez deux afficheurs 7 segments pour voir les valeurs en sortie du registre et de l'incrémenteur.

Intercalez un multiplexeur MuxBus2, pour choisir avec un DC+Toggle si l'on veut incrémenter ou remettre la même valeur dans le registre (sans toucher à l'horloge!).

En pratique, on préfère utiliser une entrée *Enable* : remplacez RegBus par RegBus-E, et branchez un DC et un Toggle sur l'entrée Enable.

Exercice 7 : Trois compteurs

Nous voudrions manipuler trois compteurs indépendants, l'idée étant qu'un seul est actif à la fois, et l'on utilise donc un seul IncBus.

Ajoutez deux autres RegBus-E à côté du premier (et branchez le même OSC dessus), branchez un afficheur 7 segments sur chaque registre, pour que l'on puisse facilement observer leurs états.

Ajoutez un DC+RotaryEncoderBus, qui va nous servir à choisir quel compteur est actif.

Il faut choisir quel RegBus-E va donner sa valeur au IncBus, utilisez un multiplexeur MuxBus pour cela.

Il faut maintenant choisir quel RegBus-E va enregistrer la nouvelle valeur, utilisez un 2to4DecoderBus pour cela.

Que se passe-t-il lorsque l'on sélectionne le choix que l'on n'a pas connecté ?

Exercice 8 : Compteur/décompteur

On veut maintenant choisir si l'on avance ou l'on recule.

Ajoutez un DecBus à côté du IncBus. Utilisez un MuxBus2 pour pouvoir choisir entre les deux en cliquant sur un Toggle.

On a ainsi un processeur très très basique. :)

Exercice 9 : Processeur complet

Observez le processeur complet (cliquez sur les circuits pour regarder à l'intérieur).

<http://dept-info.labri.fr/ENSEIGNEMENT/archi/circuits/mycpu.html>

Pour aller plus loin...

Exercice 10 : Compteur 2 chiffres

Revenons au compteur à 1 chiffre.

IncBus possède une sortie C (pour *carry*) qui est à 1 lorsque le calcul déborde et donc retombe à zéro.

Refaites une autre boucle RegBus-E+IncBus à côté de la précédente (et branchez le même OSC dessus), et branchez la sortie C du premier IncBus à l'entrée E du deuxième RegBus-E. Mettez les deux afficheurs 7 segments côte à côte.

Modifiez votre circuit pour ajouter un Toggle permettant de remettre à zéro votre compteur au prochain top d'horloge.

Exercice 11 : Compteur BCD

La plupart des gens de la vraie vie ne connaissent pas l'hexadécimal. :) Nous allons donc plutôt compter en BCD (binaire codé décimal) : chaque chiffre ne va que de 0 à 9, même s'il est codé sur 4 bits.

Au lieu d'utiliser la retenue de IncBus, on utilise EqualBus pour faire retomber le compteur à zéro lorsque l'on atteint 10.

Ajoutez un deuxième chiffre à votre compteur BCD, pour obtenir une sortie décimale de "00" à "99".