

TP 3

LOGISIM- Circuits séquentiels, registres, mémoire

3.1 Bascules

EXERCICE 1 ► Bascules

Le but est de bien comprendre le principe des **bascules D (flip-flop)**.

- Montez une **bascule D (flip-flop)** de la librairie (en mode front montant *Rising Edge* puis testez son comportement. Montez ensuite deux bascules D **en série** en les reliant au même signal d'horloge ; vérifiez le comportement sur deux cycles d'horloge successifs.
- Construisez un **chenillard à 5 leds** en utilisant 5 bascules D montées en série. Le principe est qu'à chaque cycle d'horloge, une seule led est allumée, et la led allumée est décalée d'un cran vers la droite à chaque cycle d'horloge. Le cycle suivant l'allumage de la led 5, c'est la led 1 qui doit de nouveau être allumée. Testez votre circuit.

3.2 Registres, compteurs et mémoire

EXERCICE 2 ► Registre

Un registre n -bits est une mémoire constituée d'un assemblage en parallèle de bascules D. Construisez un **registre 4 bits**, que vous testerez, puis que vous comparerez avec celui de la bibliothèque (toujours sur front montant).

EXERCICE 3 ► Compteur

Réalisez un **compteur 4 bits** en utilisant les outils suivants de la bibliothèque : un registre 4 bits et un additionneur. Testez en mettant une horloge, avec l'option `Simulate->Ticks Enabled`, puis ajoutez un signal reset qui permet de remettre le compteur à 0.

EXERCICE 4 ► Utilisation de la RAM

En vous aidant de la documentation :

- Instanciez une **RAM** avec adressage 4 bits et contenu 8 bits (en mode `One synchronous load/store port`, comme dans le cours).
- Faites la fonctionner en lecture. Pour remplir la mémoire avant de tester la lecture, on pourra faire un clic droit sur le composant de mémoire, puis `Edit Contents`. Il n'est pas nécessaire de sauvegarder le contenu de la mémoire dans un fichier.
- Dans un autre onglet, faire fonctionner une mémoire en écriture.
- Comment faire pour faire fonctionner la même mémoire en écriture et en lecture? *On pourra judicieusement regarder le cours, la documentation de la RAM ainsi que le composant Buffer.*
- Modifiez votre circuit de manière à ce qu'il affiche successivement (dans deux afficheurs hexadécimaux) le contenu de chaque case de la mémoire. Pour cela, vous utiliserez un compteur 4 bits, et à chaque cycle d'horloge votre circuit affichera le contenu de la case mémoire dont l'adresse est donnée par le compteur.

On peut aussi utiliser un composant `Probe` et afficher en Hexa directement.

3.3 Banc de registre

EXERCICE 5 ► Banc de registres

Construisez un **banc de registres, avec 4 registres 4 bits** capable de lire deux registres et d'écrire un registre. On commencera par se poser la question du nombre d'entrées et de sorties d'un tel circuit.

3.4 Reconnaissance de séquence

EXERCICE 6 ► **Un automate reconnaisseur**

Construisez en LOGISIM un automate séquentiel reconnaissant le motif 111 : la sortie doit être à 1 sur un cycle si, lors des trois cycles précédents, l'entrée était à 1 (en tout cas sur le front montant de l'horloge, en fin de cycle). On utilisera la méthodologie suivante :

- Décrivez la machine voulue en langage courant. *En particulier, que veut dire "reconnaître"?*
- Quelles sont les entrées et les sorties de l'automate à construire?
- Dessinez un automate équivalent au circuit à construire.
- Construisez la table de vérité du circuit. *L'état suivant est calculé à partir de l'état courant et de l'entrée, et la sortie est calculée à partir de l'état courant uniquement.*
- Dessinez sur papier le circuit.
- Dessinez avec LOGISIM et testez (ce n'est pas si simple!)
- Dans un autre onglet, construisez un circuit qui utilise des flips-flops pour se rappeler de 3 valeurs successives de l'entrée.
- Testez l'égalité fonctionnelle de ces deux circuits.