
ASR1, TD/TP : construction des registres

Ce TP est à réaliser avec *Digital*. Vous pouvez télécharger cet outil depuis <https://github.com/hneemann/Digital/> (la requête Google qui le trouve est “digital logisim”). Cliquez sur “Download latest Release”, puis “Digital.zip”; décompressez le fichier zip. Finalement, lancez `Digital.exe` sous Windows et `Digital.sh` sous Linux.

Exercice 1: Verrou, flip-flop et chenillard :

1. Dans un sous-circuit `latch`, implantez un verrou à l’aide d’un multiplexeur 2 vers 1. Le verrou doit être passant sur le niveau bas de l’horloge, et verrouiller sur le niveau haut.
2. Dans un sous-circuit `flip-flop`, implantez une bascule à l’aide de deux `latch` comme dans le poly. Votre bascule doit être régie par le front montant de l’horloge. Inutile de chercher à recycler le composant `latch`.
3. Créez un sous-circuit `FFRS` qui, en plus du signal d’horloge, a une entrée `reset` qui force la valeur stockée à 0 de façon synchrone (i.e. au prochain front montant de l’horloge).
4. (optionnel, plus difficile) Créez un autre circuit `FFRA` qui, en plus du signal d’horloge, a une entrée `reset` qui, au niveau haut, force la valeur stockée à 0 de façon *asynchrone*, c’est-à-dire instantanément. Il faudra aller modifier les latches eux-même.
5. Dans le circuit `main`, réalisez un chenillard à 5 leds : une seule led allumée à la fois, et elle se déplace cycliquement. Votre circuit doit comporter une horloge, ainsi qu’un bouton `start` :
 - lorsque `start = 0` en fin d’un cycle d’horloge (front montant), le circuit est ré-initialisé (seule la première led est allumée, et le circuit est prêt à redémarrer),
 - lorsque `start = 1` la led allumée avance dans le chenillard.Les spécifications du circuit ne sont pas très précises... Faites donc un chronogramme pour bien décrire le comportement au démarrage du circuit, puis sur une dizaine de cycles.

Exercice 2: Du registre à la mémoire :

Construire un registre n bits à partir d’un registre 1 bit.
Construire une mémoire adressable de 16 cases de 4 bits à partir de registres 1 bit et de multiplexeurs.