

TD 2 - CIRCUITS SÉQUENTIELS

Comparaison des circuits séquentiels élémentaires, banc de registres et registres à décalages.

Page web : <http://www-rocq.inria.fr/~acohen/teach/archi.html>

Support : `wire.lgf` comporte des « macros » pour faciliter le dessin de circuits répétitifs et `reg_mask.lgf` propose un masque du banc de registres (avec les circuits séquentiels de base).

Exercice 2.1 - Banc de registres

On souhaite réaliser un banc de registres en *DigLog*.

Le *banc de registres* vu en cours comporte :

1. un circuit séquentiel de mémorisation, généralement une *bascule D*, pour chaque bit de chaque registre du banc ;
2. un *port d'écriture* comportant un décodeur pour sélectionner le numéro de registre qui doit être modifié (e.g., le résultat d'une addition) ;
3. deux *ports de lecture* constitués de multiplexeurs pour sélectionner les deux registres que l'on souhaite lire (e.g., les opérandes d'une addition).

D'autre-part, on rappelle qu'une bascule D est constituée de deux latches en séquence avec une horloge inversée ; ce circuit joue le rôle d'un « sas » dont les « portes » s'activent lors des phases descendantes et ascendantes de l'horloge.

Question 2.1.1

Construire un banc de 8 registres 16-bits, avec un port d'écriture et deux ports de lecture (pour la lisibilité du circuit, on construira en fait un banc de 8 registres 1-bit que l'on utilisera 16 fois). Penser à utiliser efficacement la composition hiérarchique des éléments (bascules D, décodeur, multiplexeurs) en *DigLog*.

Exercice 2.2 - Registre à décalage

Outre les registres classiques, un processeur contient souvent des registres à décalage utilisés pour des manipulations de bits, des calculs d'adresses et des opérations logiques. On veut maintenant construire un tel registre à décalage.

Question 2.2.1

En utilisant le registre 1-bit défini dans le fichier `reg_mask.lgf`, réaliser un registre 4-bits capable d'effectuer un décalage de 1 bit vers la gauche ou vers la droite en un cycle (à partir de la valeur enregistrée). On effectuera un décalage « logique », c'est-à-dire que l'on introduira des 0 à droite ou à gauche, respectivement. Le registre doit pouvoir se comporter soit comme un registre classique, soit comme un registre à décalage ; on dispose pour cela d'un signal *NEW* (1 = registre classique, 0 = registre à décalage). On dispose également d'un signal *DIR* indiquant la direction du décalage (1 = décalage à droite, 0 = décalage à gauche).

Question 2.2.2

En utilisant un circuit combinatoire réalisé à la question précédente, construisez maintenant un circuit capable d'effectuer d décalages, $0 \leq d < 4$, à droite ou à gauche sur un mot de 4 bits, en un seul cycle. Un tel circuit s'appelle un *barrel shifter*.

Question 2.2.3

Pour un mot de n bits, quel est le nombre de multiplexeurs qu'un bit donné doit traverser lors d'un décalage de d bits, $0 \leq d < n$?

Quelles seraient les conséquences si l'on cherchait à réduire encore la latence de traversée du registre à décalage ?

Exercice 2.3 (facultatif) - Compléments sur le décalage

On étend le registre à décalage avec des opérations supplémentaires.

Question 2.3.1

Modifier le *barrel shifter* pour permettre le décalage logique, ainsi que la rotation de d bits vers la gauche ou vers la droite.

Question 2.3.2

Si l'on considère maintenant que le *registre à décalage* ci-dessus contient un nombre représenté en complément à 2, quel est l'effet d'un décalage à droite ? à gauche ? En quoi le décalage d'un nombre est-il différent du décalage logique ? Le décalage d'un nombre s'appelle le décalage « arithmétique ». Modifier le circuit ci-dessus pour qu'il puisse effectuer soit un décalage logique, soit un décalage arithmétique.