
6GEI228 – Systèmes Digitaux

Laboratoire #5

Conception à l'aide de flip flops

Hiver 2018

1. Objectifs

- Apprendre à utiliser les bus
- Apprendre à utiliser des modules existants
- Apprendre à se servir des flips flops

2. Méthodologie

Dans la première partie du laboratoire, vous allez suivre un tutoriel pour construire un compteur à 4 bits. Pour ce faire, vous aurez à utiliser un additionneur à 4 bits avec 4 flip flops. Chaque flip flop enregistre 1 des 4 bits du nombre en entrée. Afin d'alléger les dessins, les flip flops sont parfois regroupés. Cette situation est illustrée à la figure 1, où 4 flip flops sont regroupés en un seul ayant 4 bits en entrée et à la sortie.

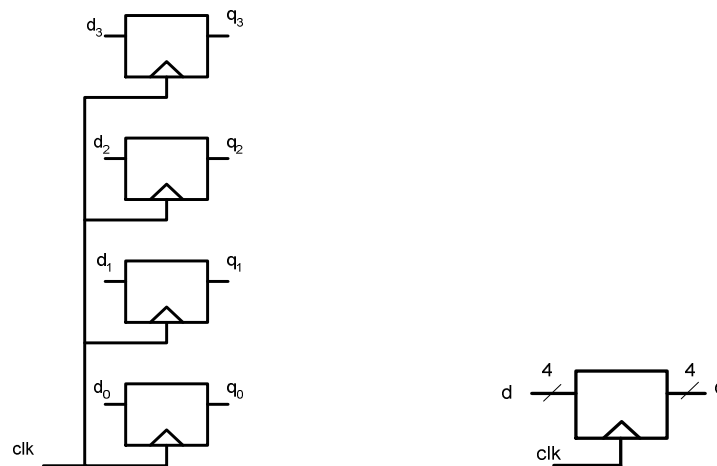


Figure 1. Flip flop à 4 entrées et 4 sorties.

Quand on a un regroupement de fils qui ont un certain lien entre eux (par exemple, les 4 bits d'un nombre de 4 bits), on va appeler ça **un bus** [Un bus informatique est un dispositif de transmission de données partagé entre plusieurs composants d'un système numérique. Le terme dérive du latin *omnibus* (à tous) ; c'est le sens, d'un usage plus ancien, du terme *bus* en électronique]. En utilisant le même principe, il est possible de représenter un additionneur de 4 bits comme ceci :

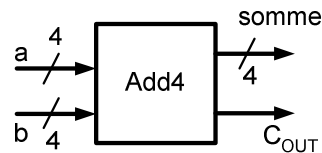


Figure 2. Additionneur de 4 bits.

Si, pour une raison quelconque, nous voulions faire un compteur, il serait possible de connecter la somme de l'additionneur à l'entrée du flip flop et la sortie du flip flop à une des entrées de l'additionneur. À chaque front actif de l'horloge, le nombre de 4 bits dans les flaps flops sera augmenté de 1.

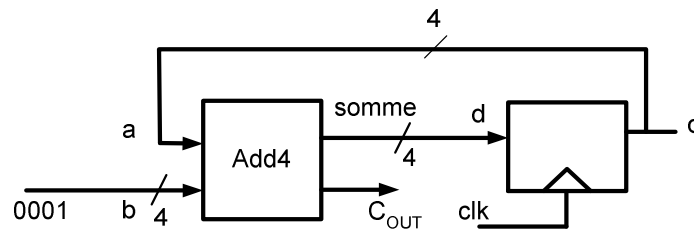


Figure 3. Schéma pour la connexion des flaps flops.

La première partie du laboratoire portera sur le compteur à 4 bits. Les instructions pour cette partie auront plus de détails puisqu'il y aura quelques éléments nouveaux. On parle notamment de la création des bus de données, de l'ajout d'un module qui a été conçu antérieurement et l'usage des flaps flops.

Dans la deuxième partie du laboratoire, il sera question d'implémenter un système de filtrage basé sur la moyenne glissante. Un tel système traite l'information provenant d'une source à flux continu : c'est-à-dire que des nouvelles données arrivent à chaque cycle d'horloge et nous devons les traiter de façon continue.

Un système de moyenne glissante forme un genre de filtre passe-bas. Il est capable d'atténuer les fluctuations qui sont trop rapides tout en laissant passer les transitions lentes. Pour ce faire, il prend la moyenne de plusieurs échantillons pour nous donner la moyenne. Imaginons que les 6 premières données qu'on reçoit du système sont 7, 12, 9, 11, 6 et 14 (avec 7 étant le premier nombre reçu). Les 3 premières sorties ne

Pour implémenter la moyenne glissante, il suffirait d'additionner la valeur en entrée avec les sorties des 3 flaps flops pour ensuite diviser le résultat par 4. Cependant, cette façon de faire nous demanderait d'avoir des additionneurs de plus grandes tailles. Imaginer qu'on additionne $15 + 15 + 15 + 15$, ça donne 60 ce qui demande 6 bits pour le représenter. Si, à la place, on divisait chaque entrée par 4 avant on n'aurait besoin qu'un additionneur de 4 bits. A cause de ce changement, cependant, nous allons perdre un peu de précision (mais ce n'est pas grave).

3. Travail demandé

Première partie

Commencez le laboratoire en ouvrant Quartus Prime et en vous créant un projet. Dans la première partie du laboratoire, nous allons créer un module qui sera simplement une connexion de 4 flaps flops. Ceci servira pour ce premier tutoriel et pour la partie à remettre. Il est donc bien important de bien le faire.

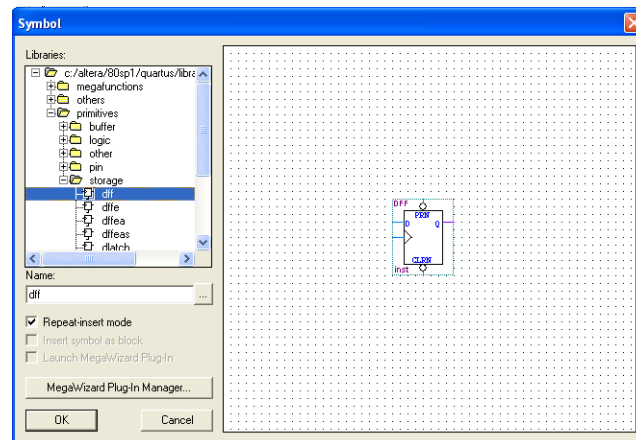


Figure 6. Flip flop de type D (dff).

Insérez 4 flaps flops de type D (DFF) dans votre schéma. Les flaps flops se trouvent dans *PRIMITIVES*→*STORAGE*. La DFF contient 5 pins : la donnée D, l'horloge (avec le triangle), la sortie Q, une remise à 0 et une remise à 1. Les remises à 0 et à 1 servent à forcer la sortie à 0 ou la forcer à 1 au prochain coup d'horloge. Ça peut servir dans plusieurs applications mais, pour nos besoins, nous allons les désactiver en les connectant à 1 (il y a une boule dans le diagramme qui indique que l'entrée est inversée et donc, 1 le désactive).

Connectez les 4 flaps flops ensemble. Pour ce faire, il faut faire les connexions suivantes :

- Le signal d'horloge est le même pour les 4 flaps flops et donc, le signal est partagé.
- Toutes les entrées avec des boules sont connectées à 1 pour les désactiver.
- On fait sortir des lignes pour les entrées D et les sorties Q, mais on ne les connecte pas immédiatement aux broches (pins).

Le circuit résultant devrait ressembler à ceci :

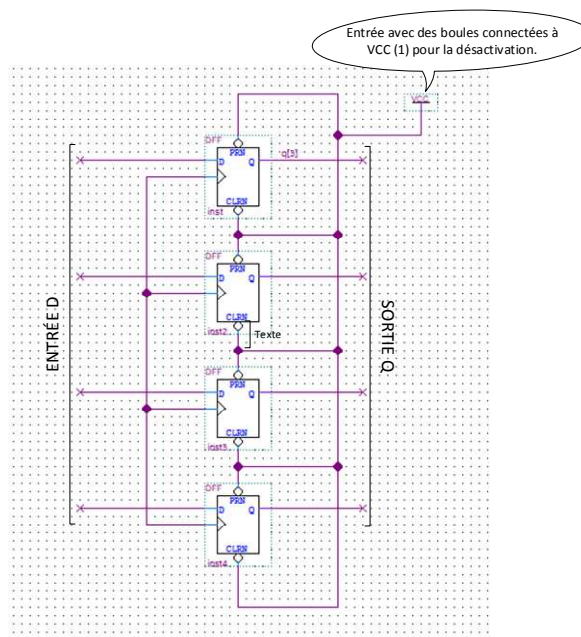


Figure 7. Module à 4 flaps flops.

Afin de simplifier les connexions, nous utiliserons le concept de bus. Par exemple, à la place d'avoir un fil différent pour a_3 , a_2 , a_1 et a_0 (qui sont les bits différents d'un nombre donné), il serait possible de les regrouper en une seule connexion (bus) à 4 bits qui va s'appeler a .

Dans Quartus prime, les bus sont représentés par des fils plus épais. Pour amalgamer les fils pour former un bus, ou pour séparer un bus en fils indépendants, on doit modifier le nom des fils.

On va commencer par donner un nom à chaque fil de données. Celui du haut sera $d[3]$, celui après est $d[2]$, celui après ça c'est $d[1]$ et celui du bas sera $d[0]$. Pour changer le nom, on fait un clic droit sur le fil et on change les propriétés voir figure 8. Il est aussi

possible (même plus facile) de cliquer sur le fil et simplement taper le nom qu'on veut lui attribuer.

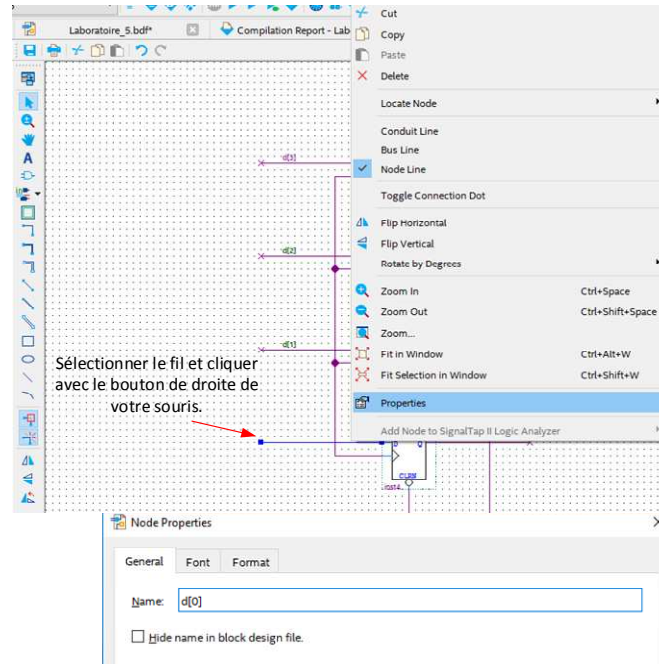


Figure 8. Identification des fils.

Par la suite, on crée un bus. C'est un fil qui est plus épais que les fils qu'on est habitué d'utiliser. Pour vraiment le définir comme étant un bus, on doit donner un nom en respectant une règle. Dans notre cas, on va vouloir utiliser le nom $d[3..0]$. Ça veut dire que le bus s'appelle d et il contient 4 bits allant de 3 à 0. Il faut vraiment respecter cette convention pour que le logiciel l'accepte. Donc, on clique sur le symbole de bus (montré à la figure suivante 9), on trace une ligne au milieu de nulle-part (proche des fils) et on lui donne le nom $d[3..0]$.

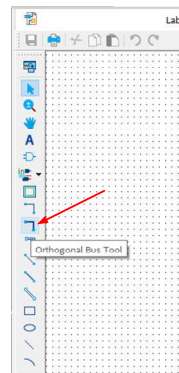


Figure 9. Symbole de bus.

Le résultat devrait nous donner quelque chose comme ceci :

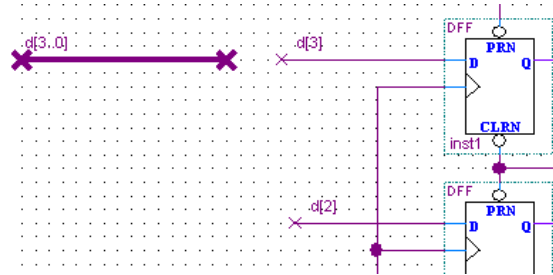


Figure 10. Identification du bus.

Maintenant, en connectant les fils individuels au bus, le système devrait comprendre la relation entre les différents fils. Ceci est présenté à la figure 11 ci-dessous.

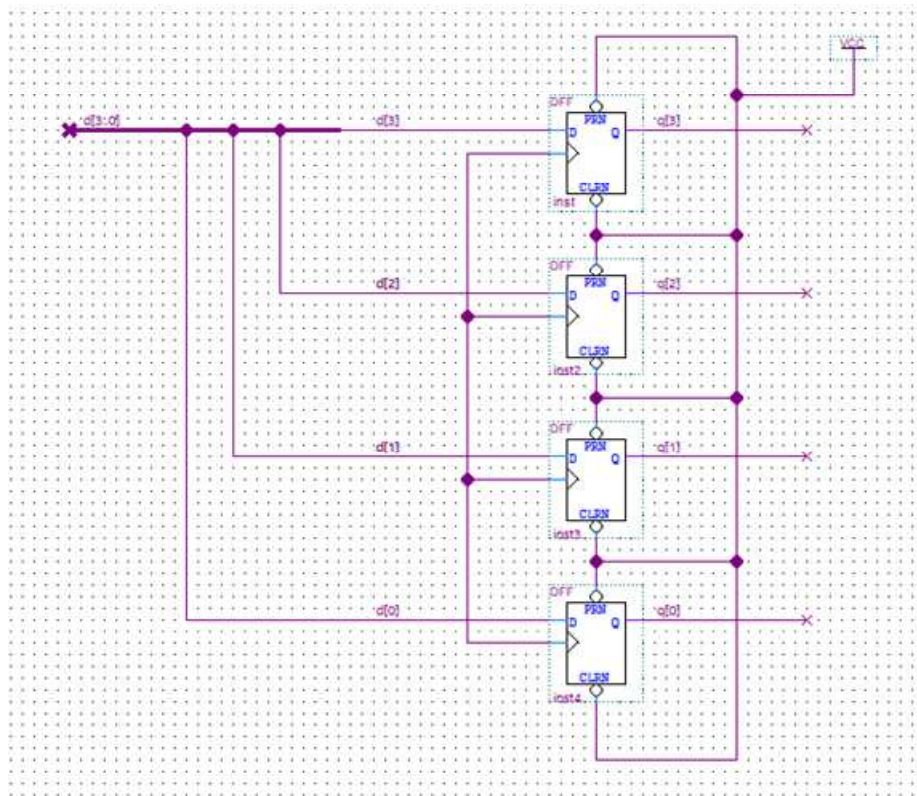


Figure 11. Branchement des fils au bus.

Finalement, on va se créer une broche pour une entrée (PIN) qui, elle aussi, sera un bus de 4 bits. Pour ce faire, on ajoute une broche pour l'entrée comme on le ferait normalement mais on doit changer son nom pour refléter la présence de 4 bits. Ici, on décide de donner le nom *entree [3..0]* (ne pas mettre d'accent).

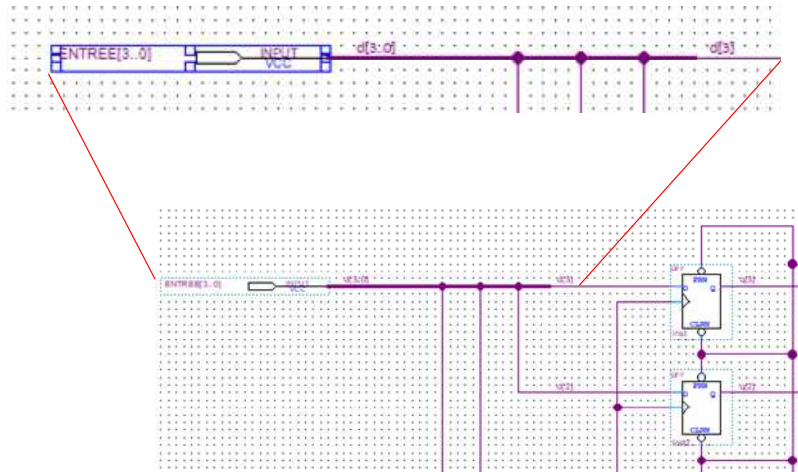


Figure 12. Entrée de 4 bits.

Faites le travail semblable pour la sortie. On aimerait rejoindre les 4 bits de sortie en un bus de 4 bits et on aimerait que la pin de sortie soit appelée *sortie [3..0]*. Ça devrait nous donner quelque chose comme :

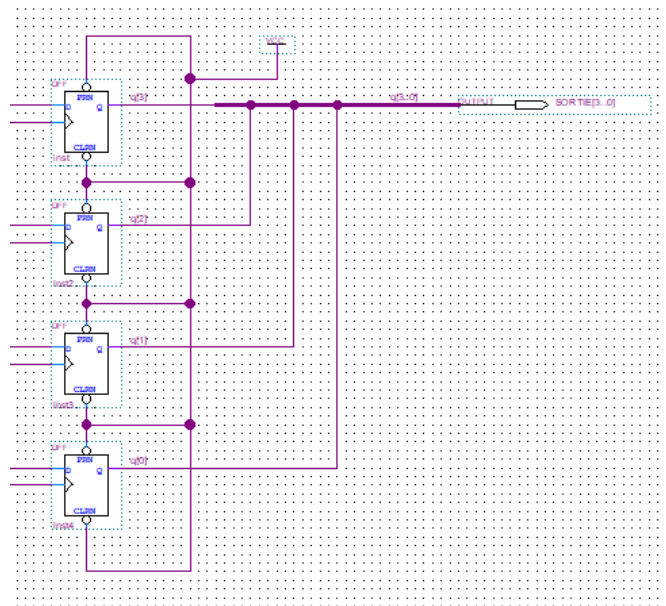


Figure 13. Sortie de 4 bits.

Finalement, ajoutez une broche (pin) d'entrée pour l'horloge. Puisque l'horloge n'est qu'un seul bit et que ce même bit est distribué aux 4 flaps, on n'a pas besoin de bus. Donnez-lui le nom *clk*.

Sauvegardez le design sous le nom *dff4* et faites-vous un symbole avec pour qu'on puisse s'en servir.

Votre symbole devrait ressembler à ceci :

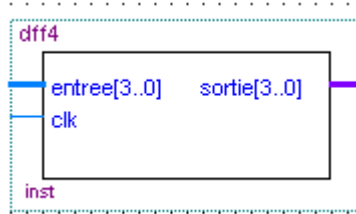


Figure 14. Symbole d'un bus constitué de 4 flaps flops

Sur la page web du cours, vous devriez trouver 2 fichiers : `add4.vhd` et `add4.bsf`. L'un contient le circuit et l'autre contient juste le symbole. Copiez ces deux fichiers à votre répertoire de travail où se trouvent vos fichiers. Si tout se fait bien, vous devriez être en mesure d'utiliser l'additionneur à 4 bits dans vos designs (il devrait se retrouver dans la section *Project* lorsque vous ajoutez les portes logiques).

La prochaine étape est de créer le compteur. Pour ce faire, nous allons utiliser l'additionneur et le module à 4 flaps flops.

Créez un nouveau fichier et ajoutez un module `dff4` que vous venez de créer et un module additionneur que nous vous avons fourni. Placez le `dff4` à droite et l'additionneur à gauche (pas grave, mais c'est plus simple quand tout le monde suit la même convention). Connectez la sortie de l'additionneur à l'entrée du `dff4`. Le fil qui les connecte devrait être un fil de type bus ce qui veut dire que les 4 fils se connectent de façon automatique. La sortie de `dff4` devrait aller à une des entrées de l'additionneur. Ceci est illustré à la figure suivante.

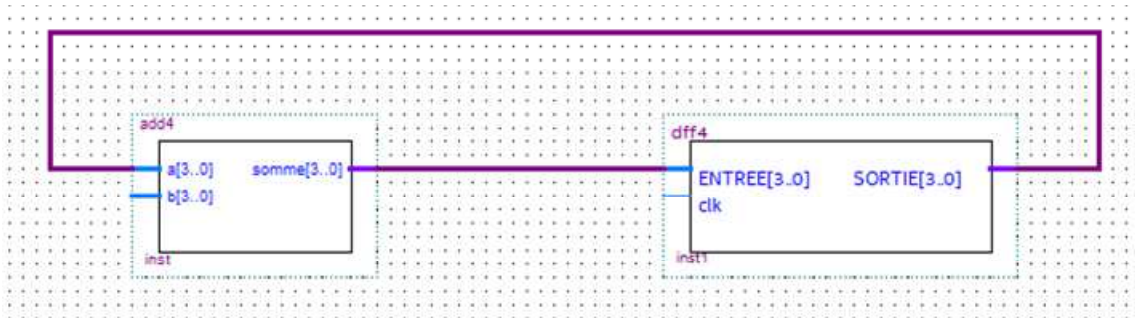


Figure 15. Branchement des blocs.

L'autre entrée de l'additionneur devrait être connectée au chiffre "0001" pour indiquer qu'on additionne 1 à notre compteur à chaque front d'horloge. **Je vais vous laisser le soin de faire cette connexion vous-même.**

Ajoutez une broche (pin) d'entrée pour l'horloge et connectez une broche (pin) de sortie de 4 bits à la sortie du *dff4*. Votre circuit résultant devrait ressembler à ceci (NOTE : la connexion de l'entrée b [3..0] n'est toujours pas faite dans la figure ci-dessous)

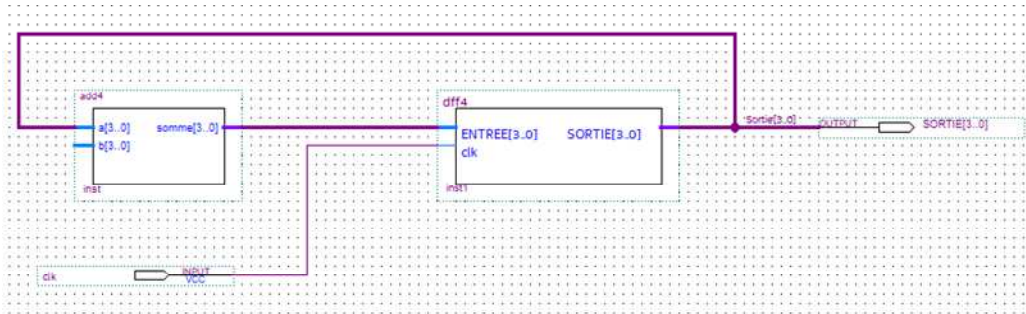


Figure 16. Branchement de l'horloge et de la sortie.

Une fois que le système est complété, sauvegardez-le et créez un banc de test (simulez-le) pour confirmer que le système fonctionne bien. Mettez une horloge de 20ns par exemple en entrée et examinez la sortie. Vous devriez vous retrouver avec quelque chose qui ressemble à :

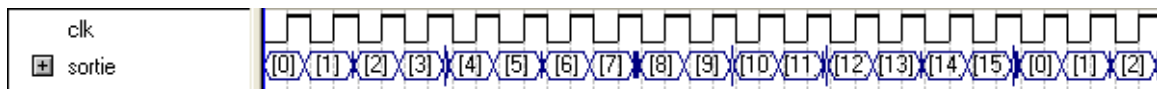


Figure 17. Simulation de la clk.

Implémentez le système sur la plaquette DE10 et connectez l'horloge à l'un des commutateurs et la sortie du compteur à des DELs pour vérifier que le tout fonctionne. Soyez au courant, qu'il est possible qu'un seul changement du commutateur fasse changer le compte de plus que 1. La raison est que le commutateur mécanique cause certaines vibrations électriques qui peuvent être perçues comme étant des signaux d'horloge. Pour en connaître plus, faites une recherche sur le "switch debouncing".

Deuxième partie

Dans la deuxième partie du laboratoire, faites un système de moyenne glissante à 4 bits tel que décrit dans la section théorique du laboratoire. Implémentez le système sur la carte DE10 et montrez au chargé de laboratoire pour avoir vos points.