

Fiche 2 : Séquentiels/Combinatoires



Romain Kamdem

🔗 EXERCICE 1 (Incrémenteur) 📝

On se propose dans cet exercice de faire un incrémenteur N bits, c'est à dire pour une entrée X représentée sur N bits on a une sortie $Xplus1$ qui est l'addition de l'entrée X et de la constante 1.

1. Décrire en VHDL l'entité de cet incrémenteur.
2. Décrire en VHDL l'architecture structurelle de l'incrémenteur en utilisant l'additionneur N bits.
3. Construire un modèle de test pour valider votre incrémenteur

🔗 EXERCICE 2 (Multiplexeur Deux vers Un) 📝

Un multiplexeur deux vers un est circuit permettant de choisir une des deux entrées ($D0, D1$) et la mettre sur la sortie suivant un bit de selection s (La sortie prend $D0$ si s est égal à 0 dans le cas contraire $D1$).

1. Décrire en VHDL l'entité d'un multiplexeur donc les entrées et la sortie sont sur N bit (on utilisera un paramètre générique pour le nombre de bit des entrées/sorties.
2. Décrire en VHDL l'architecture du multiplexeur en utilisant l'instruction concurrente `when ...else`.
3. Construire un modèle de test de pour valider votre multiplexeur

🔗 EXERCICE 3 (Registre) 📝

On se propose de modéliser un registre de N en VHDL ayant les entrées/sorties suivantes :

- Une l'horloge `clk`, un port `load` (actif haut) qui permet de charger la valeur d'entrée sur la sortie, un port asynchrone `nrst` représentant le reset (actif bas) de type `std_logic`.
- Une entrée `D` de type `std_logic_vector` contenant la valeur à stocker.
- La sortie `Q` de type `std_logic_vector` sera initialisée à zero pendant la phase de reset

Le comportement de votre registre sera implémenté à l'intérieur d'un processus sensible sur les ports `clk` et `nrst`.

1. Ecrire en VHDL l'entité et l'architecture en utilisant un paramètre générique `N` pour le nombre de bits de votre registre
2. Construire en VHDL un modèle de test pour simuler votre registre avec `N = 16`

🔗 EXERCICE 4 (Inverseur) 📝

On se propose dans cet exercice de faire un inverseur `N` bits, c'est à dire pour une entrée `X` représentée sur `N` bits on a une sortie `InverseX` qui inverse tous les bits de l'entrée `X`.

1. Décrire en VHDL l'entité de cet inverseur.
2. Décrire en VHDL une architecture structurelle de l'inverseur en utilisant comme composant de base un inverseur `1` bits.
3. Décrire en VHDL une architecture flot de données de l'inverseur en utilisant une instruction concurrente.
4. Construire un modèle de test de pour valider votre inverseur
5. construire une configuration pour choisir une des architectures