

Exercice généricité

Additionneur BCD combinatoire

semestre printemps 2023 - 2024

Composant à développer

Nous souhaitons développer un module capable d'additionner deux nombres codés en BCD (Binaire Codé Décimal). Le nombre de digits de l'entrée est un paramètre générique `NDIGITS` (dénoté N dans la suite du document).

Spécifications

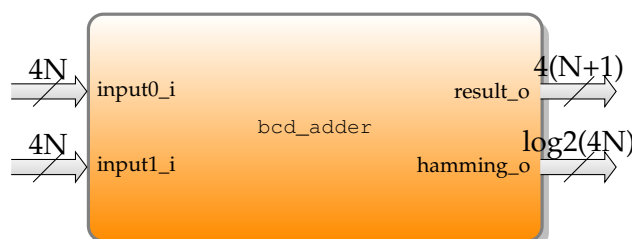
Le système doit réaliser deux opérations. La première est l'addition des deux entrées, fournies en BCD. Chaque entrée, sur $4N$ bits, code donc une valeur comprise entre 0 et $10^N - 1$. Le résultat sera sur $4(N + 1)$ bits BCD.

La deuxième opération du système consiste à fournir la distance de Hamming séparant ces deux nombre. La distance de Hamming correspond au nombre de bits différents entre deux vecteurs de bits. Pour ce faire considérez les bits représentant les deux opérandes d'entrée.

Entrées/sorties

Les entrées/sorties sont :

Nom	Taille	Dir	Description
input0_i	$4N$	in	Premier nombre
input1_i	$4N$	in	Deuxième nombre
result_o	$4(N + 1)$	in	Résultat
hamming_o	$\log_2(4N)$	out	Distance de Hamming



Il est important de noter que pour des raisons de bonne pratique de design, les ports ne sont pas de simples vecteurs, mais des `bcd_number`. Un `bcd_number` étant défini dans le fichier `bcd_pkg.vhd` comme :

```
type bcd_digit is std_logic_vector(3 downto 0);
type bcd_number is array(<>) of bcd_digit;
```

L'additionneur possède un paramètre générique, `NDIGITS`, qui permet de définir le nombre de digits de l'additionneur.

Enfin, vous disposez d'un composant `GenericFullAdder`, qui implémente l'addition d'un digit. Exploitez-le pour développer votre solution.

Exercice

Implémentez le composant tel que décrit et vérifiez en synthèse qu'il ait l'air correct.