

TP 9 ELECTRONIQUE : VHDL (Composants Imbriqués)

Réalisé par :

- PENDU Louis
- PELTIER Alexandre

SOMMAIRE

PAGE 1 : Sommaire

PAGE 2 : Exercice 1 : Demi-Additionneur

PAGE 3 : Exercice 2 : Additionneur Complet

Dans ces exercices à chaque fois nous avons réalisé différents fichiers portant l'extension .vhd. Les fichiers sont donnés en annexe. Voici les différentes sources VHDL utilisées :

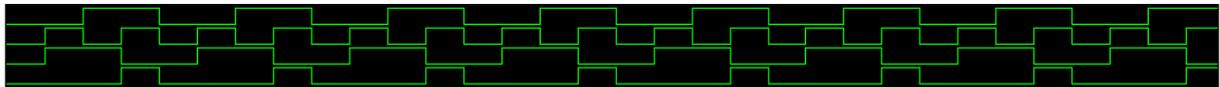
- Ou.vhd
- Xor.vhd
- Et.vhd
- Test_demi_add.vhd
- Test_full_add.vhd

Dans ce compte-rendu nous ne mettrons que les chronogrammes obtenus que nous commenterons. Désolé pour la lisibilité de notre chronogramme mais faire des impressions écran sur linux et l'importer vers Windows endommage la qualité !! Et nous ne maîtrisons pas Latex.

Exercice 1 : Demi-Additionneur

Nous avons à présent réalisé un composant demi-additionneur 1 bit (HalfAdder). Ce demi-additionneur a pour caractéristique un temps de propagation de 1 ns à chaque porte logique de base. Pour réaliser ce composant nous avons besoin des portes XOR et ET. En effet la sortie sera construite à l'aide de la porte XOR et la retenue sera faite avec la porte ET. Ce travail correspond aux sources VHDL suivantes : xor.vhd, et.vhd et test_demi_add.vhd.

Chronogramme :

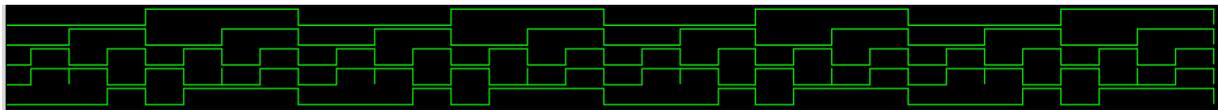


Sur ce chronogramme nous avons en entrée A (1° Ligne) et B (2° Ligne). En sortie nous avons S (3° Ligne) et R (4° Ligne). Lorsque A et B valent 0 alors S vaut 0 et R vaut 0. Ensuite lorsque A vaut 0 et B vaut 1 alors on a S à 1 et R à 0. Après on fait passer A à 1 et B à 0 ce qui donne 1 pour S et 0 pour R. Enfin lorsque A et B valent 1 alors S est à 0 et R à 1. Nous avons donc bien les caractéristiques d'un demi-additionneur 1 bit.

Exercice 2 : Additionneur Complet

Dans ce dernier exercice nous avons réalisé un composant additionneur complet 1 bit (FullAdder). Pour cela on réutilise le demi-additionneur élaboré ci-dessus. Cet additionneur complet a comme caractéristique un temps de propagation de 1 ns à chaque porte logique de base. Donc les sources VHDL liées à cet exercice sont : xor.vhdl, et.vhdl, ou.vhdl et test_full_add.vhdl. De plus comme vous pouvez le voir nous avons ajouté une porte OU dans notre réalisation : elle nous servira à relier les deux retenues sortantes des deux demi-additionneurs en faisant $R1+R2$. On se sert du OU créé dans le TP8.

Chronogramme :



Sur ce chronogramme en entrée nous avons Rin (1° Ligne), B (2° Ligne) et A (3° Ligne) puis en sortie nous avons S (4° Ligne) et Rout (5° Ligne). Quand nos entrées sont initialisées à 0 alors nos sorties valent 0. Ensuite lorsque l'on passe A à 1 on a S à 1 et Rout à 0. Ensuite on a A à 0, B à 1 et Rin à 0 on obtient toujours S à 1 et Rout à 0 et cela ainsi de suite pour arriver au cas où toutes nos entrées sont à 1 et alors nos sorties sont à 1. Par conséquent on constate bien que l'on obtient un additionneur complet 1 bit à l'aide de deux demi-additionneurs 1 bit réalisé dans l'exercice précédent.