

VHDL – TP1

VHDL – Environnement ModelSim

Objet :

Prise en main de l'environnement ModelSim 6.3c
Programmation et simulation de composants de base
Utilisation sur un exemple de multiplieur

1 Documents

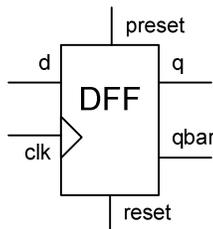
1. Simulation tutorial ([pdf](#))
2. Stimulus Creation Tutorial ([pdf](#))
3. Full ModelSim tutorial ([pdf](#))
4. ModelSim User's guide ([pdf](#))

Utiliser essentiellement les documents 1 & 2.

2 Programmation d'un DFF

2.1 Modèle idéal

(sans retards)



Les signaux sont types BIT. Le **reset** et le **preset** sont asynchrones et actifs au niveau '1'. **reset** a priorité sur **preset**.

Programmer, tester (par simulation). Ceci suppose d'avoir lu et assimilé le document 1. Le document 2 peut aussi rendre service!

2.2 Modèle avec durées

Enrichir le modèle précédent avec des paramètres temporels génériques (**tpdhl**, **tpdlh**, **tsu**, **th**). Faire des exécutions montrant des violations de set-up ou hold time.

3 Utilisation des DFF

On veut réutiliser le composant DFF idéal pour réaliser divers circuits contrôlés sur front montant d'horloge :

3.1 Reg4

Il s'agit d'un registre 4 bits, avec une entrée **en** (enable) synchrone.

3.2 SIPO3

C'est un registre à décalage entrée série, sortie parallèle, 3 bits, avec entrée **clear** asynchrone.

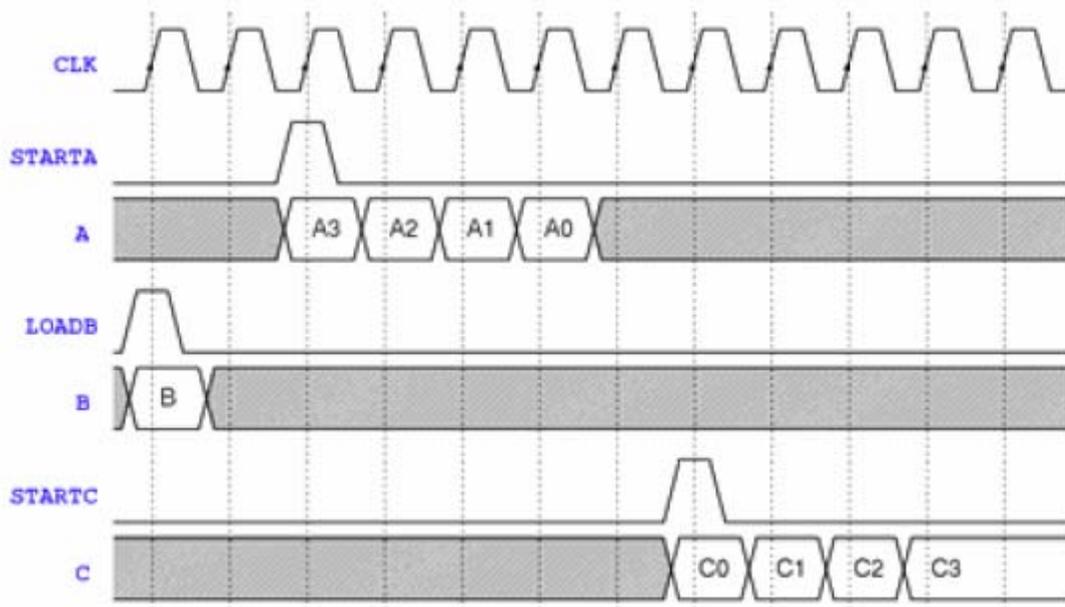
3.3 PISO4

C'est un registre à décalage entrée parallèle, sortie série, 4 bits, avec entrée **clear** asynchrone.

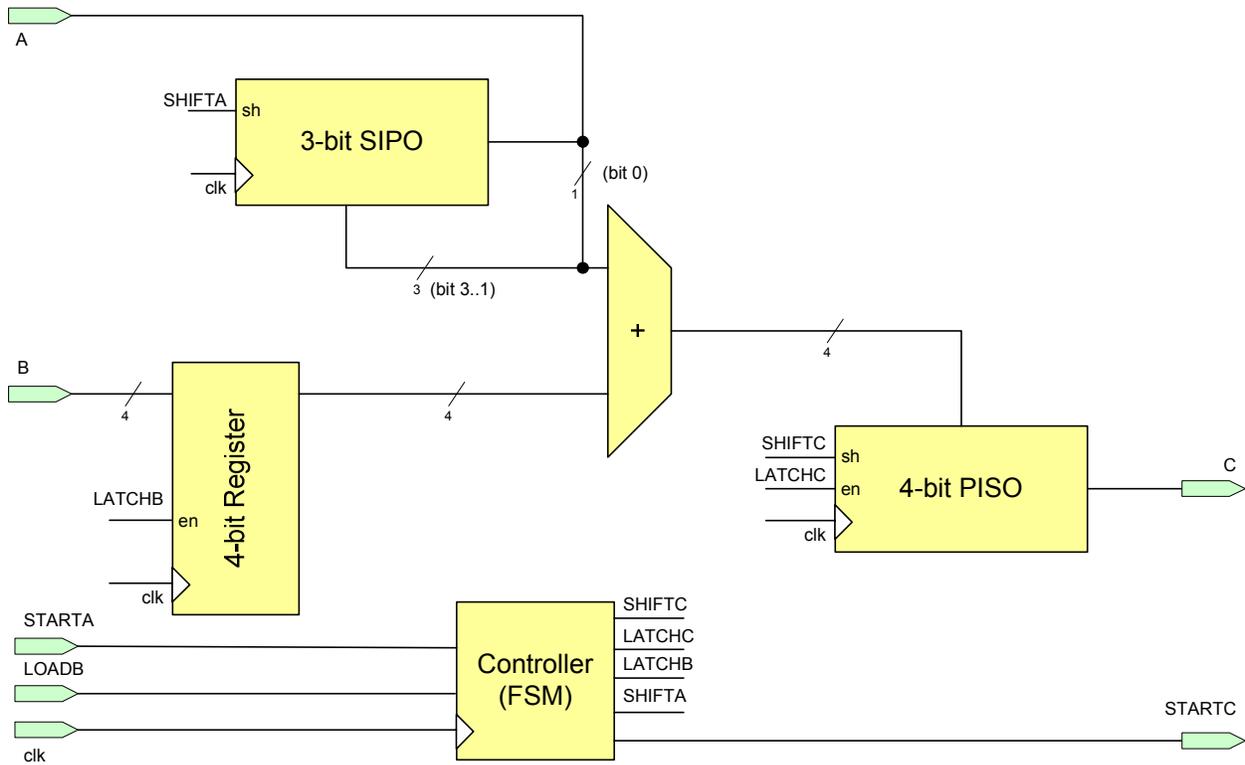
4 Conception d'un additionneur séquentiel complexe

Cet additionneur 2 fois 4 bits, reçoit la donnée A en série (MSB en premier) et la donnée B en parallèle. Le résultat C sort en série (LSB en premier).

Le comportement temporel est donné dans la figure ci-dessous.



L'architecture retenue est

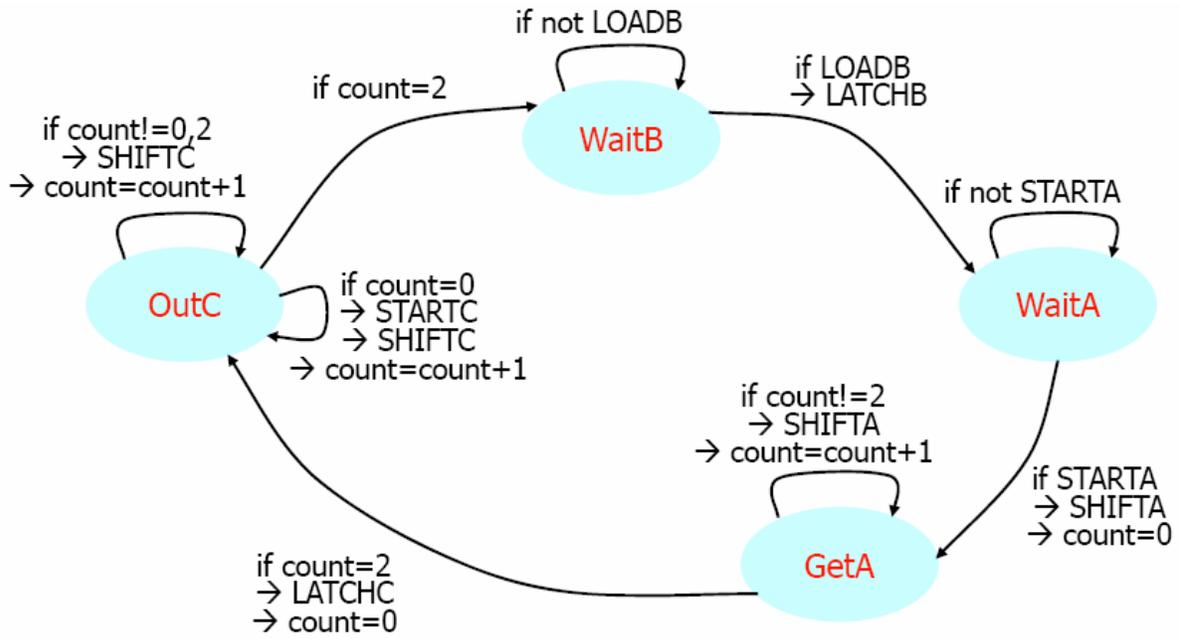


4.1 Additionneur

Programmer et tester un additionneur 4 bits sans retenue (le faire sous forme binaire en utilisant le fau (full adder unit, 1 bit).

4.2 Programmation du contrôleur

Son comportement est spécifié par la machine à états suivante :



Programmer ce contrôleur en VHDL.

4.3 Assemblage

Réaliser l'additionneur en assemblant les circuits développés en 3 + le contrôleur + l'additionneur.

That's all folks!