

## Labo 9

### Communications asynchrones

*Le but de ce laboratoire est de vous confronter aux problèmes inhérents aux horloges multiples et les précautions à prendre en pareille situation pour transférer de l'information d'un monde synchrone à un autre.*

Un système P (ressource Proc.vhd) doit utiliser un coprocesseur C (ressource SQRT\_OPT3.vhd) qui calcule une racine carrée. Lorsque P doit envoyer une donnée à C, il place la donnée sur son bus de sortie «*toc\_dout\_main*» et active un signal «*toc\_dout\_sync\_rts*» simultanément pendant un cycle d'horloge. Ensuite, il attend la réponse sur son bus d'entrée «*from\_din\_main*» avec le signal de validation «*from\_din\_sync\_rts*».

C commence son calcul quand son signal «*start*» est activé. Son signal «*ready*» est alors désactivé au coup d'horloge suivant et il commence le calcul. Lorsque son calcul est terminé, il place sa donnée sur le bus «*result*» et réactive son signal «*ready*» simultanément.

En cours de synthèse, on s'aperçoit que C ne peut pas travailler à une fréquence supérieure à 20 Mhz. On décide alors de faire travailler P à 25Mhz et C à 20 Mhz.

Réalisez une interface entre C et P qui garantisse le bon fonctionnement des transferts.

Les ressources P et C vous sont fournies dans deux fichiers VHDL. Il ne faut pas les modifier. P affiche le nombre de transferts réussis sur «*passed*» et ceux qui ont échoué sur «*failed*».

Le fichier main contient P,C et «*adapter*», pour vos simulations. La version de adapter est une version (très simpliste) qui ne fonctionne que pour une horloge unique à 20 Mhz.

1. Dans un premier temps, utilisez une seule horloge à 20Mhz et vérifiez qu'il n'y a pas de «*failed*».
2. Ensuite, utilisez deux horloges différentes travaillant toutes les deux à 20Mhz. Que constatez-vous ?
3. Finalement utilisez une horloge à 25Mhz et l'autre à 20Mhz et modifiez la ressource «*adapter*» pour qu'il n'y ait plus aucun «*failed*»

**Bon travail !**