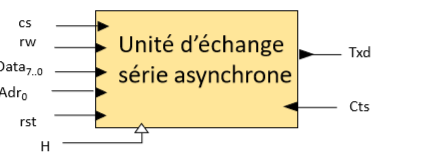
Projet d’Asics : Transmetteur de série

**Introduction**

Le projet consiste à réaliser un transmetteur de série qui permet de faire la liaison entre le modem ook et le CPU. L’ensemble transmetteur et modem forment ce que l’on qualifie de couche physique dans les systèmes de transmission de données. Cette couche est responsable de la sérialisation des données sur le support physique de transmission. Dans ce premier rapport, nous allons construit le modèle sur vhdl puis le simuler sur modelsim.



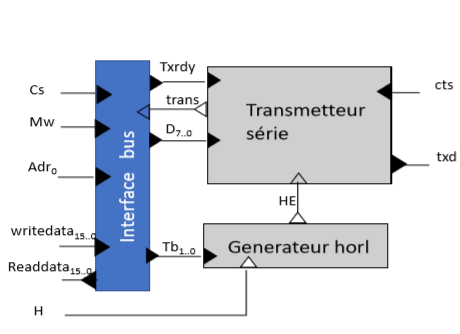
Structure du composant :

Le composant Unité d’échange série asynchrone comporte 3 entités :

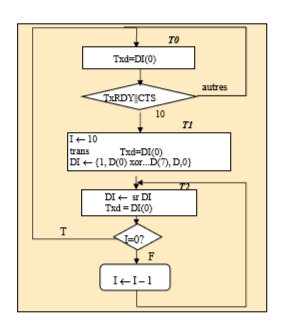
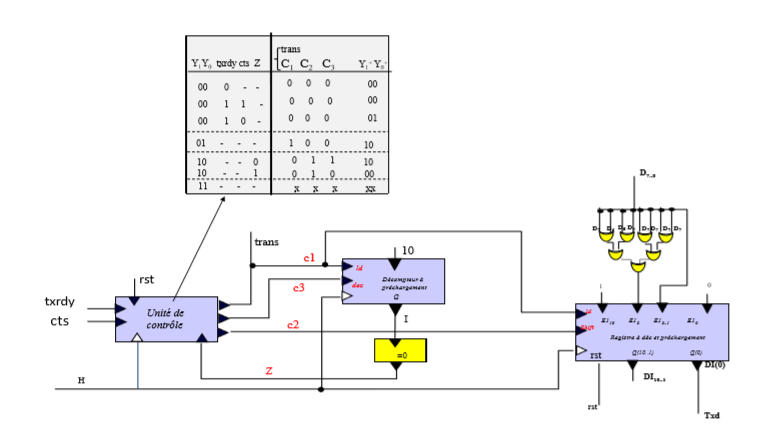
+ Interface du bus : La fonction de transmission série est interfacée avec le bus d’un microprocesseur pour permettre l’accès par logiciel aux registres internes du processeur.

+ générateur d’horloge : permet de générer les horloges selon les débits 400, 800, 1200

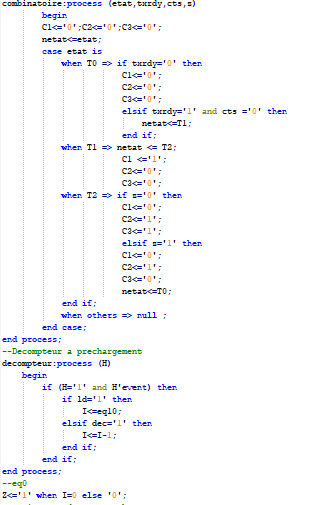
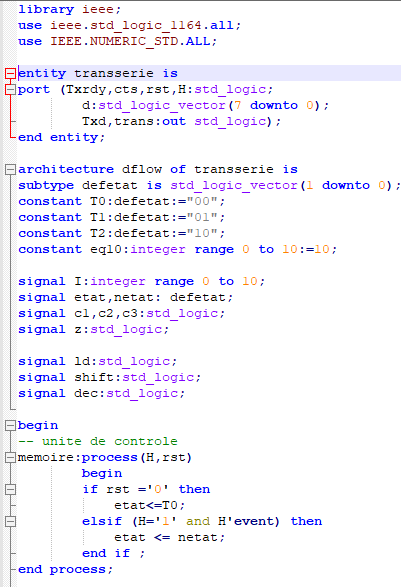
+ Transmetteur de série : faire la transmission des bits de donnée en série à partir de donnée sur 8bits.

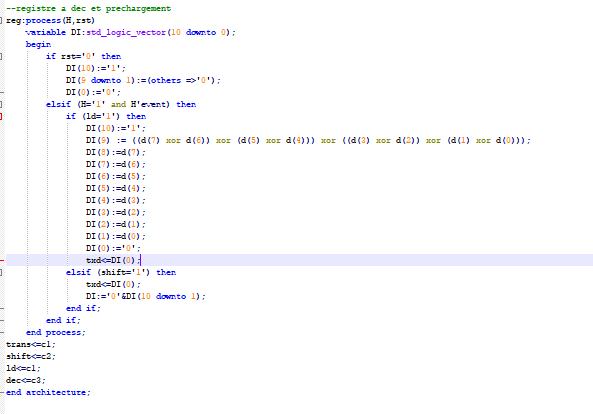


1. **Le bloc de transmission série**



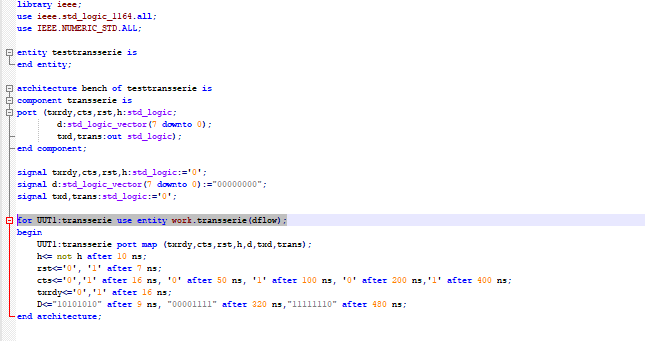
D’après les tables de transition et les schémas RTL des composants, nous écrivons les algorithmes de VHDL comme le suivant.

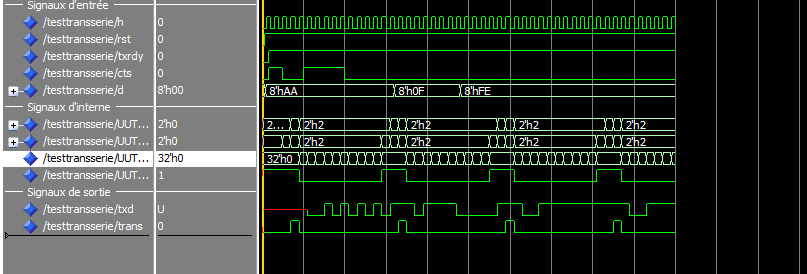




La simulation ModelSim

Testbench

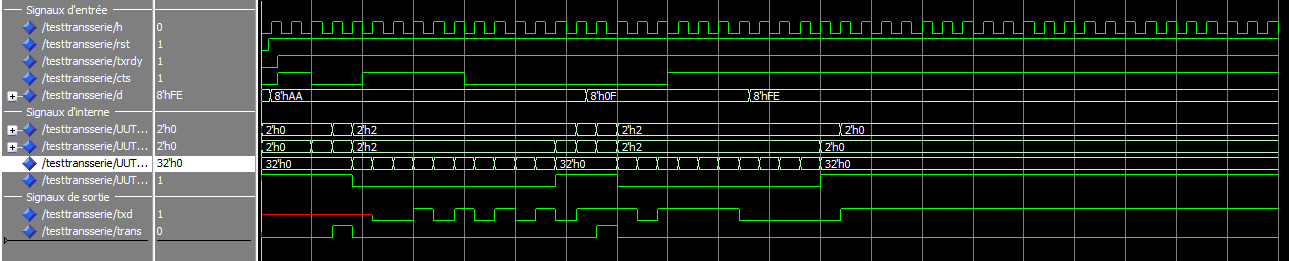




Nous voyons que la transmission des bits se fait à l’état T2 et la transition entre les états se fassent quand cts=’0’ et txrdy =’1’.

Quand nous somme à l’état T2, le compteur de rechargement va décompter de 10 à 0 et quand le compteur est égal à 0, z= ‘1’ et le contrôleur va passer de l’état T2 à T0 pour recommencer le cycle. Sur la sortie txd, nous voyons que le signal de sortie correspond à des bits de data avec le bit de start stop et le bit de parité.

Prenons exemple où le data = 10101010, le bit de partie est à 0, ou data = 11111110 et le bit de parité est à 1. Le bit de start est 0, le bit de stop est ‘1’. Quand le signal cts est à ‘0’, le récepteur n’est pas encore prêt, donc nous avons le txd est à 1, et l’état à T0



Script de simulation

vlib work

vcom transserie.vhd

vcom testserie.vhd

vsim -gui work.testtransserie(bench)

add wave -noupdate -divider {Signaux d'entrée} h rst txrdy cts

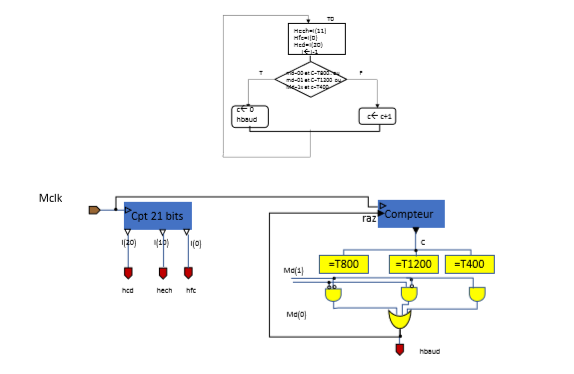
add wave -noupdate -divider {Signaux d'interne} sim:/testtransserie/UUT1/etat sim:/testtransserie/UUT1/netat sim:/testtransserie/UUT1/I sim:/testtransserie/UUT1/z

add wave -noupdate -divider {Signaux de sortie} txd trans

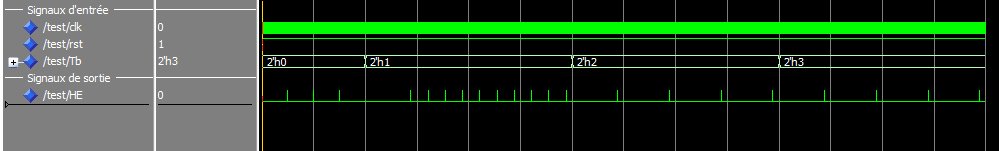
run 1000 ns

wave zoom full

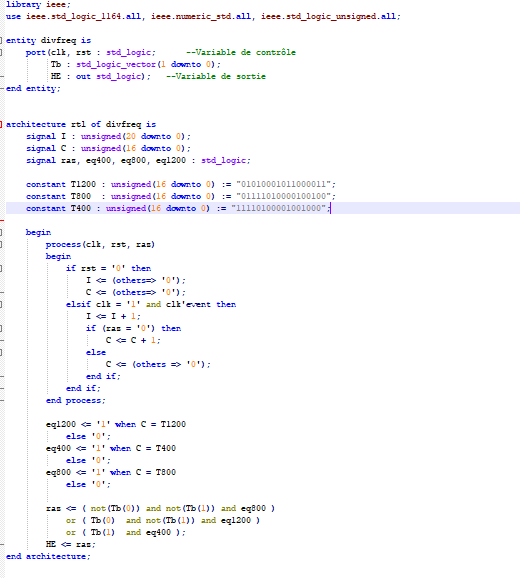
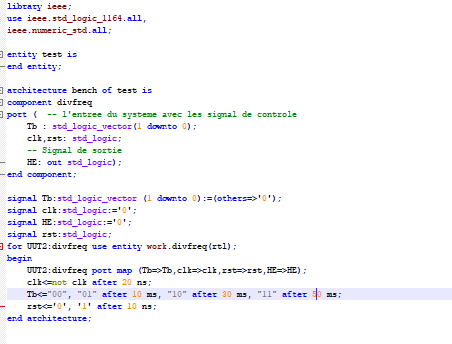
1. Générateur d’horloge



Nous reprenons les architectures du générateur d’horloge que nous avons fait dans les séances avants, et le hbaud est exactement la sortie HE de générateur d’horloge.



Le code du générateur d’horloge



Nous avons les 3 modes différent, le mode T1200, T800 et T400.

|  |  |  |  |
| --- | --- | --- | --- |
| Tb | Mode | Fréquence | Période |
| 00 | T800 | 62.5 Khz | 16000ns |
| 1x | T400 | 125 Khz | 8000ns |
| 10 | T1200 | 41.67 KHZ | 24000ns |

Script de simulation :

vlib work

vcom divfreq.vhd

vcom test.vhd

vsim -gui work.test(bench)

add wave -noupdate -divider {Signaux d'entrée} clk rst Tb

add wave -noupdate -divider {Signaux de sortie} HE

run 70 ms

wave zoom full

C.