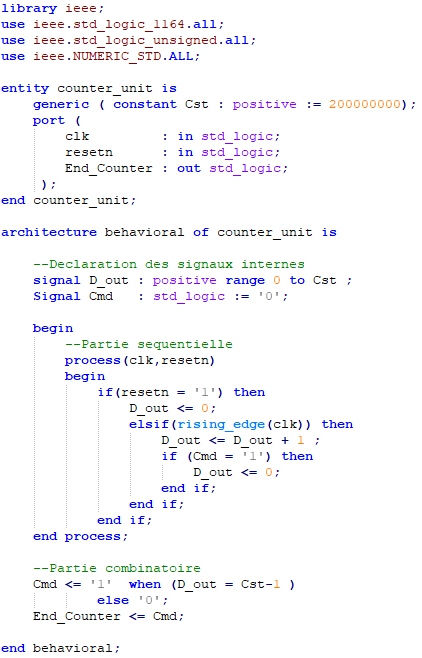
TP03 - Machine d’états (Kamal KHERCHOUCH)

1. **Dans un fichier *.vhd*, créez un module *Counter\_unit* à partir du compteur du TP1. Le module prendra en entrée un signal d’horloge et de resetn, et donnera en sortie le signal *end\_counter*. Utilisez un paramètre *generic()* pour définir le nombre de coup d’horloge à compter.**

Le module Counter\_Unit est decrit dans le fichier Counter\_Unit.vhd.



1. **En schéma RTL, créez un compteur du signal *end\_counter*. Ce compteur doit permettre de déterminer le nombre de cycles allumé/éteint qui ont été effectués par la LED. Le compteur doit pouvoir être remis à 0, maintenir sa valeur actuelle ou s’incrémenter.**

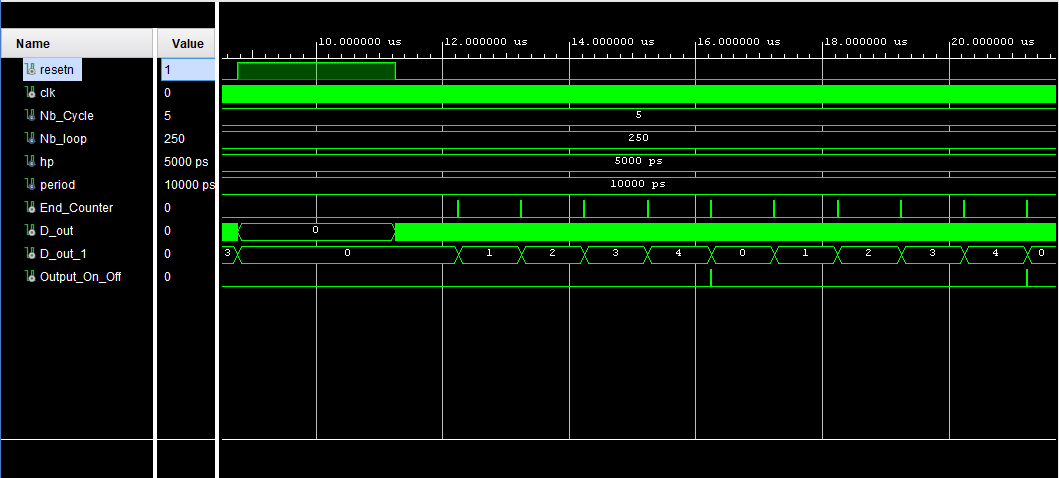
**POSEY LA PHOTO DU SCHEMA RTM**

1. **Ecrivez un code VHDL décrivant ce compteur de cycle, vous utiliserez le module *Counter\_unit*.**

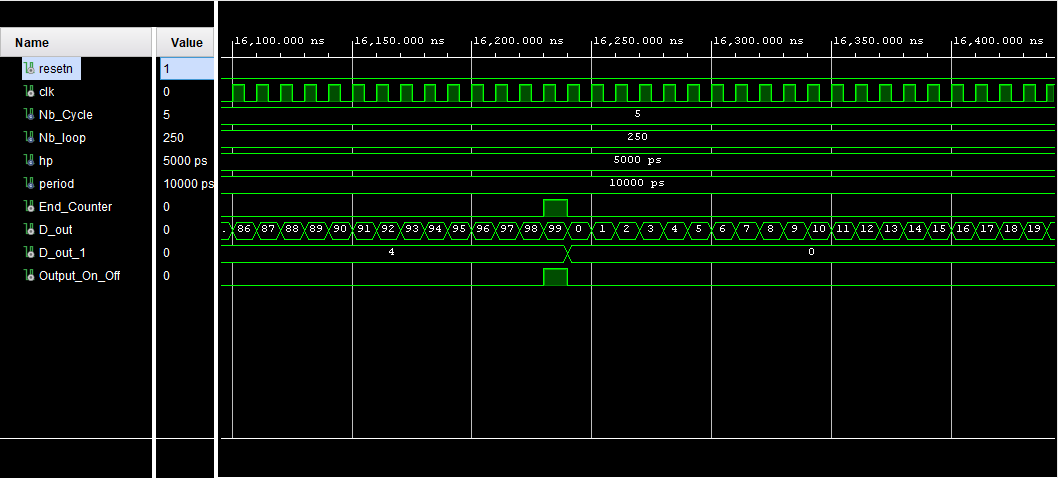
Une copie du code VHDL (à cette étape) est placée dans le dossier sous le nom « tp\_fsm - Question 3.vhd »

Il en est / sera de même pour son test bench associé « tb\_tp\_fsm - Question 3.vhd »

1. **Tester votre architecture avec un testbench.**



Ici, on valide bien le fonctionnement du reset. Lorsque le reset est activé, le compteur s’arrête de compter. Et lorsque le compteur redémarre, on attends bien 5 cycle de comptage du signal End\_Counter avant d’activer le signal de sortie Output\_On\_Off (ce dernier se fait lors de la dernière période du dernier cycle demandé). Voir image ci-dessous.



1. **Créez en RTL une machine à états (FSM) permettant de faire clignoter une LED RGB en rouge puis bleu et enfin en vert avant de recommencer le cycle (rouge, bleu, vert, …). Dans chaque état la LED devra clignoter 3 fois. De plus, si le bouton restart est appuyé, on retourne dans l’état initial quel que soit l’état dans lequel on se situe. L’état initial est l’état dans lequel on se situe au démarrage, on passe à l’état rouge après 3 clignotements de la LED en blanc (rouge, vert et bleu actifs en même temps).**

