

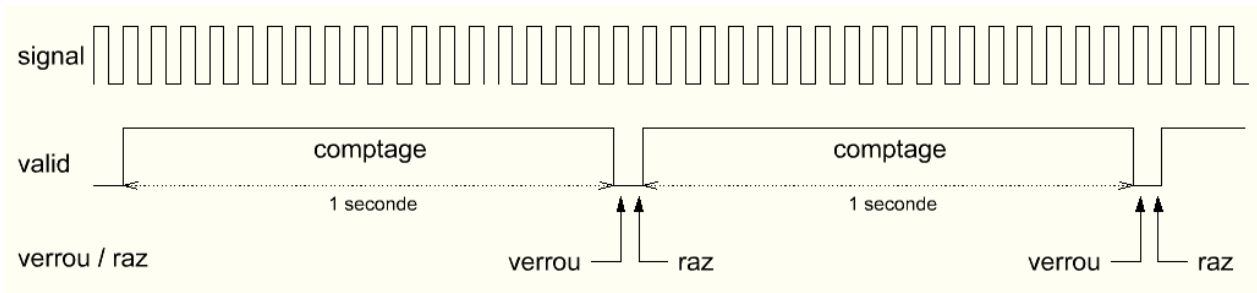
# SYNTHESE LOGIQUE

## Le fréquencemètre

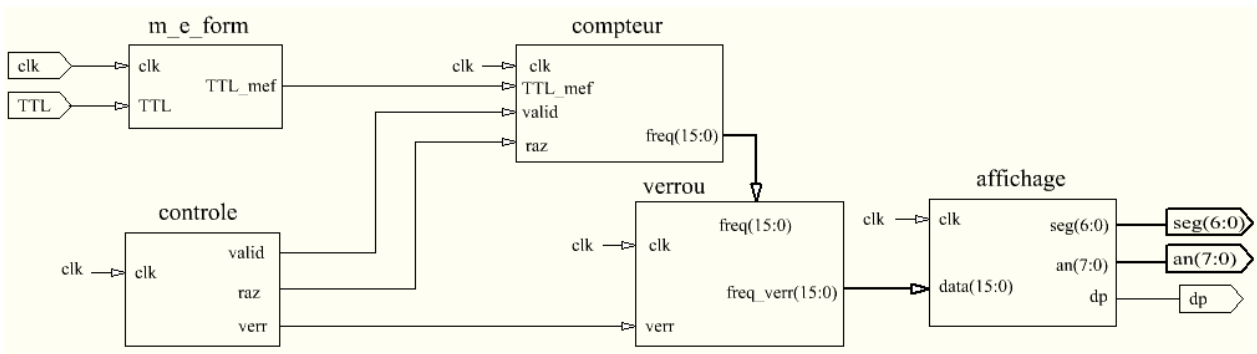
### 1. Principe

Par définition, la fréquence d'un signal est égale au nombre de périodes par seconde :  $f = \frac{1}{T}$

Pour mesurer la fréquence d'un signal, il suffit donc de compter le nombre de périodes pendant une seconde. C'est ce principe qui est utilisé dans ce TP.



### 2. Synoptique



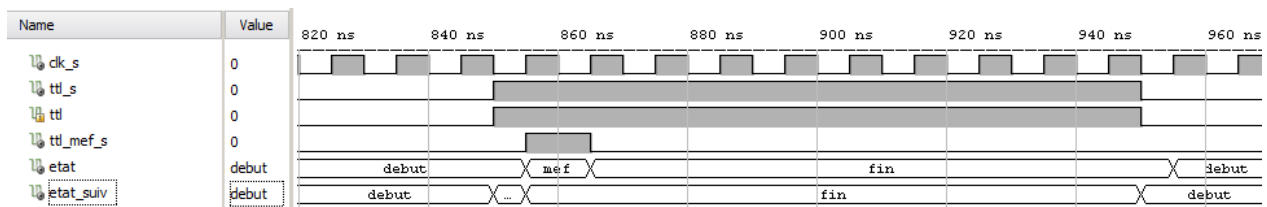
### 3. Cahier des charges

- ☒ Le signal dont il faut mesurer la fréquence est compatible TTL.
- ☒ L'affichage de la fréquence se fait sur 4 afficheurs 7 segments à LED.
- ☒ La gamme de fréquences s'étend de 10 Hz à 9998 Hz, au-delà une LED s'allume et on affiche 0000.
- ☒ Le rafraîchissement de l'affichage se fait toutes les secondes.

### 4. Description des différents modules

#### 4.1. m\_e\_form :

Le bloc « m\_e\_form » génère une impulsion de largeur une période d'horloge après chaque front montant du signal TTL.



C'est une machine à trois états

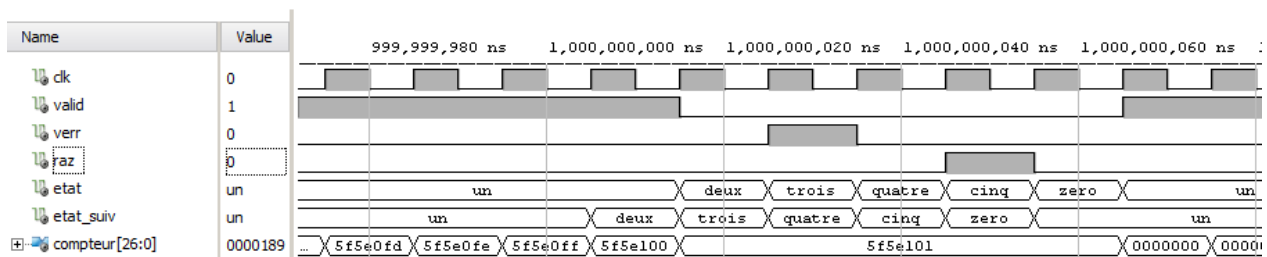
#### 4.2. controle :

Ce module s'occupe de la gestion des signaux « valid », « raz » et « verr » qui doivent arriver dans une séquence précise.

Le signal « valid » reste à '1' pendant une seconde exactement pour valider le comptage, revient à '0' pour 50 ns avant de repartir pour un nouveau cycle.

Le signal « verr » passe à '1' pendant une période d'horloge après le front descendant de « valid », il permet de verrouiller (memoriser) la valeur du compteur afin d'être affichée.

Le signal « raz » arrive ensuite pour remettre à zéro le compteur et le préparer pour le prochain comptage.



C'est une machine à six états

#### 4.3. compteur :

Ce bloc compte, pendant une seconde, le nombre d'impulsions du signal « TTL\_mef ». Le compteur est constitué de 4 décades (mille, centaines, dizaines et unités) et compte en BCD. Arrivé à 9999, il s'arrête de compter, allume une LED et affiche « 0000 ».

#### 4.4. verrou :

Verrouille le résultat du compteur sur front montant d'horloge lorsque le signal « verr » est à '1'. Le signal « verr » passe à '1' toutes secondes, c'est la période de rafraîchissement.

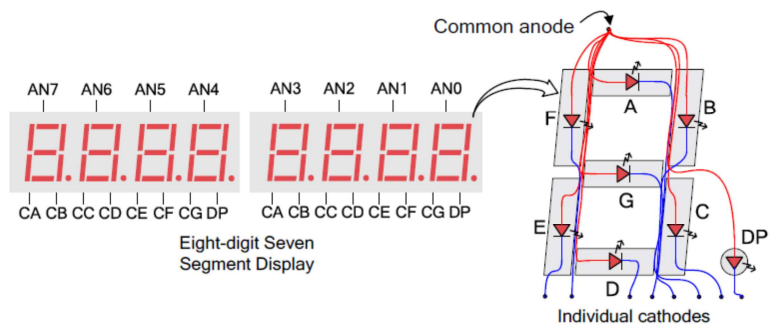
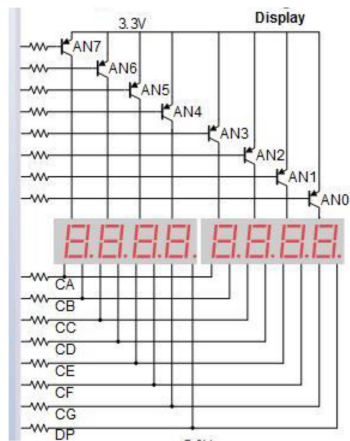
#### 4.5. Affichage :

Affiche la fréquence sur les 4 afficheurs à 7 segments de droite. Les 4 afficheurs de gauche sont éteints.

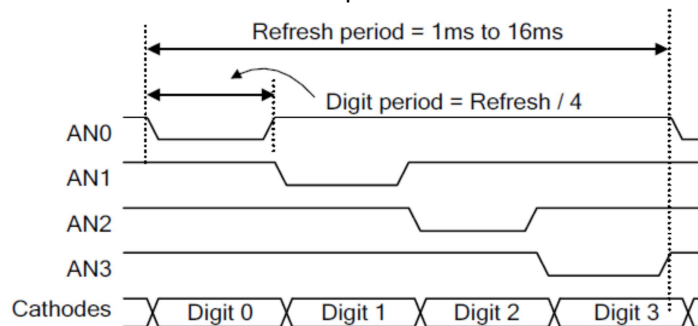
Les afficheurs partagent les mêmes broches du FPGA, il faudra multiplexer les données transmises aux 4 afficheurs.

La période de rafraîchissement doit se situer entre 1 ms et 16 ms.

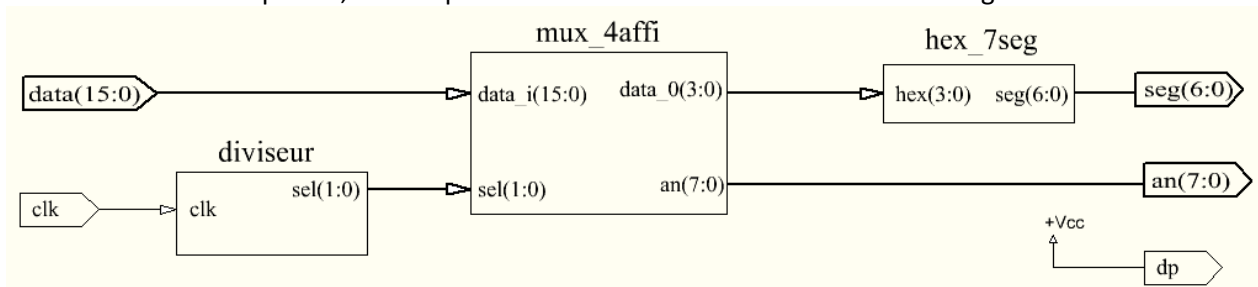
Les afficheurs reçoivent tous les mêmes données à afficher, mais ils possèdent une broche de sélection chacun.



Seuls les 4 afficheurs de droite seront utilisés dans ce TP. Pour afficher des valeurs différentes sur les 4 afficheurs, il faut sélectionner chaque afficheur individuellement en lui envoyant la valeur à afficher. Cette sélection se fera de façon circulaire à une fréquence qui donnera l'impression à l'observateur que les 4 afficheurs sont allumés en permanence.



Le schéma suivant donne le synoptique du module d'affichage qui est constitué de trois blocs : le diviseur de fréquence, le multiplexeur et le décodeur hexadécimal vers 7 segments.



- ☑ **Le diviseur** : fabrique un signal sel(1) de fréquence  $f = \frac{100MHz}{2^{18}}$  et un signal sel(0) de fréquence double.
- ☑ **Le décodeur hex\_7seg** : allume les segments qui permettent d'afficher le symbole correspondant au code hexadécimal.
- ☑ **Le multiplexeur mux\_4affi** : choisit la donnée à afficher en même temps qu'il sélectionne l'afficheur concerné, à la cadence imposée par les signaux sel (1 :0).

## 5. **Pratique :**

Consignes : Le projet doit respecter les techniques synchrones. Chaque module créé doit être validé en simulation avant de passer au suivant.

- 5.1. Concevoir le module « m\_e\_form » sous forme d'une machine d'états (voir les chronogrammes).
- 5.2. Faire le même travail pour le module « contrôle » (voir chronogrammes).
- 5.3. Passer ensuite au compteur, c'est le bloc le plus compliqué de cette application, un organigramme est nécessaire avant de commencer à coder.
- 5.4. Le bloc « verrou » ne doit poser aucun problème, c'est une ligne de code !
- 5.5. Le module « affichage » : déjà vu en TP de Logique, la différence est qu'ici tout est codé en VHDL au lieu de l'outil schématique.
  - 5.5.1. Concevoir le bloc « diviseur » en utilisant un compteur.
  - 5.5.2. Les modules « mux\_4affi » et « hex\_7seg » sont ceux vus en TP de Logique, ils sont donnés dans le répertoire « partage ».
  - 5.5.3. Assembler ces 3 blocs dans un module nommé « affichage ».
- 5.6. Ecrire le programme du niveau 1 « freq.vhd ».
- 5.7. Ecrire le fichier de contraintes pour imposer la position des signaux sur les broches du composant.
- 5.8. Compiler le projet et relever les ressources du composant utilisées ainsi que la fréquence maximale de l'horloge.
- 5.9. Générer le fichier de programmation, le télécharger dans le composant et tester le fonctionnement du fréquencemètre avec un générateur de signaux TTL.