

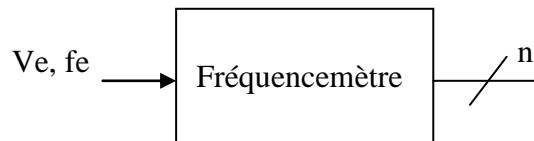
TP2 et TP3

Mini-Projet FPGA

Conception d'un frquencemtre numrique  calibrage automatique

I. Principe du frquencemtre

Un frquencemtre numrique est un appareil qui permet de mesurer la frquence d'un signal d'entre. Ici, notre signal d'entre V_e  analyser sera un signal numrique de rapport cyclique 50% (rapport entre le temps au niveau haut du signal /  la priode).



Le principe de fonctionnement du frquencemtre que vous allez raliser repose sur l'utilisation d'une fentre de comptage (figure 1) correspondant au niveau haut du signal d'entre. En comptant le nombre de priodes d'une frquence rfrence f_h (de frquence plus grande que le signal d'entre), dans cette fentre, nous pouvons alors en dduire la frquence du signal d'entre. Une demi-priode du signal  analyser comporte $N.T_h$

Connaissant T_h et en mesurant N , on peut dterminer la frquence du signal d'entre par la formule suivante :

$$T_e = 2.N.T_h$$

$$f_e = \frac{f_h}{2.N}$$

Exemple 1: La frquence de rfrence est de 50MHz ($T_h=20\text{ns}$). On mesure 4323 priodes de rfrence. La frquence d'entre est $5.10^7/(2*4323)=5783\text{Hz}$.

Exemple 2: La frquence de rfrence est de 5MHz ($T_h=200\text{ns}$). On mesure 4323 priodes de rfrence. La frquence d'entre est $5.10^6/(2*4323)=578\text{Hz}$.

Exemple 3: La frquence de rfrence est de 500kHz ($T_h=2\text{ms}$). On mesure 4323 priodes de rfrence. La frquence d'entre est $5.10^5/(2*4323)=57\text{Hz}$.

Exemple 4: La frquence de rfrence est de 50kHz ($T_h=20\text{ms}$). On mesure 4323 priodes de rfrence. La frquence d'entre est $5.10^4/(2*4323)=5\text{Hz}$.

A partir des exemples, on comprend que la fréquence de référence va fixer le calibre de mesure et la précision de celle-ci.

La fréquence de référence f_h est obtenue par sélection d'une des quatre horloges de référence (f_1, f_2, f_3 ou f_4). Pour une fréquence du signal d'entrée, la fréquence de référence sera celle qui permettra d'obtenir la meilleure résolution concernant N .

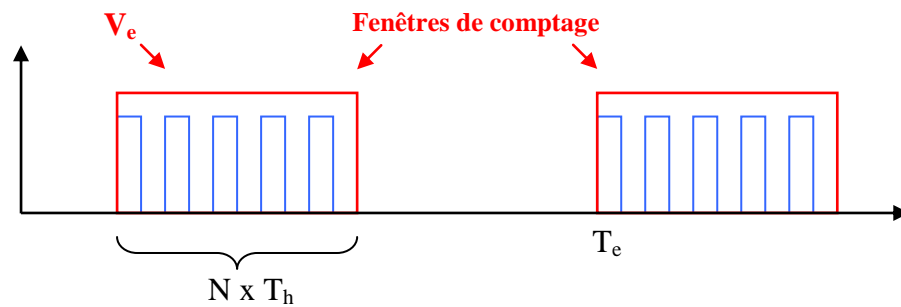


Figure 1 : principe du fréquencemètre

Le fréquencemètre possède ici deux modes. Un mode manuel où l'utilisateur pourra sélectionner lui-même l'horloge de référence, et un mode automatique où en fonction de la valeur de N , un organe de décision déterminera le meilleur calibre (la meilleure fréquence parmi les quatre disponibles) pour N .

On se limitera ici à un fréquencemètre à sélection manuel du calibre. La partie automatique est considérée comme optionnelle et pourra être abordée par l'étudiant lorsque le fréquencemètre en mode manuel sera opérationnel.

Le schéma de principe d'un tel montage (figure 2) est donné ci-dessous et comporte plusieurs blocs que nous nous proposons de dimensionner par rapport au cahier des charges donné.

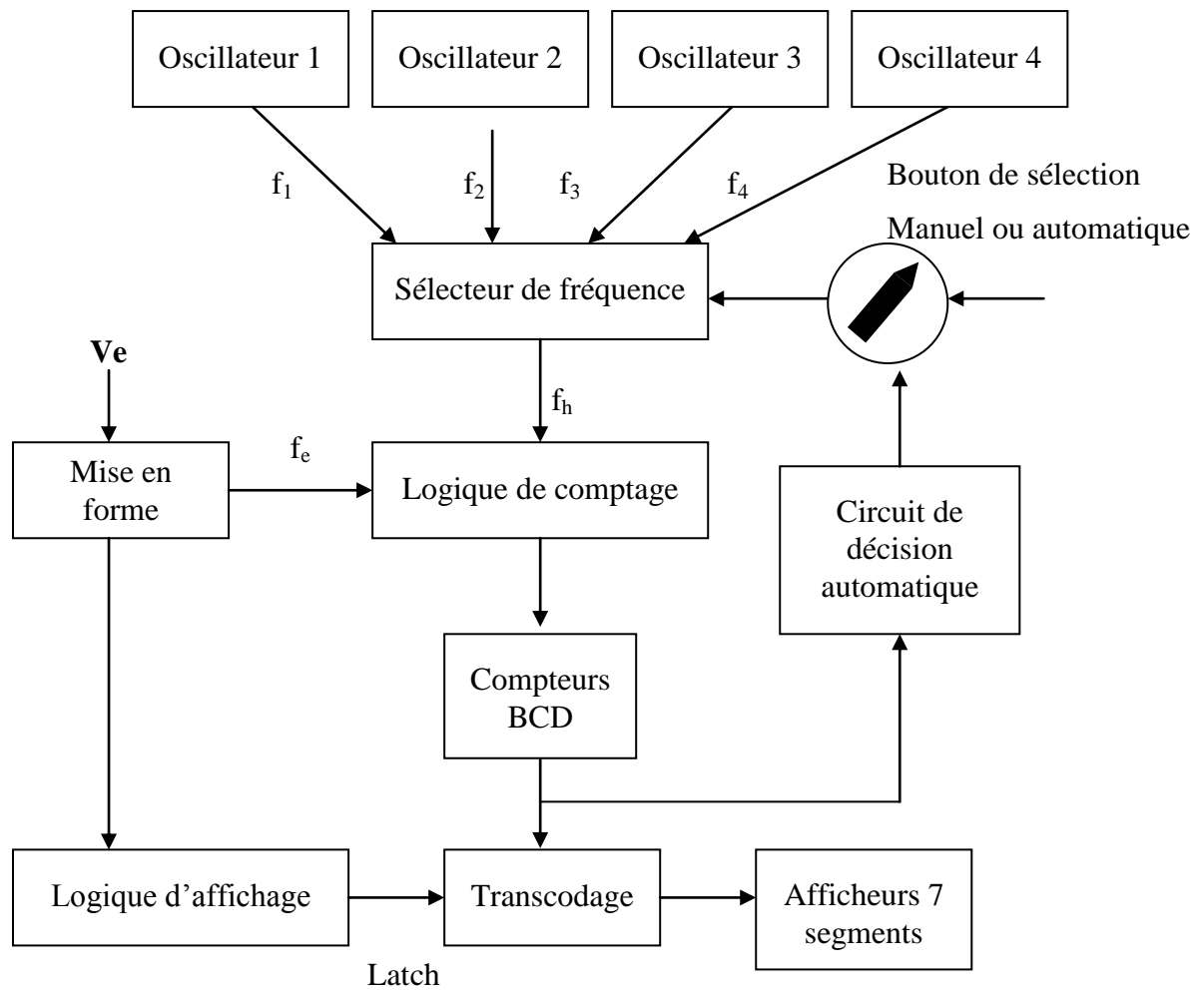


Figure 2 : principe du montage

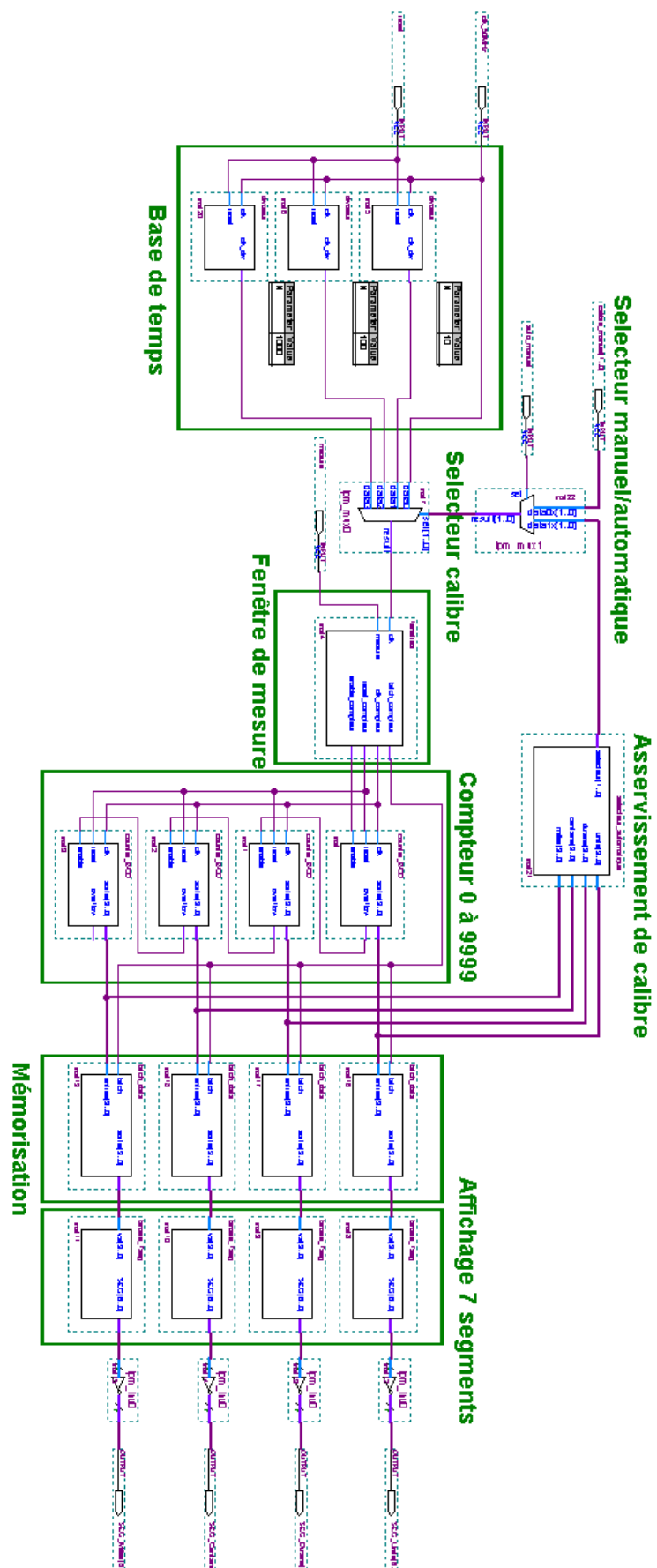


Figure 3: Implémentation sous Quartus

II. Spécifications

On désire réaliser un fréquencemètre numérique répondant au cahier des charges suivant :

Amplitude du signal d'entrée : $V_e \in [0 ; 5V]$: TTL

Fréquences de références :

- F_1 : 50 MHz
- F_2 : 5 MHz
- F_3 : 500kHz
- F_4 : 50 kHz

Plage de mesure de N : 0 à 9999

Affichage de N : 4 digits, 4 afficheurs 7 segments

III. Questions théoriques préliminaires

Question 1. En fonction des quatre fréquences de références disponibles, déterminer quelles seront les plages de mesures concernant la fréquence d'entrée. Pour cela compléter le tableau ci-dessous.

Fréquence de référence	Min	Max
$F_h = F_1 = 50\text{MHz}$		
$F_h = F_2 = 5\text{MHz}$		
$F_h = F_3 = 500\text{kHz}$		
$F_h = F_4 = 50\text{kHz}$		

Question 2. En déduire la plage de mesure de ce fréquencemètre

Fréquence Min	Fréquence Max

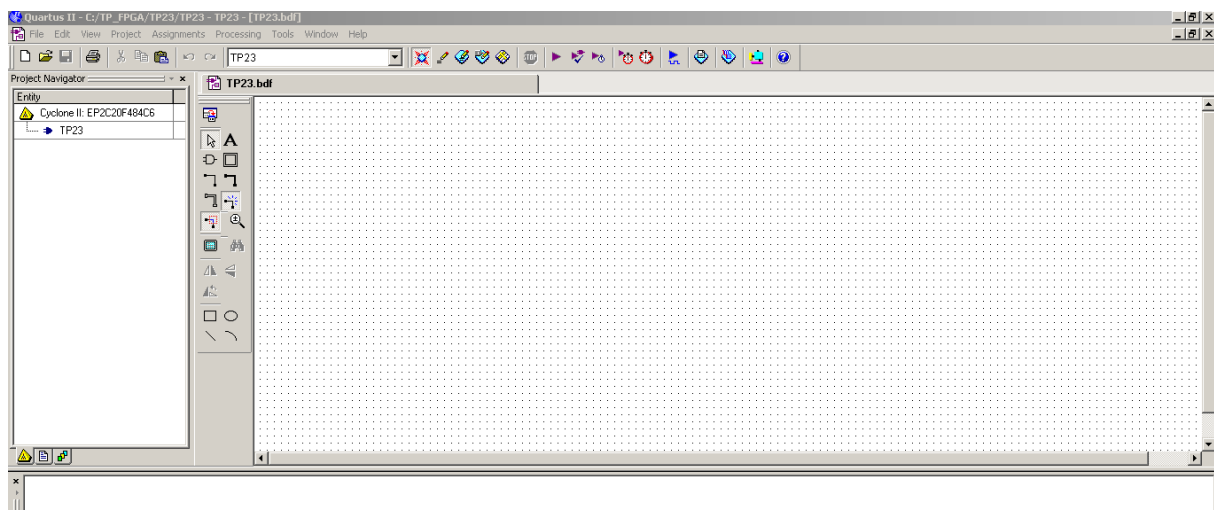
Question 3. Pour chaque calibre (fréquence f_1 , f_2 , f_3 et f_4), quelle sera la précision théorique de l'appareil. Cette estimation sera faite sur la base ou la plus petite valeur mesurable sur un calibre correspond à $N=1$.

Fréquence de référence	Précision théorique
$F_h = F_1 = 50\text{MHz}$	
$F_h = F_2 = 5\text{MHz}$	
$F_h = F_3 = 500\text{kHz}$	
$F_h = F_4 = 50\text{kHz}$	

Préliminaires

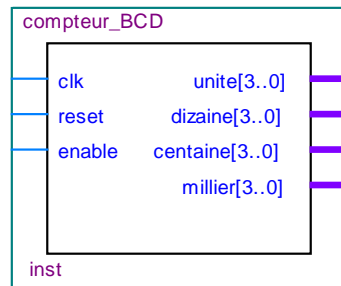
Pour ce TP, nous allons créer un nouveau projet.

1. Ouvrir le programme Quartus
2. Créer sur votre compte, un répertoire TP_FPGA
3. Créer un sous-répertoire TP23
4. Sous Quartus, créer un nouveau projet. Dans le menu file, sélectionner new projet wizard.
5. Remplissez les champs suivants
 - a. What is the working directories for this project ?
../TP_FPGA/TP23
 - b. What is the name of this project ?
TP23
 - c. What is the name of the top-level design entity for this project ...?
TP23
6. Page 3 du wizard
 - a. Dans la fenêtre Family, sélectionnez **Cyclone II**
 - b. Dans la liste des circuits disponibles, sélectionnez : **EP2C20F484C6**
7. Appuyer sur finish. Vous venez de créer votre premier projet Quartus.



IV. Compteur N

Le compteur N est un compteur décimal (BCD) dont les sorties évoluent entre 0 et 9999.



Son entité comprend trois entrées ; une horloge, un signal de remise à zéro du compteur et des sorties et un signal enable qui autorise le compteur de compter. Les quatre sorties sur 4 bits évoluent de 0 à 9 (compteur BCD) et donnent respectivement la valeur de N sur 4 digits (millier, centaine, dizaine et unité). Les évolutions des sorties seront conditionnées aux fronts montants de l'horloge.

Pour réaliser ce comptage entre 0 et 9999, 4 compteurs sont nécessaires. Un compteur unité, un compteur dizaine, un compteur centaine et un compteur millier. Le compteur dizaine est incrémenté de 1 lorsque le compteur unité passe de 9 à 0. Idem pour le compteur centaine qui s'incrémente lorsque les compteurs unité et dizaine valent 9. Le compteur millier est incrémenté lorsque les compteurs unités, dizaines et centaines sont tous à 9.

Le code VHDL ci-dessous donne l'entité et l'architecture à compléter de ce compteur N.

- Définir un nouveau fichier VHDL
 - Menu File/New et sélectionner le type VHDL
 - Sauver le fichier sous le nom compteurN.vhd
- Compléter l'architecture afin que le comportement du circuit respecte les spécifications fonctionnelles.
- Simuler et vérifier le comportement.
 - On testera que les sorties évoluent bien entre 0000 et 9999.

Question 4. Consigner dans votre rapport :

- a. Le code VHDL de l'architecture
- b. Les résultats de simulation.

Validation 1. Faites valider par l'enseignant dans la salle.


```
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all;
3  USE ieee.numeric_std.all;
4
5  ENTITY compteurN IS
6      PORT
7      (
8          --ENTREES
9          clk      : IN      STD_LOGIC;
10         reset    : IN      STD_LOGIC;
11         enable   : IN      STD_LOGIC;
12         --SORTIES
13         unite    : OUT      STD_LOGIC_VECTOR(3 DOWNTO 0);
14         dizaine  : OUT      STD_LOGIC_VECTOR(3 DOWNTO 0);
15         centaine : OUT      STD_LOGIC_VECTOR(3 DOWNTO 0);
16         millier  : OUT      STD_LOGIC_VECTOR(3 DOWNTO 0)
17     );
18 END;
19
20 ARCHITECTURE myarchitecture OF compteurN IS
21     SIGNAL count_u : UNSIGNED(3 DOWNTO 0);
22     SIGNAL count_d : UNSIGNED(3 DOWNTO 0);
23     SIGNAL count_c : UNSIGNED(3 DOWNTO 0);
24     SIGNAL count_m : UNSIGNED(3 DOWNTO 0);
25
26 BEGIN
27     unite <= count_u;
28     dizaine <= count_d;
29     centaine <= count_c;
30     millier <= count_m;
31
32     PROCESS (clk, reset)
33     BEGIN
34         IF reset = '1' THEN
35             count_u <= (OTHERS => '0');
36             count_d <= (OTHERS => '0');
37             count_c <= (OTHERS => '0');
38             count_m <= (OTHERS => '0');
39         ELSIF (clk'EVENT AND clk = '1') THEN
40             count_u <= count_u + 1;
41
42         A COMPLETER
43     END PROCESS
44 END myarchitecture;
```

V. Affichage et mémorisation

Cette partie concerne la mémorisation et l’affichage des valeurs en sorties du compteur N. La figure ci-dessous montre les connexions des blocs de mémorisation, d’affichage ainsi que la définition des broches du FPGA connectées aux afficheurs 7 segments.

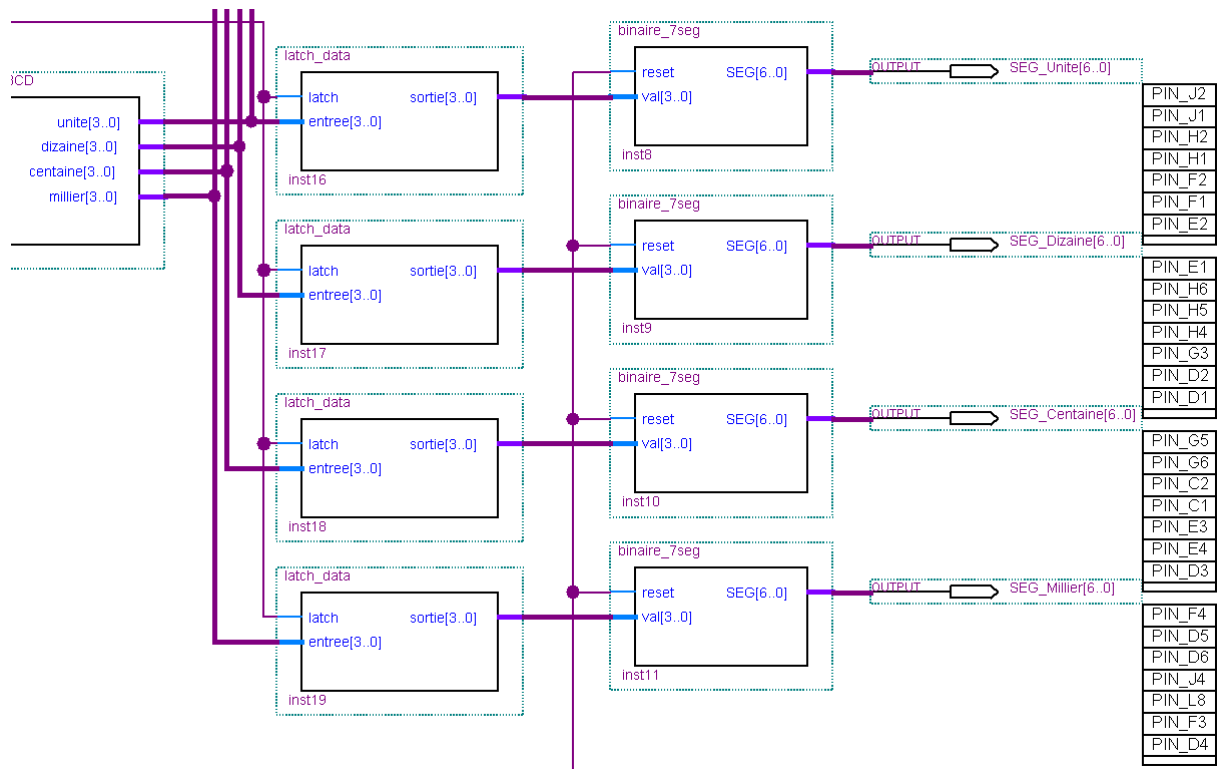


Figure 4

L'entité latch_data est un circuit latch 4 bits qui recopie la valeur de l'entrée définie sur 4 bits sur une sortie 4 bits lorsque le signal de commande latch est valide. La table de vérité de ce circuit est ci-dessous :

latch	sortie[3..0] à n+1	Etats
↑	entree[3..0]	Recopie
0	sortie[3..0] à n	Mémoire
1	sortie[3..0] à n	

- Créer un fichier VHDL (File/New)
- Enregistrer le sous le nom latch_data.vhd
- Ecrire en VHDL, l'entité et l'architecture du circuit latch_data
- Simuler et vérifier le comportement.

Question 5. Consigner dans votre rapport

- a. Les codes VHDL
- b. Les résultats de simulation.

Validation 2. Faire valider par l'enseignant

L'entité binaire_7seg est un circuit qui transcode la valeur de l'entrée 4 bits sur une sortie 7 bits pour afficher le résultat d'un des digits du compteur sur un afficheur 7 segments.

- Créer un fichier VHDL (File/New)
- Enregistrer le sous le nom binaire_7seg.vhd
- Ecrire en VHDL, l'entité et l'architecture du circuit latch_data
- Simuler et vérifier le comportement.

Question 6. Consigner dans votre rapport

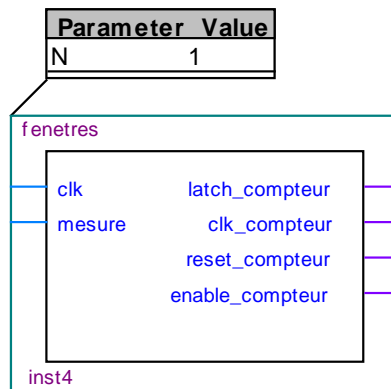
- c. Les codes VHDL
- d. Les résultats de simulation.

Validation 3. Faire valider par l'enseignant

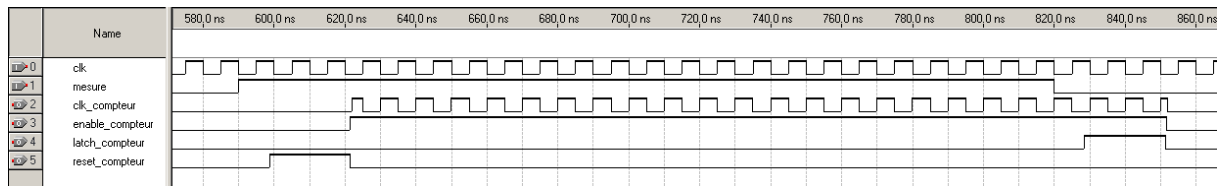
Afin de tester cette première partie, connecter le compteur, le circuit de mémorisation et l'affichage. On utilisera un bouton de type switch button comme signal de commande enable du circuit latch.

VI. Logique de comptage et fenêtre de mesure

On désire mesurer uniquement le nombre N sur la fenêtre de comptage. Pour cela, un bloc de fenêtrage doit être développé. L'entité fenêtrage prend en entrée, le signal de mesure à analyser et l'horloge de référence (une des quatre ; f1, f2, f3 ou f4). Les sorties du circuit, sont la commande de mémorisation du circuit de mémorisation, l'horloge du compteur, la commande de remise à zéro du compteur et l'autorisation de compter.



Les chronogrammes de ce circuit sont :



Lorsque la fenêtre de comptage passe de 0 à 1, on commence par remettre à zéro le compteur, puis on autorise le compteur à compter. Avant que la fenêtre de mesure repasse à l'état bas, on mémorise la valeur du compteur en activant le signal latch_compteur. Le paramètre generic N, permet de définir la largeur temporelle des signaux enable_compteur et latch_compteur, en nombre de périodes d'horloge.

Question 7. Analyser le code VHDL donné

Question 8. Faite un organigramme correspondant au fonctionnement du circuit

- Simuler et vérifier le comportement.

Question 9. Consigner dans votre rapport, les développements et les résultats de simulation.

Validation 4. Faite valider par l'enseignant

Afin de tester cette première partie, nous allons connecter les circuits fenêtre, compteur, le circuit de mémorisation et l'affichage. Pour effectuer cette description schématique :

- Créer un fichier BDF (File/New)
- Enregistrer le sous le nom TP23.bdf
- Créer les symboles des composants, compteurN, latch_data et binaire_7seg
- Relier les composants entres eux conformément à la figure 4
- Synthétiser le fichier BDF
- Simuler et vérifier le comportement.

Question 10. Consigner dans votre rapport les résultats de simulation.

On vérifiera pratiquement le bon fonctionnement de ces quatre circuits.

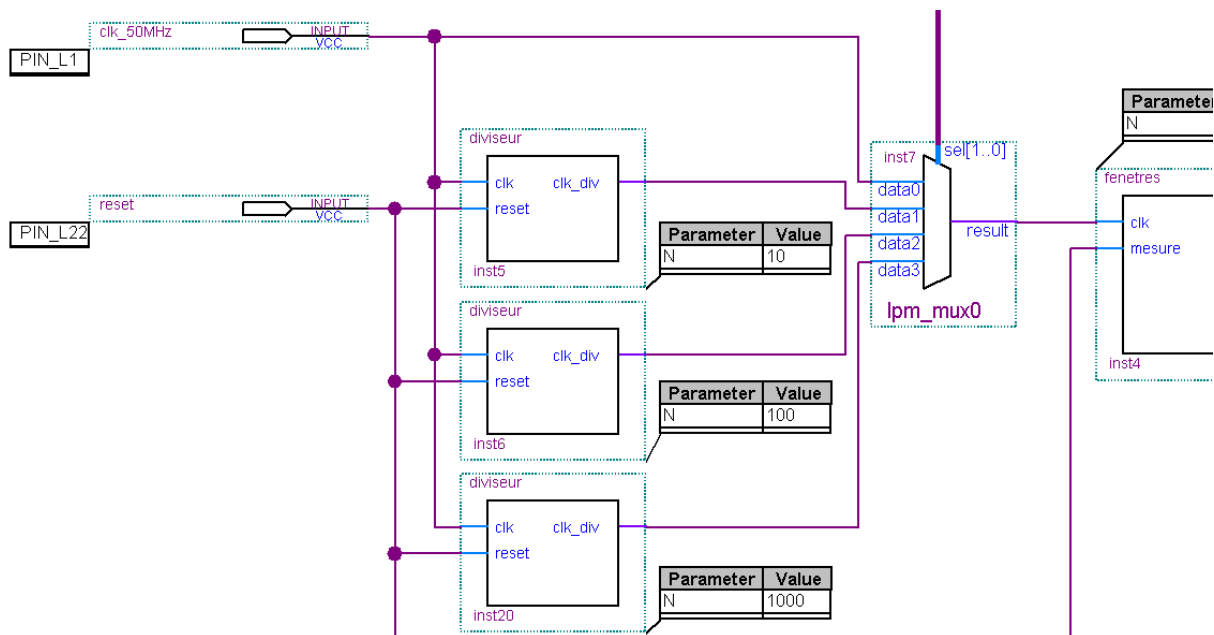
Validation 5. Faite valider par l'enseignant

VII. Base de temps

Le fréquencemètre intègre quatre calibres définis au moyen de quatre horloges. Ces horloges (50MHz, 5MHz, 500kHz et 50kHz) sont produites au moyen de divisions fréquentielles d'une horloge de référence à 50MHz. La figure ci-dessous illustre ce principe. La sélection d'une des quatre fréquences se fait au moyen d'un multiplexeur de 4->1. Ce dernier sera conçu à partir du composant lpm_mux disponible dans la librairie du constructeur.

Dans le fichier TP23.bdf

- Insérer un composant multiplexeur via la librairie lpm_mux
- Définir les paramètres du multiplexeur (4 entrées sur 1 bit)
- Relier le composant mux au reste du circuit conformément à la figure ci-dessous



L'entité diviseur est un compteur (diviseur de fréquence) générique. Il permettra de diviser le signal d'horloge par un facteur N programmable. Le rapport cyclique du signal de sortie clk_div est de 50%. Les fréquences f_1 , f_2 , f_3 et f_4 seront obtenues par division du quartz à 50MHz de la carte FPGA. Les rapports 10, 100 et 1000 permettront d'obtenir des fréquences de 5MHz, 500kHz et 50kHz de références.

- Créer un fichier VHDL (File/New)

- Enregistrer le sous le nom diviseur.vhd
- Ecrire en VHDL, l'entité et l'architecture du circuit diviseur
- Simuler et vérifier le comportement.

Question 11. Consigner dans votre rapport

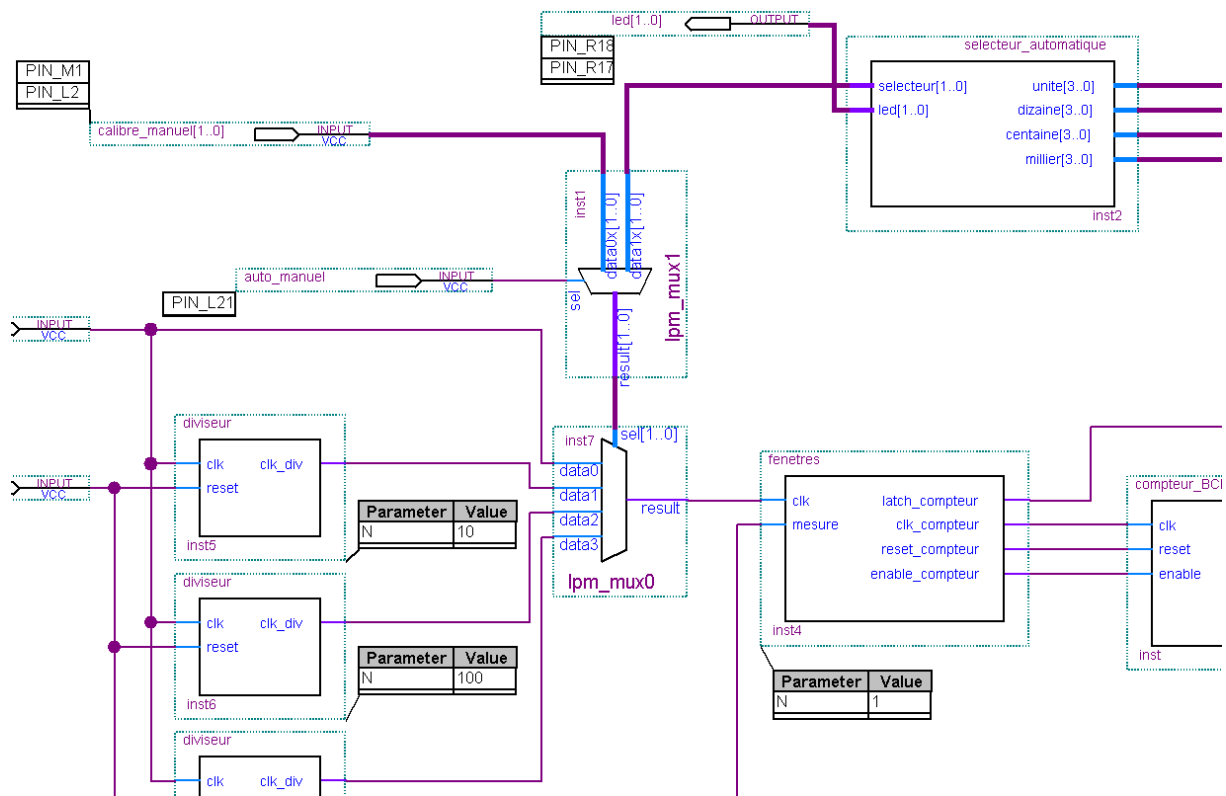
- a. Les codes VHDL
- b. Les résultats de simulation.

Validation 6. Faites valider par l'enseignant

Au moyen de cette entité et des composants disponibles dans la librairie constructeur, réaliser l'architecture de la figure ci-dessus. Connecter cette partie aux précédents développements.

VIII. Mesures expérimentales

De manière à sélectionner le calibre disponible pour déterminer la fréquence du signal à mesurer, on désire que le fréquencemètre possède deux modes. Le premier manuel laisse la liberté à l'utilisateur de choisir l'une des quatre fréquences de références disponibles, tandis que le second mode automatique, utilise les sorties du compteur pour déterminer si N est représenté avec suffisamment de digits.



Dans un premier temps, on ne s'occupera que de la partie manuelle. On utilisera pour cela les switch SW8 et SW9 pour commander le multiplexeur lpm_mux0 (sélecteur de la fréquence).

Après avoir câbler votre montage, blocs par blocs, on se propose de mesurer les performances de ce montage vis à vis du cahier des charges initial. Pour cela, il est nécessaire d'injecter un signal dont on va mesurer la fréquence. Une carte fille comprenant un connecteur BNC est connectée sur la carte FPGA. Le signal à analyser est disponible sur la broche **PIN_A13**

- Modifier votre fichier BDF pour injecter le signal disponible sur la carte fille
- Faire valider par l'enseignant
- Synthétiser et programmer la carte FPGA
- Injecter via le GBF **un signal TTL**

Validation 7. Faire valider par l'enseignant

Répondre aux questions suivantes :

- Question 12.** Comment peut-on étalonner ce montage ?
- Question 13.** Quelle est la plage de fréquence mesurable précisément ?
- Question 14.** Quelles sont les incertitudes de mesure ?

Question 15. Quels sont les inconvénients de ce montage ?

Question 16. Remplir le tableau de mesure ci-dessous

Fréquences	N théorique	N pratique
5Hz		
50Hz		
100Hz		
500Hz		
1kHz		
5kHz		
10kHz		
50kHz		
100kHz		
500kHz		
1MHz		
5MHz		

Question 17. Tracer sur un même graphique, les courbes $N_{\text{théorique}}=f^{\circ}(\text{fréquence})$ et $N_{\text{expérimental}}=g^{\circ}(\text{fréquence})$.

Question 18. Conclusion

Partie optionnelle

L'architecture du fréquencemètre proposée, permet de mesurer la fréquence à un facteur d'échelle 2 près. Si on veut une mesure directe, quelles sont les modifications à apporter ?

- Proposer les modifications à apporter
- Test et valider votre solution si le temps vous le permet.

Dans un second temps, on désire réaliser la partie automatique du sélecteur de calibre. On utilisera pour cela un second multiplexeur `lpm_mux1` qui prendra en entrée, soit les commandes manuelles de sélection du calibre (les switch SW8 et SW9), soit les sorties du circuit `selecteur_automatique`. La commande du multiplexeur `lpm_mux1` sera le switch SW1. pour commander le multiplexeur `lpm_mux0` (sélecteur de la fréquence).



Le circuit `selecteur_automatique` calcule le meilleur calibre possible en fonction du N en sortie du compteur BCD. Les sorties sont, la valeur du sélecteur de calibre et l’affichage sur 2 leds de la valeur du sélecteur.

- Question 19.** Proposer un algorithme
- Question 20.** Proposer une architecture VHDL de cette entité
- Question 21.** Simuler et vérifier le comportement.
- Question 22.** Consigner dans votre rapport, les développements et les résultats de simulation.
- Question 23.** Insérer et tester pratiquement les performances du fréquencemètre à calibre automatique