

# VHDL - FIR

## Approche Système de la Conception

Printemps 2012

### 1. Filtre à Réponse Impulsionnelle Finie (FIR)

On désire réaliser un système capable de calculer un filtre passe-bas à l'aide d'un opérateur type FIR. L'équation de ce type de filtre est

$$y(n) = \sum_{k=0}^{k=M-1} x(n-k) * a(k)$$

Dans cette équation,  $x(n-k)$  représente l'échantillon d'un signal  $x$  à l'instant  $(n-k)$ ,  $a(k)$  représente le  $k^{ième}$  coefficient,  $M$  représente le nombre de coefficients du filtre et  $y(n)$  l'échantillon filtré à l'instant  $n$ .

Afin de calculer ce filtre on réalise un système visible sur la figure 1.

Dans ce système il y a 2 mémoires pour stocker les coefficients et les échantillons, un compteur par  $M$ , un opérateur de calcul du FIR et une machine à états pour gérer l'ensemble. Afin d'assurer un bon fonctionnement de l'ensemble, une machine à états est réalisée afin d'effectuer les deux phases nécessaires au filtrage :

- La phase d'initialisation des coefficients
  - La phase de calcul à proprement parler
- (a) Phase d'Initialisation : pour initialiser les coefficients, il faut respecter le chronogramme 1, visible sur l'annexe
- i. Expliquez qualitativement la phase d'initialisation
  - ii. A l'aide du chronogramme 1, déduisez une fonction combinatoire simple de WRC
  - iii. A l'aide du chronogramme 1, déduisez une fonction combinatoire simple de HC
  - iv. Indiquez sur la ligne Etats du chronogramme 1, les différents états de la phase d'initialisation pour une machine de Moore.
  - v. Indiquez les entrées et les sorties de la machine à états
  - vi. Réalisez le diagramme des transitions
  - vii. Décrivez à l'aide de l'outil de saisie de diagrammes d'états cette machine à états

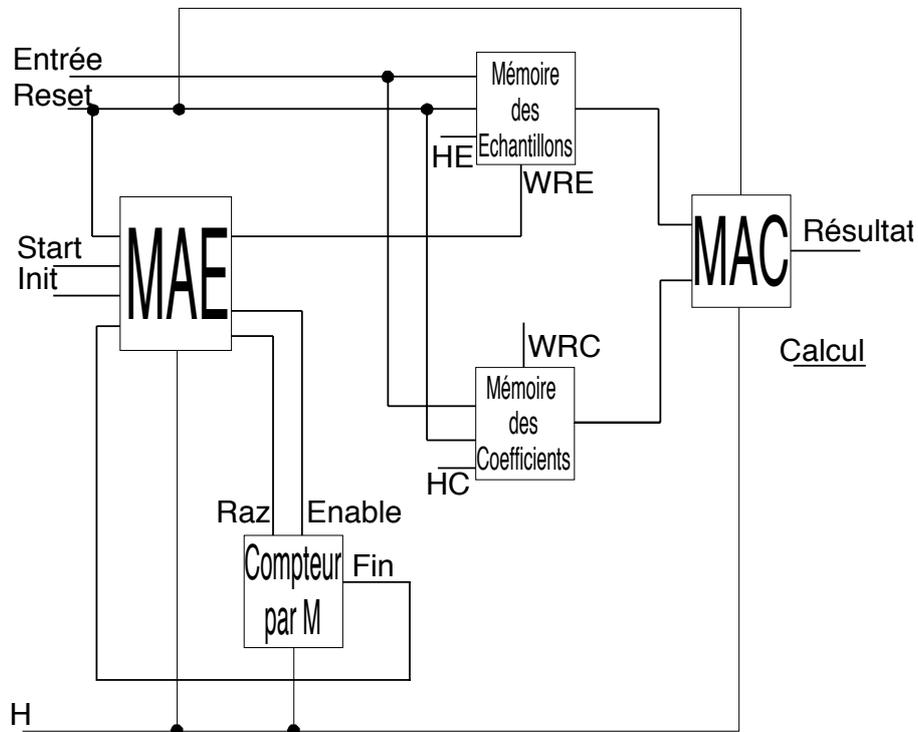


Figure 1: Synoptique du Système de Calcul du FIR

- viii. Vérifiez le code VHDL généré par l'outil
  - ix. Simulez et vérifiez le fonctionnement de cette machine .
- (b) Phase de Calcul : pour effectuer la phase de calcul il faut respecter le chronogramme 2, visible sur l'annexe
- i. Expliquez qualitativement la phase de calcul
  - ii. A l'aide du chronogramme 2, déduisez une fonction combinatoire pour HE
  - iii. A l'aide du chronogramme 2, déduisez une fonction combinatoire pour HC
  - iv. Indiquez sur la ligne Etats du chronogramme 2, les différents états de la phase de calcul pour une machine de Moore.
  - v. Indiquez les entrées et les sorties de la machine à états
  - vi. Réalisez le diagramme des transitions
  - vii. Décrivez à l'aide de l'outil de saisie de diagrammes d'états cette machine à états
  - viii. Simulez et vérifiez le fonctionnement de cette machine.
- (c) Réalisez un diagramme de transition d'une machine à états de Moore capable de réaliser les 2 phases
- (d) Que pensez-vous de la fonction WRC déduite précédemment ? Qu'en concluez-vous ?

- (e) Décrivez à l'aide de l'outil de saisie de diagrammes d'états cette machine à états
- (f) Simulez et vérifiez le fonctionnement de cette machine.
- (g) Afin de pouvoir avertir qu'un nouvel échantillon est filtré, un signal Calcul passe à 1 pendant 3 périodes d'horloges à la fin des calcul. Modifiez votre machine à états pour pouvoir générer ce signal.
- (h) Décrivez à l'aide de l'outil de saisie de diagrammes d'états cette machine à états
- (i) Simulez le fonctionnement de cette machine et vérifiez son bon fonctionnement
- (j) Faites la synthèse de la machine sur une cible FPGA Altera APEX 20K
- (k) Visualisez les schémas en technologie générique et en technologie fabricant

## 2. Mémoire

Afin de pouvoir stocker les coefficients et les échantillons, nous avons besoin d'une mémoire ayant un adressage de type circulaire, à l'instar des DSP. Cette mémoire est synchrone, elle possède en interne un compteur d'adresse nommé `ComptAdr`. Elle a en entrée une donnée sur 8 bits, un signal de remise à zéro de `ComptAdr`, un signal d'horloge et un signal de lecture-écriture. En sortie cette mémoire fournit une donnée sur 8 bits. Son fonctionnement est le suivant:

- Quand le signal de remise à zéro vaut 0, la valeur de `ComptAdr` est 0
- Quand le signal de remise à zéro vaut 1 alors
  - Si le signal de lecture-écriture vaut 0 alors
    - \* à chaque front montant de l'horloge la donnée en entrée est stockée à l'adresse fournie par le contenu de `ComptAdr`, puis `ComptAdr` est incrémenté de 1.
  - Si le signal de lecture-écriture vaut 1 alors
    - \* à chaque front montant de l'horloge la donnée en sortie prend la valeur de la donnée stockée à l'adresse fournie par le contenu de `ComptAdr`, puis `ComptAdr` est incrémenté de 1.

- (a) Décrivez en VHDL une mémoire de  $M$  octets qui possède en entrée un signal `donnee` sur 8 bits, un signal `reset` sur 1 bit, un signal d'horloge `H` sur 1 bit et un signal de lecture-écriture `WR` sur 1 bit et en sortie un signal `result` sur 8 bits. Cette mémoire doit vérifier le comportement décrit plus haut.
- (b) Simulez et vérifiez le fonctionnement de cette unité.

## 3. Unité MAC (Multiplication ACumulation)

Afin de réaliser le FIR, il est nécessaire d'avoir une unité MAC permettant de réaliser le calcul du produit et de l'accumulation. Cette unité est représentée par la figure 3.

- (a) Décrivez en VHDL une unité MAC correspondant au synoptique de la figure 3.

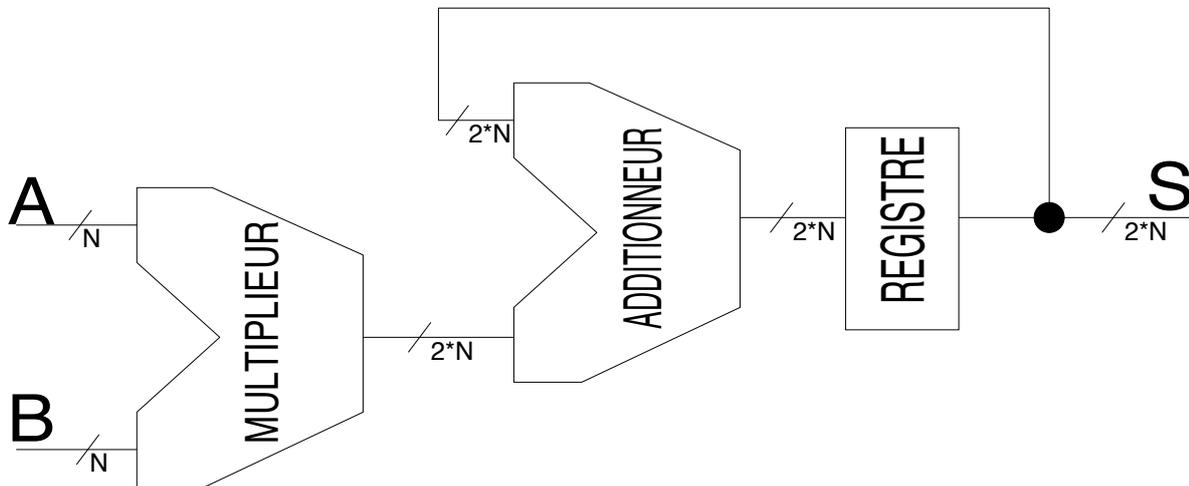


Figure 2: Synoptique de l'unité MAC

(b) Simulez et vérifiez le fonctionnement de cette unité.

#### 4. Compteur

Pour le bon déroulement du filtrage il est nécessaire d'avoir un compteur par M. Ce compteur possède, une entrée *Enable* d'autorisation de comptage, une entrée *Raz* de remise à zéro et une sortie *Fin* qui est à 1 quand le compteur a atteint la valeur M.

(a) Décrivez en VHDL un compteur par M ayant les entrées et sortie pré-citées.

(b) Simulez et vérifiez le fonctionnement de cette unité.

#### 5. Assemblage

Une fois toutes les unités décrites, il nous faut les assembler les différents composants précédemment décrit.

(a) Décrivez en VHDL un composant qui réalise le filtre FIR tel que décrit par la figure 1.

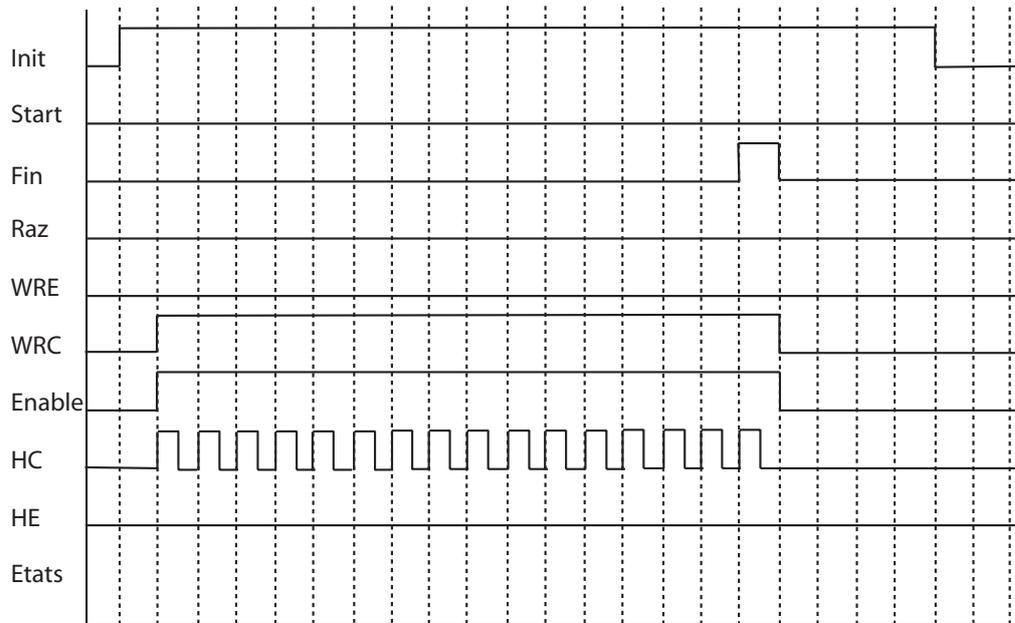
(b) Simulez et vérifiez le fonctionnement de cette unité..

#### 6. Test

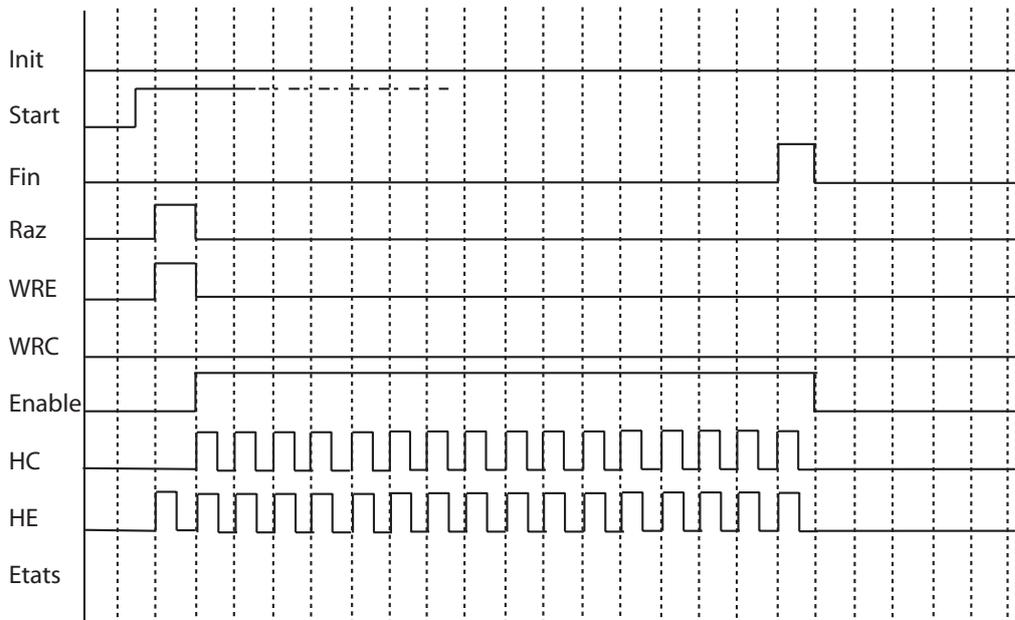
Afin de pouvoir tester votre composant, on prendra comme valeur numérique  $M=16$ .

Annexe

Nom :  
Prénom :



Chronogramme 1



Chronogramme 2