

Master 2 EEA SME, Systèmes et Microsystèmes Embarqués

Par :

Kossi Pascal SEWODA, Kahina ACHARI

Université Toulouse III Paul Sabatier

Encadre par : Thierry PERISSE et Pedro CARVALHO-MENDES

Date : 29 novembre 2021

**RAPPORT DE BE VHDL**

Année universitaire : 2021-2022

Barre franche

Département EEA - Faculté Sciences et Ingénierie

Table des matières

[I. TP de base 2](#_Toc90205220)

[1. Porte ET 2](#_Toc90205221)

[2. Additionneur 3](#_Toc90205222)

[3. Mini projet 3](#_Toc90205223)

[4. Circuit de génération de PWM 3](#_Toc90205224)

[II. BE : Interfaces pilote de barre franche 3](#_Toc90205225)

[5. Conception d’une fonction simple 3](#_Toc90205226)

[6. Conception d’une fonction compliqué 3](#_Toc90205227)

Introduction

1. TP de base

# Porte ET

Dans premier projet il s’agit de prendre en main l’environnement de développement Quartus qui sera utilisé tout au long de ce TP d’électronique numérique avec du VHDL.

Le VHDL est un langage de description matériel qui permet défaire de la synthèse logique sur FPGA.

La création de la porte ET, comme n’importe quelle composant en VHDL, est décomposé en deux parties : l’entité et l’architecture.

L’entité : l’entité représente la vue boîte noire du composant ou du système avec ses ports d’entrées sorties.

L’architecture : Représente la façon dont le composant ou le système est implémenté. Un système peut avoir plusieurs architectures.

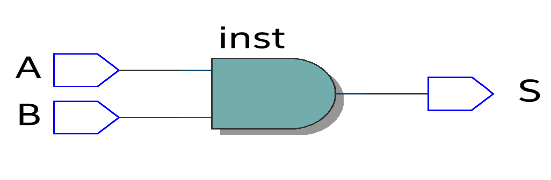
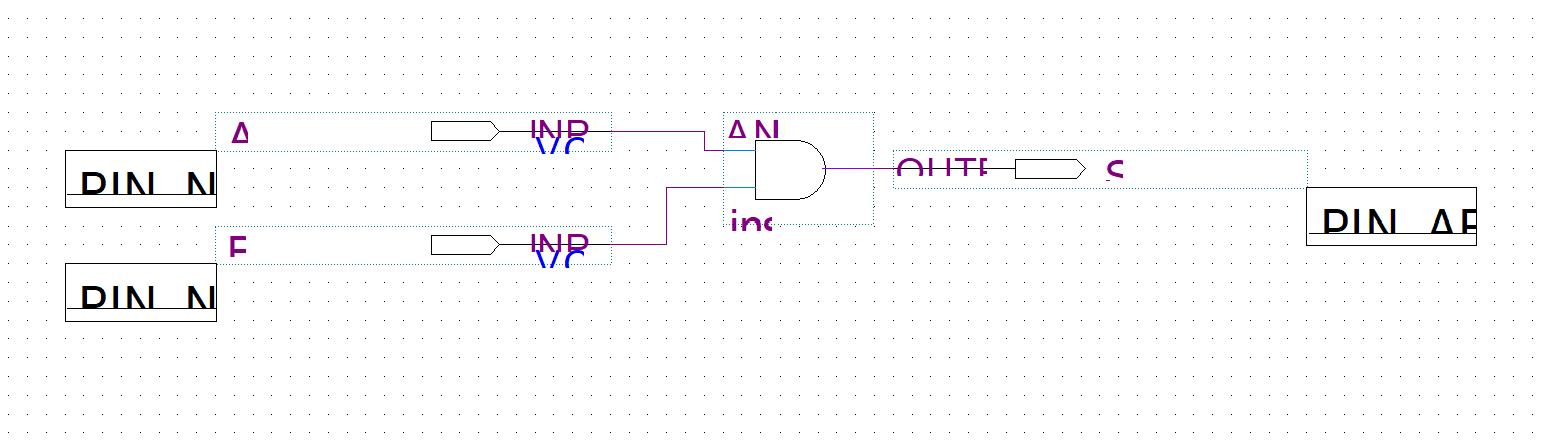
Une image contenant texte, horloge

Description générée automatiquement

Figure 1: Entité de la porte ET

Pour réaliser cette porte ET on utilise l’outil de saisie graphique de Quartus.

Figure 2: Saisie sur Quartus et schéma électrique résultant



Simulation fonctionnelle : Ce type de simulation permet de ne pas tenir compte des temps de propagation et de simuler seulement la fonction.

Une image contenant regardant, fixant

Description générée automatiquement

Figure 3: Résultat de simulation de la porte ET

La simulation temporelle : Elle permet de tenir compte des temps de propagation des composants matérielles.

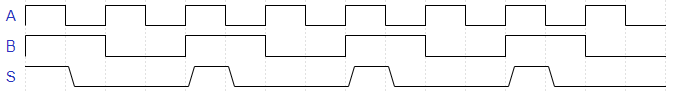


Figure 4: Simulation fonctionnelle

# Additionneur

Il s’agit de réaliser trois additionneurs (2 mots 1 bits, 2 mots 3 bits et 2 mots 5 bits) en utilisant le langage de description VHDL et aussi l’outil de saisie graphique disponible sur Quartus.

**Additionneur 2x1 bit**

Une image contenant carré

Description générée automatiquement

Figure 5 : Vue boite noire

Implémentation avec du VHDL

--Debut de declaration de l'entite

entity ent\_add1 is

port(

--Entrée logique

A: in std\_logic ;

B: in std\_logic ;

Cin: in std\_logic ;

--Sortie logique

Cout: out std\_logic ;

S: out std\_logic );

end;

--Fin de declaration de l'entite

--Début de la desciption de l'archictecture

architecture arch\_add1 of ent\_add1 is

begin

-- mise en oeuvre de l'architecture de l'additionneur 2x1 bit par des équations

S <= (A xor B )xor Cin;

Cout <= (A and Cin) or (B and Cin) or (A and B);

end arch\_add1;

--Fin de la desciption de l'archictecture

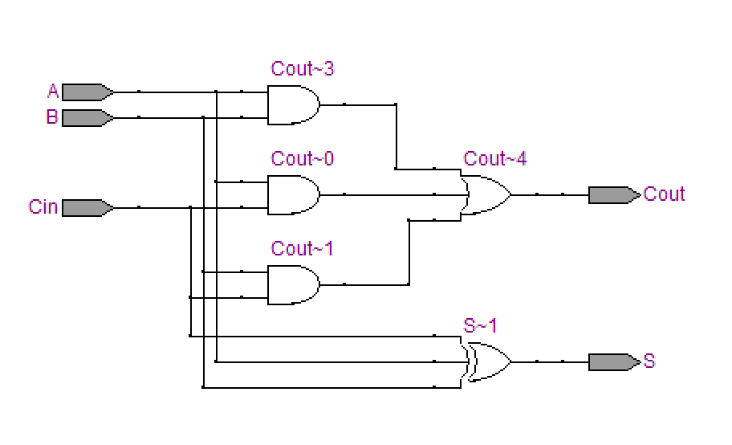


Figure 6: Circuit réalisé par Quartus

**Additionneur 2x3 bits**

Sur ce même principe on à réaliser l’additionneur 2x3bits en utilisant 3 additionneurs 2x1bit montés en cascade en utilisant l’outil graphique.

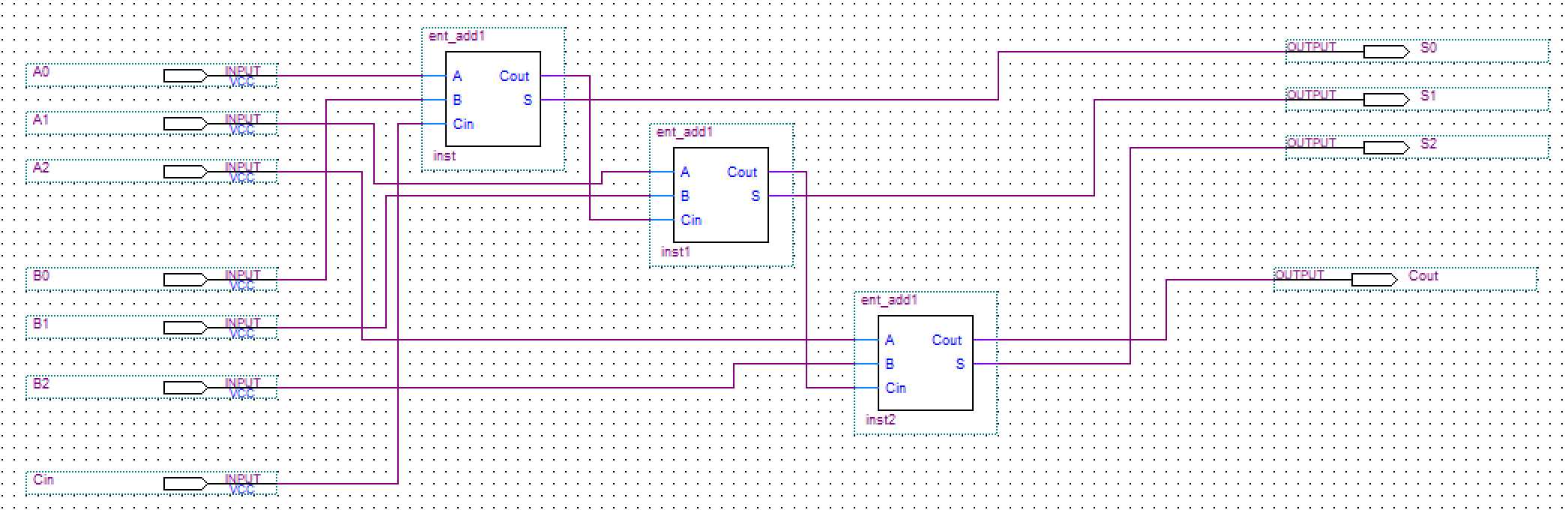


Figure 7: ADD2x3bits avec des ADD2x1

**Additionneur 2x5 bits**

Pour réaliser l’additionneur ADD2x5bits on utilise une autre approche qui consiste à utiliser des vecteurs sur 5 bits en VDHL, à l’aide de la bibliothèque :

use ieee.numeric\_std.all;

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

use ieee.std\_logic\_unsigned.all;

use ieee.numeric\_std.all;

entity add5\_q9 is

port(

-- Vector

a, b: in std\_logic\_vector(4 downto 0);

Cin: in std\_logic ;

s: out std\_logic\_vector(4 downto 0);

Cout: out std\_logic );

end;

architecture arch\_add5 of add5\_q9 is

signal result: std\_logic\_vector(5 downto 0);

begin

-- Mise en oeuvre de l'architecture de l'additionneur 5 bits

result <= '0'&a + b + cin;

s <= result(4 downto 0);

cout <= result(5);

end arch\_add5;

Schéma résultant (voir figure 8 )

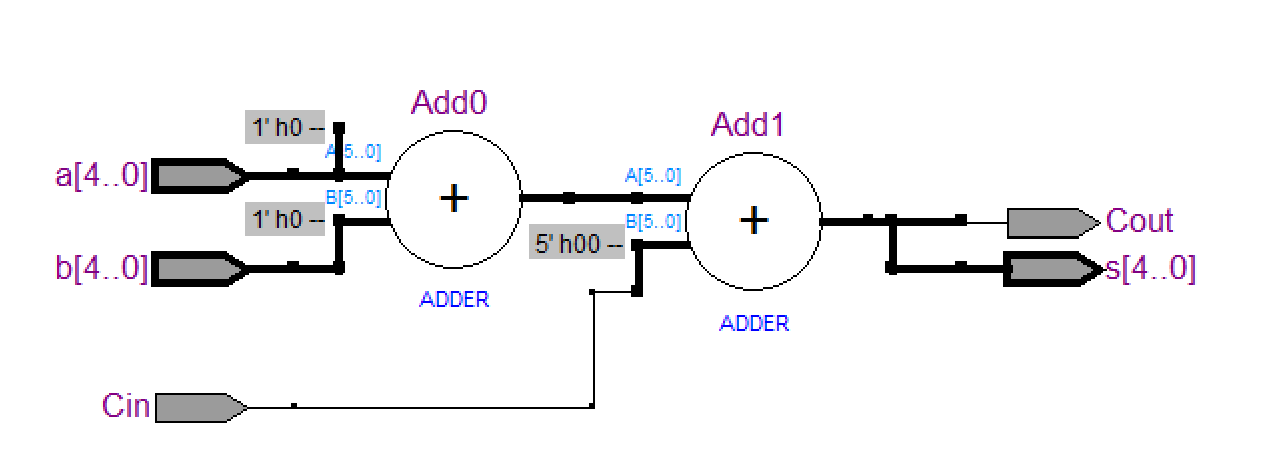


Figure 8. Schéma de connexions réalisé par Quartus

**Simulation**

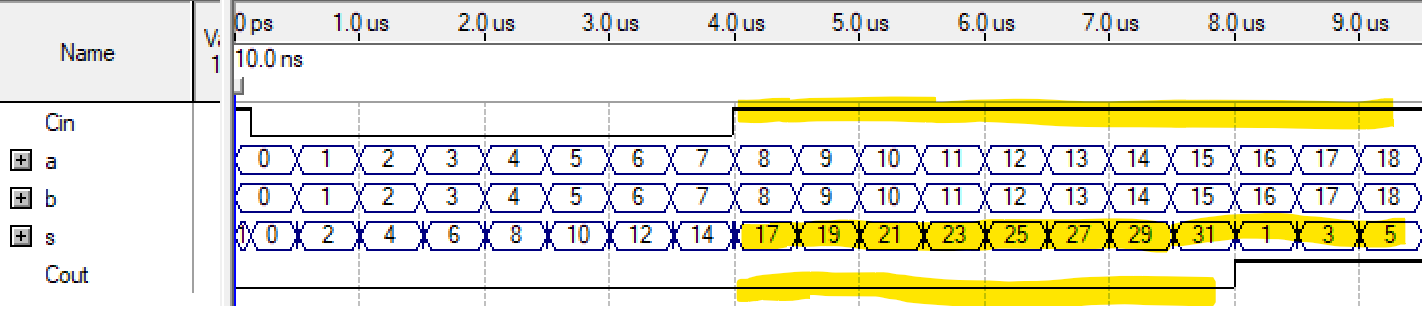


Figure 9: Simulation de ADD2x5bits

# Mini projet

Dans ce mini projet il s’agit de réaliser un compteur BCD comptant des secondes en utilisant l’oscillateur interne du DE2 NANO et afficher le résultat de comptage sur un afficheur 7-segments.

Il s’agit concrètement de réaliser :

* Diviseur de fréquence
* Un compteur BCD
* Décodeur 7-segments

La réalisation de ce mini projet a permis de découvrir de nouveau concept du langage VHDL :

* La création et l’instanciation des composants pour réaliser une tâche
* L’utilisation d’un process
* L’utilisation des boucles et de certaines instructions combinatoires.

La figure 9 montre le schéma fonctionnel du système.

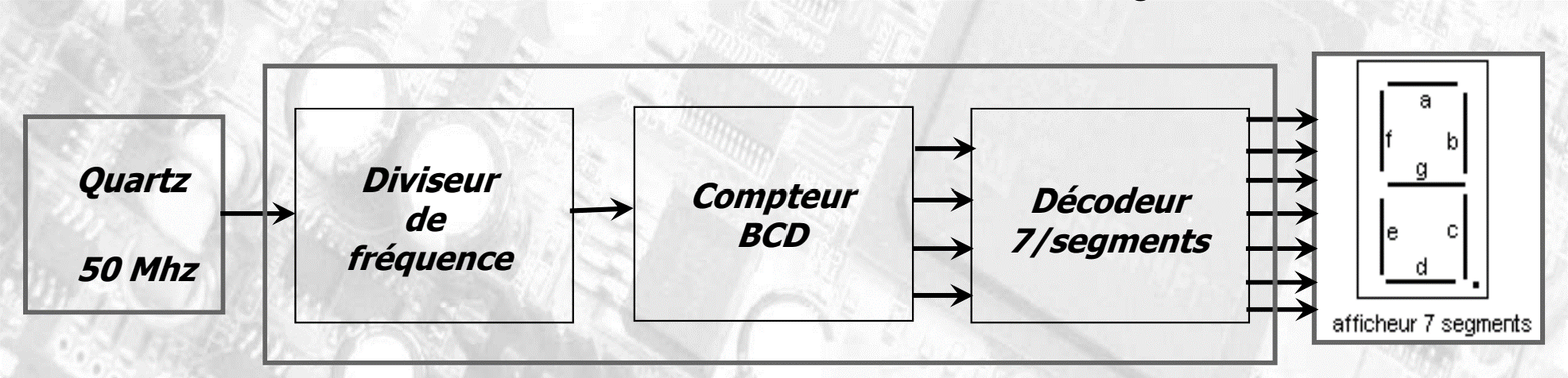


Figure 10: Schéma fonctionnel de l'ensemble

Le quartz utilisé est celui présent sur la carte DE2 en salle de TP.

Diviseur de frequence permet d’obtenir une fréquence de 1Hz correspondant à la seconde à partir de l’oscillateur interne de 50MHz. Voir le schéma fonctionnel la figure 11.

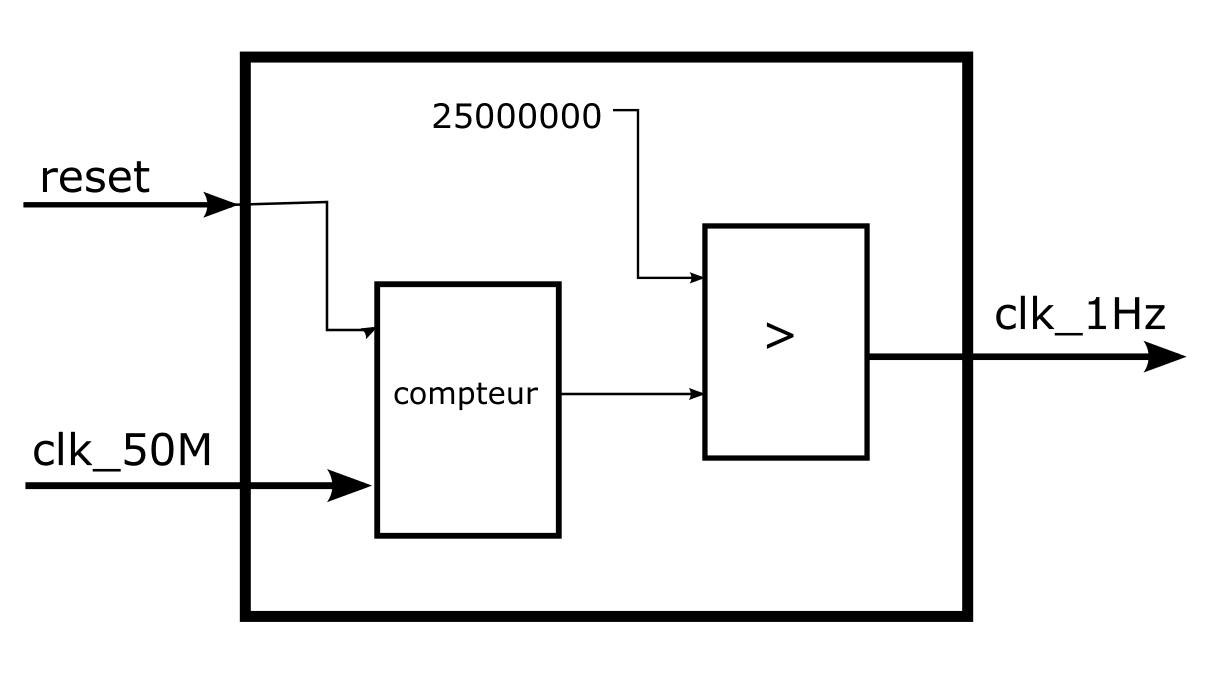


Figure 11: Schéma fonctionnel du diviseur

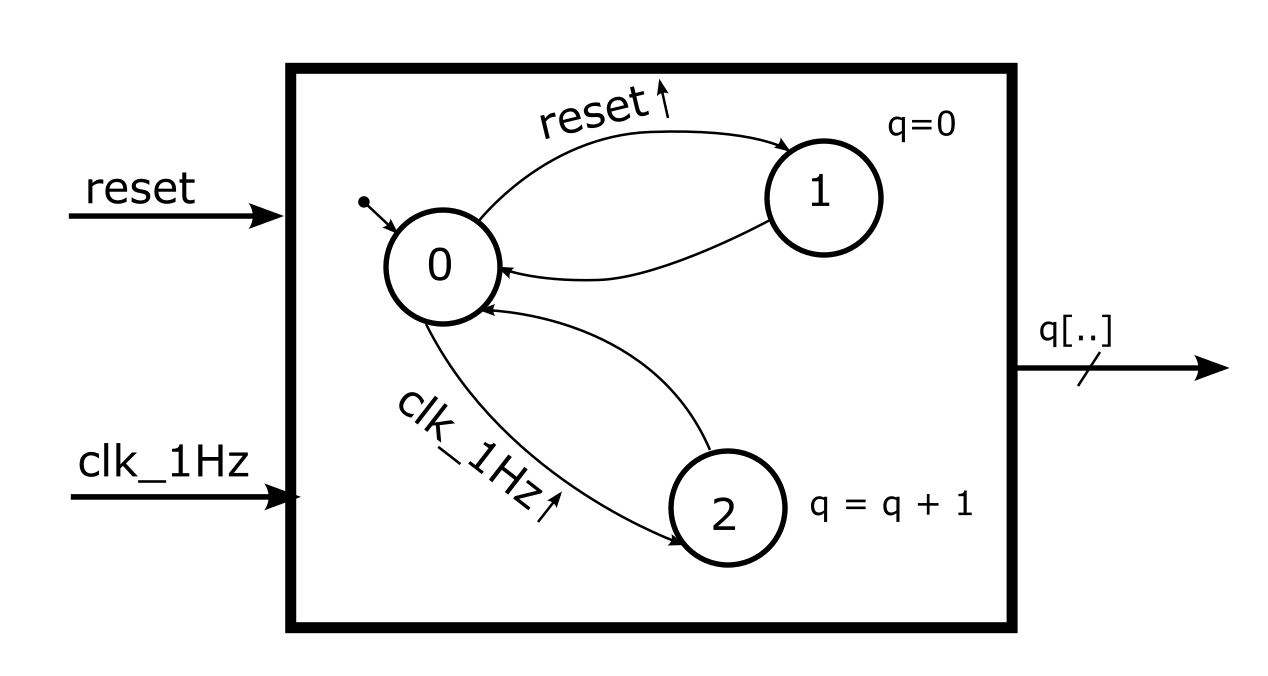


Figure 12: Principe de fonctionnement du compteur

Le décodeur est un système combinatoire qui permet d’associer à chaque valeur de q une lettre [abcdefg] avec un multiplexeur.

# Circuit de génération de PWM

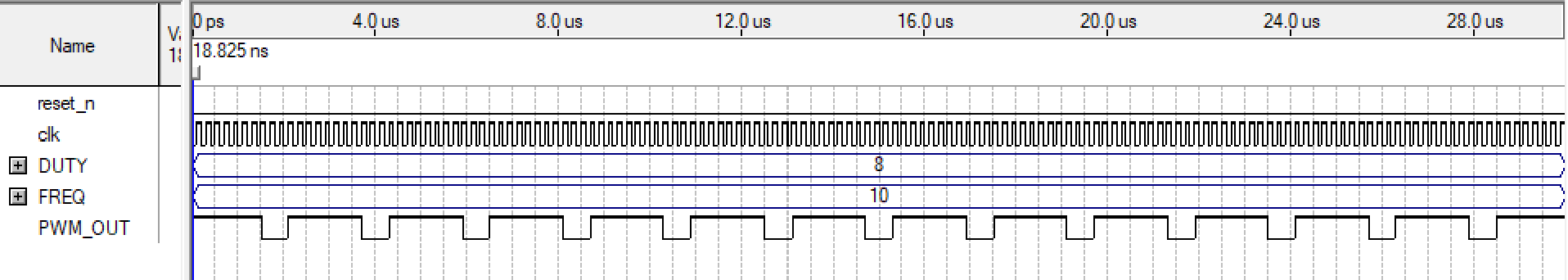


Figure 13: Simulation PWM

1. BE : Interfaces pilote de barre franche

L’interface de barre franche est constituée de hardware et de software permettant de remplir chacune une fonction particulière. Dans ce BE il s’agit de concevoir quelques fonctions de faisant partie de l’interface de pilote de barre franche. Une fonction dite simple et une fonction complexe.

# Conception d’une fonction simple

## Spécifications de l’anémomètre

Le système doit être capable de mesurer une frequence entre 0 et 250Hz

Le temps de réponse du système doit être maximum de 1 seconde.

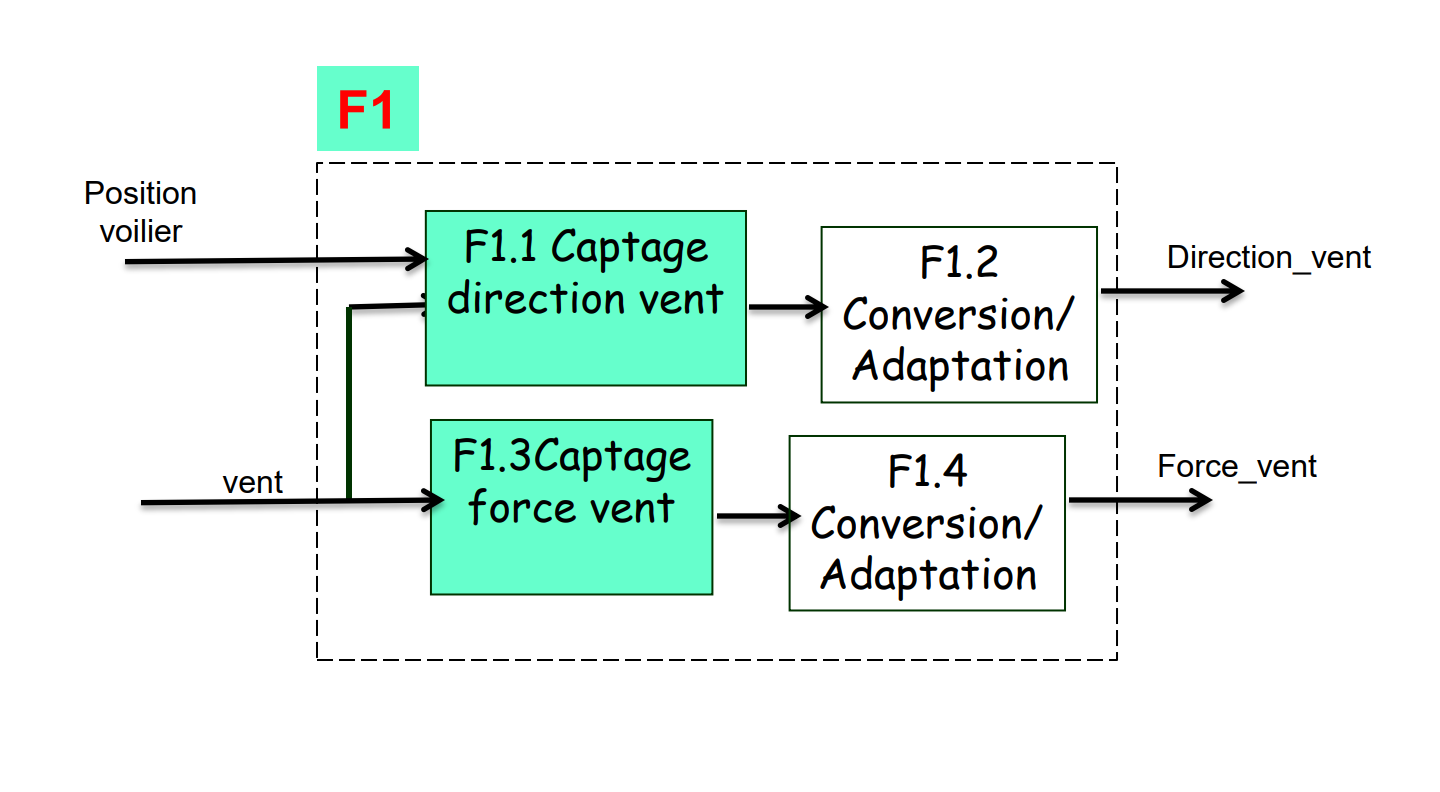
Les données doivent être dans un registre config et data

Communiqué en utilisant le bus ava

## Description

Notre choix de fonction simple s’est porté sur l’anémomètre. Cette fonction permet de mesurer la vitesse du vent entre 0 et 250km/h en mesurant la frequence issu d’un capteur de vent. Cette fréquence varie entre 0 et 250Hz. La valeur de 0 et 250km/h correspondent respectivement à la fréquence de 0 et 250Hz.

La fonction doit permettre de mesurer cette fréquence et renvoyé la valeur vers le bus Avalon pour réaliser des asservissements.



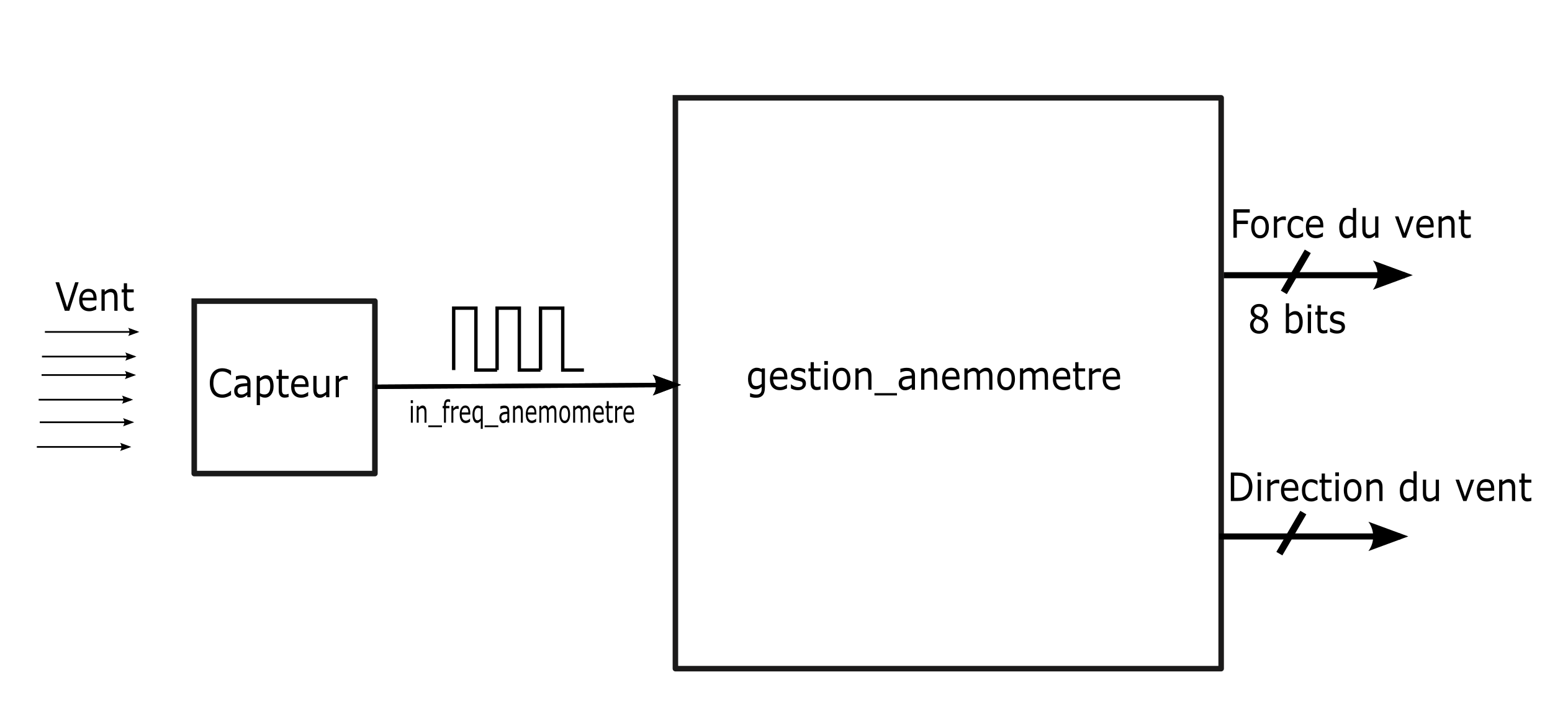


Figure 14: Schéma fonctionnel simplifié de l’anémomètre

Principe de mesure : Principe de la mesure de in\_freq\_anemo basé sur le comptage des fronts montants de in\_freq\_anemo pendant une période de 1s. Ce qui permet d’avoir directement la fréquence. Voir figure 15.

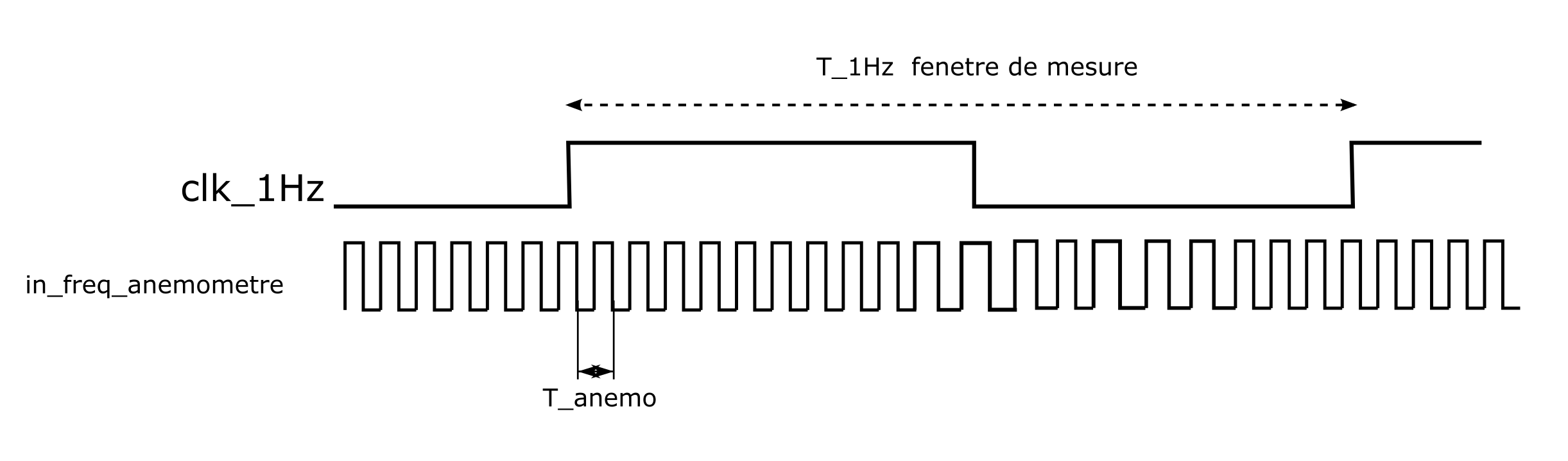


Figure 15: Principe de mesure in\_freq\_anemometre

**Description de l’interface**

Clk\_50M : Horloge de 50MHz présente sur la carte DE2 NANO. Elle permet de piloter tous les éléments du système.

In\_freq\_anemometre : la fréquence à mesurer

Raz\_n : actif à l’état haut. Permet de réinitialiser le système.

Continu : Anémomètre réalise des mesures de façon continue lorsque actif.

Start\_stop : Permet de lancer une mesure lorsqu’on n’est pas en mode continu.

Data\_anemometre : Contient les données de mesures sur 8bits.

Data\_valid : Actif lorsqu’une donnée est disponible.

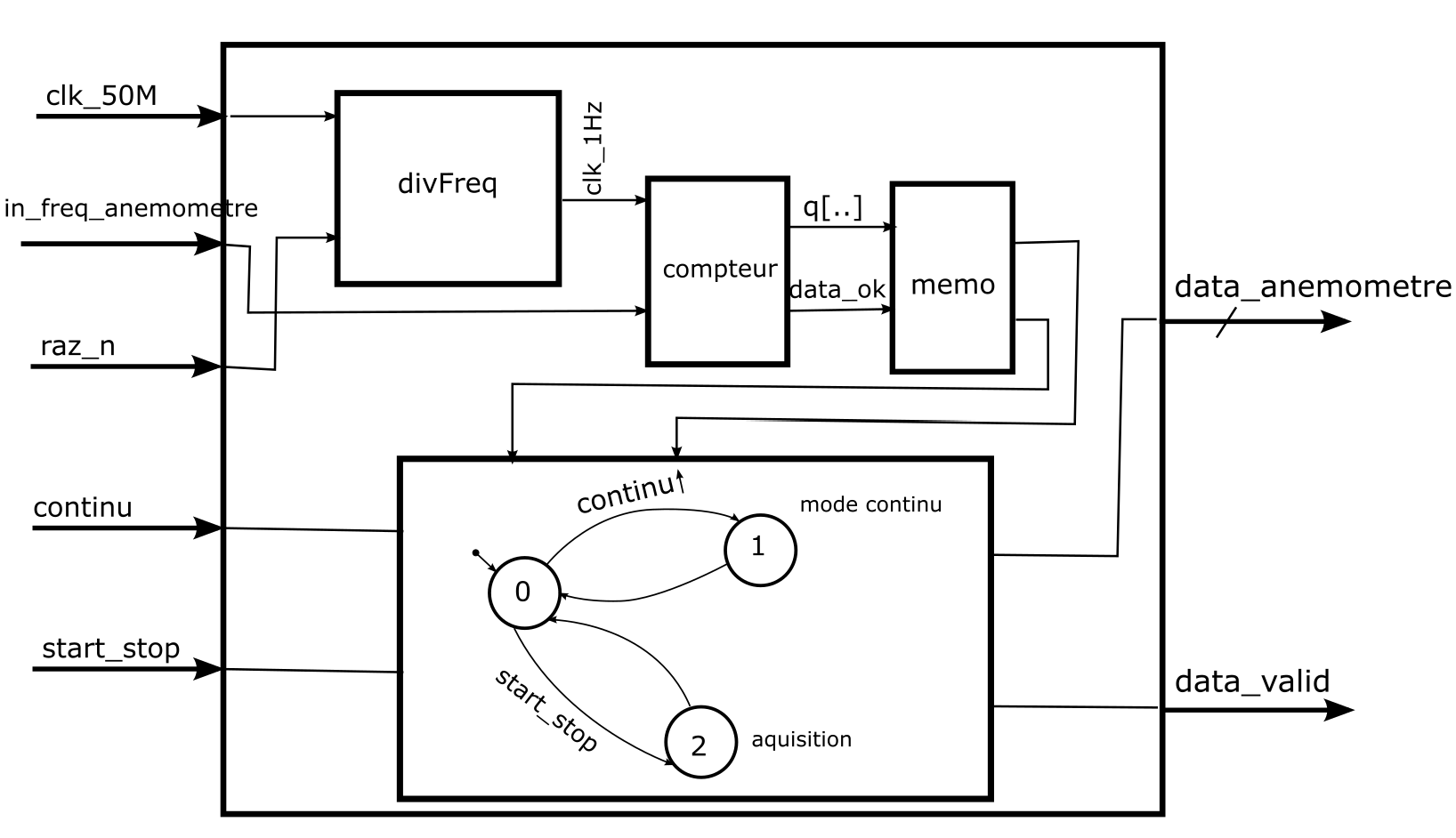


Figure 16: Architecture finale de l'anémomètre

# Conception d’une fonction compliqué

## Spécifications du bouton

Le clavier du Tillerpilot a été conçu pour une utilisation aussi simple et intuitive que possible.

Il doit etre capable de passer du mode manuel au mode automatique et aussi etre capable de régler le CAP de direction :

1. **Le mode manuel**

* La led STBY clignote
* Un appui 1 fois sur Babord (<) ou Tribord (>) modifier le cap de 1° dans la direction indiquée, ce réglage est confirmé par un bip et un éclat de led Babord ou Tribord.
* Un appui long sur Babord (<) ou Tribord (>) modifier le cap de 10° dans la direction indiquée, ce réglage est confirmé par un double bip et double clignotement de la led Babord ou Tribord.

1. **Le mode automatique**

* Un appui sur STBY pour passer du mode manuel au mode automatique.
* Led STBY allumer suivi d’une confirmation par un bip.
* Un appui 1 fois sur Babord (<) ou Tribord (>) (le cap de 1° dans la direction indiquée, ce réglage est confirmé par un bip et un éclat de led Babord ou Tribord.

Conclusion