

Communication CPU-FPGA

Implémentation d'un compteur pour la qualification de tâches périodiques

Gwenhaël GOAVEC-MEROU

17 janvier 2019

Slides disponibles sur

<http://www.trabucayre.com/enseignement/presentationCompteur.pdf>

Sources disponibles sur

http://www.trabucayre.com/enseignement/tp_fpga_sources.tgz

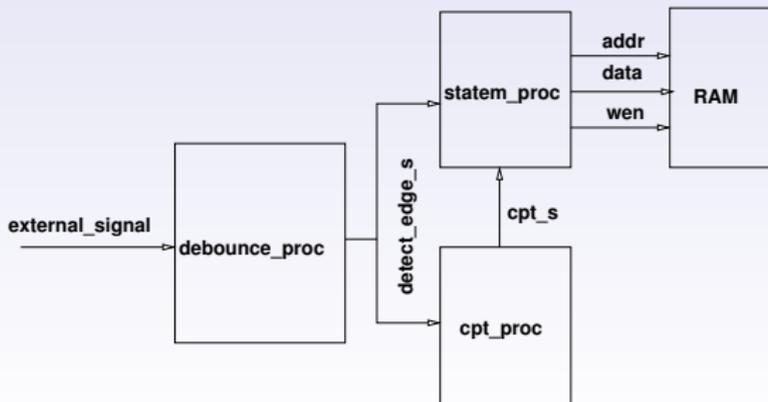
Objectif

- utiliser le FPGA pour mettre en œuvre un compteur capable de mesurer un ensemble de périodes d'un signal généré par une tâche logicielle.
- remplacer un oscilloscope
- à partir des données produire des histogrammes présentant les caractéristiques des tâches dans un environnement temps-partagé (Linux) et temps-réel logiciel (Xenomai)

Motivation

Trois blocs principaux :

- 1 **debounce_proc** : nettoie le signal des *glitches* potentiels et détecte les fronts montants ;
- 2 **cpt_proc** : déclenchement/réinitialisation sur un front montant du signal. Compte la durée jusqu'au prochain front montant dudit signal ;
- 3 **statem_proc** : à chaque détection d'un front montant stocke dans une RAM la valeur du compteur.

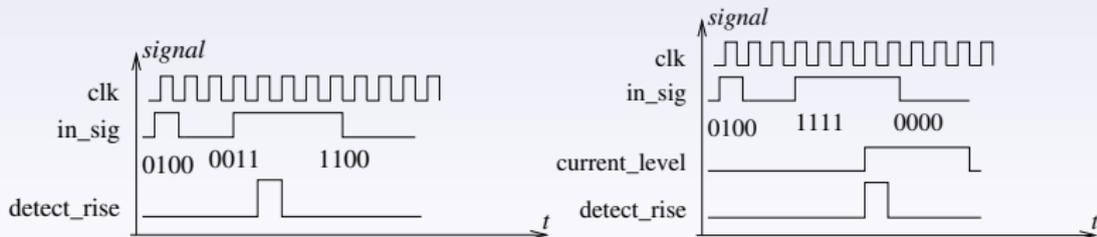


debounce

Utilisation d'un registre à décalage

Deux solutions pour la détection des fronts :

- 1 détection directe de la transition : 0011 signifie un état bas pendant 2 cycles suivi d'un état haut stable pendant 2 cycles ;
- 2 mémorisation du niveau courant et test du registre pour détecter un état pendant n cycles :
 - si **current_level == haut** et **shift_reg == "0000"** alors détection front descendant et **current_state = bas** ;
 - si **current_level == bas** et **shift_reg == "1111"** alors détection front montant et **current_state = haut** ;

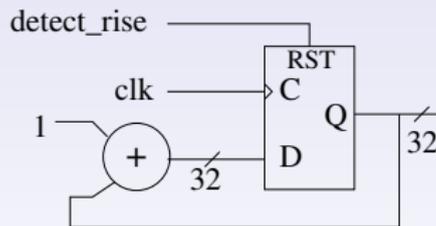


compteur

Simple compteur :

- s'incrémente à chaque cycle d'horloge ;
- remis à zero sur la détection d'un front montant.

Le **std_logic_vector**, utilisé comme compteur, est sur 32 bits pour obtenir des durées supérieures à la seconde.

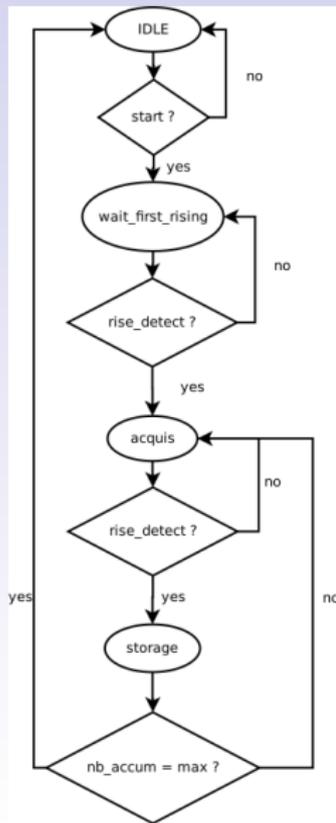


statem_proc

Une machine à état

- comme pour l'exercice du pseudo compteur, attend (IDLE) un ordre de démarrage ;
- sur réception de cet ordre, se place dans un autre état et attends un premier front montant ;
- lors de la réception passe dans un dernier état qui mémorise et attends un front montant. Sur réception du front stocke la valeur courante du compteur et incrémente l'adresse. Passe dans l'état IDLE lorsque n périodes ont été détectées.

Se référer au précédent exercice pour la logique globale.



Gestion de la communication

Écriture : un registre pour le démarrage de l'acquisition ;

Lecture :

- un registre de statut ;
- un registre pour obtenir une valeur ;

Réalisation des histogrammes

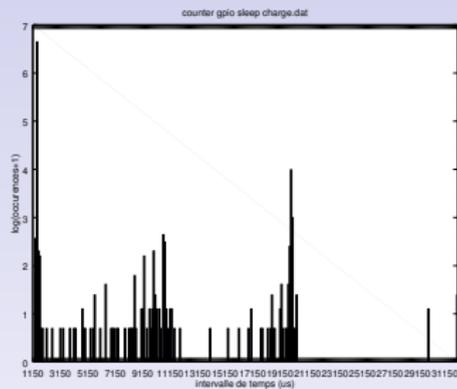
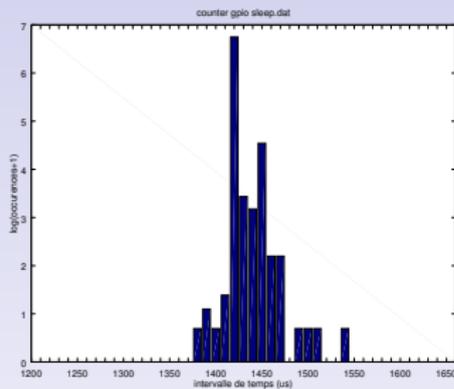
À l'aide de **GNU/octave**

```
a = load('nom_du_fichier.dat');  
a = a ./ 125; % conversion base 8ns en 1us  
[xx, nn] = hist(a, [1150:100:2000]);  
bar(nn, log(xx+1))  
xlim([1150 2000]);  
xlabel('intervalle de temps (us)');  
ylabel('log(occurences+1)');  
title('un titre');
```

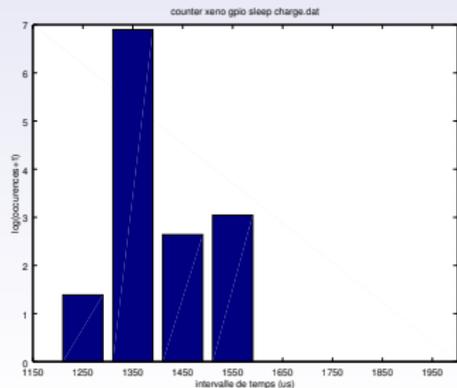
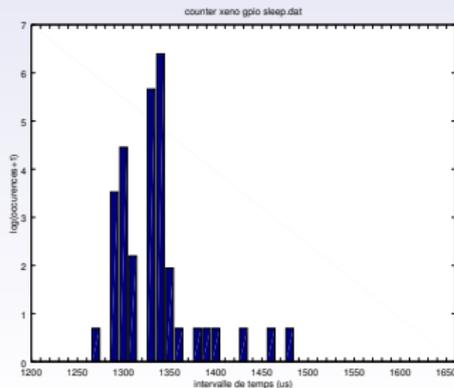
Les limites **[1150 :100 :2000]** sont à adapter en fonction des mesures.

Résultats : fonction *sleep*

Linux (gauche : non chargé de 1370 à 1540 μ s, droite : chargé 1150 à 31150) :

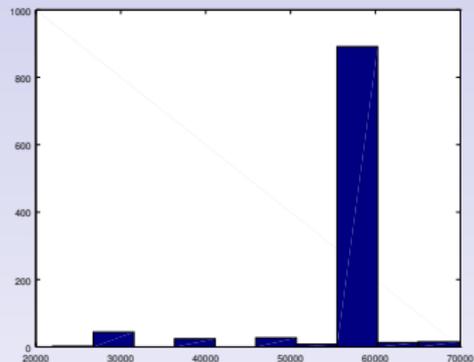
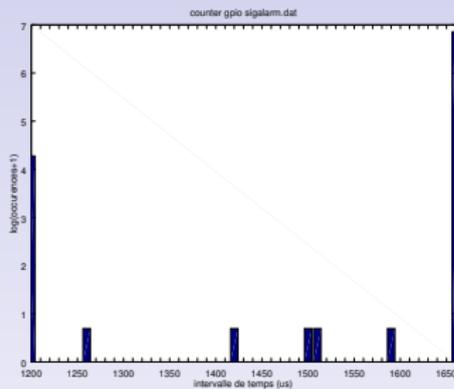


Xenomai (gauche non chargé entre 1250 et 1470, droite : chargé entre 1250 et 1550 μ s) :



Résultats : *timer*

Linux (gauche : non chargé de 1200 à 70000 μ s, droite : chargé 1150 à 31150) :



Xenomai (gauche non chargé entre 1270 et 1450, droite : chargé entre 1250 et 1650 μ s) :

