



I. PRÉSENTATION DE LA FONCTION FP5

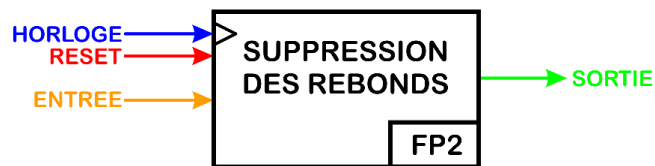
Nous avons vu précédemment que si les signaux A et B issus du codeur incrémental sont porteurs de rebonds le compteur chargé de compter les impulsions valides comptera aussi ces rebonds, d'où un résultat erroné.

Il faut donc débarrasser les signaux A et B de leurs éventuels rebonds.

En analogique cela peut être fait à l'aide d'un filtre passe-bas ou de monostables et portes logiques.

En numérique, dès que le temps intervient on doit faire apparaître des compteurs, c'est ce qui va être fait.

I.1. DESCRIPTION DU COMPOSANT SUPPRESSION DES REBONDS



Une entrée HORLOGE active sur front montant permettra le fonctionnement de l'ensemble du composant en full synchrone.

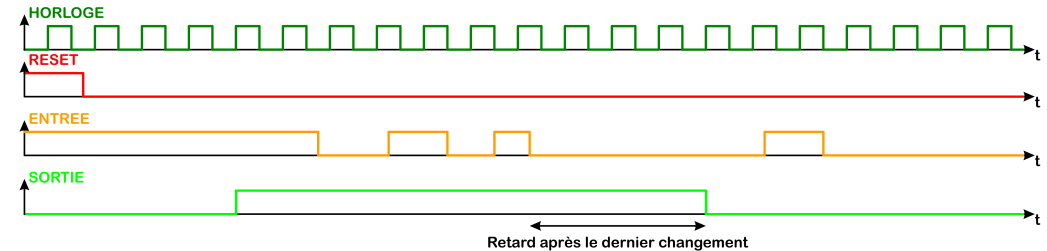
Une entrée RESET active à 1 permettra d'initialiser le composant : SORTIE prendra arbitrairement la valeur 0.

Une entrée ENTREE sera analysée afin de la recopier sur la sortie SORTIE avec un retard temporel mais sans les éventuels rebonds.

Remarque : il faut que la durée des rebonds soit beaucoup plus petite que la période des signaux A et B. De même il faut que la période du signal d'horloge soit plus petite que la durée d'un rebond.

Il faut donc que la période d'horloge soit beaucoup plus grande que la période des signaux A et B, ce qui est toujours le cas avec un circuit logique programmable.

I.2. CHRONOGRAMMES ATTENDUS



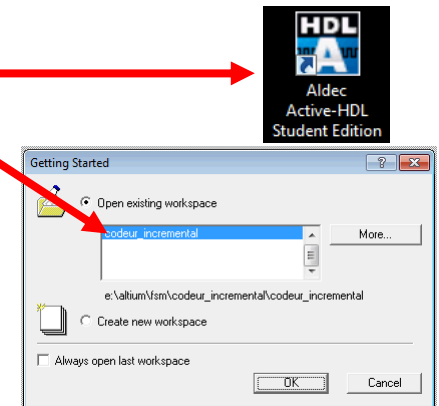
Un compteur sera mis en route dès qu'un changement aura lieu sur l'entrée ENTREE. Si dans un temps imparti il n'y a pas eu de nouveau changement la sortie SORTIE recopiera la valeur d'ENTREE. S'il y a eu un changement avant la fin du temps imparti alors le compteur est remis à zéro.

II. ÉDITION D'UNE MACHINE À ÉTATS

Une machine à état peut permettre de synthétiser facilement ce composant.

II.1. ESPACE DE TRAVAIL

Lancer le logiciel Aldec Active-HDL Student Edition et ouvrir l'espace de travail créé précédemment : Codeur_Incremental.



II.2. CRÉATION DE LA MACHINE À ÉTATS

Créer une nouvelle machine à états (State Diagram) Comme vu précédemment en la nommant : **Supression_Rebonds.asf**.

Créer l'entité à l'aide du Wizard en fonction des entrées et sorties du composant.

II.3. DESSIN DE LA MACHINE À ÉTATS

La machine à états complète est donnée en annexe page 4 pour aider au dessin.

II.4. TRADUCTION DU GRAPHIQUE EN VHDL

Générer le fichier VHDL à partir de la machine à états, corriger les éventuelles erreurs puis afficher celui-ci. Copier le texte source.

III. RETOUR VERS ALTUM

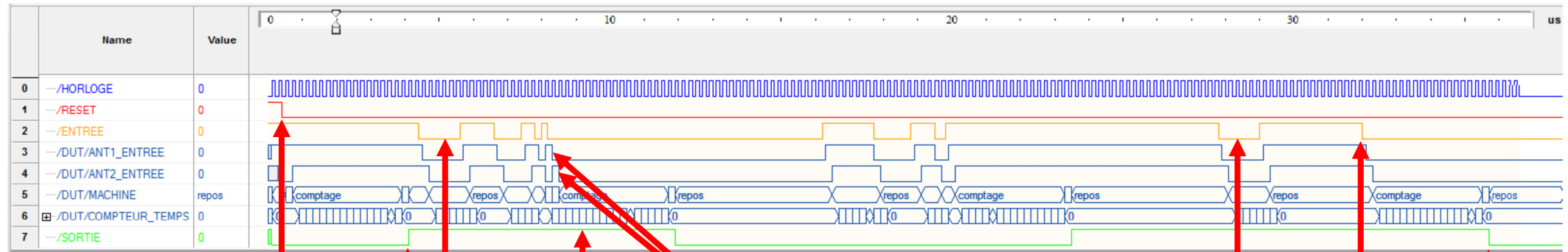
III.1. AJOUT AU PROJET D'UN NOUVEAU FICHIER VHDL

Ajouter un nouveau fichier VHDL au projet. Renommer-le **Supression_Rebonds.Vhd**.

II.2. ÉDITION ET COMPILATION DU FICHIER VHDL

Coller le texte source généré par Aldec Active-HDL dans le nouveau fichier. Sauvegarder celui-ci puis effectuer la compilation.

Remarque : si des erreurs apparaissent, corriger la machine à états et reprendre l'ensemble des opérations afin d'avoir une machine à états à jour (fichier source).



On peut remarquer que dès la fin du RESET, un premier comptage à lieu pour actualiser SORTIE

Ici les rebonds d'ENTREE n'ont pas de d'influence sur le signal SORTIE

La période d'HORLOGE est ici de 200 ns.

Remarque : dans la machine à états créée, il faut 33 coups d'horloges sans changement d'ENTREE pour actualiser SORTIE. Afin de ne pas avoir un chronogramme trop long le compteur a été réduit à 4 bits pour la simulation, ce qui donne 17 coups d'horloges (2 pour les deux registres à décalage ANT1_ENTREE et ANT2_ENTREE et 16 pour le compteur). Ce paramètre est facilement modifiable en fonction des besoins.

IV. SIMULATION FONCTIONNELLE AVEC ALDEC OEM

IV.1. PRODUCTION DU FICHIER DE TEST

Créer le fichier VHDL de test.

Ajouter à celui-ci les lignes des process de tests afin de faire évoluer les ports d'entrées suivant un jeu d'essai permettant de vérifier tous les cas. (Voir listing partiel en annexe page 3).

Enregistrer celui-ci en gardant le nom proposé.

Compiler ce fichier puis compiler le projet complet.

IV.2. ASSOCIATION DES FICHIERS

Ajouter le fichier de test dans le Manager Testbenches du simulateur.

IV.3. SIMULATION

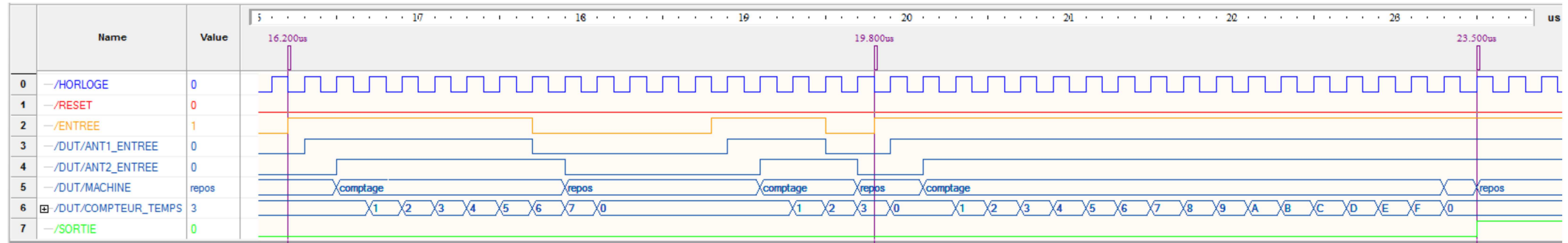
Lancer la simulation et cocher les signaux internes : MACHINE, ANT_ENTREE et COMPTEUR_TEMPS.

Ici un parasite sur ENTREE est supprimé par le composant

Bien sûr s'il n'y a pas de rebonds SORTIE prend la valeur d'ENTREE mais avec un retard de 17 coups d'horloge.

On remarque les 2 registres à décalages qui recopient l'entrée à chaque coup d'horloge.

Ce chronogramme permet de voir en détail le fonctionnement du composant :



On peut noter le temps auquel un premier changement de ENTREE à lieu : 16,2 μ s
Des rebonds ont lieu jusqu'à 19,8 μ s. Le signal ENTREE est donc instable pendant ce temps.
Ce chronogramme permet de voir en détail le fonctionnement du composant : 3,6 μ s (soit environ 18 coups d'horloge).
Le signal SORTIE est actualisé à 23,5 μ s, soit 7,1 μ s après le premier changement et 3,7 μ s après que le signal ENTREE soit stable, ce qui fait 17,5 coups d'horloge, cela correspond donc à ce qui était prévu. Le retard maximum étant la valeur maxi du compteur + 2 multiplié par la période de l'horloge.

La fonction FP5 est donc validée.

V. ANNEXES

```

60  GENERATION_HORLOGE:process 76      wait for 1200 ns; 92      wait for 1100 ns;
61  begin 77      ENTREE <= '1'; 93      ENTREE <= '1';
62      HORLOGE <= '0'; 78      wait for 1000 ns; 94      wait for 700 ns;
63      wait for 100 ns; 79      ENTREE <= '0'; 95      ENTREE <= '0';
64      HORLOGE <= '1'; 80      wait for 800 ns; 96      wait for 300 ns;
65      wait for 100 ns; 81      ENTREE <= '1'; 97      ENTREE <= '1';
66  end process; 82      wait for 400 ns; 98      wait for 8000 ns;
67  83      ENTREE <= '0'; 99      ENTREE <= '0';
68  STIMULUS0:process 84      wait for 200 ns; 100      wait for 1200 ns;
69  begin 85      ENTREE <= '1'; 101      ENTREE <= '1';
70      RESET <= '1'; 86      wait for 150 ns; 102      wait for 3000 ns;
71      ENTREE <= '1'; 87      ENTREE <= '0'; 103      ENTREE <= '0';
72      wait for 400 ns; 88      wait for 8050 ns; 104      wait;
73      RESET <= '0'; 89      ENTREE <= '1'; 105      end process;
74      wait for 4000 ns; 90      wait for 1500 ns;
75      ENTREE <= '0'; 91      ENTREE <= '0';

```

HORLOGE

RESET

ENTREE

ANT1_ENTREE

ANT2_ENTREE

COMPTEUR_TEMPS[4:0]

SORTIE

