TP n° 4 : Réalisation d'instructions SIMD pour le processeur NIOS II

0. Introduction

On peut définir de nouvelles instructions (« customization ») pour le processeur NIOS II (cœur logiciel pour FPGA d'Altera), selon le schéma

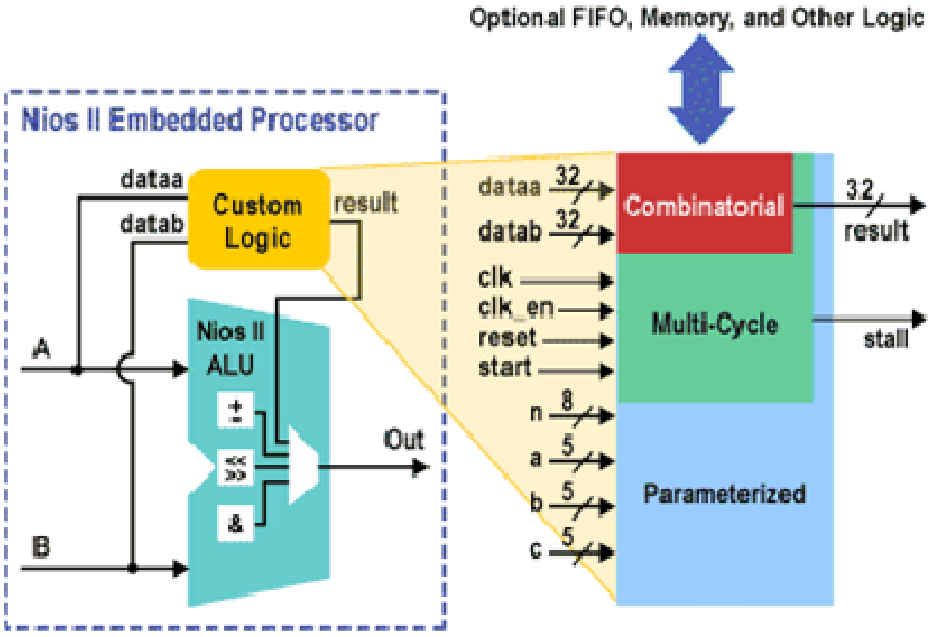

Figure 1 : Spécialisation d'instructions pour le processeur NIOS II

L'objectif de ce TP est de définir des instructions SIMD pour accélérer l'exécution d'une version « octets » d'AES 256. Le code complet est fourni à l'adresse http://www.lri.fr/~de/ArchiM1-1112.htm

1. Fonctions encryptage

Le code VHDL des fonctions

- Fonction aes_addRoundKey.
- Fonction aes_subBytes
- Fonction aes_mixcolumns

est fourni à l'adresse http://www.lri.fr/~de/ArchiM1-1112.htm
Compiler séparément le code VHDL de chacune de ces fonctions et vérifier le fonctionnement correct par simulation logique.

2. Fonctions décryptage

Pour les fonctions

Fonction aes_subBytes_inv
Fonction aes_mixcolumns_inv

Ecrire le code VHDL, compiler séparément le code et vérifier le fonctionnement correct de ces fonctions par simulation logique.

3. Autres fonctions

Déterminer d'autres fonctions appartenant à la fonction `aes256_encrypt_ecb` ou la fonction `aes256_decrypt_ecb` pour lesquelles on peut définir des instructions spécialisées NIOS accélérant l'exécution.
Pour ces fonctions, écrire le code VHDL, compiler et simuler pour vérifier par simulation logique le fonctionnement correct.

4. Annexe : utilisation de Quartus II

Compilation de code VHDL

Pour compiler un code VHDL, les étapes sont les suivantes :

- Ouvrir un nouveau projet à l'aide de *New Project Wizard*, en définissant un répertoire pour le projet, un nom de projet et le nom de l'entité la plus élevée dans la hiérarchie (ces deux derniers noms doivent être identiques).
- S'assurer que le projet contient bien le fichier .vhd à compiler
- Lancer la compilation dans le menu *Processing* | *start compilation*

Simulation

Lorsque le code VHDL est compilé, on peut le simuler l'exécution du circuit correspondant.
Les étapes sont les suivantes :
Génération des configurations d'entrée pour la simulation

- Ouvrir *New* | *Other files* | *Vector Waveform File*
- Aller dans le menu *Edit* |*Insert Node or Bus* | *Node Finder* | *List* : Vous obtenez alors à gauche la liste des signaux correspondant aux entrées et sortie du circuit en cours de conception
- Sélectionner dataa, datab et result à envoyer dans la fenêtre de droite à l'aide de la flèche >. Appuyer sur OK, puis OK après apparition d'une nouvelle fenêtre. Vous obtenez alors les formes de signaux correspondants. Sauvegarder le fichier .vwf sous le même nom que le nom de projet.
- Avec un clic à droite (de la souris) sur les signaux d'entrée, dataa ou datab, vous pouvez leur affecter des valeurs. Cliquez sur *Value* pour affecter des valeurs. Deux méthodes sont utiles : *arbitrary value* (fournir 8 chiffres hexadécimaux) ou *count value* (valeur de départ, incrément sur la première fenêtre, puis « count every » pour déterminer la fréquence du comptage (utilisez 100 ns pour pouvoir voir les 8 chiffres hexadécimaux).
- Attention, il faut souvent, après le clic droit sur les signaux, cliquer sur *zoom* | *fit in window* pour obtenir l'évolution complète des signaux du début à la fin de la simulation.

Simulation logique

- On peut alors simuler le circuit à l'aide du menu *processing* | *start simulation.*
- Les résultats sont obtenus dans le rapport de simulation.