

**INF3610 Systèmes Embarqués**

**Lab 4 – Conception logicielle/matérielle**

**Automne 2015**

**Soumis à : Arnaud Desaulty**

**Nom de l’étudiant :** Maxime Pérusse

**Matricule :** 1685946

**Nom de l’étudiant :** Gabriel Bilodeau-Périard

**Matricule :** 1681165

**Nom de l’étudiant :** Alexis St-Onge

**Matriculte :** 1688297

**Polytechnique Montréal**

**5 Décembre 2015**

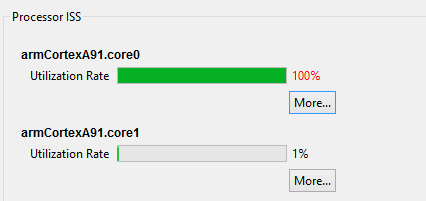
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 10x10 | 50x50 | 300x300 | 500x500 | 1000x1000 |
| FIFO | 0.000391602 s | 0.00146513 s | 0.0309216 s | - | - |
| DMA | 0.00309262 s | 0.00310812 s | 0.00509784 s | 0.0101342 s | - |
| Memory Stream | 0.00610035 s | 0.00615156 s | 0.00712582 s | 0.00717798 s | 0.00313181 s |

La configuration FIFO est la plus lente peu importe la taille des matrices car il n’y a aucune exécution parallèle, le processeur a à effectuer les transactions mémoires (read et write). La configuration DMA est plus rapide que la FIFO car le composant matériel DMA prend en charge les transactions mémoires, ce qui permet une exécution parallèle. La configuration la plus rapide est Memory Stream car elle utilise une mémoire matérielle.

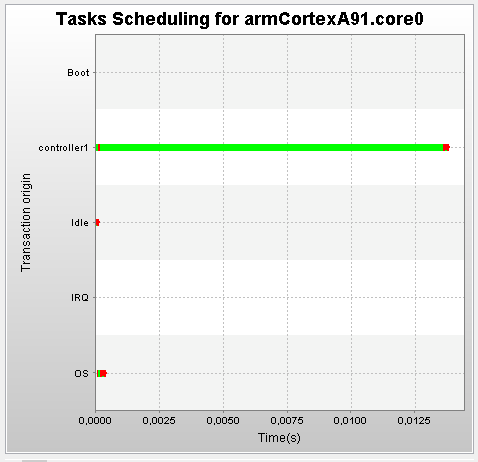
**QUESTION 1**

Voici la présentation des résultats du monitoring de l’exécution des configurations 1 et 2 pour la matrice 200x200.

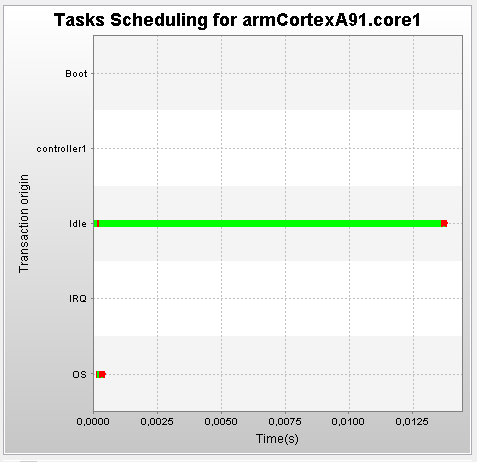
**Configuration 1 – FIFO Hardware**



Utilisation des cœurs du processeur lors de l’exécution de la config. 1

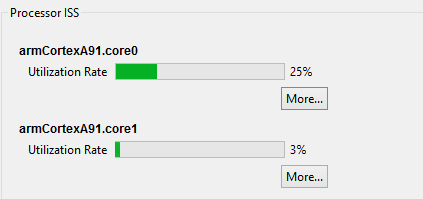


Utilisation détaillée du cœur 0 du processeur lors de l’exécution de la config. 1

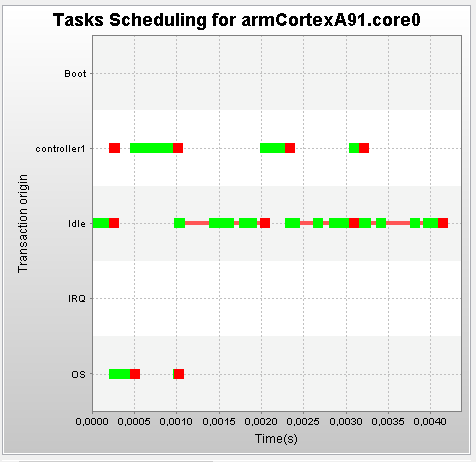


Utilisation détaillée du cœur 1 du processeur lors de l’exécution de la config. 1

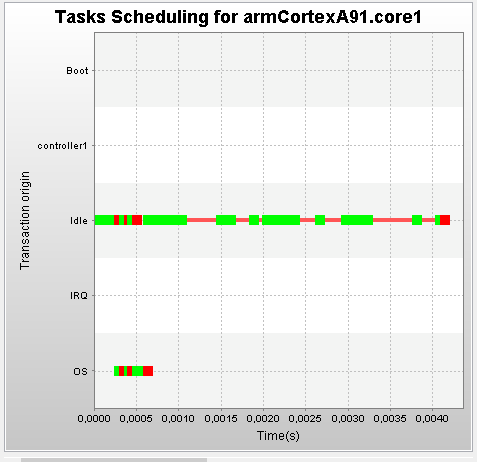
**Configuration 2 - DMA**



Utilisation des cœurs du processeur lors de l’exécution de la config. 2



Utilisation détaillée du cœur 0 du processeur lors de l’exécution de la config. 2

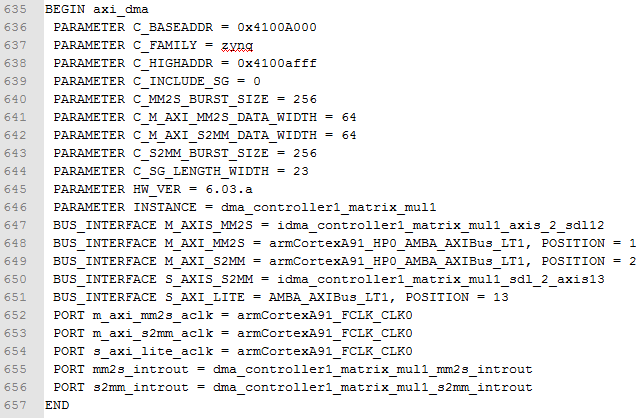


Utilisation détaillée du cœur 0 du processeur lors de l’exécution de la config. 2

En analysant l’utilisation des cœurs du processeur pour la configuration 1 (FIFO Hardware) et 2 (DMA), on remarque que la configuration FIFO nécessite beaucoup moins de calculs que la configuration DMA. L’utilisation du cœur 0 et celle du cœur 1 est de 100% et 1% respectivement pour la configuration FIFO tandis qu’elle est de 25% et 3% pour la configuration DMA. Ceci correspond à ce qu’on s’attendait car la configuration DMA n’a pas à prendre en charge les transactions mémoires (read et write) que le processeur doit effectuer dans la configuration FIFO, résultant en une utilisation moindre du processeur.

En étudiant l’utilisation détaillée des cœurs individuels, on remarque que pour la configuration FIFO, le cœur 0 n’effectue que des opérations sur le controller1 et n’est jamais *idle* tandis que pour la configuration DMA, le cœur 0 n’est pas constamment utilisé pour des opérations sur le controller1. Effectivement, on voit que le processeur effectue des opérations sur l’OS et est souvent *idle*.

**Question 2**



Extrait de system.msh

Connections (nous avons omis la clock) :

AXI Lite AMBA\_AXIBus\_LT1, POSITION = 13

MM2S armCortexA91\_HP0\_AMBA\_AXIBus\_LT1, POSITION = 1

S2MM armCortexA91\_HP0\_AMBA\_AXIBus\_LT1, POSITION = 2

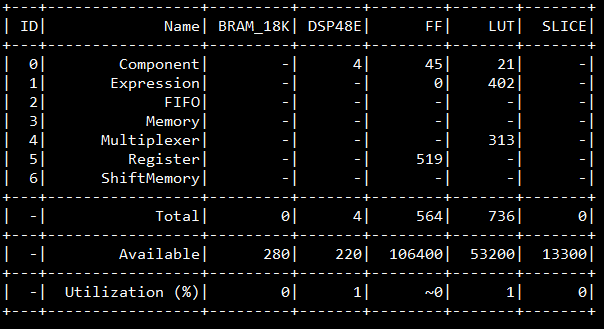
**Question 3**

Pour le mode parallèle, le nombre de parcours du processeur en à la boucle intérieure L2 est de 300 avec une *latency* de 48600. Le nombre de cycles requis est 48600 / 300 donc 162.

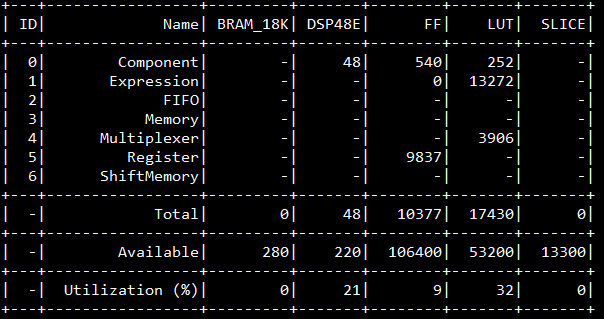
**Question 4**

L’accélération correspond à la *latency* de la boucle L2 sans pragma de déroulement moins la *latency* de la boucle L2 avec pragma déroulement. On obtient alors une accélération de 810600 – 48600 = 762000.

**Question 5**



Area Estimates sans pragma de déroulement



Area Estimates avec pragma de déroulement

L’utilisation du pragma de déroulement nécessite en effet d’avantage de matériel afin d’optimiser le nombre de cycles. On remarque que le nombre de LUT, FF et de DSP48E ont tous augmenté.

**Question 6**

Nous avons obtenus les temps de simulation suivants :

*Sans déroulement sans computeFor*  0.00509784 s

*Sans déroulement avec computeFor* 2.43579 s

Ceci nous permet de conclue que le délai causé par le matériel reviens environ à 2.43 secondes.