



**HAL**  
open science

## Relecture de bitstream appliquée à la relocation de tâches matérielles dans les SoC reconfigurables

Frederic de Melo, Laurent Gantel, Romain Sachaux

► **To cite this version:**

Frederic de Melo, Laurent Gantel, Romain Sachaux. Relecture de bitstream appliquée à la relocation de tâches matérielles dans les SoC reconfigurables. GDR SOCSIP, Jun 2010, Cergy-Pontoise, France. hal-00551374

**HAL Id: hal-00551374**

**<https://hal.science/hal-00551374>**

Submitted on 3 Jan 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Relecture de bitstream appliquée à la relocation de tâches matérielles dans les SoC reconfigurables

F. de Melo

L. Gantel

R. Sachaux

Astre Team  
UMR 8051 CNRS  
Cergy-Pontoise, 95014  
demelo@ensea.fr

Embedded System Lab  
Thales RT  
Palaiseau, 91767  
laurent.gantel@thalesgroup.com

Astre Team  
UMR 8051 CNRS  
Cergy-Pontoise, 95014  
sachaux@ensea.fr

**Abstract**—Les FPGAs permettent aujourd’hui d’intégrer à la fois des traitements matériels mais aussi logiciels, avec l’ajout d’un ou de plusieurs processeurs. Ces nouvelles architectures ouvrent la voie à l’exploration de systèmes hétérogènes où les tâches matérielles et logicielles doivent posséder les mêmes propriétés de préemption vis à vis d’un système d’exploitation temps réel d’un nouveau genre. Cet article présente une méthodologie permettant d’effectuer la relecture du bitstream d’une tâche matérielle afin d’en effectuer sa relocation sur une architecture de type Xilinx Virtex-5.

## I. INTRODUCTION

Dans un système sur puce SoC<sup>1</sup> hétérogène, on doit être capable d’interrompre une tâche matérielle et de sauvegarder son contexte à l’instant  $t$  pour ensuite relancer cette tâche à l’instant  $t + \delta t$ . On doit également être capable de déplacer une telle tâche sur le FPGA, au cas où la zone initiale de placement de celle-ci serait occupée. Pour préempter ou reloger cette tâche la lecture du bitstream devient nécessaire. Dans ces travaux, qui sont menés dans le cadre du projet FOSFOR<sup>2</sup>, nous nous intéressons à la relocation de tâches matérielles 2D sur les FPGAs de nouvelle génération (Virtex-5 et Virtex-6).

Des travaux [1] ont déjà été réalisés sur des anciennes architectures de FPGA Xilinx (Virtex-IIPro, Virtex-4). Notre démarche concernant la reconstruction de bitstreams partiels suit les travaux de Sedcole et al. [2]. Enfin, les méthodes de calcul du CRC<sup>3</sup> [3] nous serviront à finaliser la reconstruction du bitstream, ce dernier étant amené à être modifié en fonction de la zone de relocation de la tâche.

## II. RELECTURE D’UN BITSTREAM

La frame constitue le plus petit élément de donnée de configuration du FPGA, 41 mots de 32 bits la composent. La relecture du bitstream se fait frame par frame, en respectant la structure du Virtex-5 (figure 1).

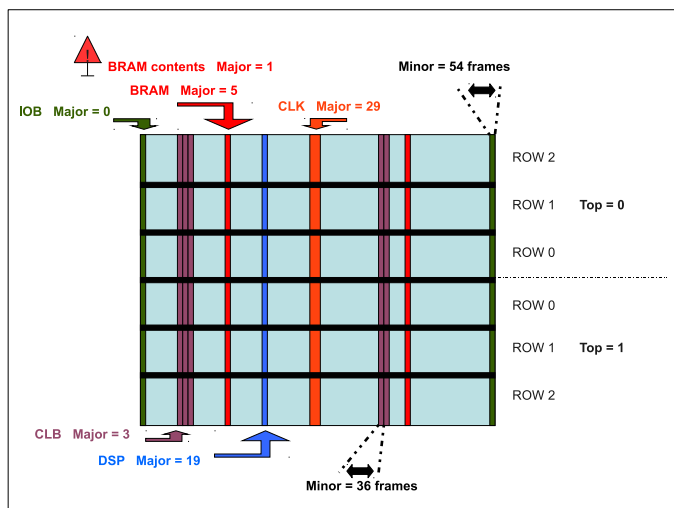


Fig. 1. Structure du Virtex-5 LX110

Pour effectuer la relecture il faut envoyer la séquence de *readback* définie dans [4]. Notons que pour la relecture d’un bitstream complet la valeur du FAR<sup>4</sup>, correspondant à l’adresse de la première frame lue, sera nulle. La valeur du FDRO<sup>5</sup>, incluse dans la commande de lecture des frames, sera, pour une seule frame :

$$FDRO \text{ Read Length} = (\text{words per frame}) * (\text{frames to read} + 1) + 1 = 41 * 2 + 1 = 83 .$$

Un exemple est donné ici pour la relecture de tous les blocks CLB du Virtex-5 LX110:

Pour: Top de 0 à 1, Block = 0, Row de 0 à 2;

Pour: Minor de 0 à 35;

Pour: Major de 1 à 4, 6 à 15, 17 à 18, 20 à 27, 30 à 41, 43 à 52, 54 à 57, 59 à 62;

Faire:  $XHwIcap\_DeviceReadFrame(\text{Top}, \text{Block}, \text{Row}, \text{Major}, \text{Minor})$

La fonction  $XHwIcap\_DeviceReadFrame$  décrit la lecture

<sup>1</sup>System on Chip

<sup>2</sup>Flexible Operating System FOReconfigurable platform, users.polytech.unice.fr/~fmuller/\*fosfor\*/

<sup>3</sup>Cyclic Redundancy Check

<sup>4</sup>Frame Address Register

<sup>5</sup>Frame Data Output Register

d'une frame avec son adresse comme paramètre d'entrée, elle est incluse dans le driver software de l'ICAP<sup>6</sup> fourni par l'outil EDK.

La génération du bitstream de debug (option -DebugBitstream:Yes du bitgen), où l'adressage unitaire des frames est décrit, nous permet de vérifier et valider la relecture des bitstreams complets et partiels. Pour relire un bitstream partiel, l'algorithme est simplement réduit à la zone de configuration avec bien-sûr la valeur du FAR[4] associé. Le nouveau bitstream de configuration contiendra les commandes du bitstream original et les données de configuration du bitstream relu, ainsi que le nouveau CRC recalculé. L'architecture du système nous permettant de réaliser la relecture d'un bitstream, qu'il soit partiel ou complet est la suivante :

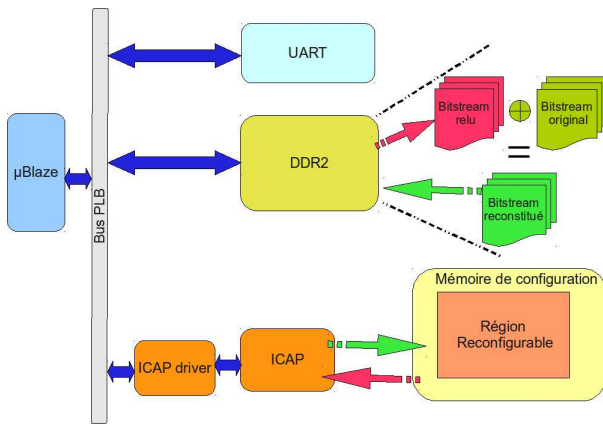


Fig. 2. Architecture utilisée

### III. RELOCATION D'UNE TÂCHE MATERIELLE

Nous présentons la méthode employée (figure 3) pour dans un premier temps valider la relocation d'une tâche sur sa zone de placement initial puis dans un deuxième temps affecter cette tâche à une autre zone inoccupée du FPGA. La validation s'effectue en chargeant d'abord le bitstream complet, puis en effectuant une lecture du bitstream partiel avant de configurer la zone de placement avec son blank (black-box) associé, interrompant alors la tâche. Enfin le chargement du bitstream reconstitué permet de valider le bon fonctionnement de la tâche.

Pour effectuer une relocation de tâche, il nous faut assurer une certaine compatibilité sur les zones de placement éventuel, car les entrées/sorties doivent être positionnées au même endroit pour chaque module. Une attention particulière doit donc être portée sur le routage des *bus-macro* (interfaces entre parties statique et dynamique). Dans cette optique, on contraindra le routage de ces *bus-macro* à l'aide de scripts XDL (langage de programmation des contraintes de routage)[5].

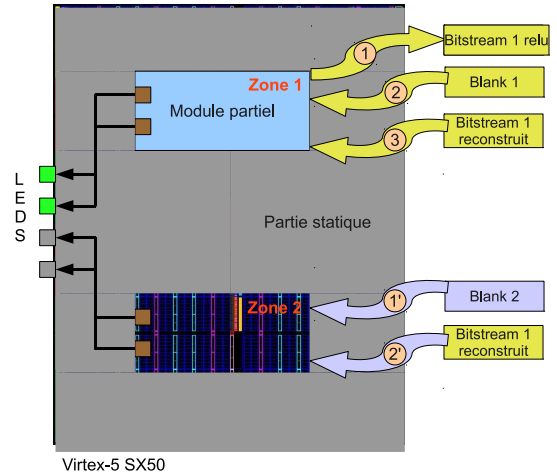


Fig. 3. Validation pour la relocation d'une tâche matérielle

### IV. CONCLUSIONS ET TRAVAUX FUTURS

Dans ce papier nous présentons une méthode pour la relecture du bitstream associé à une tâche matérielle afin d'en effectuer sa relocation. Nous sommes donc aujourd'hui capables de préempter et reloger une région matérielle avec son contexte d'exécution. Ces travaux permettent d'envisager l'intégration de la procédure utilisée au sein de nouveaux services d'OS<sup>7</sup> dédié à l'ordonnancement de tâches matérielles.

### REFERENCES

- [1] Becker, Tobias and Luk, Wayne and Cheung, Peter Y. K., *Enhancing Relocatability of Partial Bitstreams for Run-Time Reconfiguration*, FCCM '07: Proceedings of the 15th Annual IEEE Symposium on Field-Programmable Custom Computing Machines.
- [2] Sedcole, P.; Blodget, B.; Becker, T.; Anderson, J.; Lysaght, P., *Modular dynamic reconfiguration in Virtex FPGAs*, Computers and Digital Techniques, IEE Proceedings - vol.153, no.3, pp. 157-164, 2 May 2006.
- [3] Corbetta, S.; Morandi, M.; Novati, M.; Domenico Santambrogio, M.; Sciuto, D.; Spoletini, P., *Internal and External Bitstream Relocation for Partial Dynamic Reconfiguration*, IEEE transactions on very large scale integration (VLSI) systems ISSN 1063-8210 - 2009, vol. 17, no11, pp. 1650-1654
- [4] Xilinx *UG191 : Virtex-5 FPGA Configuration User Guide*, v2.6, 11 December 2007.
- [5] C. Claus, B. Zhang, M. Huebner, C. Schmutzler, J. Becker, W. Stechele, *An XDL-based busmacro generator for customizable communication interfaces for dynamically and partially reconfigurable systems*, Workshop on Reconfigurable Computing Education at ISVLSI 2007, Porto Alegre, Brazil, May 12, 2007

<sup>6</sup>Internal Configuration Access Port

<sup>7</sup>Operating System