This document is an author-formatted work. The definitive version for citation appears as:

A. Ejnioui and R. F. DeMara, " Area Reclamation Metrics for SRAM-based Reconfigurable Device," in the *Proceedings of The International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA'05),* Las Vegas, Nevada, U.S.A, June 27 – 30, 2005.

FPGA Defragmentation for Sustainable Performance in Reconfigurable Computers

A. Ejnioui and **R. F. DeMara**
Department of Electrical and Computer Engineering
University of Central Florida
Orlando, Florida 32816-2450 U.S.A
aejnioui@mail.ucf.edu

Abstract— Defragmentation is a fundamental resource management service allowing Reconfigurable Computing Systems (RCSs) to efficiently utilize resources when tasks are dispatched dynamically. Only well orchestrated interactions between these three components can sustain the highest possible performance level for applications running on these RCSs. While scheduling and placement have been extensively studied, defragmentation and its impact on overall system performance is still not well understood. This paper quantifies factors related to defragmentation that can affect performance in terms and provides upper and lower bounds on fragmentation during sustained execution.

I. INTRODUCTION

The key to exploiting an RCS is the virtualization of hardware whereby the fabric of a reconfigurable device can be reused ad-infinitum to execute many computations concurrently subject only to area and performance constraints. However, these achievements required the involvement of highly skilled digital designers in compiling and mapping these applications onto the reconfigurable resources of the RCS. Recently, several efforts went into developing software environments that ease the compiling process of the application on RCSs [1]. Other efforts went further by proposing operating systems fitted to manage the hardware resources of an RCS [2, 3]. While there are arguments for and against the development of operating systems for RCSs, their primary benefit stems from their support for multi-tasking. Multi-tasking requires facilities to schedule and place the computational tasks onto the reconfigurable fabric of the FPGA chips embedded within the RCS. In general, the operating system of an RCS provides services to support the dispatching of computation tasks to the FPGA chip in order to accelerate the running applications. These services consist of scheduling the tasks, placing the tasks on the FPGA, and performing defragmentation if task placement fails. In addition, the operating system can provide services to control configuration swapping in and out of the FPGA chip for the purpose of supporting task placement and defragmentation [4]. As the application continues its execution, tasks are added and deleted in a dynamic fashion leaving ultimately the reconfigurable fabric of the chip highly fragmented.

II. PREVIOUS WORK

Several placement studies have been previously undertaken in which different algorithmic approaches for task placement are proposed [1, 5-7]. Regardless of how efficient task placement can be, the reconfigurable fabric will eventually reach an advanced fragmented state where task placement becomes extremely difficult. Unless the tasks already placed on the chip are moved and compacted as fast as possible, task placement cannot be performed, and subsequently application performance cannot be sustained at the same level. To reach this goal, defragmentation has to be performed in the most efficient manner without severely disrupting the progress of the application execution. In fact, defragmentation should aim at bringing the overall performance back up to its previous level before the fragmentation of the FPGA chip has reached a severe level. Whereas task scheduling [1, 8] and placement [1, 7] [5, 6] have been studied in depth, no significant studies have been published to understand the impact of defragmentation on application performance [9]. Such studies could ultimately help in gaining meaningful

insights on the interplay between defragmentation, placement and scheduling.

The paper is organized as

III. Quantifying Defgramentation

One can view the reconfigurable fabric of an FPGA chip as a square area containing an array of smaller empty square areas called *cells*. In the context of FPGA chips, cells are equivalent to reconfigurable logic blocks (CLBs). Figure 2 shows tasks T_1 and T_2 occupying two and six cells respectively. The incoming task T_3, consisting of six cells, cannot be placed on the chip although there is sufficient room left on the FPGA.

A) Fragmentation Factor

Let a and A be the area of a single empty cell and the entire chip respectively. Let N x N be the number of cells in an FPGA chip. Assume that a hole i consists of k cells. This hole yields a fragmentation factor

$$f_i = \frac{1}{A}\sum_{j=1}^{k} a = \frac{ka}{N^2 a} = \frac{k}{N^2}.$$

B) Fragmentation Metric

Since the factor $f_i = \frac{k}{N^2}$ gets smaller as many cells are made empty in the chip, it is scaled to reflect maximum fragmentation by subtracting it from 1 as $F = 1 - \left(\prod_i f_i\right)$. F represents the fragmentation metric of the FPGA chip at any moment.

C) Lowest Possible Fragmentation

An empty chip represents the lowest possible degree of fragmentation. In an empty chip, there is only a single empty area consisting of one hole whose area is N^2a. In this case, $F = 1 - \left(\prod_{i=1}^{1} f_i\right) = 1 - f_1 = 1 - \left(\frac{N^2 a}{N^2 a}\right) = 0$.

D) Highest Possible Fragmentation

A highly fragmented chip resembles the checkerboard layout shown in Figure 3. Assuming N is even, the number of holes in the chip is $\frac{N^2}{2}$ where each hole occupies a single cell. In this case,

$$F = 1 - \prod_{i=1}^{\frac{N^2}{2}} f_i = 1 - \prod_{i=1}^{\frac{N^2}{2}} \left(\frac{a}{A}\right) = 1 - \frac{1}{(N)^{N^2}}.$$

Although F does not reach exactly 1 as shown, it nevertheless approaches 1 as N gets larger. While this fragmentation metric is similar to the one proposed in [1], its semantics are totally different. Given this formulation of the fragmentation metric, any event that modifies the state of the reconfigurable fabric of the chip can affect the value of F. Events which can do so consist of placing a task on the chip, purging a task from the chip, or moving a task from location to location on the chip. As a result, it is the responsibility of the placement and defragmentation process to constantly update F when these events are witnessed.

References

[1] J. Tabero, J. Septien, H. Mecha, D. Mozos, and S. Roman, "Efficient Hardware Multitasking through Space Multiplexing in 2D RTR FPGAs," *Euromicro Digital System Design Conference*, September 2003.

[2] G. Wigley and D. Kearney, "The management of Applications for Reconfigurable Computing Using an Operating System," *The 7th Asia-Pacific Conference on Computer System Archicture*, 2002, pp. 73-81.

[3] G. Wigley and D. Kearney, "The First Real Operating System for Reconfigurable Computers," *Australian Computer Systems Architecture Conference*, Goaldcoast, Queensland, Australia, 2000, pp. 129-136.

[4] O. Diessel and G. Wigley, "Opportunities for Operating Systems Research in Reconfigurable Computing," University of South Australia ACRC-99-018, Aug. 1999, available at http://citeseer.ist.psu.edu/diessel99opportunities.html.

[5] H. Walder, C. Steiger, and M. Platzner, "Fast Online Task Placement on FPGAs: Free Space Partitioning and 2D-Hashing," *International Parallel and Distributed Processing Symposium*, April 2003.

[6] H. Walder and M. Platzner, "Non-preemptive Multitasking on FPGAs: Task Placement and Footprint Transform," *The 2nd International Conference on Engineering of Reconfigurable Systems and Architectures*, June 2002, pp. 24-30.

[7] M. Handa and R. Vemuri, "An Efficient Algorithm for Finding Empty Space for Online FPGA Placement," *Design Automation Conference*, San Diego, CA, June 2004, pp. 960-965.

[8] H. Walder and M. Platzner, "Online Scheduling for Blockpartitioned Reconfigurable Devices," *Design, Automation and Test in Europe*, Mar. 2003, pp. 290-295.

[9] M. Handa and R. Vemuri, "Area Fragmentation in Reconfigurable Operating Systems," *Engineering of Reconfigurable Systems and Algorithms*, Las Vegas, NV, June 2004.

[10] M. G. Gericota, G. R. Alves, M. L. Silva, and J. M. Ferreira, "Run-Time Management of Logic Resources on Reconfigurable Systems," *Design Automation and Test in Europe*, March 2003, pp. 974-979.

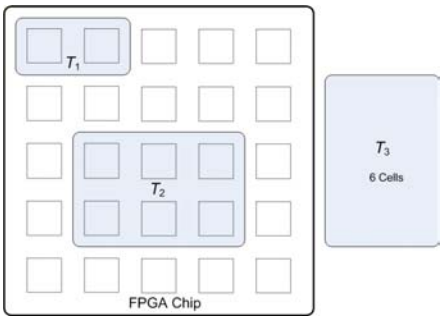

Figure 2. Fragmentation of FPGA chip.

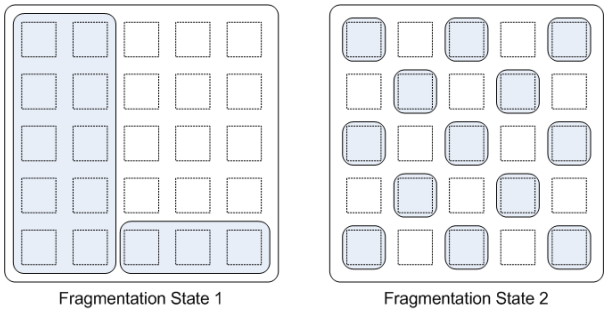

Figure 3. Fragmentation states with equal empty areas.