Microprocessors and Microsystems 47 (2016) 231–243

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

Leakage aware resource management approach with machine learning optimization framework for partially reconfigurable architectures

Nam Khanh Pham[a,b,*], Akash Kumar[c,*], Amit Kumar Singh[d], Mi Mi Aung Khin[b]

[a] Department of Electrical and Computer Engineering, Faculty of Engineering, NUS, Singapore
[b] Data Storage Institute, A*STAR, Singapore
[c] TU Dresden, Center for Advancing Electronics Dresden (cfaed), Germany
[d] School of Electronics and Computer Science, University of Southampton, UK

ARTICLE INFO

Article history:
Received 22 December 2015
Revised 30 April 2016
Accepted 28 September 2016
Available online 14 October 2016

Keywords:
Scheduling
Mapping
Resource management
Design space exploration
Machine learning

ABSTRACT

Shrinking size of transistors has enabled us to integrate more and more logic elements into FPGA chips leading to higher computing power. However, it also brings a serious concern to the leakage power dissipation of the FPGA devices. One of the major reasons for leakage power dissipation in FPGA is the utilization of prefetching technique to minimize the reconfiguration overhead (delay) in Partially Reconfigurable (PR) FPGAs. This technique creates delays between the reconfiguration and execution parts of a task, which may lead up to 38% leakage power of FPGA since the SRAM-cells containing reconfiguration information cannot be powered down. In this work, a resource management approach (RMA) containing *scheduling, placement* and *post-placement* stages has been proposed to address the aforementioned issue. In scheduling stage, a leakage-aware priority function is derived to cope with the leakage power. The placement stage uses a cost function that allows designers to determine the desired trade-off between performance and leakage-saving. The post-placement stage employs a heuristic approach to close the gaps between reconfiguration and execution of tasks, hence further reduce leakage waste. To further examine the trade-off between performance (schedule length) and leakage waste, we propose a framework to utilize the Genetic Algorithm (GA) for exploring the design space and obtaining Pareto optimal design points. Addressing the time-consuming limitation of GA, we apply Regression technique and Clustering algorithm to build predictive models for the Pareto fronts using a training task graph dataset. Experiments show that our approach can achieve large leakage savings for both synthetic and real-life applications with acceptable extended deadline. Furthermore, different variants of the proposed approach can reduce leakage power by 40–65% when compared to a performance-driven approach and by 15–43% when compared to state-of-the-art works. It's also proven that our Machine Learning Optimization framework can estimate the Pareto front for new coming task graphs 10x faster than well-established GA approach with only 10% degradation in quality.

1. Introduction

Field-programmable gate arrays (FPGAs) are promising candidates for digital circuit implementation because of their growing density and speed, short design cycle, and steadily decreasing cost. Furthermore, most of the FPGA devices nowadays can be partially reconfigured at run time, i.e., a configuration can be loaded into part of the device while the rest of the system continues operating. This feature obviously provides greater flexibility and more powerful computing ability. However, these advantages come with additional problems related to reconfiguration time and power dissipation.

A drawback of FPGA due to its hardware redundancy is its inefficiency in term of power consumption when compared to ASIC components [1,2]. In practice, an FPGA circuit implementation may use only a fraction of the hardware resource but the power is dissipated in both the used and the unused components. The power consumption in FPGA includes static (leakage) and dynamic power and their contribution into the total power consumption heavily depends on the circuit technology. Beyond 65 nm technology, leakage power becomes an increasingly dominant component of total power dissipation [3]. This has motivated us to focus our work on reducing the leakage power dissipation.

* Corresponding author.
E-mail address: phamnamkhanh@u.nus.edu (N.K. Pham).

http://dx.doi.org/10.1016/j.micpro.2016.09.012
0141-9331/

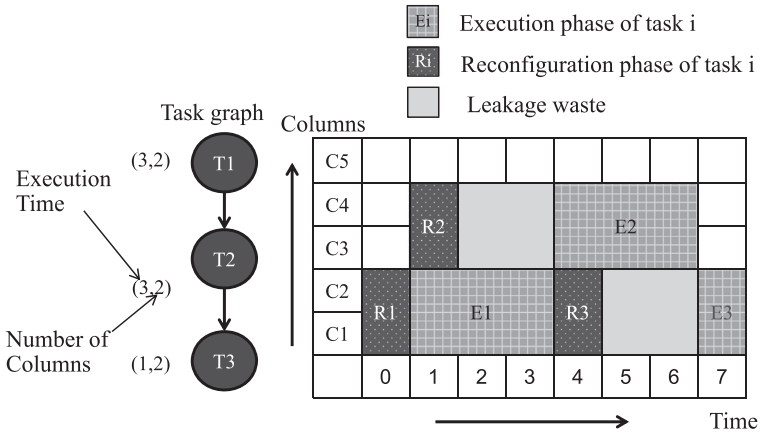

Fig. 1. Example of leakage waste caused by prefetching technique.

Configuration prefetching [4] is a widely adopted technique for reducing the reconfiguration delay in Partially Reconfigurable (PR) FPGA. In prefetching, a task is loaded into the FPGA as soon as possible and this may result in overlap between the configuration part of the waiting task (to be executed) with the execution part of operating tasks, facilitating for reduced reconfiguration overhead (time). However, even after the task is loaded (prefetched), it may not execute and has to wait until few other tasks complete due to involved dependencies. Such waiting introduces delays between the configuration and execution part of the same task. During the delay interval, the SRAM-cells of the FPGA (containing bits of the waiting task to be executed) cannot be powered down to avoid the loss of configuration data from the cells. Therefore, the cells dissipate a significant amount of power.

Motivational Example: Fig. 1 presents an example to demonstrate aforementioned issues. In this example, the task graph on the left-hand side is scheduled on an FPGA platform with prefetching technique. During the interval between R3 and E3, the logic blocks of columns 1 and 2 can be powered down to remove leakage wastes. However, since the SRAM-cells of these columns cannot be powered down as the configuration data will be lost, they consume a considerable amount of power. As SRAM cells leakage contributes ≈38% to FPGA leakage [5], reducing FPGA SRAM leakage is of paramount importance.

In order to reduce leakage, a scheduling approach needs to be developed aiming at allocating reconfiguration and execution parts as close as possible while keeping task dependencies, timing and architecture constraints into account. Several works have been proposed to solve this problem [6,7]. However, these works attempt to address the leakage problem in a single phase of the resource management process (details in later sections). As a result, the leakage power cannot be significantly reduced. It has also been observed that there exists a trade-off between leakage waste and performance [6]. However, the trade-off analysis by employing the existing approaches is not efficient. A high degradation in performance is noticed in order to achieve small amount of leakage savings. To tackle the problem in a comprehensive perspective towards achieving high leakage reductions, we propose a multi-stage **Leakage-aware resource management approach (RMA)** consisting of three stages. Our main contributions to each stage are as follows:

- **Scheduling:** A list-scheduling algorithm has been developed with a specific priority function that is customized for addressing the leakage power reduction.
- **Placement:** A cost function has been derived for the placement stage to further reduce the leakage power. This function provides designers a flexibility to manage the trade-off between performance and leakage waste.
- **Post-placement:** A post-placement heuristic has been proposed to improve the scheduling results (leakage savings) from previous stages.

As our multi-stage Leakage-aware RMA utilizes two cost functions in scheduling and placement stage with various parameters, these parameters form a multidimensional design space with 2 objectives on performance and leakage saving. To further examine the trade-off between these two objectives, we propose an Optimization Framework with Genetic Algorithm to help the designers to efficiently traverse the design space and generate a set of points that are superior in one of the objective dimensions. These points form the Pareto front, which is the Holy Grail for system designers since it not only provides the insight into the trade-off between different objectives but also allows them to choose the most efficient design for different purposes. However, the process of traversing the design space with GA is usually very time-consuming due to the exponential increase in the number of design points with the dimension of the space, which is the number of coefficients in the priority/cost functions. In attempting to solve the time consuming problem of GA optimization, we develop a Machine Learning (ML) component for our Optimization Framework that can accurately estimate the Pareto fronts of new incoming tasks in a fraction of time when compared to GA approach. To achieve such a superior performance, our ML component utilizes Linear Regression to build predictive models for Pareto fronts from a training set of task graphs (TG) at training phase and applies these predictive models together with Density-based Clustering algorithm at prediction phase. Main components of our **ML Optimization Framework** are summarized as follows:

- A comprehensive framework for integrating GA and ML techniques to optimize our **Leakage-aware RMA**: from generating data to building predictive models and predicting Pareto fronts for new TGs;
- A Linear Regression model describing the dependency between the range of Pareto front and TGs features;
- A Density-base Clustering Algorithm to generate near-Pareto-optimal design points.

Paper Organization: Section 2 presents state-of-the-art related to leakage power reduction and existing works on GA and ML techniques in scheduling domain. Section 3 provides the targeted FPGA architecture, application model and problem definition. Our **Leakage-aware RMA** are presented in Section 4. The details of our **ML Optimization Framework** are presented in Section 5. In Section 6, experimental results are reported and Section 7 provides the conclusion.