Expert Systems with Applications 41 (2014) 4691–4703

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Automated exploration of datapath and unrolling factor during power–performance tradeoff in architectural synthesis using multi-dimensional PSO algorithm

Anirban Sengupta *, Vipul Kumar Mishra

Discipline of Computer Science and Engineering, Indian Institute of Technology, Indore, India

ARTICLE INFO

Keywords:
Unrolling factor
Multi-dimensional
Particle encoding
Power
Execution time

ABSTRACT

A novel algorithm for automated simultaneous exploration of datapath and Unrolling Factor (UF) during power–performance tradeoff in High Level Synthesis (HLS) using multi-dimensional particle swarm optimization (PSO) (termed as 'M-PSO') for control and data flow graphs (CDFGs) is presented. The major contributions of the proposed algorithm are as follows: (a) simultaneous exploration of datapath and loop UF through an integrated multi-dimensional particle encoding process using swarm intelligence; (b) an estimation model for computation of execution delay of a loop unrolled CDFG (based on a resource configuration visited) without requiring to tediously unroll the entire CDFG for the specified loop value in most cases; (c) balancing the tradeoff between power–performance metrics as well as control states and execution delay during loop unrolling; (d) sensitivity analysis of PSO parameter such as swarm size on the impact of exploration time and Quality of Results (QoR) of the proposed design space exploration (DSE) process. This analysis presented would assist the designer in pre-tuning the PSO parameters to an optimum value for achieving efficient exploration results within a quick runtime; (e) analysis of design metrics such as power, execution time and number of control steps of the global best particle found in every iteration with respect to increase/decrease in unrolling factor.

The proposed approach when tested on a variety of data flow graphs (DFGs) and CDFGs indicated an average improvement in QoR of >28% and reduction in runtime of >94% compared to recent works.

1. Introduction

When digital systems are built, designers undergo numerous decision making steps at various levels of design abstraction (register transfer level, system/high level etc.) such as the type of architectural framework (datapath) required, exploring the best possible implementation alternative and managing hardware–software tradeoff. More formally, the above process is termed as design space exploration. Design space exploration when performed during high level synthesis becomes a non-trivial task as it involves multiple convoluted design decisions specially when simultaneously dealing with conflicting parameters such as power, area and performance. The above process becomes further intricate when an auxiliary variable called 'loop unrolling factor' joins the decision making process. Owing to the reasons above, architecture exploration suffers from exponential order of complexity with the increase in number of alternative solutions, thereby making it impossible to perform an exhaustive search (Coussy, Gajski, Takach, & Meredith, 2009; Coussy & Morawiec, 2008; De Micheli, 1994; Gajski, Dutt, Wu, & Lin, 1992; Mohanty, Ranganathan, Kougianos, & Patra, 2008; Zhang & Ng, 2000).

Other DSE approaches in HLS employed very recently such as genetic algorithm (GA) (Sengupta, Sedaghat, and Sarkar (2012), Harish Ram, Bhuvaneswari, & Prabhu, 2012; Krishnan & Katkoori, 2006; Gallagher, Vigraham, & Kramer, 2004; Mandal, Chakrabarti, & Ghose, 2000) used for solving similar problem (but of lesser complexity as it did not include exploration of unrolling factor as well as tradeoff between power and execution time) have not been found suitable candidates owing to the computationally expensive runtime (growing exponentially) and lower guarantee of reaching optimal result. This has also been found after comparison with Krishnan and Katkoori (2006) and Sengupta, Sedaghat, and Sarkar (2012) where proposed PSO based approach produces better QoR with lesser exploration run time. Moreover, we have attained real optimal solutions (by comparing with golden solutions) for almost all cases unlike other heuristic approaches. Therefore, in order to combat the problem of exploration, a novel framework using PSO

* Corresponding author. Tel.: +91 7324240735.
E-mail address: asengupt@iiti.ac.in (A. Sengupta).

http://dx.doi.org/10.1016/j.eswa.2014.01.041
0957-4174/© 2014 Elsevier Ltd. All rights reserved.

for automated parallel exploration of datapath and loop unrolling factor is utilized.

To the best of the authors' belief, this is the first work that proposes a completely automated parallel exploration of datapath and loop unrolling factor using novel hyper-dimensional particle encoding as discussed in Section 3. So far in the related works, none has performed simultaneous exploration of UF and resource combination. Further, no approach exists in the literature that transforms PSO (Kennedy & Eberhart, 1995) for solving MO-DSE problem through an automated process for CDFGs. Besides, following are the highlights/contributions of this work:

(a) Simultaneous exploration of datapath and loop UF through multi-dimensional PSO.
(b) An estimation model for delay computation of a loop unrolled CDFG used in most cases.
(c) Balancing tradeoff between power–performance as well as control states and delay.
(d) Sensitivity analysis of swarm size and its impact on exploration time and QoR of DSE.

The rest of the paper is organized as follows: Section 2 discusses the related works. The problem formulation and proposed framework are explained in Section 3. The demonstration of the proposed algorithm is introduced in Section 4. Section 5 reports and analyzes the experimental results while Section 6 concludes the paper.

2. Previous work

Design space exploration in HLS has been addressed by many researchers so far. However, there has been no previous work so far on automated parallel exploration of datapath and unrolling factor during HLS, besides many other shortcomings. The paragraph below highlights some of the major/minor drawbacks with the previous works. For example in Mishra and Sengupta (2014), authors have proposed the use of particle swarm optimization (PSO) for exploration process. The drawback of this approach is the lack of high level transformation techniques during exploration process of CDFGs. The problem of DSE was also addressed in Sengupta et al. (2012) by proposing a multi structure genetic algorithm (GA), which assists in deciding Pareto fronts amongst the different design variances. However, power optimization performed by the approach is not accurate and incurs major power violations for some applications. Also, the approach is inherently slow in speed and gives local optimal solutions in most of the cases. The GA has been used by the authors in Gallagher et al. (2004), Dhodhi, Hielscher, Storer, and Bhasker (1995) and Heijlingers, Cluitmans, and Jess (1995) to yield better quality results for the DSE process. However, the GA used suffers from poor implementation runtime and has no guarantee for yielding superior design points always.

Besides above, authors have proposed an open-source high-level synthesis tool called LegUp in Canis et al. (2013), which is for FPGA-based processor/accelerator systems. LegUp is able to synthesize C language to hardware, thereby providing a useful platform to perform high level synthesis. Different FPGA architectures are supported by this tool, and allows of new scheduling algorithm and parallel accelerators. Furthermore, tools such as ROCCC have also been proposed in Villarreal, Park, Najjar, and Halstead (2010), which is an open-source high-level synthesis tool for generating register transfer level (RTL) structure from C. It is designed for kernels that perform computation intensive tasks such as digital signal processing (DSP) applications. Therefore, ROCCC applies to a specific class of applications (streaming-oriented applications), and is not a general C-to-hardware compiler, unlike LegUp (Canis et al., 2013) which compiles larger C programs than is possible with ROCCC. Furthermore, tool such as CatapultC from Mentor Graphics (now acquired by Calypto) (Calypto, 2013) uses C/C++ to describe the functional intent and generating Register Transfer Level (RTL) structure. This tool provides two options: (a) automated tool flow that disables the registers to make them inactive and (b) manual work flow that allows changing the code. In Palermo, Silvano, and Zaccaria (2008), the authors have proposed a discrete particle swarm optimization based design space exploration in high level synthesis. The major drawback of the approach is the lack of consideration of local best (cognitive factor) during exploration. Further, while updating the particles' velocity, the authors updated only direction (as step length was kept constant). In Di Nuovo, Palesi, and Patti (2006), the authors have recommended the identification of some superior design points from the Pareto set. Further, a tool called AutoPilot was introduced in Zhang et al. (2008) to address the problem of exploration in HLS. It performs C/C++/systemC-to-RTL synthesis. The tool was targeted for FPGA's. Authors in Harish Ram et al., 2012 describe an approach to solve DSE problem which is based on GA and weighted sum particle swarm optimization (WSPSO). They have not considered the actual velocity function to update the particle position. In WSPSO, the authors also have not considered user constraints for power and execution time in cost function. In Gupta, Dutt, Gupta, and Nicolau (2004), the authors introduced a tool called SPARK for performing HLS. The shortcoming of this approach is that the unrolling factor for the loop being user-directed is not able to automatically determine the optimal combination of UF and datapath together. In Krishnan and Katkoori (2006), a framework based on node priority scheme has been suggested for DSE of datapaths in high level synthesis. The work in Krishnan and Katkoori (2006) produces the balanced trade-off between latency and area. Though the results are promising, but it is computationally expensive and does not consider power constraint in their approach. Moreover, the authors used exploration capability of evolutionary approach in Holzer, Knerr, and Rupp (2007) for DSE. The main drawback of this approach besides requiring manual intervention to decide the optimal UF, considers only UFs as potential candidates which evenly divided the iteration count. Work presented in Ascia, Catania, Di Nuovo, Palesi, and Patti (2007) for DSE suggests that authors used an evolutionary algorithm for successful evaluation of the design for an application specific SoC. In addition to above, in Mandal et al. (2000), authors have applied GA to the binding and allocation phase. One of the key features of their approach is the use of multiport memories. However, its main drawback is the input format as scheduled data flow graph, thereby being unable to handle the scheduling phase during exploration. In addition, authors in Liu and Carloni (2013) have proposed machine learning method: random forest for DSE and introduced an experimental design which can wisely sample micro-architecture choices and use for training in the learning model. However, authors in Liu and Carloni (2013) only explore different high level transformation variants like function inline, loop breaking, array implementation without focusing on automated exploration of resource combination and high level transformation simultaneously. Further, the authors in Liu and Carloni (2013) have also not considered power and data pipelining. In Dhodhi et al. (1995), the authors have used the concept of heuristic problem pair to convert a data flow graph into a valid schedule. The approach suffers from extremely slow speed due to inherent nature of genetic approaches during searching. In Das (1999), the problem of exploration was also addressed by suggesting order of efficiency, which assists in deciding preferences between different Pareto optimal points. However, the approach lacks considering of data pipelining during exploration.

Moreover, authors in Zuluaga, Bonilla, and Topham (2012) proposed a machine learning strategy that models the impact of the parameterization of the tool on the target objectives given the