Microprocessors and Microsystems 35 (2011) 392–404

Contents lists available at ScienceDirect

Microprocessors and Microsystems

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

Multi-objective efficient design space exploration and architectural synthesis of an application specific processor (ASP)

Anirban Sengupta, Reza Sedaghat *, Zhipeng Zeng

Ryerson University, Department of Electrical and Computer Engineering, Toronto, Canada

ARTICLE INFO

Article history:
Available online 6 March 2011

Keywords:
Application specific processor
Design space exploration
Multi-objective
Compound operational constraints
RTL

ABSTRACT

As the growth of system complexity rapidly increases, the gap between Electronic System Level (ESL) and the Register Transfer Level (RTL) must be filled. Currently, Very Large Scale Integration (VLSI) and System-on-Chip (SoC) designs are multi-objective in nature, requiring simultaneous fulfillment of multiple parameters. Extensive research on Design Space Exploration (DSE) problems and synthesis of an application specific processor (ASP) design have been done until now but none of the prior works have focused explicitly on integrating a fast multi-objective architecture exploration mechanism with the architectural synthesis stages to formalize the design methodology of an application specific processor in case of multiple objectives. This paper proposes a design methodology of a multi-objective application specific processor by integrating an efficient multi-objective (area occupied, execution time and power consumption) exploration approach with the architecture synthesis process, useful for portable devices and many high end applications. The formalized steps of the design methodology for the ASP guarantees the designer an error free approach to design the system with strict limitations on compound operational constraints. The results of implementation of the designed ASP using the proposed design methodology in FPGA and ASIC have also been shown.

1. Introduction

The design procedure of general purpose processors has traditionally been quite different from the design of application specific processors. The ASP is only meant to cater a certain specific application or for a certain class of applications, while on the contrary, the primary requirement for general purpose processors are sustained high performance across a broad spectrum range of general user applications. The design of ASP demands pure high performance based on the working constraints provided, thereby being used as a function specific system. ASP cores are being increasingly used to address the need for high performance, low area, minimum cost and timely operation in many embedded systems. For example, elements used in mobile phones such as the DSP cores must be of low cost and occupy minimal area than the general purpose counterparts. The distinction clearly reflects the use of ASP in areas where power consumption, cost, area and speed are of paramount importance [1,2].

Due to the emergence of various sophisticated devices with increased complexity, the development of a formal design methodology with concurrent satisfaction of multiple performance parameters in the design process has become extremely significant. Traditional ASP designing using high-level synthesis procedure uses only area and delay as their objective parameter. However, recent trends show that power consumption has become one of the most important design constraints, aside from area and delay. This paper proposes a design flow for multi-objective architectural synthesis of an ASP integrated with an efficient and fast multi-objective design space exploration approach for architecture selection. Many design methodologies optimize traditional area and delay as performance metrics. The design methodology for most modern automated synthesis tools starts from the behavioral model of an application, determines the performance parameters such as required hardware resources, throughput and latency or estimated software execution time. This is followed by design space exploration for finding the most suitable micro-architecture and then RTL generation. The tool developed by the researchers in [3] clearly reveals that the design flow only considers objectives such as required hardware resources, throughput and software execution times for designing their system. Although these parameters can considerably affect system performance, modern trends suggest that considering only these parameters during the high-level synthesis design process is insufficient for high-performance requirements. Moreover, due to the rapid emergence of handheld devices, power and energy consumption have now become one of the major contributing factors toward system performance. For computation-intensive applications, power and energy con-

* Corresponding author. Tel.: +1 416 979 5000; fax: +1 416 979 5280.
E-mail address: rsedagha@ee.ryerson.ca (R. Sedaghat).

0141-9331/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.micpro.2011.02.004

sumption increases heavily thereby degrading the system reliability and increasing design cost. The survey shown in [4] reveals that the current research trend is toward finding ways to minimize power consumption as much as possible to improve system reliability. Therefore, the overall design flow for high-level synthesis must take into consideration power consumed in addition to the area occupied and execution time when developing automated high-level synthesis tools. Thus, there needs to be a structured design methodology for automated high-level synthesis tools that takes into account constraints like area of the hardware resources and execution time, while minimizing power dissipation and increasing the reliability.

This paper is organized as follows: Section 2 highlights previous works related to design space exploration and application specific processor design. Section 3 of the paper gives a brief overview on the general multi-objective ASP design procedure. Section 4 describes about the proposed design flow for design space exploration of a multi-parametric objective application specific processor. It describes in detail the design space exploration approach. Section 5 describes the architectural synthesis process of an ASP designing. The implementation of the proposed design flow on an FPGA, simulation results, and chip layout generation are discussed in Section 6. Finally, Section 7 concludes the paper with a discussion on the utility of the proposed approach either as a standalone ASIC at silicon level or in System-on-chip (SoC) design.

2. Previous works

In general, an engineering problem can be described as a phenomenon of analyzing and managing the trade-offs between contradictory design objectives. A commanding method of mitigating this conflict and arriving eventually at the final design is the multi-objective exploration. The problem of obtaining a comprehensive Pareto-optimal set has been the focus of many researchers. In [5], the problem was addressed by suggesting order of efficiency, which assists in deciding preferences among the different Pareto-optimal points. The research work shown in [6] suggests that identification of a few superior design points from the Pareto set is enough for an excellent design process. In [7], an evolutionary algorithm, namely the Genetic Algorithm (GA), has been suggested to yield better results in the design space exploration process. The use of GA has also been suggested as a framework for DSE of data paths in high-level synthesis [8]. Another approach was introduced in [9] that were based on Pareto-optimal analysis. According to that work, the design space was arranged in the form of an Architecture Configuration Graph (ACG) for architecture variant analysis of the performance parameters. Those results proved quiet promising for architectural synthesis of digital systems. Furthermore in [10] and [11], authors described another approach for DSE in high-level systems based on binary encoding of the chromosomes. Work shown in [12] for DSE suggests an evolutionary algorithm for successful evaluation of the design for an application specific SoC. Although works such as [7,8,10–12] have all successfully performed design space exploration, but the above-mentioned approaches all use GA for arriving at the optimal result. GA has a tendency to produce good solutions in an acceptable time span, but requires lot of iterations (and hence time) to yield a superior result. Hence, although efficient, GA is inherently slow in nature. Thus due to time to market pressure, approaches such as [9] based on Pareto-optimal analysis are preferred.

Furthermore lot of works has been done in the area of design and synthesis of an application specific processor. For instance, authors in [13] discusses about the challenges faced in designing retargetable compilers and synthesis tools for an ASP for portable digital communications and multimedia systems. Authors in [14] have researched in the area of system design. They have tried to reach a level of application specific system integration. They have mainly focused on going beyond the level of circuit integration into system integration by integrating application specific processors with different architectures. Besides above, works such as [15] present a methodology for custom processor core construction using C code by quantifying the characteristics of C code in terms of operation types, available parallelism and other metrics. Moreover authors in [16] clearly outline the general design steps of a synthesis tools for realizing application specific processors that can be used in many scientific applications.

3. Overview on multi-objective ASP design

Efficient ASP design from the high level involves the following steps: formulation of the problem, exploration of the design space by analysis of the various possible architectures and selection of the best possible configuration, final scheduling and binding of the best architecture, and ultimately development of the system structure at RTL consisting of data path and control path. Scheduling refers to the determination of the actual utilization of a specific resource at a certain instance of time. Hence, the job of synthesis is to take a given required specification for the behavior of the system as well as the set of parametric goals and operational constraints and to finally find such a structural block that reflects the given required behavior. The initial step of synthesis is to compile the behavioral specification into an internal representation. The next step is to apply high-level transformation techniques with the aim of optimizing the behavior as per the desired performance [2]. Since the trend is toward fast and accurate designing with rapid time to market requisite, the design process should be the most optimum and efficient in terms of utility. Earlier researchers in [17,18] also investigated this trend and highlighted its significance. Since most of the system design has now become heavily dependent on accurate and structured high-level design flow, the current need is to formalize this design methodology for systems such as ASP. For example, power, area, and execution time are the most recognized constraints considered for judging the performance of the system. In this work hardware area, execution time and power consumption are used as performance metrics and operating constraints.

$$(1)\ \text{Area}(A_R) = f_A(A_1, A_2, \ldots, A_k) \tag{1}$$

where $A - i = f_i(N_{Ri}, A_{Ri}), i = 1, 2, \ldots, k;\ 1 \leqslant i \leqslant k,\ N_{Ri}$ – number of resource R_i; A_{Ri} – area per unit resource R_i, 'k' – total number of available resource.

The total hardware area can be expressed as the function of the sum of hardware area of all available resources, where area of each available resource is the function of the number of resources and the area per unit occupied by each resource. Therefore Eq. (1) above represents the sum of the area occupied by each resource, e.g., adder/subtractor, multiplier, divider, etc., and also clock frequency oscillator.

$$(2)\ \text{Power}(P_i) = f_P(f_c, A_R) \tag{2}$$

f_c – clock frequency of the system.

Power consumption can be expressed as the function of the total area occupied by the resources (A_R) and the power consumed at particular frequency of operation (f_c). Therefore, total power consumption is the product of the total area occupied by the resources (A_R) and the power consumed at a particular frequency (f_c).

$$(3)\ \text{Execution time}(T_R) = f_T(N_D, L, T_{cycle}, fc) \tag{3}$$

N_D – number of processing data elements, L – individual latency of the processing element, T_{cycle} – cycle time.