Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

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

Customizable embedded processor array for multimedia applications

Mehmet Tükel[a,*,1], Arda Yurdakul[b], Berna Örs[a]

[a] Department of Electronics & Communication Engineering, Istanbul Technical University, Istanbul, Maslak TR-34469, Turkey
[b] Department of Computer Engineering, Boğaziçi University, Istanbul, Bebek TR-34342, Turkey

ARTICLE INFO

Keywords:
Customizable Processor Array
Flexible instruction
Image processing hardware
Domain specific computing
Time-to-market

ABSTRACT

We are proposing a Customizable Embedded Processor Array for Multimedia Applications (CPAMA). This architecture can be used as a standalone image/video processing chip in consumer electronics. Its building blocks are all designed to achieve low power and low area, thus it is a good candidate for low cost consumer electronics. Our contribution is, designing a configurable embedded multimedia processor array considering the nature of image/video processing applications. This approach is considered in all the basic blocks of the architecture. Because of its configurable architecture and ability to connect with other devices, it may be used in a large domain of applications. Our architecture is purely implemented with VHDL. It is not dependent on any technology or design software. We have implemented our architecture for different applications on a Xilinx Virtex-5 device and as a number of Application Specific Integrated Circuits (ASIC) by using 90 nm CMOS technology. Experimental case studies show that CPAMA has better or comparable results to the existing similar architectures in terms of performance and energy consumption. Our studies show that throughput of CPAMA is 0.3x–2.4x times better than ADRES. Energy consumption of CPAMA is 31–50% less than ADRES. On the other hand, in one configuration of IDCT application, CPAMA provides 56% less throughput and consumes 55% more energy than ADRES.

1. Introduction

Computing hardware design methodology has evolved significantly over the years. As chips get larger and complexity of each design increases, flexibility and quick time to market in the form of reprogrammable/reconfigurable chips and systems increase in importance [1]. Several Multi Processor System on a Chip (MPSoC) and Coarse-Grained Reconfigurable Architectures (CGRA) have been proposed in recent years [2–4]. Using CGRAs may be preferred for several reasons such as speed, area, power or IP re-usability [3]. Furthermore, comparing to Field Programmable Gate Arrays (FPGA), CGRAs have a shorter reconfiguration time. CGRAs are suitable for systems that require intensive computations. By adjusting the number and structure of processing elements on a CGRA, we can obtain an architecture that meets the requirements of the computation.

Image/video processing is an area where algorithms need intensive computation with high performance. Handling this kind of computation usually requires custom hardware [5]. Considering today's technology, every portable device tends to have a camera, e.g. glasses, watches, smart phones, etc. Each device has its own configuration and requires mostly different features. Designing dedicated hardware for image processing tasks for every device is time consuming and not economically feasible at all. In most devices, image processing tasks are handled using System-on-Chips (SoC) with DSP or GPU cores. If a designer chooses to use commercial SoCs, he/she has to accept what the chip offers, in terms of speed and power dissipation. Those architectures may include redundant parts that might not be used at all. This redundancy leads to extra chip area usage and power dissipation. On the other hand, implementing an image processing task on a CGRA yields efficient results in terms of area, power dissipation, or speed comparing to commercial SoCs [6]. Time-to-market of an image/video processing system, which is implemented on customizable cores like CGRAs, is less than that of a custom Application Specific Integrated Circuit (ASIC) [7]. Besides, it is easy to adopt such systems for later alterations. Consequently, we can say that CGRAs are suitable for image/video processing tasks of low power, low cost consumer electronics.

In this paper, we introduce a Customizable Embedded Processor Array for Multimedia Applications (CPAMA). CPAMA consists of a processor array for intensive computation, and a host processor for control and coordination with other devices. Our configurable architecture is designed by considering the nature and requirements of image processing algorithms:

[*] Corresponding author.
E-mail addresses: tukel@itu.edu.tr, mehmet.tukel@ankasys.com (M. Tükel), yurdakul@boun.edu.tr (A. Yurdakul), orssi@itu.edu.tr (B. Örs).
[1] Anka Microelectronic Systems.

http://dx.doi.org/10.1016/j.vlsi.2017.09.009
Received 25 January 2017; Received in revised form 6 August 2017; Accepted 29 September 2017

- CPAMA processes a multimedia application in sequences of image blocks. Hence, we design a configurable processor array which concurrently processes all pixels in an image block.
- Each processor of CPAMA can also be configured according to the position of a pixel in an image block depending on the application.

This architecture can be used for domains that require intensive computation such as image/video processing, and scientific computations that can be mapped onto a 2 dimensional (2D) processor array.

This paper is organized as follows: In Section 2 we mention the related architectures in literature and demonstrate the differences with the proposed CPAMA. In Section 3, we explain the basic concepts that we refer in CPAMA design. In Section 4 we present the configurable hardware architecture of CPAMA in details. In Section 5, we present our case study implementations and make comparisons with the existing similar architectures. Finally in Section 6, we make our remarks on the CPAMA architecture and conclude the paper.

2. Related works

Mei et al. [3] proposed a template-based CGRA called Architecture for Dynamically Reconfigurable Embedded System (ADRES). Coarse grained reconfiguration refers to reconfiguration in relatively high level modules, not in logic blocks or in Look Up Tables(LUT) as in an FPGA. A design tool, namely Dynamically Reconfigurable Embedded System Compiler (DRESC) [8], is used for this architecture to generate the design. Propagating data, in other words performing iterations, is implemented in a stream manner. Total performance of the array is strictly related to the effectiveness of scheduling and mapping of the application code onto processing elements, which is handled by DRESC tool. It is known that optimum scheduling in DRESC is an NP-Hard problem. Therefore, the outcome of the scheduler, which is implemented using a heuristic method, is expected to be a sub-optimal solution. Another work related to ADRES [6] suggests that a failure in performance increase despite increasing the size of the array may be caused by a lack of scalability of the scheduling algorithm.

Marshall et al. [9] proposed another CGRA called CHESS, where multimedia applications are taken into account. Despite the reconfiguration word in its definition, most of the features of this array are kept fixed, e.g., number of registers, processors, instructions, etc. The reconfiguration is performed by only changing the program memory. CHESS can be considered as predecessor of ADRES.

Related to CGRAs, data partitioning and instruction scheduling techniques are also studied [10]. The target architecture is a variant of ADRES. Moreover, a recent study [11] focuses on power optimizations on the same target architecture. In these two studies [10,11], the emphasis is not on proposing a new architecture, but on instruction scheduling, data partitioning techniques for a CGRA like ADRES in order to achieve better speed and power consumption results.

Eichel [12] proposed MEP architecture for developing multimedia applications. The architecture consists of a RISC processor and an accompanying VLIW co-processor. The architecture has only instruction level parallelism. The configurable part of the architecture is the VLIW part. It is explained that, the RTL definition of the configurable part is generated based on a customised instruction-set architecture.

Chu et al. [13] proposed a programmable architecture called UniCore. This design is optimised for MPEG4 encoding. The whole architecture is not reconfigurable. It is composed of a 32-bit conventional processor, DSP like units and 4 co-processors. Programmability is achieved by the firmware that runs on the processor and co-processors.

Başsoy et al. [14] proposed an FPGA based customizable processor architecture called SHARF. SHARF has multiple ALU units controlled by the same control unit. ALUs receive instruction addresses from the same bus which is driven by the control unit. ALUs are tightly coupled with the control unit. In this architecture, tightly coupling may cause communication overhead, and moreover may restrict scalability.

Masselos et al. [15] concentrated on low power mapping of multimedia applications on VLIW multimedia processors. They searched methods for mapping tasks on the commercial processors rather than designing their own architectures.

Sanghai and Gentile [16] explored software parallelism in multimedia applications using a dual-core DSP. It is expressed that developing scalable parallel software greatly depends on the efficient use of the interconnect network, memory hierarchy, and the peripheral resources. While designing our CPAMA architecture, we have considered the methods which are proposed for software parallelism in [16].

Rashid et al. [17] proposed implementation of an application specific instruction set processor using a software called LISATek, which is now owned by Synopsys [18]. In this study, the speed-up relies on instruction level parallelism only. A RISC processor provided by LISATek is extended by processor data-path extension using the mechanisms in the tool. In CPAMA, not only instruction level parallelism, but also processor level parallelism is targeted.

Göhringer and Becker [19] proposed a runtime reconfigurable architecture called Runtime Adaptive Multi-Processor System-on-a-Chip (RAMPSoC). Parallel processing elements of the architecture are connected through a Network-on-Chip (NoC) called Star Wheels. This network is composed of groups that may have different number of processing elements. Processing elements of a group can communicate with each other through a switch, and processing elements of different groups can communicate through other larger switches.

Different digital implementations of Cellular Neural Network, which is an analog image processing structure, are proposed in several studies [20–23]. These architectures are capable of filter based image/video processing algorithms.

A configurable video decoder architecture [7] was proposed for mobile terminals. This architecture consists of a conventional application processor accompanied by a co-processor. The co-processor is composed of basic functions (hardware blocks) of H.264 decoder such as loop filter, motion compensation and integer transformation. Communication between the application processor and co-processor is implemented with a bus architecture. This study aims to decrease the time-to-market of a system that requires video decoding, and proposes an adaptable architecture for future standards by making configurable parts.

STP engine [24] is a multi processor accelerator IP, currently provided by Renesas Inc. It is used with a compiler tool called Musketeer [25]. Different stream applications [25,26] are implemented using STP engine. The current version [24] has 256 processing cores with 8-bit word length. STP engine can be considered as a fixed size CGRA with fixed word length.

The literature can be classified into three groups: (1) CGRAs, (2) architectures essentially built for specific applications, and (3) technology/device dependent architectures.

The difference between proposed CPAMA and CGRAs [3,9–11] is that CPAMA consists of fully customizable processors, whereas CGRAs consist of configurable functional units, like Arithmetic Logic Units (ALU). Besides, data are shared by multi-port register files in CGRAs, yet this task is handled through NoC with packets in CPAMA. Some CGRAs [3] have scalability issues. It is hard to comment on performance values in some studies [10,11], because the results are given as normalized values. Last but not least, the problem that needs to be solved on ADRES [8] is stated as a loop expansion problem. Instead in CPAMA, the nature of image processing algorithms is considered as explained in Section 3. Since STP [24] is a hard IP, the number of processing cores and the word length are fixed. On the other hand, in CPAMA, the full architecture is compile-time configurable, including the number of processing cores, word-length, array size and dimensions. Yet, contexts of CPAMA are also generated offline as it is done in STP.

The architectures mentioned in [7,13,20,21,22,22] are essentially