Micro-TASK Processing in Heterogeneous Reconfigurable Systems

Sebastian Wallner
Department of Distributed Systems, Technical University of Hamburg-Harburg, Schwarzenbergerstrasse 95
D-21073 Hamburg, Germany
E-mail: wallner@tu-harburg.de
Revised July 14, 2005.

Abstract New reconfigurable computing architectures are introduced to overcome some of the limitations of conventional microprocessors and fine-grained reconfigurable devices (e.g., FPGAs). One of the new promising architectures are Configurable System-on-Chip (CSoC) solutions. They were designed to offer high computational performance for real-time signal processing and for a wide range of applications exhibiting high degrees of parallelism. The programming of such systems is an inherently challenging problem due to the lack of an programming model. This paper describes a novel heterogeneous system architecture for signal processing and data streaming applications. It offers high computational performance and a high degree of flexibility and adaptability by employing a micro Task Controller (mTC) unit in conjunction with programmable and configurable hardware. The hierarchically organized architecture provides a programming model, allows an efficient mapping of applications and is shown to be easy scalable to future VLSI technologies. Several mappings of commonly used digital signal processing algorithms for future telecommunication and multimedia systems and implementation results are given for a standard-cell ASIC design realization in 0.18 micron 6-layer UMC CMOS technology.

Keywords system-on-chip, reconfigurable heterogeneous architectures, configuration instructions, descriptors, parallel processing system, signal processing

1 Introduction

Today’s and future telecommunication and multimedia services require high flexibility and high speed signal processing. The flexibility requirement points to the need of various communication, audio and video algorithms which differ in complexity. They have mostly a heterogeneous nature and comprise several sub-tasks with real-time performance requirements for data-parallel tasks. At present, their complex and time consuming algorithms are implemented in dedicated Application-Specific Integrated Circuits (ASICs) which come along with their lack of flexibility, long design cycles and high design cost. As time-to-market is increasingly critical, system adaptivity and algorithmic complexity grows, a consequent reuse of hardware components is necessary. A way to solve the flexibility and adaptability demands has been to use General Purpose Processors (GPP) or Digital Signal Processors (DSP), i.e., trying to solve all kinds of applications running on a very high speed microprocessor. A major drawback of using these general-purpose devices is that they are extremely inefficient in terms of utilizing their resources to best take the advantage of data-level parallelism in the algorithms. From there, a combined requirement on providing flexibility, adaptability and high performance is a challenging design issue for future architectures. Today’s demands motivate the use of Configurable Systems-on-Chip solutions which can be realized by integrating reconfigurable (re-usable) and programmable hardware components. In contrast to processors, they totally lack programming models that would allow forward compatibility to other architecture families and device independent compilation. Applications are compiled (synthesized) to a fixed size hardware structure. The resulting configuration instructions cannot be reused for devices of different types or sizes. In comparison, Instruction Set Architectures (ISA) defines instructions which decouples the compiler from the underlying hardware organization. This allows the ISA to change as long as the defined instructions are still supported. Programs can be used on new ISA processors without a recompilation task.

The approach in this paper describes a heterogeneous reconfigurable system approach which provides a programming model to overcome the restrictions mentioned above. It combines a wide variety of macro modules resources including a MIPS-like scalar processor core, coarse-grained reconfigurable processing arrays, embedded memories and custom modules supervised by a micro Task Controller. In the architecture, functions can be dynamically assigned to physical hardware resources such that the most efficient computation can be obtained. For these purpose, a set of basic configuration instructions (descriptors) with different functionalities are defined. Together with Micro-Tasks and streams (connections between them), a programming model is presented that compilers or mapping tools can target. As a result, the architecture can be forward compatible to other architecture families with variable numbers of reconfigurable processing cells for different performance features. A further key issue includes the system partitioning in order to provide efficient mapping of application kernels, the macro-module resource implementation and transfer of data. This paper is organized as follows. Section 2 gives an overview of the related work. Section 3 presents the reconfigurable processing array and the
hardware virtualization concept which based on previous research activities. Section 4 introduces the architecture composition, the system control mechanism and the scalar processor implementation in detail. Section 5 presents the programming paradigm. Algorithms mapping and performance analysis are presented in Section 6. Finally, Section 7 discusses the design and physical implementation while conclusions and future work are drawn in Section 8.

2 Related Work

There have been several research efforts as well as commercial products that have tried to explore the use of reconfigurable and System-on-Chip architectures. They integrate existing components (IP-cores) into a single chip or explore new architectures.

In the Pleiades project at UC Berkeley[4], the goal is to create a low-power high-performance DSP system. A general purpose microprocessor is surrounded by a heterogeneous array of autonomous special-purpose processing units (satellites) communicating over a reconfigurable communication network. The reconfigurable architecture CS2112 Reconfigurable Communication Processor[5] from Chameleon Systems couples a processor with a reconfigurable fabric composed of 32-bit processor tiles. The fabric holds a background plane with configuration data which can be loaded while the active plane is in use. A small state-machine controls every tile. The embedded processor manages the configuration and streaming data. The chameleon chip has a fixed architecture targeting communication applications. MorphoSys from the University of California Irvine[6] has a MIPS-like “TinyRISC” processor with extended instruction set, a mesh-connected 8 by 8 reconfigurable array of 28-bit ALUs. The “TinyRISC” controls system execution by initiating context memory and frame buffer loads using extra instructions via a DMA controller. MorphoSys offers dynamic reconfiguration with several local configuration memories. The architecture model proposed here provides coarse-grained reconfigurable processing cells and a pipelined configuration concept with descriptors. It provides a programming model based on micro-tasks and data streams for communication. The architecture includes a micro Task Controller with a simple instruction set and a single local configuration memory. The micro task instructions and the descriptor set are device independent. They can be directly reused in other architecture families with different performance characteristics (different numbers of reconfigurable processing cells) without a recompilation task. In contrast to the other architectures, the implemented MIPS like scalar processor core is not involved in the configuration process.

3 Reconfigurable Processing Cell Array

High performance requirements in the application domains point to the need for reconfigurable architectures to better exploit the parallelism in algorithms. Coarse-grained reconfigurable architectures can enhance the performance of critical loops and computation intensive function. They offer abundant parallelism, high computational density and flexibility of changing behavior during runtime. A reconfigurable architecture has been constructed, which is built on previous research activities in identifying reconfigurable hardware structures[7].

A Reconfigurable Processing Cell Array (RPCA) has been designed which targets applications with inherent data-parallelism, high regularity and high throughput requirements. It supports data level parallelism as used in vector and MMX architectures and has the potential to perform signal processing tasks more efficiently in terms of performance. The architecture is based on a synchronous multifunctional pipeline flow model using reconfigurable processing cells and configurable data paths. A configuration manager allows run-time and partial reconfiguration. It provides efficient hardware virtualization to remove the fixed-size constraints present in conventional reconfigurable devices.

The array consists of configurable coarse-grained processing cells linked to each other via broadcast and pipelined data buses. It is fragmented into four parallel stripes which can be configured in parallel. A stripe includes four reconfigurable processing cells. It has available an independent 128 x 48-bit dual port register-pad memory and a 64 x 24-bit descriptor memory for the configuration instructions. Fig.1 shows a single cluster with overall 16 processing cells.

![Structure of a cluster with the configuration manager, the processing cells (Cell) with the switch boxes for routing, the dual-port scratch-pad memories and the descriptor memory. In order to adjust configuration cycles, two pipeline registers for every stripe are implemented. (Pipe Register).](image)

In order to lower bus bottlenecks and to broadcast data words simultaneously to the processing cells, four independent 24-bit broadcast data buses for every stripe are implemented. In addition, the processing cells in every stripe are connected via 48-bit pipelined data buses.