A Study on a Multitasking Environment for Dynamically Reconfigurable Processors

(Abstract)

VU MANH TUAN

A dissertation submitted in partial fulfillment of
the requirements for the degree of

DOCTOR OF PHILOSOPHY

School of Science for Open and Environmental Systems
Graduate School of Science and Technology
Keio University

2009

Abstract

This thesis presents fundamental work in the area of multitasking on dynamically reconfigurable processors. The main objective is to investigate necessary mechanisms in order to build a true multitasking execution environment on such devices.

Reconfigurable devices promise to be a valuable alternative to conventional computing devices such as general-purpose processors and application specific integrated circuits. The hardware circuit of reconfigurable architectures is not static but adapted to the target applications at the run time. Through dynamic hardware customization, reconfigurable architectures potentially achieve a better performance than microprocessors while maintaining a higher level of flexibility than application specific integrated circuits.

Due to the increase of logic capacity of modern fields-programmable gate arrays expanding beyond millions of system gates, and through the broad use of coarse-grained dynamic reconfiguration, it has become feasible for several applications to share a single high density device. In other words, reconfigurable hardware devices can now execute several hardware tasks in parallel. However, developing applications that share a device is difficult as the current design flow assumes the exclusive use of the underlying reconfigurable resources, and critical mechanisms for a multitasking environment on such devices are still immature.

The use and implementation of similar mechanisms modeled on a microprocessor-based software operating system would allow the full benefits of multitasking on dynamic reconfiguration systems to be realized. In this thesis, based on equivalent concepts found in the software domain with specific characteristics of hardware implementation, several necessary mechanisms to support a multitasking environment on dynamically reconfigurable processors are presented.

Specifically, I consider following issues: (1) how to improve the throughput in data flow driven applications by using an appropriate application model and a mapping algorithm to map a target application onto a underlying coarse-grained reconfigurable device consisting of multiple hardware execution units, (2) how to implement a preemption mechanism for hardware tasks, (3) how to extend the multi-thread execution model in a tile-based architecture into a multicore architecture, and the impact of the core size on performance and internal fragmentation in a multicore reconfigurable architecture.

Efficiently mapping applications partitioned into multiple threads of control or tasks onto a coarse-grained dynamically reconfigurable processing array is crucial for further exploiting the parallelism of applications. Basically, a certain speedup of reconfigurable devices over traditional microprocessors can be achieved by exploiting the inherent parallelism of a target application at a larger scale apart from instruction-level parallelism. However, in many cases, the parallelism of an application is not enough for the reconfigurable array to be utilized efficiently. In such a condition, one of the methods to improve performance is to make the best use of stream-level pipelined execution; or

in other words, to exploit the task-level parallelism. That is, a total process is partitioned into small independent sequential processes, which are separately mapped onto a coarse-grained underlying reconfigurable array architecture comprising of multiple hardware execution units. By investigating the trade-off between the size of a tile group and delay as well as execution time, an optimized mapping algorithm is proposed and evaluated compared with the single-process execution model and other mapping versions.

Effectively implementing a hardware task preemption scheme, which is a critical mechanism in multitasking operating systems where an executing task is temporarily suspended, and a new task is executed or a previously interrupted task is resumed, on coarse-grained dynamically reconfigurable processors is another mechanism toward a multitasking environment. Different from well-known task preemption in software operating systems based on microprocessors, hardware task preemption accompanies problems on how to suspend and resume hardware execution, how to save and restore context data within a certain overhead. Based on the resource requirement and the state transition graph of a target application, the concept of preemption points is introduced and a static preemption approach is proposed. Basically, an application can only be preempted at certain computation states where resource usage is small. Moreover, the steps of the proposed method are integrated into the design flow in order to help developers at the design time.

The development of multicore architectures in microprocessors has recently become prevalent and achieved advantages over traditional single-core devices. Multicore architectures have proved to be a suitable platform for a true multitasking environment at the task level where tasks can execute in parallel. I introduce a multicore reconfigurable architecture in which cores have a certain size and are connected by an interconnection network composed of routers. Mapping multiple hardware tasks onto the proposed architecture and designing an efficient inter-task communication mechanism become important research problems. The proposed on-chip network will act not only as an efficient communication mechanism between tasks (both hardware and software tasks), but also as a support method for mapping processes onto the underlying reconfigurable array. First of all, a tile-based architecture and the proposed multicore architecture are examined to see how each of those affects on the performance of target applications. The channel bit width to transfer data in the interconnection network of the multicore architecture is taken into account. Next, the size of core in the proposed multicore reconfigurable architecture is an important factor in balancing performance and resource utilization. Therefore, this is evaluated by implementing real applications on the proposed architecture and measuring correspondent parameters.