CCIT Report #579
March 2006

Architecture and Routing in NOC Based FPGAs

ABSTRACT

We present a novel network-on-chip architecture for future programmable chips (FPGAs). We examine the required capacity allocation for supporting a collection of typical traffic patterns on such chips under a number of routing schemes. Since in FPGA, traffic patterns are determined at configuration time, after physical links have been synthesized, it is important to employ routing schemes that allow for high flexibility in the permissible traffic pattern during configuration. We propose a new routing scheme, Weighted Ordered Toggle (WOT), and show that it allows for high design flexibility with low per-link capacity. Moreover, WOT utilizes simple, small-area, on-chip routers and has low memory demands.

Categories and Subject Descriptors

Multiprocessors and Network-on-chip
• Network-on-Chip (NoC)

Keywords

Network-On-Chip (NoC) architecture, FPGA, routing, capacity.

1. INTRODUCTION

In recent years, much attention has been dedicated to Networks-on-Chip (NoCs) (see, e.g., [1],[3],[4],[5],[6],[8],[11], [12]), which provide scalable solutions for on-chip communication in the sub-micron era. However, there is no "one size fits all" NoC architecture, as different silicon systems have very different requirements from their NoCs. For example, in a System-on-Chip (SoC), the network usage patterns are known a priory. Hence, the NoC can be synthesized with exactly the right link capacities for supporting the required usage [7]. In contrast, in a Field Programmable Gate Array (FPGA), the communication pattern is determined when the chip is configured to implement some specific functionality, and hence its physical layout must provide the flexibility to support a variety of traffic patterns.

In this paper, we focus on NoC design for FPGAs. A distinguishing feature of FPGA systems is that they include a combination of *hard* and *soft* functionalities. The hard functionality is implemented in silicon; it typically includes special purpose modules like processors, multipliers, external network and memory interfaces, etc. The soft functionality is configured using programmable elements (gate arrays, flip-flops, etc.). Modern FPGAs contain hundreds of thousands of programmable elements, in addition to special purpose modules (0). As technology scales, the sheer number of logic units will render a flat FPGA chip design unmanageable. We thus envision a future FPGA that is organized hierarchically; whereby the chip is divided into high-level regions (some programmable and some hard), interconnected by a NoC. Our novel architecture is detailed in Section 2.

When architecting an FPGA NoC, one has to decide which functionalities to implement as hard cores and which to leave soft. There is a tradeoff between the flexibility offered by soft designs and the better performance offered by hard ones. Since inter-module communication is often a bottleneck, it is important to design the NoC architecture for high performance. We therefore advocate laying out the network infrastructure, including metal wires and hard-coded routers in silicon. At the same time, in order to allow for maximum flexibility, the NoC infrastructure should be able to accommodate multiple routing schemes and a large variety of traffic patterns. To this end, we allow network interfaces to be soft. That is, each module has a *configurable network interface* (CNI) Simplistic routing schemes, like XY, can employ small interfaces, whereas more elaborate source-routing schemes ([1],[9]) may have the interfaces store large routing tables.

The main challenge is exploiting network resources efficiently, i.e., supporting a large number of program designs while investing minimal resources (wires and logic). In this context, there is an inter-play between the link capacity requirements and the routing scheme used to route packets between modules. A routing scheme that balances the load over all links readily supports more designs using smaller link capacities than an unbalanced one. Traditional algorithms like XY lead to unbalanced capacities, and are therefore not suitable for