

A DAMQ SHARED BUFFER SCHEME FOR NETWORK-ON-CHIP

Jin Liu and José G. Delgado-Frias
School of Electrical Engineering and Computer Science
Washington State University
Pullman, WA 99164-2752
{jinliu, jdelgado}@eecs.wsu.edu

ABSTRACT

In this paper we present a novel shared buffer scheme for network on chip applications. The proposed scheme is based on a dynamically allocated multi queue self-compacting buffer. Two physical channels share the same buffer space. This in turn provides a larger available buffer space per channel. The proposed scheme has similar performance using only sixty three percent of the buffer size that is used in traditional implementation for NoCs. In addition, using same size buffers the proposed scheme outperforms existing approaches by 1% to 2% in throughput. The proposed scheme also has a better utilization of the available buffer space.

KEY WORDS

DAMQ, Self-compacting, buffer and Network on Chip.

1. Introduction

As technology allows greater integration system-on-chip (SoC) designs have a great potential. By the end of this decade SoC will have up to four billion transistors [1]. SoCs are anticipated to have a number of components (or modules) that will interact to compute a solution. These components will need to communicate to transfer data and/or control information. Having dedicated connections between any given modules could be extremely complex as the number of modules increases. An alternative could be use an interconnection network within the chip. An interconnection network on chip needs be restricted in terms of area due to the constraints of being in a single chip. Thus, it is extremely important to use schemes that require less hardware resources as well as provide a good performance.

Virtual channel multiplexing across a physical channel is extensively used to boost performance and avoid deadlock. Each virtual channel is realized by a pair of buffers located on adjacent communicating nodes. As virtual channels are not equally used in many applications, if they share a common buffer, the whole buffer space will be better utilized. In this paper we present a novel shared buffer scheme that is based on a dynamically allocated multiple queue (DAMQ) buffer. This scheme provides similar performance as statically allocated multiple-queue (SAMQ) and DAMQ buffers using less buffer space and, therefore, requiring less hardware.

This paper has been organized as follows. In Section 2, background information is provided. In Section 3, the proposed buffer scheme is described. Performance evaluation results are reported in Section 4. Some concluding remarks are included in Section 5.

2. Background

Routing and buffer organization algorithms are two important design factors of a modern high performance wormhole switch for an interconnection network on chip. In this section we present a brief review of existing routing algorithms and DAMQ buffer schemes which are widely accepted as an efficient way to organize a switch's input buffer.

2.1. Existing routing algorithms

Routing algorithms are used to determine the messages path from source to destination. They can be implemented in two ways, which are deterministic and adaptive ways. Deterministic routing protocol chooses the path for a message only by its source and destination. All packets with the same source and destination pair will follow one single path. The packet will be delayed if any channel along this path is loaded with heavy traffic, and if a channel along this path is faulty the packet cannot be delivered. Thus the deterministic routing protocols are prone to suffer from poor use of bandwidth [8]. A common deterministic routing algorithm is dimension-order routing [9]. Adaptive routing protocols are proposed to make more efficient use of bandwidth and to improve fault tolerance of interconnection network. In order to achieve this, adaptive routing protocols provide alternative paths for communicating nodes. Several adaptive routing algorithms have been proposed, showing that message blocking can be considerably reduced, thus strongly improving throughput. [10] Among them, routing algorithms based on Duato's design methodology [11] are widely used. These routing algorithms split each physical channel into two virtual channel sets, the adaptive and the deterministic channels. When the paths of adaptive channels are blocked, a packet uses an escape channel at the congested node. If there is any free adaptive channel available at subsequent nodes, the packet can go back to the adaptive channels. As Duato's adaptive algorithms are

