DESIGN AND IMPLEMENTATION OF ARGO AND FALP METHOD ON NOC AND THEIR ANALYSIS

A. Chezhiyan, Mrs. M. R. Mahalakshmi

PG Scholer, Dept. of Electronics and communication Engineering, Sri Muthukumaran Institute Of Technology, Chennai, Tamilnadu, India, chelichezhiyan@gmail.com.
Asst Prof, Dept. of Electronics and communication Engineering, Sri Muthukumaran Institute Of Technology, Chennai, Tamilnadu, India, mahapras2005@yahoo.co.in

KEYWORDS

NOC, RRB, Argo, FALP

ABSTRACT

Most communication traffic in today's Network on Chips (NOC) is based on Router for volatile memory based designs. The NOC should be designed to efficiently handle the many-to-one communication pattern, data access to and from the routing controller. In this paper, we motivate the use of a separate network for the routing traffic and justify the power consumption and performance improvement obtained by using Argo and FALP methods, when compared to traditional round robin method. This new algorithm supports irregular topologies caused by faulty components in a network. Argo method shows how the congestion is avoided and state transition based Finite State Machine memory controller is designed for on-chip cores which optimize area up to 50% and 70% in power reduction using FALP Method.
I. INTRODUCTION

Network-on-Chip is a general purpose on-chip communication concept that offers better throughput, which deals with complexity of modern systems. Network-on-chip (NOC) is a complex interconnection of various functional elements. Thus there was need of such system that can modularity and parallelism, network-on-chip provide many such attractive properties and solve the problem of communication bottleneck. It basically works on the idea of interconnection of cores using on chip network.

II. NETWORK ON CHIP

Network-on-chip (NOC), comparatively a new concept that comeforth as a system-on-chip (SOC) communication methodology, take over many ideas from the computer networks, the knowledge domain in which the research on routers and packet switching has matured. A scheduling algorithm computes which packet has to be forwarded prior to the other packets. NoC structured as a 4-by-4 grid which provides global chip-level communication. It employs a grid of routing nodes spread out across the chip, connected by communication links. For now, we will accommodate a simplified perspective in which the NoC contains the following fundamental components. Network adapters implement the interface by which cores (IP blocks) connect to the NoC. Their function is to decouple computation (the cores) from communication.

Fig 1: NOC 4-by-4 grid structured NoC

The NoC in the figure could thus employ packet or circuit switching or something entirely different and be implemented using asynchronous, synchronous, or other logic.

III. REVIEW OF EXISTING METHOD

An effective adaptive routing algorithm can help minimize path congestion. However, conventional adaptive routing schemes use only channel-based information to notice the congestion status. Due to the deficiency of switch-based information, it is difficult to unveil the real congestion status of channel-based information along the routing path. Switch Congestion: When a packet is transmitted through the north output port, some packets that receive a failed output request must be blocked and then be queued at the input buffers. The routed packet has to wait for this channel to be released. Channel Congestion: Because of the limited input buffer size, the switch runs out of buffer space. Notably, due to the backpressure effect of the link-level flow control, switch and channel
congestion are severely correlated. That is, switch congestion in one router can reconstruct itself as channel congestion in one of the adjacent routers. Therefore, path congestion can start to build and spread from a congested switch to source nodes, which grows into a congestion tree. It highly degrades the overall system performance, especially in real-time applications with strict latency requirements.

Congestion-aware adaptive routing selects an output channel based on various types of network congestion information. Consequently, these selection functions can adjust path selection based on a time-variant congestion status. Congestion aware adaptive routing adopts two types of spatial information: local information and regional information. Local routing information considers local information such as the downstream buffer count and available flit slot to assess the traffic status.

IV. BLOCK DIAGRAM

Nios II is a 32-bit embedded-processor architecture designed specifically for the Altera family of FPGAs. Nios II architecture is a RISC soft core architecture which is implemented entirely in the programmable logic and memory blocks of Altera FPGAs. User-defined instructions accept values from up to two 32-bit source registers and optionally write back a result to a 32-bit destination register. DRAM plays vital role for system design such as RAM, cache memories etc. To increase row buffer locality, existing paper introduced a Thread row buffer to increase throughput and DRAM’s overall performance. TRB increases row hit rate by reusing row that consists of same information’s. Data access between DRAM banks are controlled by a logic controller consists of sequential elements. Logic controller is mostly accessed by every blocks in DRAM circuitry. Memory controllers contain the logic necessary to read and write to DRAM, and to "refresh" the DRAM. Without constant refresh, DRAM will drop off the data written to it as the capacitors leak their charge within a fraction of a second. Reading and writing to
DRAM is performed by selecting the row and column data addresses of the DRAM as the inputs to the multiplexer circuit, where the demultiplexer on the DRAM uses the converted inputs to select the correct memory location and return the data, which is then passed back through a multiplexer to consolidate the data in order to reduce the required bus width for the operation.

V. ARGO AND FALP METHOD

In this paper we describe the use of asynchronous routers in a time-division-multiplexed (TDM) network-on-chip (NOC), Argo, that is being developed for a multi-processor platform for real-time systems. TDM need a common time reference, and existent TDM-based Network on chip designs are either synchronous or mesochronous. We use asynchronous to accomplish a simpler, smaller and more robust, self-timed design. Our design provide the fact that pipelined asynchronous circuits also act as ripple FIFOs. Thus, it keep off the need for explicit synchronization FIFOs between the routers. Argo has interesting timing properties that allow it to tolerate skew between the network interfaces (NIs).

The paper presents Argo NOC-architecture and provides a quantitative analysis of its ability of absorb skew between the NIs by using a signal transition graph model and realistic component delays. Network-on-Chip (NoC) are known as the future communication infrastructure for many-core systems. They are capable of malfunction in the presence of the faults as technology sizes reduce proportionally the performance of the system. According to the results, links have failed 71% due to the Crosstalk fault. A new fault-adaptive and low power, calling FALP, method for Network on chip routers is presented in this article to extenuate their unexpected behavior through links. It reduces the switching power consumption overhead by keep track of the frequency and life time of faults. However, based on the VHDL execution of a Network on chip router, routing unit has a trifling area comparing to other components of an Network on chip router. In a router architecture less than 11% of an NoC router area is occupied by routing component embedded with FALP technique. The proposed fault-tolerant method for anNoC design FALP, is presented in this section.
FALP is an adaptive fault-tolerant method to increase the reliability of NoC design with considering the overhead of power consumption. This method is composed of fault detection and fault-tolerant techniques with different modes of operation depending on the frequency and lifetime of the faults in the system. It is proposed to be implemented in different zones of the network, which makes the routers to be able to utilize different fault detection methods independently. The other advantage of this technique is to prevent extra power consumption by employing the fault-tolerant method only the faulty zones rather than all the network. FALP method uses viterbi algorithm which is the most popular decoding approach for convolutional codes and it determines the minimum distance (hamming distance). It has three major parts in FALP method they are BMU(Branch Metric Unit),ACS(AddCompareSelect Unit), TBS(Trace Back Unit). A branch metric unit's function is to calculate branch metrics. The branch metrics are difference values between received code symbol and the corresponding modified branch words from the encoder trellis. Find out the difference with past value to calculate hamming distance. An ACS unit determines path costs and identifies lowest-cost paths. ACS (Add-Compare-Select) units are the most important block in FEC (Forward Error Correction) decoders. The survivor memory stores lowest cost bit-sequence paths based on decisions made by the ACS units. Trace back unit generate the original bit from BMU & ACS. Trace back unit restores an (almost) maximum-likelihood path from the decisions made by BMU.

The goal of this method is to use the benefit of different methods to have more reliable NoC design while keeping the cost low. Any fault detection method imposes area and power overhead to the system, which has a direct relation with the coverage of the fault detection method. The simple parity bit is chosen here as a fault detection method since its overhead is negligible. This method is capable of detecting odd number of errors. Retransmitting the faulty data embedded with some error correction method, with one clock cycle penalty, is chosen as the fault-tolerant method in this article. This method is repeated until the received flit has no error. The error correction method is selected based on the persistency of the fault occurrence in the system. Hamming method as a separable code is chosen in this article as the basic error correction scheme.

![Fig 6: Mode of Operations in FALP Method](image-url)

![Fig 7: Encoder Decoder of FALP Method](image-url)
The probability of having single faults are higher than multi-bit ones in a NoC design, so single faults are targeted in FALP method. The probability of having single faults are higher than multi-bit ones in a NoC design, so single faults are targeted in FALP method. Data flits are transmitted with parity bits by default in FALP method. A router changes its mode of operation once an error is detected by parity bits to retransmit the faulty flit. The retransmission sub component of the router calculate the corresponding Hamming code for the faulty flit and retransmit the data with the Hamming codes. The router repeats this type of flit transmission until there is no fault in the system and then switches back to the normal mode of operation. This scheme prevents increasing the total power consumption although it imposes some area overhead. In worst case that all of routers operate in faulty state the power consumption of the whole network is equal to a purely design embedded with Hamming method. The encoder/decoder are implemented between the output ports of the switch component and input buffer ones of the neighbor router. A counter is proposed to be implemented in the Encoder component to keep track of the number of fault occurrence in a fixed period of time. The encoder employs the value of this counter as a transition point to the faulty state from no fault state to faulty state. This value is defined based on the running application and the susceptibility of the system to the fault occurrence. For the sake of simplicity the boundary value is defined as 1 in this article. The router switches to faulty states as soon as an error is detected and it will stays in it until the Error signal changes to ‘0’ value. Assuming the boundary value of the counter is defined as 1, the maximum penalty of error correction is one clock cycle. However, there is no more penalty if there are still more errors in the system since the encoder stays in faulty state until the faults are diminished.

VII. SIMULATION RESULT

In this paper the analysis is based on typical and constant gate delay values. In fig 11 shows the output waveform of existing system. and fig 12 shows the waveform of proposed system. We have explored the use of min-max delay intervals and it leads to higher throughput values. the below table shows the performance evaluation table and therefore reduced skew tolerance.
VIII. CONCLUSION

In this paper, we emulated the performance of Path congestion aware routing and Argo on a quartus II platform. The paper extended previous work on synchronous and asynchronous TDM-based NOCs by exploring the use of asynchronous routers that allow a truly GALS-style implementation of a NOC-based multi-core platform. We compared the performance of these two routing methods in terms of number of resources utilized, throughput, area, power, and delay. Path congestion aware routing consumes more resources, which means it utilizes more silicon area. Argo has a high clock frequency than the Path congestion aware routing, which means Argo could process data more quickly. In this design, we should make a trade-off among the resource or silicon area, maximum clock frequency and delay and choose suitable arbitration mechanism according to that.

IX. REFERENCES


