5K-5

On-Chip Router Architecture for Detecting Soft Errors

Shunya IGARASHI † Ryosuke SASAKAWA † Kenji KISE †

Tokyo Institute of Technology †

1 Abstract

As the silicon integration technology has grown, the frequency of soft errors has increased. If an error occurs on the control of an on-chip router for many-core processors, it will cause some issues such as deadlock and packet loss. To prevent these issues, we can prepare additional hardware for detecting soft errors in intra-router logic. A simple answer is to use redundant router. However, it requires too much extra hardware. On the other hand, additional hardware must be large enough to detect all the errors. In this paper, we discuss what kind of hardware should be added, and propose a method for error detection named Shadow Router.

2 Introduction

With the advance of semiconductor technology, many functional modules can be implemented on a single chip[1]. Large System-on-Chips (SoC) and many-core processors require on-chip interconnection known as Network-on-Chip (NoC). However, the reduction of feature sizes and operating voltage, along with improvement in transistor density, increases the soft error rate. Soft errors, caused by cosmic rays, will cause temporary bit inversion that affects the control of on-chip routers. It results in packet loss and deadlock of the network. While these issues are critical for dependable chips, we must prevent or detect mistakes in the routers.

In this paper, we present an approach for detecting soft errors by redundant routers or Shadow Router. The idea of Shadow Router is based on dual modular redundancy (DMR). It reduces some features unrelated to the control of the router. Moreover, for independence of the method from router architecture, we do not change router-specific control modules. In short, Shadow Router has the same control modules as the original router, reduced input buffers, and simplified data path.

3 Router Architecture

The architecture of the target router is shown in Figure 1. It applies X-Y dimension order routing and wormhole switching. Network topology is a 2-D mesh. One physical channel has four virtual channels. The organization of a flit is shown in Figure 2. The upper 12 bits are involved in the control, and the address of the destination node is written to the lower 16 bits of the header flit.

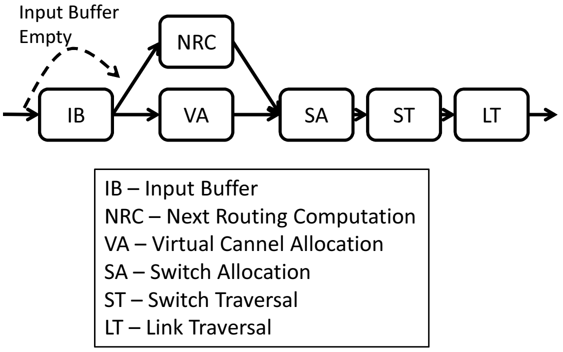

Figure 1: Baseline Router Architecture

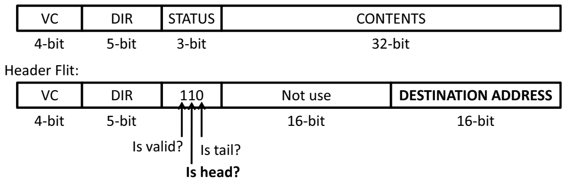

Figure 2: Flit Configuration Example

Input buffer (IB) is a queue of input flits. Next routing computation (NRC) computes the direction where a packet should go in the following node. The result is written in the direction bits (DIR) in the flits. Virtual channel allocation (VA) starts at the same time as NRC. VA computes the output virtual channel with the direction computed by the previous node. Switch allocation (SA) decides the connection of crossbar based on the result of VA. On switch traversal (ST) stage, the flits pass through the crossbar. They go to the next node by traversing the physical channel on link traversal (LT) stage.

4 Proposal of Shadow Router

The concept of Shadow Router does not depend on the router architecture. We consider the router-specific control modules as black boxes and let them unchanged in this paper. We only modify the input buffers. Shadow Router must be large enough to achieve the dependability equivalent to the redundant routers. Thus, we must keep several bits in the flit that are used by the control modules. On the other hand, while it should be as small as possible, we drop the other bits, that is, the bits unrelated to the control.

According to Figure 2, the router control needs the upper 12 bits of all the flits and the lower 16 bits of the header flit. Hence, Shadow Router requires 28 bits for the header flit and 12 bits for the other flits in the input buffers.

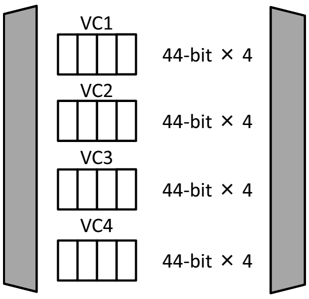

Figure 3: Original Router Input Channel Architecture

Figure 3 shows the input buffers for a single physical channel in the original routers. The original input buffer contains four flits per virtual channel and each entry has a constant (44) bits. In Shadow Router, the required bits are different between the header flit and the other flits.

We show the input buffers for a virtual channel in the Shadow Routers in Figure 4. In addition to the per-flit buffers, we prepare a buffer dedicated to the header flit. Since the original router employs wormhole switching, input buffers for a virtual channel does not have two or more header flits at once. If the incoming flit is a header flit, the lower 16 bits are stored to the dedicated buffer. When the router transfers the header flit to the crossbar in ST stage, the content of the dedicated buffer is also tranfered. If the incoming or outgoing flit is not the header flit, the dedicated buffer is not used.

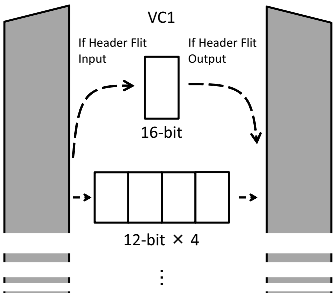

Figure 4: Shadow Router Input Channel Architecture

5 Discussion

Since we consider the control modules as black boxes, our method for error detection prepares the same control modules as the original, though the input buffers and the data path are reduced. It can detect all the errors, such as selecting wrong VC and reversing the order of packets, as long as more than one error does not occur simultaneously.

However, it may just be enough to detect some errors that cause serious effects [1]. There are dependencies between the control modules and the effects of some errors may be hidden by the other modules and nodes. Taking advantage of it, errors that has small effect for the system are ignored in such methods. What kind of errors should be detected is determined by not only router architectures but also system requirements.

If we stop considering the control modules as black boxes, we can further reduce the additional hardware for Shadow Routers with the knowledge from the methods that we have mentioned above. How to reduce the additional hardware without losing the versatility is future work.

Shadow Router considers only the control path: errors in the data path are not detected. However, there is a number of researches for detecting or correcting the errors on the data path[2][3]. We may also need to find which kind of methods fit our Shadow Router.

6 Conclusion

We proposed Shadow Router for detecting soft errors as a versatile method that can be applied to various routers. In the example that we assumed this time, the number of bits required for the input buffers was reduced to less than one-half. Moreover, if the flit size get larger, the relative size of Shadow Router becomes smaller.

We are going to evaluate the frequency and the hardware amount of Shadow Router relative to the original by implementing it with Verilog HDL. We will also reduce the additional hardware without losing the versatility by stopping considering the control modules as black boxes.

Acknowledgements

Mr. Shinya TAKAMAEDA-YAMAZAKI gives us the original router. We thank for his support.

This work is supported in part by Core Research for Evolutional Science and Technology (CREST), JST.

References

[1] D. Park, C. Nicopoulos, J. Kim, N. Vijaykrishnan, and C.R. Das. Exploring fault-tolerant network-on-chip architectures. In *Dependable Systems and Networks, 2006. DSN 2006. International Conference on*, pp. 93 –104, june 2006.

[2] Yu Kojima, Hiroki Matsutani, Michihiro Koibuchi, and Hideharu Amano. A low power network-on-chip using error correction and detection codes. In *Symposium on Advanced Computing Systems and Infrastructures*, pp. 3–10, may 2010.

[3] Wen-Chung Tsai, Deng-Yuan Zheng, Sao-Jie Chen, and Yu-Hen Hu. A fault-tolerant noc scheme using bidirectional channel. In *Proceedings of the 48th Design Automation Conference*, DAC '11, pp. 918–923, New York, NY, USA, 2011. ACM.