Fault Tolerant Asynchronous Networks-on-Chip

Guangda Zhang
Supervisor: Jim Garside
Co-Supervisor: Javier Navaridas
Advisor: Kung-Kiu Lau
School of Computer Science, University of Manchester

1 Introduction

Advanced semiconductor technology makes it possible to integrate more processing cores on a single chip, resulting in a growing demand for scalable and efficient on-chip communication fabrics. Network-on-Chip (NoC) [1] has emerged as a potential candidate to support scalable communication. Most existing NoCs are synchronous which are expected to suffer from serious problems caused by global clocks as the network scales. As alternatives, asynchronous NoCs [2] have drawn wide attention of both academic and industrial community. One of the main benefits in using asynchronous NoCs is their timing-robust nature. Quasi-delay-insensitive (QDI) circuits [3], which belong to an attractive family of asynchronous circuits, can tolerate nearly all delay variations. However, in the face of faults, asynchronous NoCs are not robust and can make errors.

Deep submicron technology is making electronic devices vulnerable to faults. Fault-tolerance has become an essential design objective for critical digital systems, especially in highly specialized fields such as aerospace, military and medical equipment.

The aim of this research is to explore different fault-tolerant techniques and propose a thorough fault-tolerant architecture for asynchronous NoCs.

2 Background

Built from routers and links, an asynchronous NoC is implemented as an asynchronous circuit where all communication is controlled by handshakes [2]. Processing elements (PEs), which are usually synchronous Intellectual Property (IP) cores, are connected to the network through network interfaces (NIs), resulting in a Globally-Asynchronous Locally-Synchronous (GALS) system [2]. NIs implement the transformation between the asynchronous domain and synchronous domains. Fig. 1 shows a GALS system which is structured by an asynchronous NoC.

A large number of asynchronous NoCs have been proposed but rarely have fault-tolerance capability, making them vulnerable to faults. Therefore, to provide fault-tolerance for asynchronous NoCs makes sense, to further extend their usage in the coming multi-core or many-core era.

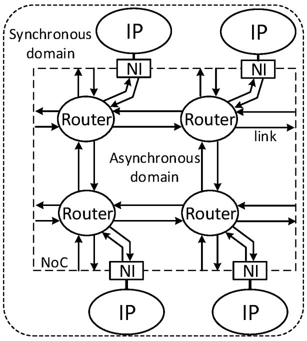


Fig. 1. A GALS system structured by an asynchronous NoC

Faults can be transient, intermittent or permanent [4]. Depending on their behaviour, intermittent faults can be taken as either transient or permanent during the fault management process, thus the fault-tolerance of asynchronous NoCs is divided into the toleration of transient and permanent faults.

Transient faults are prevalent and cause nearly 80% of failures [5]. They usually last for a short period and cause transient (or *soft*) errors when captured by memory components. Existing fault-tolerant techniques for asynchronous links usually either depend on strict timing constraints, corrupting the timing-robust feature of asynchronous circuits, or cause a large area and speed overhead making them difficult to use in practical asynchronous circuits. This research will explore a new, inexpensive fault-tolerant coding scheme to tolerate transient faults on QDI links. Besides data wires, wires transmitting acknowledgement signals are also protected, which has long been disregarded.

Permanent faults may happen at runtime due to the ageing process [4, 6] and will never be recovered by themselves, bringing lifetime reliability problems. There is rarely research on permanent fault detection and recovery on asynchronous NoCs. Besides data errors, permanent faults will violate the handshake protocol, resulting in a deadlock of the network. Many synchronous NoCs use syndromes collected from retransmissions to detect permanent faults, but in a deadlocked state, to perform a retransmission on asynchronous NoCs is difficult. This research will propose new tech-

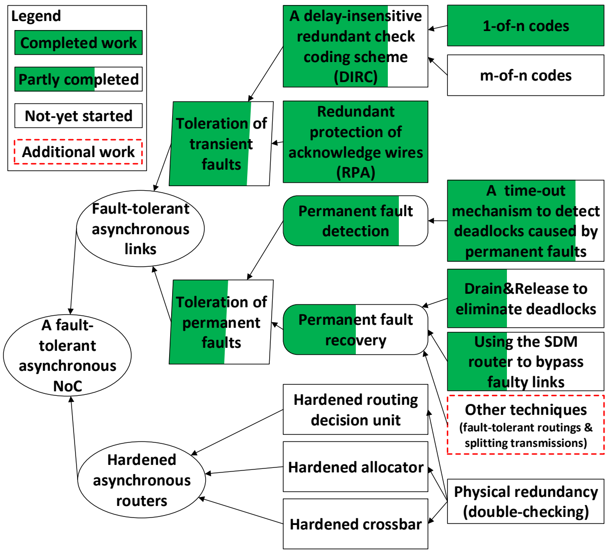

Legend
Completed work
Partly completed
Not-yet started
Additional work
A delay-insensitive redundant check coding scheme (DIRC)
1-of-n codes
m-of-n codes
Toleration of transient faults
Redundant protection of acknowledge wires (RPA)
Fault-tolerant asynchronous links
Permanent fault detection
A time-out mechanism to detect deadlocks caused by permanent faults
Drain&Release to eliminate deadlocks
A fault-tolerant asynchronous NoC
Toleration of permanent faults
Permanent fault recovery
Using the SDM router to bypass faulty links
Hardened routing decision unit
Other techniques (fault-tolerant routings & splitting transmissions)
Hardened asynchronous routers
Hardened allocator
Physical redundancy (double-checking)
Hardened crossbar

Fig. 2. Research objectives and methods

niques to detect permanent faults on links and eliminate the deadlock.

To recover from permanent faults, the Spatial Division Multiplexing (SDM) flow control technique will be explored [7]. In SDM NoCs, every link between routers is physically divided into several sub-links. Therefore, even if one or multiple sub-links encounter permanent faults, leading to deadlocks, other sub-links can normally work. By configuring the allocator of relevant routers, defective sub-links can be bypassed without adding spare wires.

Besides the protection of asynchronous links, asynchronous routers are exposed to the same fault environment as links and should also be protected. Note that this topic has rarely been studied before. Traditional redundancy techniques may incur a high hardware overhead when applied to routers. This research will explore efficient and inexpensive techniques to harden asynchronous routers. As a potential candidate, the double-check technique [8] will be studied.

3 Objectives & Contribution

The overall research objective is to provide a fault-tolerant architecture for asynchronous NoCs. A set of techniques will be proposed to protect asynchronous NoCs from different kinds of faults. The fault-tolerance capability and the incurred overhead are two main evaluation factors through the whole research.

Using a bottom-up design flow, Fig. 2 presents research objectives with detailed research methods.

The contribution of this research includes:

1. A fault-tolerant coding scheme and a new redundant technique to protect QDI interconnects from transient faults;
2. A set of permanent fault detection and recovery techniques: a time-out mechanism to detect and locate permanent faults, a *Drain&Release* technique to eliminate deadlocks caused by permanent faults, and using SDM to recover asynchronous NoCs from permanent faults;
3. Hardening techniques for critical components of asynchronous routers;
4. Co-management of transient and permanent faults to provide a fault-tolerant asynchronous on-chip communication;
5. Implementation of a fault-tolerant asynchronous NoC.

In addition, some other techniques (Fig. 2), such as fault-tolerant routings and splitting transmissions, are alternative tasks if time permits.

4 Research Progress

The current progress mainly includes (Fig. 2):

1. A new efficient delay insensitive redundant check coding scheme named DIRC providing transient fault tolerance.
2. A redundant technique, RPA, protecting the acknowledge wires from transient faults.
3. A SystemC-modelled configurable asynchronous router which supports SDM and permanent fault detection.
4. Implementation of a new asynchronous SDM NoC which uses a series of techniques to detect and recover from permanent faults.

Specifically, a fault-tolerant coding scheme DIRC (delay-insensitive redundant check codes), along with a redundant technique RPA (redundant protection of acknowledge wires), has been proposed and applied in QDI 1-of-n pipelines (named DIRC pipelines) to tolerate transient faults. Experimental results show that the 128-bit DIRC 1-of-4 pipeline is only 13% slower than the basic one without fault-tolerance. The fault-tolerance capability of DIRC pipelines increases hundreds fold even when multi-bit transient faults are considered. This work has been published in DSD 2013 (Section 7).

To tolerate permanent faults, a SystemC modelled asynchronous SDM router was first built to study the impacts of permanent faults on asynchronous links and verify the correctness of the proposed detection method.

An initial version of a permanent-fault-tolerant asynchronous SDM NoC has been implemented. A time-out mechanism is proposed to detect and locate permanent faults (this work has been submitted to DATE 2014, Section 7). A novel *Drain&Release* technique is proposed to eliminate the deadlock caused by permanent faults and release the locked error-free resources. To recover the network from permanent faults, the SDM flow control is used. By configuring the SDM router at runtime, the faulty links are bypassed. A draft paper has been completed. Because shortcomings exist within the first version of NoC, this work is being optimized with new ideas.

5 Future Research

5.1 Research plan

A detailed plan has been made, following the objectives and the research progress illustrated in Fig. 2. Some additional tasks are also given as risk management methods. A Gantt chart is presented in Fig. 3 (on page 5).

1.Topic: **Runtime permanent fault tolerant asynchronous NoCs** *(Sep. 2013 ∼ Dec. 2013)*

This topic is to implement an asynchronous SDM NoC tolerating permanent faults. An optimized version is being designed, following the previously proposed permanent fault detection and *Drain&Release* techniques. Asynchronous SDM routers are used to provide fault recovery. Thus the toleration of permanent faults on asynchronous links will be achieved.

2.Topic: **Extending the DIRC to m-of-n codes** *(Jan. 2014 ∼ Apr. 2014)*

This topic is to extend the proposed DIRC coding scheme from 1-of-n codes to m-of-n codes. m-of-n codes have a high complexity both in encoding/decoding and completion detection. Incomplete m-of-n codes [9] can be used to simplify these processes with a reasonable loss of coding efficiency. They can also apply the proposed DIRC encoding for fault-tolerance. Thus the toleration of transient faults on asynchronous links will be achieved.

3.Topic: **Co-management of transient and permanent faults** *(May. 2014 ∼ Sep. 2014)*

This topic is to co-manage the proposed transient and permanent fault tolerant techniques. Besides ensuring the transient and permanent fault tolerant techniques work correctly, the occasion of intermittent faults will be considered. In previous designs, permanent faults will block the relevant defective link, making it unusable. In case these faults are intermittent rather than permanent, a recovery mechanism is needed to unblock the false-diagnosed link. Thus the on-chip communication of the asynchronous NoC will be fully protected.

4.Topic: **Fault-tolerant asynchronous NoCs** *(Oct. 2014 ∼ Feb. 2015)*

This topic is to provide fault-tolerance for asynchronous routers, and then implement an asynchronous NoC to demonstrate the performance of all proposed fault-tolerant techniques. To harden the router, it is expensive to duplicate a whole router to achieve physical redundancy. As a potential technique, the double-check [8] will be explored to provide a fine-grained physical redundancy for routers, which only hardens key components such as routing decision units, allocators and crossbars. The implemented asynchronous NoC will be fully protected from transient and permanent faults.

Table 1. Risk assessment of the research plan

ID	Duration	Difficulty
Topic 1	4 months	⋆⋆
Topic 2	4 months	⋆⋆
Topic 3	5 months	⋆⋆⋆
Topic 4	5 months	⋆⋆⋆⋆
Topic 5	6 months	-
Task 6.1	1 month	⋆
Task 6.2	1 month	⋆⋆
Task 6.3	3 months	⋆⋆⋆
Task 6.4	2 months	⋆⋆

* The number of ⋆ represents the difficulty degree of the proposed plan. More ⋆ means the plan is more difficult.

5.Topic: **Thesis writing and viva preparation** *(Mar. 2015 ∼ Aug. 2015)*

In addition, some alternative tasks (Topic 6 in Fig. 3) are given in case that some research work encounters significant difficulty or there is additional time.

5.2 Risk assessment

A risk assessment (Table 1) is conducted to evaluate the feasibility of the research plan.

The most difficult point lies in Topic 4 on the protection of routers (Step 4.1 in Fig. 3) since it is difficult to provide fault-tolerance with a reasonable overhead (area and speed). (It should be noticed that no efficient and thorough methods have been proposed to protect a whole router.) In case the workload is not enough, Task 6.3 will be conducted to explore other permanent fault recovery techniques instead of using the SDM flow control, thus providing a general fault-tolerant architecture. Fault-tolerant routings and split-transmission are two candidates which are popular in synchronous NoCs but have been rarely studied in asynchronous NoCs.

6 Thesis Structure

1. Introduction[1]
   - 1.1 Motivation
   - 1.2 Problem description
   - 1.3 Research objectives
   - 1.4 Research contributions
   - 1.5 Thesis organization
2. Background
   - 2.1 Fault Classification

[1] Each chapter includes an introduction and a summary

- 2.2 Asynchronous circuits
- 2.3 Networks-on-chip

3. Related Work
   - 3.1 Asynchronous Networks-on-chip
   - 3.2 Transient fault tolerance
   - 3.3 Permanent Fault Tolerance
4. Delay-insensitive Redundant Check Coding Scheme
   - 4.1 Impact of transient faults on links
   - 4.2 Arithmetic rules of 1-of-n codes
   - 4.3 DIRC coding scheme
   - 4.4 Extension to m-of-n codes
   - 4.5 Protection of acknowledge signals (RPA)
   - 4.6 Hardware implementation
5. Permanent Fault Detection and Recovery
   - 5.1 Deadlock caused by permanent faults
   - 5.2 Permanent fault detection
   - 5.3 *Drain&Release* technique
   - 5.4 Permanent fault recovery using the SDM
   - 5.5 Hardware implementation
6. Fault-tolerant Asynchronous NoC
   - 6.1 Fault-tolerant on-chip communication
     - 6.1.1 Co-management of transient and permanent faults
     - 6.1.2 Recovery from intermittent faults
   - 6.2 Fine-grained router hardening
     - 6.2.1 Hardened routing decision unit
     - 6.2.2 Hardened allocator
     - 6.2.3 Hardened crossbar
   - 6.3 Fault-tolerant asynchronous NoC
7. Experiments and Performance Evaluation
   - 7.1 DIRC coding scheme
   - 7.2 Permanent fault tolerant SDM NoC
   - 7.3 Fault-tolerant on-chip communication
   - 7.4 Fault-tolerant asynchronous NoC
8. Conclusion and Future work
   - 8.1 Summary of this thesis
   - 8.2 Future work

7 Publications

- Guangda Zhang, Wei Song, Jim Garside, Javier Navaridas and Zhiying Wang. Transient fault tolerant QDI interconnects using redundant check code. In Proc. of Conference on Digital System Design (DSD), Sept. 2013.

- Wei Song, Guangda Zhang and Jim Garside. Online detection of permanent faulty links in asynchronous networks on chip. In submission to Design, Automation and Test in Europe (DATE) Conference, 2014.

References

[1] W. J. Dally and B. Towles. Route packets, not wires: On-chip interconnection networks. In *Proceedings of Design Automation Conference (DAC)*, pages 684–689. IEEE, 2001.

[2] Jens Sparsø. Asynchronous design of networks-on-chip. In *Proceedings of Norchip*, pages 1–4, 2007.

[3] Jens Sparsø and S. B. Furber. *Principles of Asynchronous Circuit Design: a Systems Perspective*. Kluwer Academic Publishers, 2001.

[4] C. Constantinescu. Trends and challenges in VLSI circuit reliability. *IEEE Micro*, 23(4):14–19, 2003.

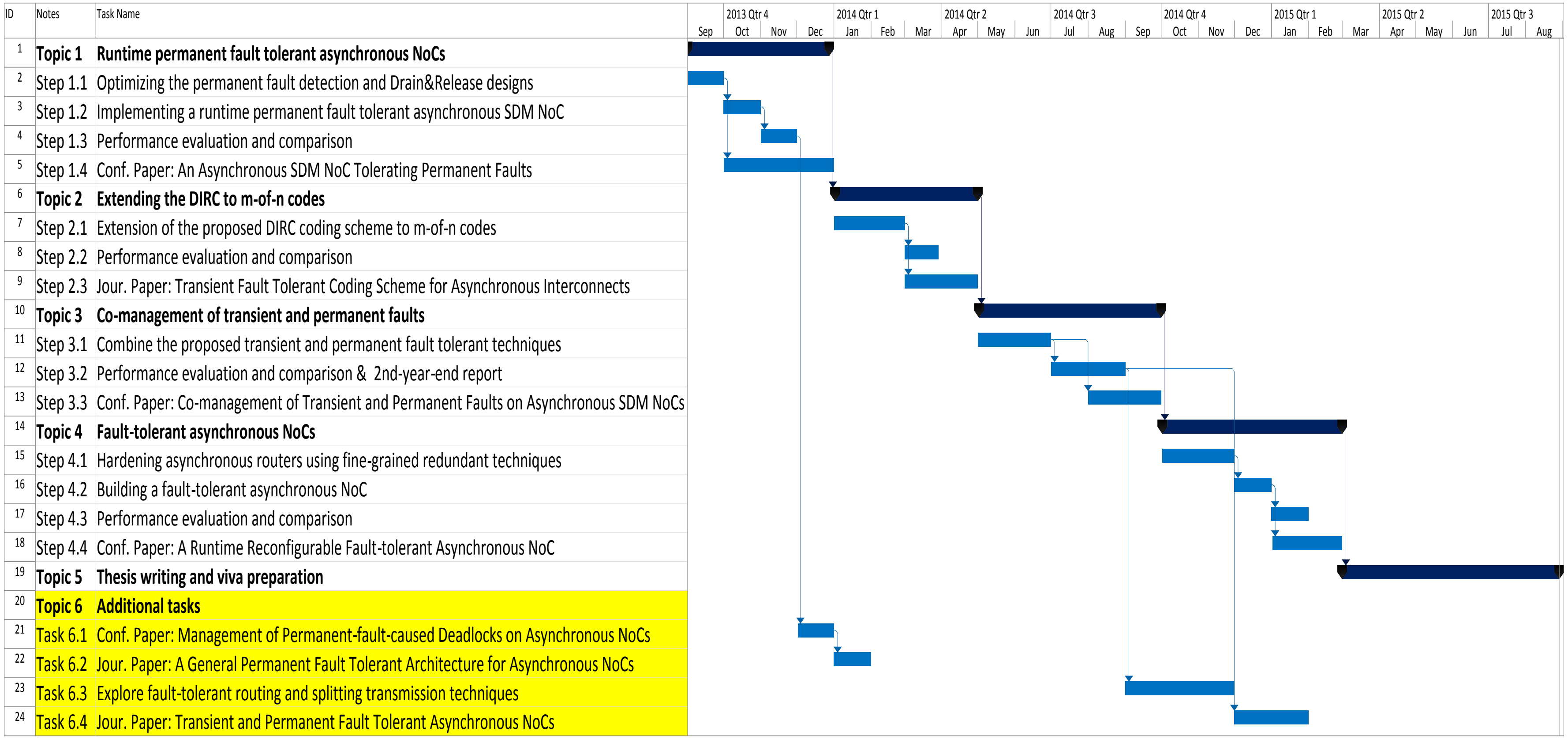
[5] T. Lehtonen, P. Liljeberg, and J. Plosila. Online reconfigurable self-timed links for fault tolerant NoC. *VLSI Design*, 2007.

[6] T. Lehtonen, D. Wolpert, P. Liljeberg, J. Plosila, and P. Ampadu. Self-adaptive system for addressing permanent errors in on-chip interconnects. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 18(4):527–540, 2010.

[7] W. Song and D. Edwards. *Spatial parallelism in the routers of asynchronous on-chip networks*. PhD thesis, University of Manchester, 2011.

[8] W. Jang and A. J. Martin. SEU-tolerant QDI circuits. In *Proceedings of 11th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, pages 156–165, 2005.

[9] W. J. Bainbridge, W. B. Toms, D. A. Edwards, and S. B. Furber. Delay-insensitive, point-to-point interconnect using m-of-n codes. In *Proceedings of IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, pages 132–140, 2003.



ID	Notes	Task Name
1	**Topic 1**	**Runtime permanent fault tolerant asynchronous NoCs**
2	Step 1.1	Optimizing the permanent fault detection and Drain&Release designs
3	Step 1.2	Implementing a runtime permanent fault tolerant asynchronous SDM NoC
4	Step 1.3	Performance evaluation and comparison
5	Step 1.4	Conf. Paper: An Asynchronous SDM NoC Tolerating Permanent Faults
6	**Topic 2**	**Extending the DIRC to m-of-n codes**
7	Step 2.1	Extension of the proposed DIRC coding scheme to m-of-n codes
8	Step 2.2	Performance evaluation and comparison
9	Step 2.3	Jour. Paper: Transient Fault Tolerant Coding Scheme for Asynchronous Interconnects
10	**Topic 3**	**Co-management of transient and permanent faults**
11	Step 3.1	Combine the proposed transient and permanent fault tolerant techniques
12	Step 3.2	Performance evaluation and comparison & 2nd-year-end report
13	Step 3.3	Conf. Paper: Co-management of Transient and Permanent Faults on Asynchronous SDM NoCs
14	**Topic 4**	**Fault-tolerant asynchronous NoCs**
15	Step 4.1	Hardening asynchronous routers using fine-grained redundant techniques
16	Step 4.2	Building a fault-tolerant asynchronous NoC
17	Step 4.3	Performance evaluation and comparison
18	Step 4.4	Conf. Paper: A Runtime Reconfigurable Fault-tolerant Asynchronous NoC
19	**Topic 5**	**Thesis writing and viva preparation**
20	**Topic 6**	**Additional tasks**
21	Task 6.1	Conf. Paper: Management of Permanent-fault-caused Deadlocks on Asynchronous NoCs
22	Task 6.2	Jour. Paper: A General Permanent Fault Tolerant Architecture for Asynchronous NoCs
23	Task 6.3	Explore fault-tolerant routing and splitting transmission techniques
24	Task 6.4	Jour. Paper: Transient and Permanent Fault Tolerant Asynchronous NoCs

Fig. 3. Gantt chart of the research plan