Nano Scale Disruptive Silicon-Plasmonic Platform for Chip-to-Chip Interconnection

Data Codecs for Error Detection and Correction

Milestone no.:	M32
Due date:	04/30/2013
Actual Submission date:	06/10/2013
Authors:	ST
Work package(s):	WP5
Distribution level:	RE[1] (NAVOLCHI Consortium)
Nature:	document, available online in the restricted area of the NAVOLCHI webpage

List of Partners concerned

Partner number	Partner name	Partner short name	Country	Date enter project	Date exit project
1	Karlsruher Institut für Technologie	KIT	Germany	M1	M36
2	INTERUNIVERSITAIR MICRO-ELECTRONICA CENTRUM VZW	IMEC	Belgium	M1	M36
3	TECHNISCHE UNIVERSITEIT EINDHOVEN	TU/e	Netherlands	M1	M36
4	RESEARCH AND EDUCATION LABORATORY IN INFORMATION TECHNOLOGIES	AIT	Greece	M1	M36
5	UNIVERSITAT DE VALENCIA	UVEG	Spain	M1	M36
6	STMICROELECTRONICS SRL	ST	Italy	M1	M36
7	UNIVERSITEIT GENT	UGent	Belgium	M1	M36

[1] **PU** = Public
PP = Restricted to other programme participants (including the Commission Services)
RE = Restricted to a group specified by the consortium (including the Commission Services)
CO = Confidential, only for members of the consortium (including the Commission Services)

Milestone Responsible

Organization:	STMicroelectronics
Contact Person:	Alberto Scandurra
Address:	STMicroelectronics, Stradale Primosole n.50 95121 Catania Italy
Phone:	+39 (0)95 – 740 4432
Fax:	+39 (0)95 – 740 4008
E-mail:	alberto.scandurra@st.com

Executive Summary

This document describes algorithms and implementations related to encoders and corresponding decoders targeting transmission errors detection and correction in on-/off- chip communication systems.

Change Records

Version	Date	Changes	Author
0.1 (draft)	2013-06-03	First version	Alberto Scandurra
1 (submission)	2013-06-10	Final version	Alberto Scandurra

Contents

1. Introduction

Layered architectures in networks-on-chip require upper levels to be provided an abstraction of the physical link into an ideal transmission channel. This is one of the tasks of the data link layer.

In general, data transmission can be affected by errors, which in digital electronic sense are wrong bit values which randomly occur. Static CMOS, though being regarded as a set of noise-resistant technologies, is still affected by channel noise and thus can be subject to errors. Important causes of these errors include the limited noise margin of the gates, however high it may be, cross-talk from integrated communication systems, interference from neighbour microwave devices and many others.

End-to-end communication is verified by the hardware modules which implement one or more techniques of error detection and correction. These techniques have been widely explored in the field of mathematics, informatics and engineering for long time[9], and particularly in the context of telecommunication for computer networks[19].

2. Error detection and correction: encoding techniques to reduce Bit Error Rate

This chapter presents some error detection and correction techniques from the point of view of hardware implementation in modern NoCs' data link.

1. Overview of traditional coding theory

Error detection and correction (EDC) is the field of coding theory which focuses on enabling reliable delivery of digital data over unreliable communication channels.

• **Error Detectors** – techniques which detect (but not correct) errors.

• **Error Correctors** – techniques which detect errors and reconstruct the original error-free data-format.

Error correction can be performed in two main ways:

• **Automatic Repeat Request (ARQ) / Backward Error Correction (BEC)** – the canonical way to ensure reliability is to inform the source through an acknowledgement and ask for data repetition in case of errors. Retransmission is asked until the data are properly received.

• **Forward Error Correction (FEC)** – the source encodes the message and the destination is able not only to determine whether or not an error occured, but also to reconstruct the original data (or what is deemed the “most likely” original data) on its own.

ARQ and FEC can be combined into **Hybrid Automatic Repeat Request (HARQ)** techniques, which is used when major errors are correct via ARQ techniques while minor guessing is performed by the target.

In general, EDC schemes consist of adding some *redundancy* (extra data) to the original message. This additional information is used by the target to check the received data.
Common techniques in the fields of informatics and telecommunication are mostly included in the following code families:

• **Repetition codes –** this simple ARQ scheme consists of transmitting the data a certain number of times. The target can guess whether the reiceved data are correct by confrontation and ask for retransmission if it is needed. Repetition codes are not very efficient, as they require an extremely large redundancy.
For example, transmitting 4 bits (useful data) 3 times requires 12 bits; redundancy is then 8 bit, which is 200% the useful data and 67% of the whole message.

• **Hamming codes**[19] – this technique makes use of *perfect codes*, which are codes that exactly match the theoretical upper bound on the number of distinct code words for a given number of bits. Adding some extra bits in key positions to the original message allows for error correction.

- **Checksum** – a message is constructed into codewords of known size and a special value, called checksum, is obtained as the modular arithmetic sum of a group of words. The checksum is then confronted to a reference, according to the specific algorithm.
Parity Word, Two's complement, Fletcher's Checksum, Adler-32 are widely used checksum algorithms.

- **Cyclic Redundancy Checks (CRC)[9]** – this error-detecting technique is well suited to detect burst errors and is widely used in Ethernet protocols. Its computation resembles a polynomial long division operation in which the quotient is discarded and the remainder becomes the result, but polynomial coefficient are calculated according to the carry-less arithmetic of a mathematical finite field. In this operation, the divisor is called *generator polynomial*. Different CRCs are defined according to the chosen generator.

- **Hash functions** – cryptographic hash functions allow the target to determine mismatches in the received message, through an autentication code, sometimes referred as digital fingerprint, hash value or checksum.

- **Convolutional codes** – the Error Correcting Codes (ECC) which are processed on a bit-by-bit basis; decoding of such codes is usually performed by the Viterbi decoder[20].

- **Block codes** – the ECCs which are processed on a block-by-block basis; Hamming codes can be considered part of this family, together with Repetition codes, Multi-dimensional Bit-Parity Checks, Reed-Solomon codes (a CRC subset, widely used in optical disks, DSL and WiMAX), Turbo codes and Low-Density Parity-Checks (LDPC).

2. Error detection and correction in hardware

In hardware, soft errors[24] (also referred to as single-event upsets) generally affect storage elements, such as memory, latches and registers, affecting the stored charge values, and subsequently the logic state of bits.
As technologies scale down, the noise margin decreases and a number of factors become relevant in their contribution to soft errors.
The most relevant ones are:

- **Neutron radiations** - they interfere with charges held within sensitive nodes in the circuit.

- **Particle collisions** – these phenomena are more and more likely to determine a critical modification in the stored charge values as the minimum feature size shrinks.

Manufacturers are increasingly taking soft error rates (SER) into serious account: they can determine increased Bit Error Rate in high-density memories and system-vulnerability to unpredictable malfunctions. Some improvements and precautions have been proposed in the manufacting process, i.e. the use of expensive silicon-on-insulator (SOI) substrates, trench capacitors and/or special single-transistor architectures to reduce the occurence of particle collisions.

Apart from the transistor-level point of view, some solutions can also be provided by hardware design at system level, taking advantage from the fact that, as of today, soft errors rates are still moderate and single-bit errors represent the most likely scenario. For example, software exceptions are implemented in some tightly-coupled memories (TCM) to perform complete system resets, while parity checks (see below) are already used for cache instructions, so that the detection of an error triggers a low-end flush/refresh of the pipeline. Hamming codes have often been proposed to provide protection for tags and other vulnerable data fields, but complete and multi-bit correction mechanisms pose a cost in terms of complexity, area and performance which is often hard to sustain.

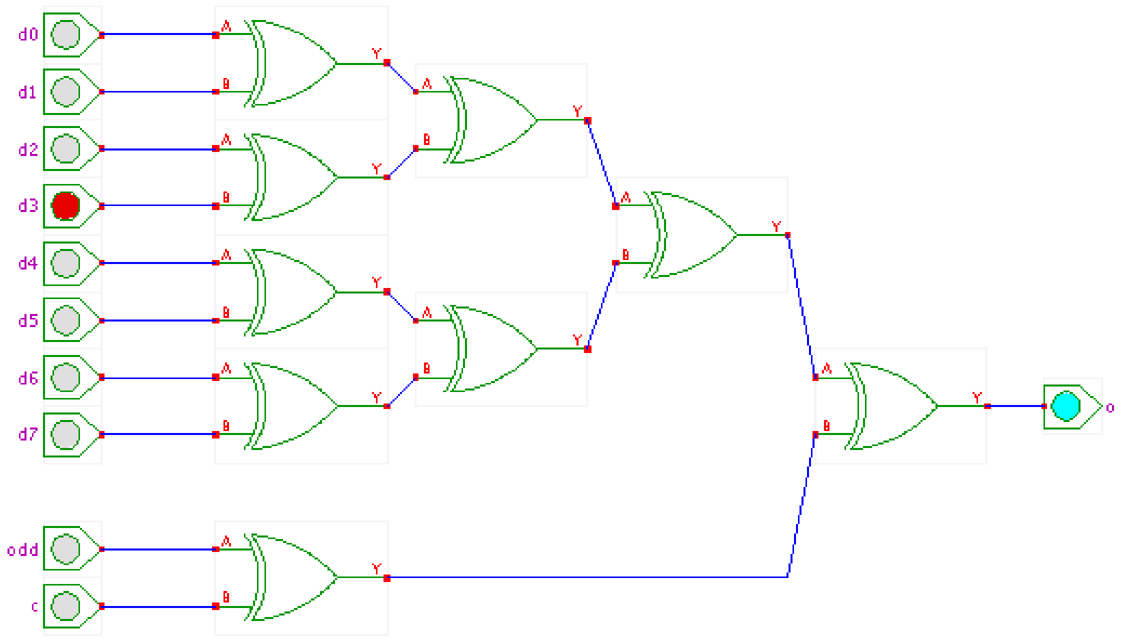
Indeed, the vast majority of the traditional techniques are designed for software implementation and mathematical research: they are usually well suited for low-level programming, but their logics are far too abstract and complex to be easily computed with the limited resources of the data link layer of a network-on-chip. They simply cannot comply to the strict requirements of high speed, moderate area occupation and low power-consumption which are required in such application.

3. Bit parity check in hardware

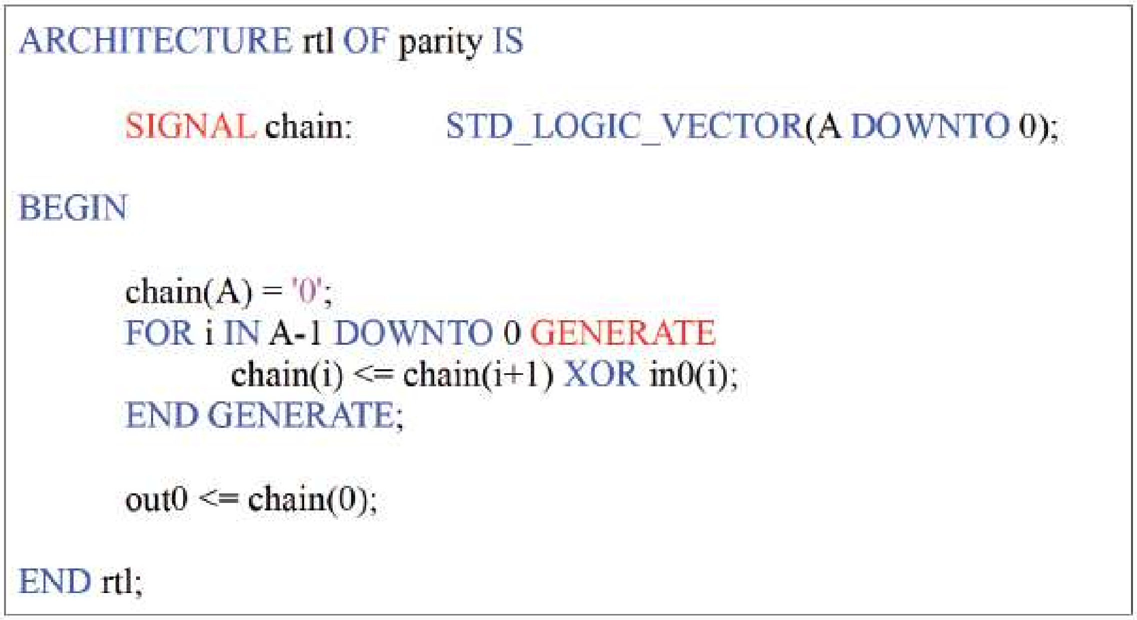
Bit Parity Check[19] is a simple 1-detector ARQ (*Automatic ReQuest*) technique.
Even parity is a special case of a Cyclic Redundancy Check (CRC)[9], where the 1-bit CRC is generated by the polynomial *x*+1.

In encoding phase, a single check line is added to the original phyt to mark whether the number of 1 in the phyt is odd or even. In decoding phase, the number of 1 in the phyt is counted again to check the correctness of the phyt. Only when the parity bit is coherent with the input lines, the phyt is considered valid.

7 bits of data (number of 1s)	**8 bits including parity**	
	even	**odd**
0000000 (0)	**0**0000000	**1**0000000
1010001 (3)	**1**1010001	**0**1010001
1101001 (4)	**0**1101001	**1**1101001
1111111 (7)	**1**1111111	**0**1111111



An example of 8-bit parity generator



```
ARCHITECTURE rtl OF parity IS

        SIGNAL chain:       STD_LOGIC_VECTOR(A DOWNTO 0);

BEGIN

        chain(A) = '0';
        FOR i IN A-1 DOWNTO 0 GENERATE
                chain(i) <= chain(i+1) XOR in0(i);
        END GENERATE;

        out0 <= chain(0);

END rtl;
```

Sample VHDL architecture of parity generator with generic number of bits A

It can be observed that an even number of errors in the same phyt brings to a non-detected error. This technique can detect (but not correct) an odd number of errors per phyt, thus it can be considered suitable only for single-error protocols.
Single-error detectors can be useful in NoCs, because of their simple logic (small area overhead and limited power consumption) and fast execution.

The table below shows synthesis results of a single-block 77-bit version and the equivalent 8-bit segmented (10 modules for a total of 80 bits).
Segmented version is obtained in the same way of power modules, and is expected to reduce critical paths: propagation delay of smaller modules is significantly reduced at the cost of further redundancy lines.

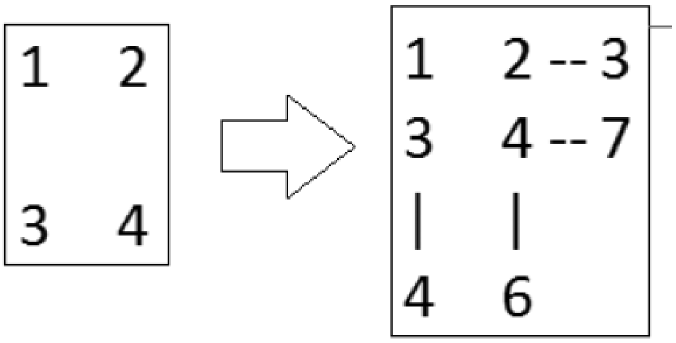
Bus size	**DSM Technology**	**Clock limit***	**Power overhead****	**Area overhead**	**Redundancy**
77	65 nm	633 MHz	1.82 mW	2866 standard cells	1 parity line
77	40 nm	750 MHz	1.46 mW	1664 standard cells	1 parity line
77	32 nm	867 MHz	1.34 mW	948 standard cells	1 parity line
8 x 10	65 nm	900 MHz	3.06 mW	3110 standard cells	10 parity line
8 x 10	40 nm	1033 MHz	2.48 mW	2190 standard cells	10 parity line
8 x 10	32 nm	1200 MHz	2.30 mW	1120 standard cells	10 parity line

* assuming input delay = 15% clock period, output delay = 10% clock period
** comprehensive of dynamic and static power consumption at the maximum working frequency (thus values are not directly comparable)

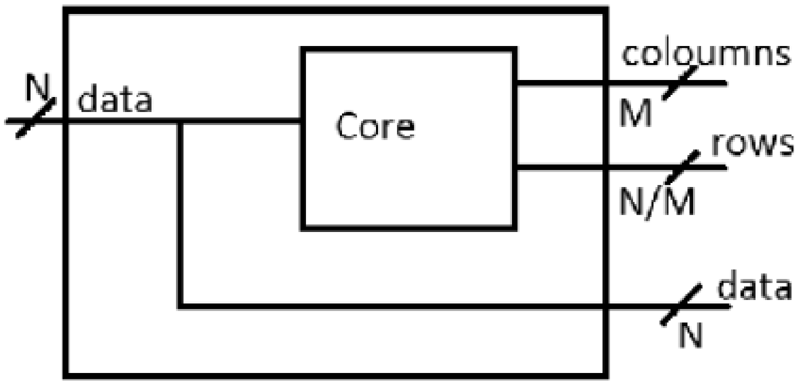
4. Multi-dimensional bit parity check in hardware

Multi-dimensional Bit Parity Check is a 1-corrector FEC (*Forward Error Correction*) technique. Traditional FEC techniques, such as *Hamming Distance Check*[9] and *Reed-Solomon Algorithm*[9], are not suitable for hardware implementation, due to their complexity (area, power and critical path would not be acceptable). Multi-Dimensional Bit Parity Check is usually discarded in the field of telecommunication, as it requires more redundancy than the other common techniques. In the context of electronics, however, this technique is far lighter to implement.

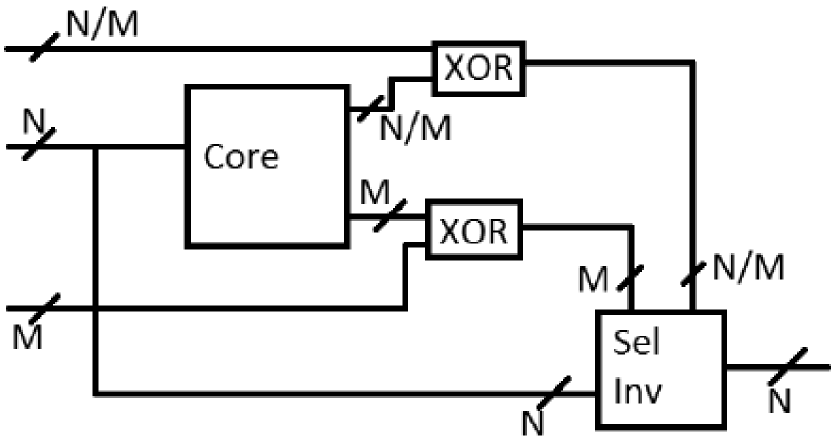
In the encoder, the original phyt is redistributed in a 2D matrix and parity checks are performed for each row and for each coloumn. The phyt is then transmitted with the row and coloumn parity check (which then become redundancy lines), which are verified by the decoder. If a bit in the phyt is affected by an error, then both its row and coloumn parity checks will provide an incoherence. A conditional inverter is instructed to invert only the bit whose parity checks are incoherent, thus correcting the error.



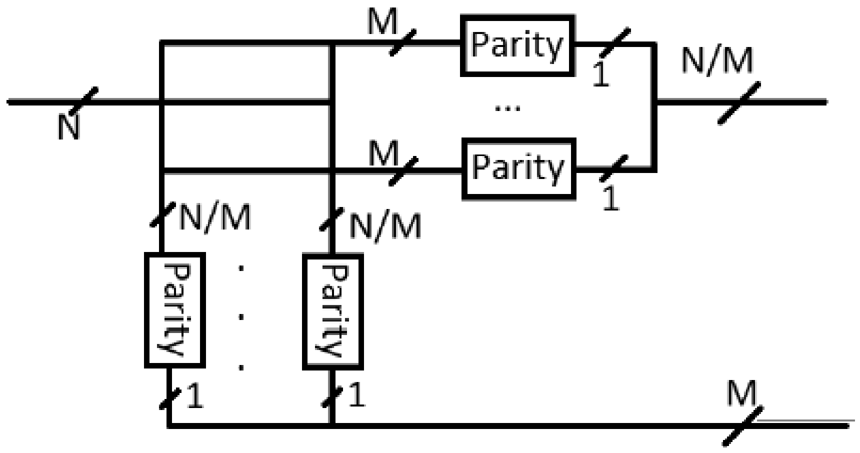
Encoder working principle



Encoder scheme
minimum redundancy is obtained with$M_{opt} = \sqrt{N}$



Decoder scheme
“Sel Inv” block inverts the bit whose row and coloumn are flagged wrong by the “Core” block



Core scheme

The described technique works only for single-error protocols: if two errors occure in the same phyt, then up to four bits are inverted, but only two of them were wrong. A multi-dimensional matrix can be used to correct a greater number of errors. In general, a Q-dimensional parity scheme can correct up to Q/2 errors.

Error-correctors require, in general, more redundancy bits and more complex logic than error-detectors.

Version	**DSM Technology**	**Clock limit***	**Power overhead****	**Area overhead**	**Redundancy**
72-bit single-block	65 nm	633 MHz	2.74 mW	3774 standard cells	17 parity lines
72-bit single-block	40 nm	833 MHz	2.80 mW	2590 standard cells	17 parity lines
72-bit single-block	32 nm	967 MHz	2.44 mW	1265 standard cells	17 parity lines

* assuming input delay = 15% clock period, output delay = 10% clock period
** comprehensive of dynamic and static power consumption at the maximum working frequency (thus values are not directly comparable)

References

[1] – L. Benini, G. De Micheli; "Networks on Chips: A New SoC Paradigm"; IEEE January 2002

[2] – W. J. Dally, B. Towles; "Route Packets, Not Wires: On-Chip Interconnection Networks"; IEEE Computer 2001

[3] – L. M. Ni, P. K. McKinley; "A Survey of Wormhole Routing Techniques in Direct Networks"; IEEE Computer, February 1993

[4] – M. R. Stan, W. P. Burleson; "Bus Invert Coding for Low Power I/O"; IEEE Transactions on VLSI Systems; March 1995

[5] – M. Palesi, R. Holsmark, X. Wang, S. Kumar, M. Yang, Y. Jiang, V. Catania; "An Efficient Technique for In-Order Packet Delivery with Adaptive Routing Algorithms in Networks on Chip"; IEEE 2009

[6] – M. Palesi, F. Fazzino, G. Ascia, V. Catania; "Data Encoding for Low-Power in Wormhole-Switched Networks-On-Chip"; IEEE 2009

[7] – G. Ascia, V. Catania, F. Fazzino, M. Palesi; "An Encoding Scheme to Reduce Power Consumption in Networks-On-Chip"; IEEE 2009

[8] – A. Ganguly, P. P. Pande, B. Belzer; "Crosstalk-Aware Channel Coding Schemes for Energy Efficient and Reliable NoC Interconnects"; IEEE Transactions on VLSI Systems 2009

[9] – D. Hankerson, D. Hoffman et al.; "Coding Theory and Cryptography"

[10] – J. A. Davis, R. Venkatesan et al.; "Interconnect limits on gigascale integration"; Proceedings of the IEEE 2001

[11] – E. Ho, K. W. Mai, M. A. Horowitz; "The future of wires"; Proceedings of the IEEE 2001

[12] - "International Technology Roadmap for Semiconductors – Interconnect"; Semiconductor Industry Association 2006

[13] – ISO/OSI; www.iso.org

[14] – D. Sylvester, K. Keutzer; "A Global Wiring Paradigm for Deep Submicron Design"; IEEE Trans. CAD/ICAS, February 2000

[15] – A. Agrawal; "Raw Computation"; Scientific Am. August 1999

[16] – J. C. S. Palma, L. S. Indrusiak et al.; "Inserting data encoding techniques into NoC-based systems"; IEEE Computer Society Annual Symposium on VLSI, March 2007

[17] – D. Bertozzi and L. Benini; “Xpipes: a network-on-chip architecture for gigascale systems on-chip”; IEEE Circuits and Systems Magazine

[18] – K. W. Kim, K. H. Baek et al.; “Coupling-driven signal encoding scheme for low-power interface design”; IEEE/ACM International Conference on Computer-Aided Design 2000

[19] – A. S. Tanenbaum; “Computer Networks”

[20] – A. J. Viterbi; “Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm”; IEEE Transactions on Information Theory, April 1967

[21] – PICMOS project; http://picmos.intec.ugent.be

[22] – D. J. Lockwood, L. Pavesi; “Silicon Photonics II: Components and Integration”; Springer

[23] – Wikipedia; http://en.wikipedia.org

[24] – R. Phelan; “Solutions for Soft Errors in System on Chip Designs”; http://www.design-reuse.com