THE INTERNATIONAL JOURNAL OF SCIENCE & TECHNOLEDGE

CMS Scheme in Self-Adaptive System for Addressing Permanent Errors in On-Chip Interconnects

S. Shanthi
PG Scholar/ VLSI Design, Karpagam University, Coimbatore, India
G. R. Mahendra Babu
Assistant Professor, Department of ECE, Karpagam University, Coimbatore, India

Abstract:
Current-mode signaling (CMS) with dynamic overdriving is one of the most promising scheme for high-speed low power communication over long on-chip interconnects. We propose a self-sufficient adaptive system for identifying by giving CMS circuit as input, transmitting it and avoiding permanent errors in on-chip interconnects. The proposed system redirects data on erroneous links to a set of spare wires without interrupting the data flow. For detecting the permanent errors , a novel in-line test (ILT) method using spare wires is proposed. In the presence of permanent errors, probability of correct transmission in the proposed systems is improved by 38% area, 64% energy, and 61% latency improvements.

Key works*: Current-mode circuit, dynamic overdriving, on-chip global interconnects, process variation*

1. Introduction

SPEED and power consumption of on-chip interconnect network have become significant in advanced CMOS technologies. It is difficult to meet preferred power and performance provisions of modern system-on-chips (SoCs) and multicore processors with buffer implanted long interconnects [1].Many alternate repeater circuits and signalling schemes have been recommended in recent past to achieve high-speed low-power communication over long on-chip interconnects. THE NUMBER of faults in on-chip links is expected to increase as technology scales further into the nanoscale regime. While most faults are temporary, about 20% of all errors are caused by permanent or intermittent faults [1]. Many permanent faults are a result of manufacturing defects, which can be detected during manufacture testing; however, these errors can also occur at runtime (e.g., from electromigration or aging). Error control coding (ECC) techniques are commonly used to address reliability issues in on-chip interconnects [2]–[8], but these techniques generally target transient errors rather than permanent errors. A single perpetual fault can severely reduce or even eliminate the correction capabilities of the commonly used codes. We first review the most promising signalling schemes for energy-efficient communication over long wires. Current-mode signalling (CMS) scheme has the potential to improve both speed and dynamic power consumption. It devours much less power compared to the upgraded repeater circuits (such as self-timed repeaters [6] and boosters [3]) and signalling schemes such as near-speed of light communication [7] and transition-aware signalling [5] as shown in [10]. However, the basic current-mode signalling scheme consumes static power and exhibits a direct trade-off between speed and static power. Capacitive coupled driver based low-swing signalling schemes are proposed in [11] and [12]. The capacitive coupled driver-based signalling scheme consumes much less energy than the basic CMS scheme.

Further, the scheme in [13] engages channel equalization technique at the receiver end to improve eye-opening. This scheme attains 85% improvement in energy consumption over the safeguard insertion scheme for a 10 mm line in 90-nm process. We have earlier proposed a CMS scheme with equalization at transmitter end (with dynamic overdriving). This scheme achieves 87% improvement in energy/bit over the conventional buffer insertion scheme for a 6 mm long line in 180-nm process. The scheme is more efficient for long wires carrying high activity signals than short wires carrying low activity signals. Hence the scheme is expected to give more than 87% improvement for a 10 mm line. The dynamic overdriving CMS schemes proposed in [11]–[14] also offer .more than 60% improvement in energy/bit. Hence, we have adopted dynamic overdriving CMS scheme for variation analysis. The huge reduction in energy consumption offered by the low-swing signalling schemes and the CMS schemes is mainly due to the reduced voltage-swing on the line. However, low voltage swing on the line reduces the noise margin of the data communication system. Hence, CMS schemes are more susceptible to parameter variations than the voltage-mode repeater insertion scheme. In highly ascended technologies, process variations cause important variations in device parameters. The variations in the transistor parameters can be categorized as either inter-die variations or intra-die parameters. In the case of inter-die variations similar devices on a chip have identical electrical parameters but the device parameters vary from die to die, wafer to wafer and batch to batch. In current expertise, variations in the parameters of devices on the same chip are also significant. This class of variations are referred to as intra-die variations. The variations can cause the voltage swing on the line to change which can lead to significant changes in the performance of a scheme.

The schemes in [11] do not discuss the impact of intra-die and inter-die variations on performance of their schemes. Further, the schemes proposed in [12]–[14] do not consider parametric variations in the designs of the transmitter and the receiver circuits. The transmitter and the receiver circuit of the CMS scheme proposed in [11] employ feedback which makes the scheme less sensitive to inter-die variations. However, this scheme is robust only as long as the transistor-parameters of the transmitter and the receiver are identical. In a SoC, the driver circuit and the receiver circuit of a repeaterless CMS scheme are quite likely to be in different parts of the chip. In this case, the performance of the scheme proposed in [11] can degrade significantly due to mismatch between transistor parameters in the transmitter and the receiver. In this paper, we propose a dynamic overdriving CMS scheme that is robust against intra-die and inter-die variations. It also offers significant improvement in delay, energy and EDP over voltage-mode schemes for a wide range of line lengths, data rates and data activity factors. The proposed scheme employs a smart bias circuit in the transmitter which makes it robust against inter-die variations. The operation of the circuit does not rely on matching of the transistor parameters of the transmitter and the receiver. In [8] and [9], we have introduced our basic CMS scheme and showcased its energy-efficiency and robustness in 180-nm technology. The CMS scheme proposed in this paper employs an improved version of the bias circuit.

With the new bias circuit the proposed CMS scheme is more robust against inter-die variations. Section II describes the principle of operation of a dynamic overdriving CMS scheme.

2. Introduction to CMS Scheme with Driver Pre-Emphasis

Driver pre-emphasis is the technique of supplying large current/voltage to the line during transitions of input and very small current/voltage in the steady state. In the frequency domain, it means amplifying high frequency components of the input signal. Fig. 1(a) shows typical waveforms of voltage and current of a CMS system with driver pre-emphasis. The line voltage swings around the voltage defined by the receiver. Fig. 1(b) shows the transmitter and the receiver circuits used in CMS-Fb scheme. The transmitter reported in [11] has a strong driver and a weak driver. The strong driver supplies a large current to the line during transition and the weak driver provides a small steady-state current to the line. NAND and NOR gates turn on the strong driver for a short duration. This duration is controlled by a feedback inverter [see Fig. 1]. The strong driver is turned off after the line voltage (at the transmitter end) crosses the switching threshold of the feedback inverter . At the receiving end, the line voltage is held near [see Fig. 1]. The receiver employs a feedback which makes the line voltage (at the receiving end) swing around , which is the switching threshold of inverter-amplifier. The steady-state voltage swing on the line is given by the product of the static current supplied by the weak driver and the small signal input impedance of the receiver. The inverter following the inverter-amplifier takes the output to CMOS logic levels.

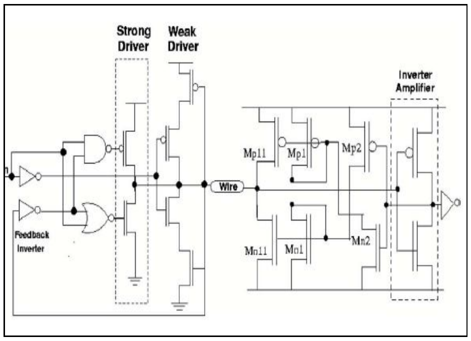

Figure 1: CMS scheme with driver pre-emphasis CMS scheme proposed in [11] (CMS-Fb)

3. Proposed Scheme

In order to maintain coding strength in the presence of permanent errors, spare wires can be used to replace permanently erroneous wires. The introduction of spare wires requires the following: 1) reconfiguration control and logic for bypassing erroneous wires and 2) a protocol for synchronizing information between receiver and transmitter.

We present a system that uses spare wires to replace permanently erroneous wires without interrupting the data flow. To detect these permanent errors, we propose a novel *in-line test* (ILT) method to test each adjacent pair of wires in a link for opens and shorts. These tests can be run periodically to ensure that each link’s ECC capability is not being crippled by permanent errors. By testing every wire in the link, the ILT method also recovers resources from intermittent errors that were incorrectly flagged as permanent. In addition to the ILT method, we describe a number of important improvements to an alternative *syndrome storing-based error detection* (SSD) method, which is based on evaluation of consecutive code syndromes at the receiver [9]. Syndromes are calculated during the decoding procedure and contain information on errors in the received words, the remainder of this paper is organized as follows. Associated work is discussed in Section II. In Section III, we present the intended permanent-error detection methods. In Section IV, the reconfigurable link framework is presented. In Section V, an implementation case study is presented along with simulation results. Section VI contains a discussion of the design tradeoffs with our approach. Concluding remarks are presented in Section VII.

4. Related Work

ECC research applied to on-chip networks typically only considers transient errors; protection against permanent or intermittent (lasting several cycles) errors is rarely discussed. Some methods of protecting against transient errors can also be used to protect against permanent and intermittent errors; unfortunately, this can severely limit the code's ability to protect against transient errors.

4.1. Coding-Based Error Recovery

In general, two recovery techniques are used when an error on a link has been detected. In *automatic repeat query* (ARQ), a retransmission is requested, while in *forward error correction* (FEC), check bits transmitted together with the data are used to correct errors without the need for retransmission [10]. Hybrid approaches combine the best properties of ARQ and FEC [2], [6], [11]. A code's error detection capability td is larger than its error correction capability t_c ($t_d = d-1$ and $t_c = \lfloor (d-1)/2 \rfloor$, where is the minimum code distance); thus, because the probability of a link error depends on the link voltage, a lower link voltage can be used to detect errors than is required to correct them. As a result, error detection plus ARQ can be more power efficient than FEC [2], particularly when dynamic voltage scaling is applied [12]. The disadvantage of ARQ is the retransmission potential, in untiring noise environments, a huge number of retransmissions may result, as the number of retransmissions be contingent on error conditions; making ARQ less energy effective than FEC.

More importantly, the ARQ method flops in the occurrence of permanent errors (retransmission is only useful for avoiding transient errors). FEC codes can perceive and spot on the permanent errors, but each permanent error will decrease the code's capability to bear transient or intermittent errors. FEC codes that can perceive and spot on the multiple errors (e.g., Bose–Chaudhuri–Hocquenghem (BCH) codes) have large power and area overhead costs, motivating the need for a different type of solution to handle permanent errors.

4.2. Spare Resources for Handling Permanent Errors

The use of spare modules to replace erroneous ones, particularly uses a synchronous design in array structures, is a long-known fault-tolerant approach [13]. Spare cells and wires have been used in field-programmable gate arrays to bypass defective components [1]. Refan *et al.* use spare wires to recover from switch failure by connecting each processing element to two switches in a network- on-chip (NoC) [10]; if a permanent fault occurs in one switch, processing elements share the working switch, and the system reroutes its data accordingly. Grecu *et al.* have analysed the use of spare wires in NoCs to increase manufacturing yield; reconfiguration of the links used crossbar switches with redundant channels. Unfortunately, the authors did not discuss the error detection procedure. In another work, Grecu *et al.* presented a built-in self-test methodology for NoC interconnects and thoroughly discussed manufacture testing methods for NoCs, but they did not address runtime failures. Reick *et al.* discuss dynamic I/O bitline repair using spare wires [19], but their detection and correction processes are not specified.

Runtime reconfiguration has been presented in the authors' previous work [9], in which spare wires and the SSD method were applied to a self-timed system using ARQ. That prior work had a few significant limitations; for example, the system could tolerate only one permanent error per code interleaving section, and the data flow had to be stopped for the reconfiguration. In contrast, the ILT method proposed in this paper handles as many permanent errors as there are spare wires. Furthermore, it allows the link to be reconfigured without interrupting data transmission. The presented reconfiguration system uses a synchronous design methodology and FEC to achieve higher throughput.

This paper also presents an improved non-interrupting implementation of the SSD method with additional analysis of design tradeoffs.

5. Permanent-Error Correction

Permanent-error correction in on-chip links using spare wires is a two-step process. First, the permanent error must be detected; then, the link must be reconfigured to avoid transmit-ting over the faulty wire. The proposed adaptive link framework is shown in Fig. 1 and consists of a transmitter, a link, and a receiver. The incoming -bit-wide data word is encoded in the transmitter to a codeword of width , which is transmitted through the link and decoded in the receiver. The decoder is responsible for correcting any errors and outputs the original –bit data word. A number of spare wires are available.

Reconfiguration units at the transmitter and receiver determine which of the lines carry data and which are left idle. The reconfiguration control units pass reconfiguration information between the receiver and transmitter and synchronize reconfiguration.

The *error detection and reconfiguration central control unit* detects permanent errors and initiates reconfiguration. The inputs to this unit depend on which detection method is used.

Tthe syndrome and error vector from the decoder are needed for the SSD method,. For the ILT method, test outputs from the spare wires under test are needed. The ILT method requires a *test pattern generator* (TPG) block and test inputs to produce test signals.

We apply our techniques to permanent and intermittent errors in the *link*; the logic units are assumed to function correctly. Two methods are presented here, namely, the proposed ILT method and the improved SSD method.

5.1. ILT Method

The proposed ILT method sequentially routes data from each pair of adjacent wires to a set of available spare wires, permitting each pair to be tested for intermittent and permanent faults. Through normal operation, without interfering data transmission, by making use of the reconfiguration system to be labelled in Section IV. To protect against runtime permanent errors, the ILT is run periodically, with a period that can be shortened to improve error resilience or increased for energy efficiency. In addition to this recurring testing, the ILT can be generated when an error is detected beyond the error correction capability of the code protecting the link. The trigger can use a simple timer or be adaptively controlled by an upper protocol layer (e.g., application) to save energy during idle periods.

5.1.1. ILT Procedure

To begin the test, the ILT control unit reconfigures the link such that the first pair of wires is connected to the TPG (the data on those lines are rerouted to spare wires). The TPG issues a series of test patterns using the signal. The ILT control unit compares the received signal to a lookup table (described in Section III-A-2) to determine if there is a permanent error in that pair of wires. The lookup table indicates which line(s) need(s) to be flagged as erroneous. Functional wires are reconfigured to carry data once again, and the process is repeated for each pair of wires (i.e., the test is shifted from wires 1 and 2 to wires 2 and 3, etc.). Note that, during each test, wires that were flagged as faulty are retested to prevent intermittent errors from wasting wire resources.

To provide even greater protection against permanent errors, the system is designed to take into account the number of spare wires when running a round of tests. The ILT control unit determines the number of remaining spares by checking the status of the last wire in the link (described in more detail in Section IV).

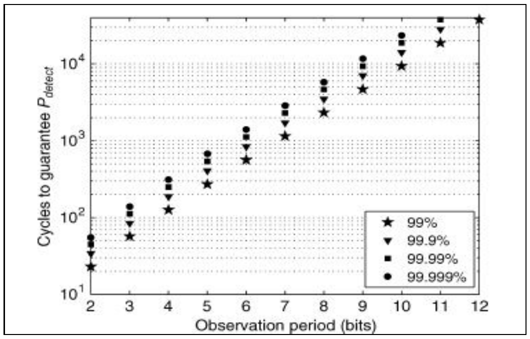

Figure 2: Effect of observation on the period of detection of errors

If one spare remains, the upper layer system can be alerted to the lack of spare resources, and the system only reroutes one wire at a time to that remaining spare. Instead of the two-wire test, the system performs a single-wire test that can detect opens in the line but cannot detect a short between the wire under test and its neighbors. If a wire adjacent to the wire under test is disabled due to a previously detected error, the ILT unit will perform the two-wire test on that pair. If no spare wires remain, the system will periodically retest each disabled wire in an effort to recover from intermittent errors.

5.1.2. Analysis of Open and Shorted Wires

Here, we analyse the impact of open and shorted interconnect faults on circuit operation to determine the test patterns that will be used in the test procedure. Based on these patterns, a lookup table is created to determine if wires are erroneous using the signal. The simulation schematics in Fig. 2 were used to analyze the behavior of opens and shorts on a link, with an open circuit simulated using an open resistance between repeaters and a short circuit simulated using a short resistance between two adjacent lines.

Values for the link resistances 's, substrate capacitances 's, and metal-to-metal capacitances 's are taken from an STMicroelectronics 90-nm technology report . The link resistances and the capacitances were computed for 200- m wire sections using the sixth metal layer M6. The expected output of a broken wire is fixed regardless of the input value (ignoring coupling effects); however, the failure mode of the wire as its resistance increases (e.g., because of electromigration) is interesting and highlights an important feature of the ILT system. The output of the open wire, labelled in Fig. 2(a), depends on and the system operating frequency. To analyze the behavior of a faulty wire, was varied between 1 and 100M . A plot of the resistance versus output voltage at the receiver is shown in Fig. 3 for a variety of operating frequencies.

6. Adaptive System Framework

In the following sections, the structure and operation of all the modules from Fig. 1, aside from the previously described fault exposure units, are presented.

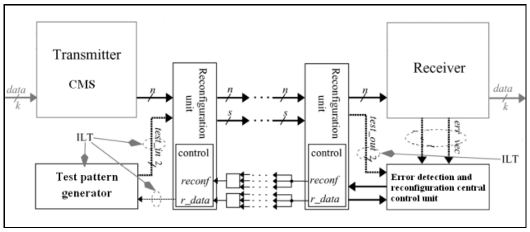

Figure 3: Reconfigurable link system. The parts required only for ILT or implementation are marked in the figure.A

6.1. Encoder and Decoder

The encoder and decoder are responsible for implementing the tolerance against transient faults. The encoder calculates check bits that are transmitted together with the data word over Fig. 7. Reconfiguration unit. (a) No permanent errors. (b) Permanent error at location _. c) Permanent errors at locations _ and _ _ _.

the link, and these check bits are used in the decoder to detect and correct possible errors. The reconfiguration system is designed as a separate layer from the underlying data transmission. This means that there are no ECC requirements for the reconfiguration system to be able to bypass permanent errors (although if the SSD detection system is used, the link code must have a syndrome, i.e., the code should be a linear block code).

The system uses and signals to indicate when the data is valid for processing and when the receiver is prepared to input new data. These signals are needed to handle situations when there is no data to be transmitted or when the receiver cannot input new data words because it is out of buffering space (e.g., because of congestion). The control signals are protected using triple modular redundancy (TMR) and spatial separation, which makes the system more immune to local noise and intradie variations.

6.2. Reconfiguration Units

The function of the reconfiguration unit is to route the data around the erroneous wires or wires under test. To balance the delay within routed wires, the reconfiguration ripples through the bus, as shown in Fig. 7, which presents the core capability of the reconfiguration unit. The number of spare wires to be inserted into the system depends on the probability of a permanent error in a wire, the number of wires, and the desired probability for correct operation of the link. Assuming that the permanent errors are independent and that the detection and reconfiguration can be done in all cases, this relation can be described by

$$P_{\text{correct}} = \sum_{i=0}^{s} \binom{n+s}{i} (1-p_e)^{n+s-i} p_e^{i}$$

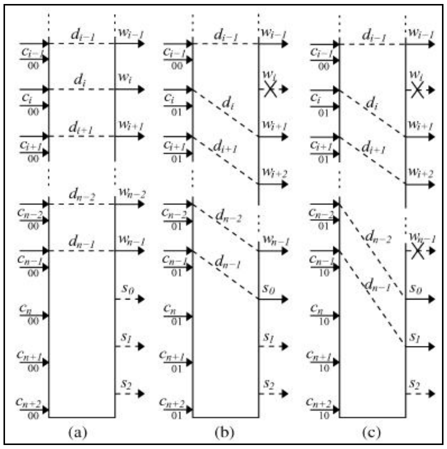

Figure 4: Reconfigurable unit (a) No errors. (b) Permanent errors at location I (c) Permanent errors at location i and n-1

where is the probability for correct link operation, is the number of wires in a link, is the number of spare wires, and is the probability of a permanent error in a wire. For example, if the probability of incorrect link operation must be less than and the probability of a permanent error in a wire is , three spare wires are needed if , while two spare wires are enough for . Interconnect error probability has been analyzed in [23]. The number of spare wires has an effect not only on the permanent- error tolerance but also on the complexity of the reconfiguration units.

It can mark a wire erroneous or remove the mark from a wire. The latter is needed for periodic testing or returning a wire to normal operation if the error on it turns out to be intermittent instead of permanent. For systems where an error mark cannot be removed (e.g., the SSD system), only the first mode is needed. The mode can be given as an input to the reconfiguration unit, or it can be extracted from the reconfiguration control signals.

The estimation of control significances involves incrementing or decrementing the control value of each wire based on the reconfiguration location and mode.

7. Conclusion

A complete reconfigurable system utilizing spare wires to replace erroneous wires and enabling reconfiguration without interfering with data transmission has been presented. Error detection have been evaluated, by rotating ILT. The results show that our approach provides tolerance against a number of permanent errors equal to the number of spare wires in the system. If any errors occur while passing, it can be reconfigured and resent through the successive lines without errors. For example, if any errors are found in 3rd line, then it can be found out through the reconfigurable unit while comparing test in 2 and test_out, error will be detected. So by correcting the error the data is sent through the next line. In addition, Hamming and BCH error tolerances decrease significantly in the presence of permanent errors, while the error tolerance of the adaptive systems is less affected. Indeed, the ILT detection method sees no degradation at all. For example, with one permanent and one transient error, the probability for correctly transmitted words is 30% higher in the ILT system than with the reference Hamming code approach; The framework can be constructed with reasonable energy, latency, and throughput overhead, i.e., 10%, 29%, and 23%, respectively, but with a considerable area increase of 250%–280%. The presented system occupies just 62%–64% of its area, consumes 36% of the energy per transmitted flit, and has latency of 39% of the BCH. Of the two error detection methods, The reconfigurations and test runs consume energy, but the total energy which is lower, if the periodic testing is done after every 46 386 data transmissions or more.

This paper shows that protection against permanent errors can be achieved using spare wires with smaller penalties than using complex coding schemes. The latter method is used in the transmitter to minimize the control transmission length from receiver to transmitter, while the former is used in the receiver since the information is already available there.

8. References

1. G. De Micheli and L. Benini, Networks on Chips: Technology and Tools (Systems on Silicon). San Fransisco, CA: Morgan Kaufmann, 2006.
2. D. Bertozzi, L. Benini, and G. De Micheli, “Error control schemes for on-chip communication links: The energy-reliability tradeoff,” IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 24, no. 6, pp. 818–831, Jun. 2005.
3. L. Li, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin, “Adaptive error protection for energy efficiency,” in Proc. ICCAD, San Jose, CA, Nov. 2003, pp. 2–7.
4. S. Murali, T. Theocharides, N. Vijaykrishnan, M. J. Irwin, L. Benini, and G. De Micheli, “Analysis of error recovery schemes for networks on chips,” IEEE Des. Test Comput., vol. 22, no. 5, pp. 434–442, Sep./ Oct. 2005.
5. D. Rossi, P. Angelini, and C. Metra, “Configurable error control scheme for NoC signal integrity,” in Proc. 13th IEEE IOLTS, Crete, Greece, Jul. 2007, pp. 43–48.
6. S. R. Sridhara and N. R. Shanbhag, “Coding for system-on-chip networks: A unified framework,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 13, no. 6, pp. 655–667, Jun. 2005.
7. Q. Yu and P. Ampadu, “Adaptive error control for reliable systems-onchip,” in Proc. IEEE ISCAS, Seattle, WA, May 2008, pp. 832–835.
8. H. Zimmer and A. Jantch, “A fault model notation and error-control scheme for switch-to-switch buses in a network-on-chip,” in Proc. 1st IEEE/ACM/IFIP CODES+ISSS, Newport Beach, CA, Oct. 2003, pp. 188–193.
9. T. Lehtonen, P. Liljeberg, and J. Plosila, “Online reconfigurable selftimed links for fault tolerant NoC,” VLSI Des., vol. 2007, pp. 1–13, 2007.
10. R. Ziemer and R. Peterson, Introduction to Digital Communication, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 2001.
11. A. Ejlali, B. M. Al-Hashimi, P. Rosinger, and S. G. Miremadi, “Joint consideration of fault-tolerance, energy-efficiency and performance in on-chip networks,” in Proc. DATE, Nice, France, Apr. 2007, pp. 1–6.
12. L. Shang, L.-S. Peh, and N. K. Jha, “Dynamic voltage scaling with links for power optimization of interconnection networks,” in Proc. 9th Int. Symp. HPCA, Anaheim, CA, Feb. 2003, pp. 91–102.
13. B. Johnson, Design and Analysis of Fault Tolerant Digital Systems. Boston, MA: Addison-Wesley, 1989