Book Review

A Scholarly Review of "Error Control for Network-On-Chip Links" (Authors: Bo Fu and Paul Ampadu, 2012)

Fu, B.; and Ampadu, P. 2012. Error Control for Network-On-Chip Links.
Springer Science+Business Media, LLC, New York, NY, USA.
Available: <http://dx.doi.org/10.1007/978-1-4419-9313-7>.

Dobri Atanassov Batovski

Department of Telecommunications Science, Faculty of Science and Technology
Assumption University, Bangkok, Thailand
E-mail: <dbatovski@au.edu>

Introduction

The term 'scholarly' in the review title is used to indicate that preliminary knowledge about error-control coding (ECC) is beneficial in following the review of the book chapters. This review is scholarly in the sense that the discussion is concerned with specialized technical material and is more appropriate to address certain parts of the book content with the use of specific terminology and abbreviations.

The book entitled "Error Control for Network-On-Chip Links" by Fu and Amadu (2012) was published in 2012 by Springer Science+Business Media, LLC, New York, NY, USA (international standard book number (ISBN): 978-1-4419-9312-0; electronic book ISBN (e-ISBN): 978-1-4419-9313-7; and digital object identifier (DOI): 10.1007/978-1-4419-9313-7).

The book was written by Bo Fu, Marvell Semiconductor, Inc., Santa Clara, CA, USA, and Paul Ampadu, Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY, USA. It contains 6 technical chapters, List of Symbols and Index. The total number of pages in the book is 163 pages including initial pages (12 pages), six chapters (144 pages), List of Symbols (4 pages) and Index (3 pages).

Connectivity is a fundamental issue in communications technology and the book provides a thorough investigation of on-a-chip (on-chip) interconnects. Interconnection networks are created when multiple processing units are interconnected with a specific configuration of links. When the processing units are identical, the resultant homogeneous networks usually serve as massively parallel computers for exhaustive computations. Such computers are utilized for simulations ranging from the expansion of the universe to interactions of subatomic particles. Weather forecasting, solar flares, brain activity, etc., are simulated to estimate the behavior of complex systems. There is a tendency at present toward the creation of heterogeneous interconnection networks in which different processing units have separate functions. In particular, tablets based on mobile platforms run a range of applications associated with the human activity requiring coordinated real-time processing of wireless transmission, multimedia content, social interaction and decision making. This poses technical challenges for the integrated circuit (IC) industry because an increasing number of processing units (cores) must be interconnected on a single IC (chip) rather than having macroscopic wires between separate chips. The size of the planar cores also becomes smaller in accordance with the tendency of shrinking their areas by redesigning them with the use of nanoscale components. The complementary metal-oxide-semiconductor (CMOS) technology continuously implements tinier and faster complementary pairs of *p*-channel and *n*-channel metal-oxide-semiconductor field-effect transistors (MOSFETs) which are the physical foundation for the construction of the basic logical components needed for the design of digital processing units.

Metal wires are used as interconnects in the single- or multi-layer planar CMOS circuits. A metal wire is characterized by its resistance *R*, capacitance *C*, and inductance *L*. The first book chapter explains how scaling down the wires affects their electrical characteristics. Table 1 provides a summary of wire-related phenomena for increased length and reduced thickness and width of the wires.

Table 1. Properties of microscopic metal wires.

Electrical characteristics	Effects
Increased resistance (due to approaching the mean free path of electrons and skin effect of the current flow at high clock frequencies)	Increased interconnect delay (high *RC* product for a fixed length), signal attenuation and heat dissipation
Increased capacitance (sidewall capacitance between adjacent wires)	Increased power consumption and capacitive coupling between adjacent wires
Increased inductance (for long wires at high clock frequencies)	Increased inductive crosstalk coupling over a long distance

In addition to the electromagnetic interference (EMI) of scaled down metal wires, there are other reliability issues arising from structural defects, external radiation (particle strikes), noise due to variations of supply voltage and temperature, etc. For instance, open (crack) and short circuits may result from structural defects. Also, the space missions are especially at risk of errors arising from deep space and solar radiation. Even the terrestrial applications are prone to particle strikes with the participation of neutrons or alpha particles (consisting of two protons and two neutrons as in the nucleus of the helium atom) of various energies.

It is to be expected that logical errors occur whenever a certain combination of factors perturbs the current flow through the wires. Such errors can be transient, intermittent or permanent and the error control methods in the book make use of redundancy and fault tolerance for error detection and error correction.

The second book chapter lists possible solutions to address the aforementioned reliability issues. It is shown that increasing the interconnect width is problematic as the capacitive coupling decreases while the inductive coupling increases. Similar is the case with the interconnect spacing because the inductive coupling dominates over the capacitive coupling after a certain spacing threshold. Shielding is considered as the straightforward solution to crosstalk coupling with the introduction of shield wires which can be connected to power or ground (passive shielding), or to the signal wires (active shielding) and follow their switching behavior. Repeater insertion methods are shown to be able to reduce the crosstalk coupling over long wires. In addition, crosstalk avoidance codes (CACs) can achieve coupling reduction by avoiding specific switching patterns. Also, the use of skewed transitions introducing relative delays in adjacent wires is an alternative way of reducing capacitive coupling by avoiding simultaneous opposite switching. Such preventive measures cannot fully eliminate the occurrence of transmission errors.

The next logical step is to introduce error recovery which would allow the detection of errors and their subsequent correction. There are two error control coding schemes which are widely used in communications technology, automatic repeat request (ARQ) and forward error correction (FEC).

The ARQ schemes rely on error detection codes to encode the message at the transmitter into blocks (packets) which contain redundancy in the form of additional parity symbols. The receiver decodes said packets by comparing the message and parity symbols to detect errors. Whenever at least one error is detected after transmission, the receiver informs the transmitter by sending a negative acknowledgement (NACK) control message with a request for retransmission. Three different types of ARQ are discussed in the book: stop-and-wait; go-back-*N*; and selective-repeat. Stop-and-wait ARQ appears to be the simplest and most inefficient ARQ type because the transmitter would wait for either positive (ACK) or negative (NACK) acknowledgement before sending a new packet

or resending the packet received in error. Go-back-*N* ARQ is a more efficient ARQ type because the transmitter would continue sending packets until an eventual NACK is received. Then a window of *N* packets would be retransmitted including the packet received in error followed by (*N* - 1) subsequent packets which have also been transmitted during the round-trip delay of receiving the NACK control message. Selective-repeat ARQ seems to be the most efficient ARQ type because only packets received in error will be retransmitted. However, a buffer is needed at the receiver to store the incoming packets and reorder them during ongoing retransmissions in accordance with the desired error-free packet sequence. Since there has to be a tradeoff between the efficiency and complexity for on-chip implementations of error control methods, go-back-*N* ARQ is the preferred ARQ type in the book. It should be noted that the performance of ARQ schemes depends on the maximum number of errors which can be detected during the decoding of received packets. There has to be another tradeoff between the minimum redundancy of message encoding (maximum code rate) and the maximum efficiency of error detection. It is assumed that there are a relatively small number of errors occurring in on-chip interconnects. The error correction methods are to be considered as a supportive part of properly functioning integrated circuit systems. Whenever the number of errors exceeds the error-detection capabilities of the decoding algorithm, corrupted packets would occur and remain unnoticed thus resulting in erroneous computations. Despite the efficiency of the ARQ schemes within the limits of their error-correcting specifications, the observed latency is an obvious disadvantage arising from multiple retransmissions.

The FEC schemes correct errors directly at the receiver and there is no need for the packets to be retransmitted. Error-correcting codes of increasing complexity have been developed over the past several decades. The simplest example of an error-correcting code is a repetition code which encodes the message by simply repeating every symbol several times. The redundancy is determined by the number of repetitions and the code rate, which is the ratio between the number of message symbols and the number of encoded symbols, is too small for the limited number of errors that can be corrected.

A much better tradeoff between code rate and error-correcting capability is achieved with block codes. A generator matrix and a parity-check matrix derived from it are used for encoding and decoding. Such codes are mainly systematic codes and the encoded block contains the message symbols in unaltered form followed/preceded by parity bits.

The convolutional codes are non-systematic codes which alter the message sequences similarly to the spectral modifications done by digital filters in signal processing. Discrete convolution is performed with at least one generator polynomial and the message can be viewed only after a subsequent decoding at the receiver. As stated in the book, the convolutional codes are not quite suitable for on-chip implementation because the discrete convolution has to be done serially which would increase the encoding latency. The decoding is usually done serially on a trellis which is also a time consuming process. The increased complexity and latency of convolutional encoding and decoding are obstacles to their implementation in on-chip communications.

Iterative FEC codes such as turbo codes and low-density parity-check (LDPC) codes are not considered at all for on-chip interconnects because of their excessive complexity and latency. Furthermore, the technologically constrained design of wired links between cores makes use of the so-called flow control unit (flit). Packets split into flits of small size which are strictly ordered and routed through virtual channels. Turbo codes and LDPC codes are applicable for encoding and decoding of very long message sequences where a significant number of errors may occur in noisy communication channels while the error control of individual flits is limited to small block size and just a handful of errors occurring randomly or in bursts of consecutive errors. This is the reason most of the book to be focused on linear block codes and their product codes.

The combined use of linear block codes

for forward error correction and go-back-*N* automated repeat request is referred to as hybrid ARQ (HARQ). Type-I HARQ is based on error detection and subsequent error correction at the receiver and if detectable but uncorrectable errors occur, then the entire encoded message sequence (codeword) is retransmitted until it is successfully decoded. This hybrid type is a simple combination of FEC and ARQ with the utilization of the full strength of FEC irrespective of the absence or presence of errors during transmission. Type-II HARQ is a more flexible hybrid type having a provision for two parity-check segments for a given message sequence which are created by the product code of two linear block codes (component codes). Only one of the parity-check segments is transmitted with the message sequence. The eventual occurrence of uncorrectable errors prompts the generation of a NACK control signal and the second parity-check segment is transmitted in an attempt to correct said errors. Therefore, the second transmission does not contain the message sequence so that a reduced latency and a higher throughput can be achieved.

The second book chapter also considers the occurrence of permanent errors which are caused by open and short circuits. A periodic in-line test (ILT) of adjacent wires is considered in combination with a subsequent configurable remapping of faulty links using spare wires. The redundancy provided for hardware fault tolerance would allow the effective reconfiguration of the parallel metal wires throughout the lifetime of the integrated circuits.

The third book chapter is an overview of existing and future networks-on-a-chip (NoC). The standard bus-based communication architecture between processing units has a limited applicability for nanoscale on-chip implementations due to the increased capacitive load of multiple parallel metal wires. Alternative NoC architectures are being developed for the NoC design of emerging systems-on-a-chip (SoC) which will contain thousands of intellectual property (IP) cores. The NoC architectures depend on the choice of suitable topologies for the interconnection between the routers and the IP cores. Logical topologies appropriate for a planar implementation such as fat tree, butterfly fat tree (BFT), mesh, octagon, torus and folded torus are provided as illustrations. It is discussed that the simple mesh topology can be used for a relatively small number of IP cores but it has the disadvantage of an increased latency between said IP cores at the opposite edges of the interconnection network as the mesh size increases. The folded torus is considered as the best prospective topology for a planar implementation.

In most cases, a router has five input/output ports with one port connected to an IP core and four remaining ports connected to neighboring routers. The router design includes a crossbar switch for the interconnection between the five input and output ports of the router. It should be noted that the crossbar remains the best topology for the optimal internal router connectivity among a small number of ports. However, the crossbar is not included in the list of prospective topologies for NoC design between IP cores because of the apparent excessive wiring resulting from the increase of the size of the interconnection network.

The routing and switching techniques for interconnection networks and the router design depend on the chosen topology. The book discusses both static (deterministic) routing and dynamic (adaptive) routing. The deterministic routing is fast but inefficient in the case of link failures and is the preferred choice for special-purpose parallel computing applications in homogeneous interconnection networks. An example of deterministic planar routing is the XY routing. The dynamic routing is more appropriate for general purpose applications in both homogeneous and heterogeneous interconnection networks in avoiding lifelock and deadlock situations. Circuit switching and packet switching (store-and-forward, virtual cut-through and wormhole) are further considered with an emphasis on wormhole packet switching for which the packets are divided into flits of equal size to be routed over virtual channels. The use of flits is beneficial because of the small buffer size requirements for router design and the uniformity of the block size for error-control coding. The next

chapter section on router design considers the first-in first-out (FIFO) buffers per virtual channel and the basic components of a router operating with virtual channels while the last chapter section on reliability in NoC links is concerned with the necessity of error control coding and the combined use of ARQ and FEC in HARQ.

The fourth book chapter is an introduction to error control coding for on-chip interconnects. It starts with the definition of a field and its six properties: closure, associative property, identity, inverse, commutative property and distributive property. The use of modular operations in a finite (Galois) field (*GF*) with *q* elements *GF*(*q*) is further justified for keeping the results of the encoding and decoding within said field. The rest of the book is mainly concerned with binary codes within the *GF*(2) field. The modulo-2 addition (logical XOR operation) and modulo-2 multiplication (logical AND operation) are sufficient for the computations associated with binary codes and can be easily implemented in CMOS circuits.

A summary of the error-control codes introduced in the fourth book chapter is shown in Table 2. Table 3 shows conventional ECC examples. The emphasis is on two-dimensional product codes for which the message bits are arranged into a matrix. The rows and the columns of said matrix are encoded/decoded using two different component codes. If the component codes are Hamming (H) or extended Hamming (EH) codes, the resultant product codes are called Hamming product codes. A three-stage pipelined decoding method is used for correcting both random and burst errors by passing row status vectors and column status vectors between the stages. A drawback of product codes is the additional redundancy resulting from the matrix approach. The excess of redundant parity bits is undesirable because the individual bits of a codeword after encoding are transmitted in parallel over separate wires. Type-II HARQ is one way to alleviate this problem by using two parity-check segments.

The fifth book chapter considers the following energy efficient error control implementations: error control coding with a low link swing voltage system; and error control coding with a dynamic voltage swing scaling (DVSS) system. Table 4 lists ECC examples shown as illustrations. Configurable error control systems are further introduced to improve the overall energy efficiency when using different codes in changing noise conditions. The practical illustrations in the book include: hardware sharing between one H(22,16) encoder and four EH(8,4) encoders; and configurable syndrome decoder design with H(21,16) or four EH(8,4) codes. Standard (*n*,*k*) notations are used for all codes, where *n* is the number of codeword bits after encoding and *k* is the number of message bits.

ARQ CRC-5, FEC H(71,64) and BCH(85,65), and HARQ EH(72,64) are the four coding schemes used for comparison with Type-II HARQ with EH product code (C_P). Generally, a CRC code can detect a single error burst, but it is inefficient in detecting multiple random errors. A Hamming code can correct a single bit error or detect double-bit errors. A BCH code can correct multiple random errors but is inefficient in correcting error bursts. EH(72,64), which is a truncated H(127,120) code with an additional parity bit, has a single-error-correcting and double-error-detecting (SEC-DED) capability. In comparison, Type-II HARQ with EH C_P can correct multiple random and burst errors. Also, it can correct at least two permanent errors in the bus wires. According to the results from the performance evaluation provided in the book, Type-II HARQ with EH C_P has the lowest residual flit error rate (RFER), the lowest required link swing voltage and the lowest link energy consumption. The RFER is the rate at which a transmitted flit is in error but remains undetected. Type-II HARQ with EH C_P has the highest throughput compared to other retransmission schemes like ARQ CRC-5 and HARQ EH(72,64). It also has the lowest average energy consumption for high noise voltage deviations. Therefore, the use of product codes is justified in high noise environments but such ECC sophistication can be avoided in low noise environments with the provision of at least two operating modes in a configurable error control system based on the hardware sharing between coding schemes and configurable encoder/decoder design.

Table 2. List of systematic error-control codes introduced in the fourth book chapter.

No.	Code	Mes-sage bits	Code-word bits	Error-detecting capability	Error-correcting capability	Comment
1	Linear block codes	k	n	d_{min} - 1	$t = \lfloor (d_{min}-1)/2 \rfloor$	
2	Shortened codes	$k - l$	$n - l$	d_{min} - 1	$t = \lfloor (d_{min}-1)/2 \rfloor$	$d_{min} \geq d_{original}$
3	Extended codes	k	$n + l$	d_{min} - 1	$t = \lfloor (d_{min}-1)/2 \rfloor$	
4	Punctured codes	k	$n - l$	d_{min} - 1	$t = \lfloor (d_{min}-1)/2 \rfloor$	$d_{min} \leq d_{original}$
5	Lengthened codes	$k + l$	n	d_{min} - 1	$t = \lfloor (d_{min}-1)/2 \rfloor$	$d_{min} \leq d_{original}$
6	Single parity check (SPC) codes	k	$k + 1$	all odd numbers of errors		$d_{min} = 2$
7	Duplicate-add-parity (DAP) codes	k	$2k + 1$		1	$d_{min} = 3$
8	Hamming codes	$2^r - 1 - r$	$2^r - 1$	2	1	$d_{min} = 3$, syndrome decoding method
9	Extended Hamming (EH) single-error-correcting and double-error-detecting codes (SEC-DED)	$2^{r-1} - r$	$2^{r-1} - 1$	2	1	$d_{min} = 4$, up to 3 errors can be detected if no error correction is attempted
				error correction and detection at the same time		
10	Shortened Hamming codes	$k - l$	$n - l$	SEC-DED codes if $d_{min} \geq 4$		$d_{min} \geq 3$
11	Hsiao codes (a special case of EH SEC-DED codes)	$2^{r-1} - r$	$2^{r-1} - 1$	2	1	$d_{min} = 4$, reduced calculation delay and less hardware requirements than that of EH codes
				error correction and detection at the same time		
12	Single-error correcting (SEC) codes with interleaving for dealing with two-bit burst errors	k	n		2-bit error burst	row-column interleaving with local hardwire connections
13	Cyclic codes	k	n	d_{min} - 1	$t = \lfloor (d_{min}-1)/2 \rfloor$	encoding with simple linear feedback shift register (LFSR) or a generator matrix
14	Shortened cyclic codes, or cyclic redundancy check (CRC) codes, CRC-t	$k - l$	$n - l$		t-bit error burst	good burst error detection, less efficient to detect multiple random errors
15	Bose-Chaudhuri-Hocquenghem (BCH) codes, $GF(2^m)$	$k \geq n - mt$, $m \geq 3$	$2^m - 1$		t multiple random errors	Berlekamp-Massey algorithm (BMA) for the error location polynomial
16	Reed-Solomon (RS) codes, $GF(2^{\boldsymbol{m}})$	$n - 2t$	$2^m - 1$		t	a class of non-binary BCH codes
17	SPC product codes	$k_1 \times k_2$	$n_1 \times n_2$		1	$n_1=k_1+1$, $n_2=k_2+1$
18	Hamming product codes	$k_1 \times k_2$	$n_1 \times n_2$		both random and burst errors	$d_{min} = d_1 \times d_2$

Note: In all tables, d_{min} is the minimum Hamming distance between the codewords in a code.

Table 3. List of ECC examples discussed in the fourth book chapter.

No.	Code	Mes-sage bits	Code-word bits	Error-detecting capability	Error-correcting capability	Comment, page number(s)
1	A (7,4) systematic linear block code	4	7	2	1	d_{min} = 3, p. 53
2	A (6,3) shortened block code	3	6			p. 56
3	A (8,4) extended block code	4	8			p. 56
4	Hamming code H(7,4)	4	7	2	1	d_{min} = 3, p. 59
5	Shortened/standard H(n, k) codes	k	n			Table 4.1, p. 62
6	Shortened/standard EH(n + 1, k) codes	k	n + 1			Table 4.1, p. 62
7	EH(8,4) SEC-DED code	4	8	2	1	d_{min} = 4, syndrome decoder, p. 62
				error correction and detection at the same time		
8	Hsiao SEC-DED code	4	8	2	1	d_{min} = 4, p. 62
				error correction and detection at the same time		
9	CRC-5, generator polynomial $g(x) = 1 + x^2 + x^4 + x^5$			5-bit error burst		also used in International Telecommunication Union (ITU) Telecommunication Standardization Sector (ITU-T) Recommendation G.704, p. 66
10	BCH codes, $m \leq 7$				t = 1, 2, 3	m = 3-7, Table 4.2, p. 66
11	BCH(15,5) code	5	15		3	m = 4, p. 67
12	Hamming product code $C_p(8 \times 8, 4 \times 4)$	16	64		5	p. 78

Two operating modes are considered in the book. The H(71,64) code is used in the low noise mode and Type-II HARQ with EH C_P in the high noise mode. The flexibility allowed by the configurable error control system to dynamically adjust the coding scheme while at the same sharing the same hardware components results in improved average energy efficiency.

The development of decision making algorithms for the choice of a coding scheme depending on the noise voltage deviation and other system characteristics is a separate subject which deserves a thorough investigation. Such algorithms must be simple enough for on-chip implementation and at the same time based on partial statistical accumulation of data and the use of intelligent technologies. With the increase of the design complexity of multi-core integrated circuits, some advanced concepts of the theory of control systems may become applicable to the nanoscale region after proper modifications. The introduction of redundancy for fault tolerance and hardware sharing for reconfigurable circuits is a preparation for the next step, the design of evolvable on-chip interconnects. This is especially important for deep space applications in the presence of damaging radiation from the sun and outer space which could damage not only the metal wires but also the encoder/decoder circuits.

Table 4. List of ECC examples discussed in the fifth book chapter.

No.	Code	Mes-sage bits	Code-word bits	Error-detecting capability	Error-correcting capability	Comment, page number(s)
1	CRC-8, generator polynomial $g(x) = 1 + x + x^2 + x^8$			8-bit error burst		inefficient to detect timing errors when used alone, also used in ITU-T Recommendation I.432.1, p. 83
2	Four EH(22,16) SEC-DED codes with row-column interleaver	64	72	2	1	p. 89
				error correction and detection at the same time		
3	EH(72,64) SEC-DED code	64	72	2	1	pp. 97, 98, 103
				error correction and detection at the same time		
4	CRC-5, generator polynomial $g(x) = 1 + x^2 + x^5$			5-bit error burst		inefficient to detect multiple random errors, also used in Universal Serial Bus (USB) token packets, p. 95
5	H(71,64)				1	pp. 98-115
6	Product code, *EH*(22,16) row encoding and EH(8,4) column encoding	4×16			multiple random and burst errors	p. 109
7	BCH(85,64)	64	85		3, multiple random errors	7-stage pipelined decoder architecture, p. 110
8	RS(85,65)	65	85		multiple errors within two symbols	7-stage pipelined decoder architecture, p. 110

The sixth book chapter goes beyond the conventional ECC techniques and addresses the uncertainty caused by crosstalk delay and the timing errors associated with it. Several coding techniques which control the binary content of adjacent bit positions to reduce the capacitive crosstalk are shown in Table 5. A list of crosstalk reduction examples is shown in Table 6. The duplication of the message bits and the inclusion of extra parity check bit(s) increase the number of metal wires following to the exclusion of undesired binary patterns in every two or three adjacent bits of the codewords as with the DAP, DSAP and BSC codes in Table 5. The decoding capabilities of the CAMEC, CADEC, JTEC and JTEC-SQED codes in Table 5 are enhanced by first encoding the message bits with Hamming or extended Hamming (EH) codes for the calculation of parity bits. The JTEC and JTEC-SQED codes with Hsiao code have smaller codec delay and area.

A separate section is provided for a unified coding framework which combines crosstalk avoidance codes (CACs) with conventional error control coding (ECC). Three CACs are used for this purpose: forbidden overlap condition (FOC) codes; forbidden transition condition (FTC) codes; and forbidden pattern condition (FPC) codes. As the efficiency of said codes is affected by the increase of the number of message bits, a hierarchical encoding method is applied by creating small groups of message bits which are encoded with FOC(5,4), FTC(4,3) or FPC(5,4) codes.

Table 5. List of error-control codes introduced in the sixth book chapter.

No.	Code	Mes-sage bits	Code-word bits	Error-detecting capability	Error-correcting capability	Comment
1	Duplicate-add-parity (DAP) codes	k	$2k + 1$		1	$d_{\min} = 3$
2	Duplicate-shield-add-parity (DSAP) codes				1	
3	Boundary shift code (BSC)	k	$2k + 1$		1	$d_{\min} = 3$
4	Crosstalk avoidance and multiple error correction code (CAMEC): H(n_H,k) + DAP	k	$2n_H + 1$		3	$d_{\min} = 7$
5	Crosstalk avoiding double error correction (CADEC) codes				2	
6	Joint crosstalk avoidance and triple error correction (JTEC) codes				3	$d_{\min} = 7$
7	Joint crosstalk avoidance and triple error correction (JTEC) and simultaneous quadruple error detection code (JTEC-SQED) codes			4	3	$d_{\min} = 8$
				error correction and detection at the same time		

One lambda code OLC(8,4) is also considered. Then a H(38,32) Hamming code (HC) is combined with the FOC, FTC and OLC codes and the performance evaluation of the resultant FOC + HC, FTC + HC, and OLC + HC coding schemes together with the BSC, DAP and DSAP codes listed in Table 5 is carried out. A comparison with the conventional Hamming codes shows that the unified coding framework has the tendency to achieve a larger speedup.

Another separate section is also provided for the combination of skewed transitions with conventional error control coding (ECC). It is shown that the HC + skewed transitions scheme has minimum link area, minimum delay uncertainty, minimum link energy consumption and minimum total energy consumption in comparison with the DAP, BSS, HC + FOC, HC + FTC, and HC + FPC coding schemes.

The book considers in order of increasing complexity the error control coding techniques which are suitable for detecting and correcting errors occurring in on-chip interconnects. Results from recently published research papers are the basis for most of the specialized technical and graphical content in all book chapters.

Concerning the use of abbreviations, some mistyping is observed occasionally. For example, the abbreviation “SEC-DEC” appears in pages 27 and 61 and also in the List of Symbols in page 146 while it should be “SEC-DED” as correctly written in pages 45, 60, 122, 134, 135 and 137. The inclusion of abbreviations in the List of Symbols and the Index is relaxed and some abbreviations are not listed there, for instance: in-line test (ILT), intellectual property (IP), system on a chip (SoC), special-purpose processing unit (SPU), ternary-error-correcting (TEC), etc.

Table 6. List of crosstalk reduction examples discussed in the sixth book chapter.

No.	Code	Mes-sage bits	Code-word bits	Error-detecting capability	Error-correcting capability	Comment, page number(s)
1	DAP(9,4)	4	9		1	d_{min} = 3, p. 118
2	BSC(9,4)	4	9		1	d_{min} = 3, p. 119
3	CAMEC: H(38,32) + DAP	32	77		3	d_{min} = 7, p. 120
4	CADEC: H(71,64) + DAP	64	143		2	p. 121
5	FOC(5,4)	4	5			p. 125
6	FOC(40,32) with 8 FOC(5,4) codes	32	40			hierarchical method, p. 124
7	FTC(4,3)	3	4			p. 125
8	FTC(53,32) with 11 FTC(4,3) codes	32	53			hierarchical method, p. 126
9	FPC(5,4)	4	5			p. 126
10	Two FPC(5,4) codes					hierarchical encoding, Fig. 6.9, p. 127
11	FPC(52,32) with 8 FPC(5,4) codes	32	52			hierarchical method, p. 127
12	One lambda code OLC(8,4)	4	8			two adjacent bit boundaries in the codebook cannot both be of 01- or 10-type, p. 128
13	H(38,32) code combined with FOC, FTC, FPC and OLC codes					p. 128
14	Data mapping algorithm with H(12,8) code	8	12			p. 134
15	Data mapping algorithm with 4 H(7,4) codes	16	29			pp. 135-136
16	H(71,64) code combined with skewed transitions	64	71		1	pp. 136-141

If the essence of the book is to be written in a short summary, the most notable coding concepts include:

- EH SEC-DED codes and Hsiao codes;
- Hamming product codes with a three-stage pipelined decoding method;
- Type-II HARQ with extended Hamming product codes and a segmentation of the parity check bits for obtaining shorter codewords;
- configurable error control systems and hardware sharing allowing switching between coding modes for energy efficiency; and
- skewed transitions combined with Hamming codes and the use of a data mapping algorithm.

Although the book content is oriented toward readers specializing in on-chip interconnects and error-control coding, the introductory information included in the first three book chapters is sufficient to understand the coding techniques. Certain details about the exact code configurations require further reading from the references given at the end of each book chapter.