High-Performance Approximate On-Chip Communication

Dissertation zur Erlangung des akademischen Grades Doktor-Ingenieur (Dr.-Ing.) im Fach Elektrotechnik und Informationstechnik

Amir Najafi

1. Gutachter:	Prof. Dr. Alberto García-Ortiz
2. Gutachter:	Prof. Dr. Antonio José Acosta-Jiménez
Eingereicht am:	02.03.2021
Tag des Promotionskolloquiums:	28.04.2021

Universität Bremen

To my family and teachers

Abstract

English version

Interconnects do not gain from technology scaling as the computation units do. There is still no substitute for copper interconnects to meet the conductivity requirements and reduce dielectric permittivity. It is becoming more expensive to guarantee the interconnect's reliable error-free operation as the gains of technology scaling are diminishing. The on-chip communication is currently a bottleneck of the high-performance integrated circuits. Fundamental improvement of on-chip communication is essential.

Approximate computing has been thoroughly investigated for the computation sub-systems. Recently, approximate computing has been extended to the communication domain. The idea is to trade transmission quality for energy or/and performance efficiency. Approximate communication, on the one hand, provides a practical solution to maintain the benefits of the technology scaling; and, on the other hand, exploits the gap between the accuracy requirements of the applications and that provided by communication sub-systems. Besides, the unfulfilled potential of full-system approximation can only be exploited by approximating all sub-systems, including the communication. However, prerequisites for the development and efficiency characterization of effective on-chip codings and approximate techniques are precise energy and delay models.

Traditionally, delay and energy models do not consider variations, resulting in inaccurate and only partially useful models. The misalignment effect, that

is, the variation in the delay of the transmitted signals has been conventionally overlooked.

In this dissertation, the first high-level misalignment-aware energy and delay models for narrow multi-segment global interconnects are proposed to address the problem of conventional models. The proposed models have been evaluated using simulations. Results show a significant improvement in the accuracy of the energy and delay estimations.

Three different techniques are proposed to exploit approximate computing's considerable potential in the communication domain. **First**, two memoryless encoding approaches for approximate communications to reduce the integer-value deviation are presented. Restricted to area-constrained applications, we propose a Combined Integer-Value (CIV) coding technique based on the input signals' swap and inversion. Moreover, a Crosstalk-Avoidance-based Integer-Value (CA-IV) coding technique for applications with a more relaxed area constraint is presented. The optimal mapping of the data words and codewords combined with the selective inversion of input signals minimizes the error magnitude in this coding technique. **Second**, taking into account the challenges of classical wave-pipelining in error-free operation, the concept of Stochastic Wave-Pipelining (SWP) communication is proposed. It relaxes the limiting constraints of the classical wave-pipelining and allows its practical implementation. Besides, extra performance improvement by accepting a tolerable error is achievable. Unlike most existing approximate techniques, SWP has the quality to be combined with other approximate techniques. **Third**, an Alternating Bit-Truncation (ABT) technique is introduced. ABT sets a certain number of LSBs to zero. The inactive lines are used as virtual shielding to shield the data lines. The inactive lines do not contribute to dynamic energy consumption, resulting in a very power-efficient technique. Besides, depending on the number of truncated lines, ABT reduces the crosstalk noise that leads to energy saving and performance improvement. This technique can be implemented with almost no overhead. A comprehensive experimental result validated the proposed approximate communication approaches.

The proposed approximate communication methods' applicability is also studied in Network-on-Chip (NoC), where they are compared with a state-of-the-art compression technique. Results show the superiority of proposed techniques for different design criteria.

German version

Im Gegensatz zu den Recheneinheiten profitieren Verbindungen nicht von der Technologieskalierung. Es gibt immer noch keinen Ersatz für Kupferverbindungen, um die Leitfähigkeitsanforderungen zu erfüllen und die dielektrische Permittivität zu verringern. Es wird immer teurer, den zuverlässigen fehlerfreien Betrieb der Verbindung zu gewährleisten, da die Vorteile der Technologieskalierung abnehmen. Die On-Chip-Kommunikation ist derzeit ein Engpass bei den integrierten Hochleistungsschaltungen. Eine grundlegende Verbesserung der On-Chip-Kommunikation ist unerlässlich.

Das ungefähre Computing wurde für die Berechnungssubsysteme gründlich untersucht. In letzter Zeit wurde das ungefähre Rechnen auf den Kommunikationsbereich ausgedehnt. Die Idee ist, Übertragungsqualität gegen Energie- oder/und Leistungseffizienz zu tauschen. Die ungefähre Kommunikation bietet einerseits eine praktische Lösung, um die Vorteile der Technologieskalierung aufrechtzuerhalten, und nutzt andererseits die Lücke zwischen den Genauigkeitsanforderungen der Anwendungen und denen, die von Kommunikationssubsystemen bereitgestellt werden. Außerdem kann das unerfüllte Potenzial der vollständigen Systemnäherung nur durch Annäherung aller Teilsysteme einschließlich der Kommunikation ausgeschöpft werden. Voraussetzungen für die Entwicklung und Effizienzcharakterisierung effektiver On-Chip-Codierungen und Näherungstechniken sind jedoch präzise Energie- und Verzögerungsmodelle.

Traditionell berücksichtigen Verzögerungs- und Energiemodelle keine Variationen, was zu ungenauen und nur teilweise nützlichen Modellen führt. Der Fehlausrichtungseffekt, d.h. die Änderung der Verzögerung der übertragenen Signale, wurde herkömmlicherweise übersehen.

In dieser Dissertation werden die ersten hochgradig fehlausrichtungsbewussten Energie- und Verzögerungsmodelle für enge globale Mehrsegmentverbindungen vorgeschlagen, um das Problem herkömmlicher Modelle anzugehen. Die vorgeschlagenen Modelle wurden mithilfe von Simulationen bewertet. Die Ergebnisse zeigen eine signifikante Verbesserung der Genauigkeit der Energie- und Verzögerungsschätzungen.

Es werden drei verschiedene Techniken vorgeschlagen, um das beträchtliche Potenzial des ungefähren Rechnens im Kommunikationsbereich auszuschöpfen. Zunächst werden zwei speicherlose Codierungsansätze für die ungefähre Kom-

munikation vorgestellt, um die Abweichung des ganzzahligen Werts zu verringern. Aufgrund von Anwendungen mit eingeschränkten Bereichen schlagen wir eine CIV-Codierungstechnik (Combined Integer-Value) vor, die auf dem Swap und der Inversion der Eingangssignale basiert. Darüber hinaus wird eine CA-IV-Codierungstechnik (Crosstalk-Avoidance-based Integer-Value) für Anwendungen mit einer entspannteren Bereichsbeschränkung vorgestellt. Die optimale Abbildung der Datenwörter und Codewörter in Kombination mit der selektiven Inversion der Eingangssignale minimiert die Fehlergröße bei dieser Codierungstechnik. Zweitens wird unter Berücksichtigung der Herausforderungen des klassischen Wave-Pipelining im fehlerfreien Betrieb das Konzept der Stochastic Wave-Pipelining-Kommunikation (SWP) vorgeschlagen. Es lockert die einschränkenden Begrenzungen des klassischen Wave-Pipelining und ermöglicht dessen praktische Umsetzung. Außerdem ist eine zusätzliche Leistungsverbesserung durch Akzeptieren eines tolerierbaren Fehlers erreichbar. Im Gegensatz zu den meisten vorhandenen Näherungstechniken kann SWP mit anderen Näherungstechniken kombiniert werden. Drittens wird eine ABT-Technik (Alternating Bit-Truncation) eingeführt. ABT setzt eine bestimmte Anzahl von LSBs auf Null. Die inaktiven Leitungen werden als virtuelle Abschirmung verwendet, um die Datenleitungen abzuschirmen. Die inaktiven Leitungen tragen nicht zum dynamischen Energieverbrauch bei, was zu einer sehr energieeffizienten Technik führt. Weiterhin reduziert ABT abhängig von der Anzahl der abgeschnittenen Leitungen das Übersprechrauschen (crosstalk noise), das zu Energieeinsparungen und Leistungsverbesserungen führt. Diese Technik kann nahezu ohne Mehraufwand implementiert werden.

Ein umfassendes experimentelles Ergebnis bestätigte die vorgeschlagenen ungefähren Kommunikationsansätze. Die Anwendbarkeit der vorgeschlagenen ungefähren Kommunikationsmethoden wird auch in Network-on-Chip (NoC) untersucht, wo sie mit einer modernen Komprimierungstechnik verglichen werden. Die Ergebnisse zeigen die Überlegenheit der vorgeschlagenen Techniken für verschiedene Entwurfskriterien.

Acknowledgement

I cannot be enough thankful for being able to pursue my Ph.D. in ITEM.ids group at the University of Bremen. During my Ph.D. I received invaluable support, guidance, and encouragement from my colleagues, friends, and family.

I am most grateful to my supervisor, Prof. Dr. Alberto García-Ortiz, for his patient mentorship and continuous guidance. I consider myself extremely lucky for being able to carry out my Ph.D. under his supervision. His teaching, leadership, and guidance have been instrumental in my academic and professional development. I also would like to thank Prof. Dr. Antonio José Acosta-Jiménez for reviewing my work and providing valuable feedback.

I am also very grateful to my colleagues. I would like to thank Kerstin Janssen, the administrator of ITEM.ids group at the University of Bremen, for her wholehearted support during my study in Bremen. Her support not only enabled me to conduct successful research work but also facilitated my integration into the new culture and society. I also would like to thank Ardalan, my brother, and my colleague, which I cannot imagine being where I am currently without him. I had his generous support whenever I needed his help all the time during my study. I would like to thank all current and former member of the ITEM group which I shared many interesting moments with them and learned a lot from them, specially Yanqiu Huang, Wanli Yu, Lennart Bamberg, Robert Schmidt, Yarib Nevarez, Daniel Gregorek, Behnam Razi, Andreas Beering, Jakob Döring, and Peter Lutzen.

Among all my beloved friends, I would like to thank Freya Gröning and

Charlotte Wagner who made me feel at home from the very beginning in Bremen. I do appreciate their presence in my life.

Finally, I would express a deep sense of gratitude to my family as well as my partner. My parents, Akram and Akbar, have always stood by my side and supported my adventures. I am and will be always grateful to my parents because of their kindness and all their sacrifices. Special thanks are due to my one and only loving sister Elahe and her husband Shahin who always strengthened my morale by standing by me in all situations. I am as well grateful for their supports and kindness. This gratitude is also extended to my sympathetic and caring friend, Hedieh Farhandi. I am happy to have her as part of our small family in Germany. I also would like to thank my partner, Stefanie Jahn. Steffi patiently accompanied me during my journey through the Ph.D. and gave me love, happiness, and support. I am grateful to her as well as her parents, Gudrun and Matthias, for being supportive and caring all the time.

Glossaries

Mathematical and Physical Symbols

C_g	Self/ground capacitance.
C_c	Coupling capacitance.
κ	Ratio of coupling capacitance to the self capacitance.
l	Wire length.
$\mathtt{t}$	Wire hight.
h	Distance between bus wires and substrate.
R	Resistance.
$R_\square$	Per-unit-length resistance.
ρ	Electrical resistivity.
k	Dielectric constant.
s	Spacing between adjacent wires.
N	Number of interconnects segments.
w	Wire width.
V_{dd}	power-supply voltage.
R_d	Driver resistance.
b_i	Logical binary values on the metal-wire i.
$v_i(t)$	Voltage in node i at time t.
Δb_i	Self switching of the metal wire i.
$\uparrow$	Logical 0 to 1 transition.
$\downarrow$	Logical 1 to 0 transition.
$\bullet$	Quiet line with no transition.
$\delta_{i,j}$	The coupling switching between the two direct adjacent metal wires i and j.
$\mathcal{L}\{\}$	Laplace transform.
τ_0	The delay of an ideal cross-talk free wire.
T_{t_i}	Self-switching factor of wire i.
T_{e_i}	Coupling switching factor of wire i.
$E_{e,i}$	Dynamic energy extracted from driver in i^{th} line of the bus.
$\mathbb{E}\{\}$	Expectation operator.
P_D	Dynamic power.

A	Average switching activity.
F	Clock Frequency.
$\uparrow_i$	Existence function specifying the existence of rising transition in wire i.
$\downarrow_i$	Existence function specifying the existence of falling transition in wire i.
$\bullet_i$	Existence function specifying the existence of no transition in wire i.
$\hat{Y}^+ \vert X^-, X^+$	The possibly received values when X^- and X^+ transferred through the bus.
$Pr.$	Output value probability.

Acronyms

MPSoC	Multiprocessor System-on-Chip.
CAC	Crosstalk-Avoidance Coding.
NoC	Network-on-Chip.
IVC	Integer-Value Coding.
CA-IV	Crosstalk Avoidance based Integer-Value coding.
SWP	Stochastic Wave-Pipelining.
ABT	Alternating Bit Truncation.
CIV	Combined Integer-Value coding.
ITRS	International Technology Roadmap for Semiconductor.
FEC	forward error correction.
ARQ	automatic repeat request.
VLSI	Very Large-Scale Integration.
IC	Integrated Circuit.
EDA	Electronic Design Automation.
SNR	Signal-to-Noise Ratio.
SoC	System-on-Chips.
BI	Bus-Invert.
FI	Full-Invert.
FPF	Forbidden Pattern Free.
FTF	Forbidden Transition Free.
LSB	Least Significant Bit.
CMOS	Complimentary metal-oxide-semiconductor.
MAA	Misalignment-Aware energy and delay models.
MF	Misalignment Factor.
MSE	Mean Square Error.
OLS	Ordinary Least Square.
MSB	Most Significant Bit.
OCR	Optical Character Recognizer.
EDP	Energy-Delay Product.
SIFT	Scale-Invariant Feature Transform.
SIMD	Single instruction, multiple data.
MIPS	Microprocessor without Interlocked Pipelined Stages.

GPS	Global Positioning System.
PE	Processing Element.
NI	Network Interface.
QoS	Quality of Service.
SAF	Store And Forward switching.
VAT	Virtual cut through.
WH	Wormhole switching.
BER	Bit-Error-Rate.

Contents

CHAPTER 1

Introduction

Contents

1.1 Motivation

The processing technology's inexorable march continues to feature sizes beyond 5 nm. The unprecedented increase in density and parallelism leads to smaller geometries of semiconductor devices and faster computations. Current technologies enable the integration of millions of transistors in a single die. However, to maintain sufficient routing densities, fast and compact interconnects are required as well.

The wiring system delivers power to each transistor, distributes the clock to latches, and transfers data and control signals throughout the chip. Aggressive scaling results in the reduced interconnect pitch and the enhanced crosstalk effect. In principle, metallic wires can be either fast or dense but not both at the same time [1]. A smaller wire pitch increases resistance, while greater height

and width increase the parasitic capacitance. In both cases, propagation delay and energy consumption of wires are adversely affected. Local interconnects' length decreases with scaling, and therefore, shrinking metal wires' pitch for a dense routing does not adversely affect their performance metrics. On the other hand, in the advent of new technology nodes, the long global wires' RC delay is comparable with logic gates.

Global interconnects are even longer in Multiprocessor System-on-Chip (MPSoC). In modern integrated circuits, billions of logic gates and memory elements are linked by up to 16 levels of stacked wires [1]. Scaling in lateral dimensions (pitch) with almost unchanged wire's length leads to large coupling capacitance between adjacent interconnects that becomes significantly larger than ground capacitance. In general, the gap between the gate delay and interconnection delay is increasing with new technology nodes.

Interconnect designs are becoming a dominant issue in high-performance integrated circuits, and if not addressed, interconnection limits can potentially undermine Moore's law [2]. Fundamental improvement of interconnects is required.

Accurate interconnect energy and delay modelings are critical for the development and evaluation of optimization techniques. Interconnect models enable chip power and performance optimization early in the design flow. There are many research works focusing on energy and delay modeling of the interconnects [3, 4, 5, 6, 7]. Traditionally, delay and energy models do not consider variations and nanometric effects, leading to inaccurate and only partially useful models. Crosstalk and noise associated with decreasing geometries should be considered for modeling of interconnects energy and delay. Almost all the existing techniques share an implicit assumption that signals are synchronized throughout the interconnect. Signals' misalignment as an intrinsic or extrinsic phenomenon should be considered for the accurate development of interconnects energy and delay models.

Bus encoding is one of the most well-known approaches that can effectively mitigate the impact of crosstalk and improve the performance, and power consumption of interconnects [8, 9, 10, 11]. Bus encoding techniques, including Crosstalk-Avoidance Coding (CAC) and low-power codes, manipulate the input data before transmitting through the bus to eliminate specific undesirable data patterns [12]. Despite considerable improvements using bus encoding,

there are two undesirable attributes associated with these techniques, especially in current dense technology nodes: first, most bus coding techniques (even memoryless coders) impose different area and energy overhead due to the coder/decoder and redundant lines; second, they are designed to ensure the reliable transmission of data and therefore, they require expensive bounds to guarantee a reliable operation. Allocating more resources does not solve the problem as rising performance demands are outpacing the resource budget growth. The on-chip communication is reaching the fundamental limits of the resources required for a fully reliable operation [13].

Approximate computing has emerged as an attractive computing paradigm for many computing applications with relaxed error constraints. In principle, approximate computing exploits the gap between the level of accuracy required by application or user and that provided by computing subsystems [14]. Many computing-intensive applications such as machine learning, signal processing, and computer vision possess a high degree of error resilience. Approximate computing has a great potential to leverage application resiliency to error. For example, it has been shown that for 5% loss of accuracy for k-mean clustering algorithm up to 50 times energy saving can be obtained [15]. However, exploiting the approximate computing's full potential is only possible by approximating all the subsystems, including the communication subsystem.

Recently, some research works expand the hardware approximation into communication subsystems [16, 17, 18, 19, 20, 21, 22]. Approximate communication techniques provide promising improvement both in terms of energy and performance for an acceptable error. Most of the existing techniques are based on lossy compression of the data. For example, Ref.[16] shows that employing a lossy compression method up to 31% reduction in the average delay of Network-on-Chip (NoC) with 10% quality loss is possible. In [19], the approximate equivalent of dictionary-based and frequent pattern compression techniques can reduce the average packet latency, respectively, by 40.7% and 46.5% in comparison with conventional baseline NoC data transmission. However, in this case, a close inspection of the additional hardware overhead is required.

1.2 Main Contributions and Publications

The core of this thesis is made based on two hypotheses: First, the effective development and evaluation of high-level optimization techniques are only achievable using precise and abstract models for power consumption and interconnects' performance. This hypothesis is made based on the observation of temporal signals alignment analysis presented in the literature [5, 3, 4, 23] and is examined and proved in Chapter 2 of this thesis. Second, the untapped potential of full-system approximation cannot be exploited without the development and utilization of effective approximate communication techniques. The advantages of the full-system approximation have been highlighted in [24, 25].

Based on these working hypotheses, in this thesis, we develop more accurate models than the existing high-level delay and energy models by considering variability and nanometric effects. We also develop multiple light-weight approximate/stochastic techniques that provide drastic improvement to the state-of-the-art.

The main contributions of this dissertation are the followings:

- We present precise high-level energy and delay models for the power and performance estimation of parallel on-chip interconnects. The proposed models provide an accurate estimation of pattern dependent energy and propagation delay considering the inherent effect of signal's alignment. The proposed models enable (1) developing effective on-chip optimization techniques; (2) evaluation of the coding technique early in the design stages.

- Exploiting the integer value-representation of signals, we propose two Integer-Value Coding (IVC) methods to minimize integer-value of timing errors caused by voltage/frequency overscaling. First, considering applications with stringent area budget, a technique combining swap and inversion of selective signals is proposed. Second, for less stringent area constraint applications, a memory-less Crosstalk Avoidance based Integer-Value coding (CA-IV) scheme is employed that permits a more aggressive voltage/frequency scaling.

- Reviving the concept of classical wave-pipelined interconnects, we devise Stochastic Wave-Pipelining (SWP). SWP offers the following major

benefits:

(1) It relaxes the selection of sampling time for wave-pipelined structures. Therefore, it enables the practical utilization of the wave-pipelining scheme.

(2) SWP as a concurrent technique to other approximate communication approaches can be integrated with the existing techniques for extra gains.

(3) Additional energy and speed improvements are achievable by voltage and frequency scaling for an acceptable error.

These benefits are achieved while SWP imposes negligible hardware overhead.

- We propose Alternating Bit Truncation (ABT) that is a simple yet effective approximate communication technique. It enhances the simple bit-truncation concept by setting truncated wires to zero. It uses the truncated wires as virtual shields to mitigate the undesirable effect of crosstalk noise. Thus, ABT provides twofold benefits for approximate data transmission:

  (1) A drastic energy saving through inactive wires. An extra improvement in energy as a by-product of inactive wires is crosstalk noise reduction, utilizing the truncated lines as virtual shielding.

  (2) Performance improvement as a result of removing worst-case patterns. A careful swap of the signals leveraging the CMOS interconnects' physical properties can also provide an additional improvement.

This thesis's outcomes are partly presented to the scientific community through internationally renowned and peer-reviewed journals and conference proceedings. The complete list of the related publications are as follows:

Modeling

1. **Amir Najafi**, Lennart Bamberg, and Alberto Garciá-Ortiz. 2020. Misalignment-aware energy modeling of narrow buses for data encoding schemes. Integr. VLSI J. 72, C (May 2020), 58-65. doi:vlsi.2020.01.001

2. **Amir Najafi**, L. Bamberg, A. Najafi and A. Garciá-Ortiz, "Misalignment-aware delay modeling of narrow on-chip interconnects considering variability," 2018 7th International Conference on Modern Circuits and Systems Technologies (MOCAST), Thessaloniki, 2018, pp. 1-4, doi: 10.1109/MOCAST.2018.8376593.

3. Lennart Bamberg, **Amir Najafi**, and Alberto Garciá-Ortiz. Edge effects on the TSV array capacitances and their performance influence. Elsevier Integration, 61:1-10, 2018, doi: vlsi.2017.10.003.

4. **Amir Najafi**, L. Bamberg, A. Najafi and A. Garciá-Ortiz, "Energy modeling of coupled interconnects including intrinsic misalignment effects," 2016 26th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS), Bremen, 2016, pp. 262-267, doi: 10.1109/PATMOS.2016.7833697.

Optimization Techniques

1. **Amir Najafi**, A. Najafi and A. Garcia-Ortiz, "Stochastic Wave-Pipelined On-Chip Interconnect," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 5, pp. 841-845, May 2020, doi: 10.1109/TCSII.2020.2984194. **(The original submission on ISCAS conference Awarded runner best paper)**

2. **Amir Najafi**, L. Bamberg, A. Najafi and A. Garciá-Ortiz, "Integer-Value Encoding for Approximate On-Chip Communication," in IEEE Access, vol. 7, pp. 179220-179234, 2019, doi: 10.1109/ACCESS.2019.2959446.

3. **Amir Najafi**, L. Bamberg, G. P. Vayá and A. Garciá-Ortiz, "A Coding Approach to Improve the Energy Efficiency of Approximate NoCs," 2019 14th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), York, United Kingdom, 2019, pp. 74-81, doi: 10.1109/ReCoSoC48741.2019.9034965.

4. Lennart Bamberg, **Amir Najafi**, and Alberto Garciá-Ortiz. Edge effect aware low-power crosstalk avoidance technique for 3D integration. Elsevier Integration, 69:98-110, 2018.

5. Alberto Garciá-Ortiz, Lennart Bamberg, and **Amir Najafi**. Low-power coding: trends and new challenges. ASP Journal of Low Power Electronics, 13(3):356-370, 2017.

6. Lennart Bamberg, **Amir Najafi**, and Alberto Garciá-Ortiz. Edge effect aware crosstalk avoidance technique for 3D integration. In International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS), pages 1-8. IEEE, 2017. **(Awarded best paper)**.

1.3 Dissertation Outline

This dissertation is organized into seven chapters.

Following the present introductory chapter, in Chapter 2, relevant background information on 2D interconnects are provided. An overview of the conventional interconnect modeling is presented. The existing energy and delay models' problem is discussed, and the necessity of high-level models that enable the incorporation of variations due to new dense technologies are explicitly emphasized. In this chapter, the most well-known and relevant exact and approximate optimization techniques are reviewed as well.

The intrinsic effect of signals' misalignment is investigated in Chapter 3. It is shown that the misalignment effect is a relevant phenomenon that has a direct impact on the accuracy of high-level energy and delay models of parallel buses. In this regard, the misalignment-aware energy and delay models are proposed and validated in this chapter.

The focus of Chapter 4 is on approximate communication techniques. Various approximate schemes are presented. The main objective is to develop techniques that enable an effective trade-off between resource usage and error. In particular, Combined Integer-Value coding (CIV) and CA-IV are developed targeting, receptively, area-stringent and looser area constraint applications. Moreover, SWP and ABT techniques are presented in this chapter.

In Chapter 5, the proposed approximate techniques are evaluated using comprehensive simulations. Proposed methods are validated using synthetic and real-world input data. Besides, several case-studies verify the effectiveness of designed techniques.

In Chapter 6, the proposed approximate techniques' applicability is evaluated in the context of NoC communication. Practical configurations to integrate the

proposed techniques into the NoC is presented. The proposed techniques are compared with a state-of-the-art compression technique.

Finally, Chapter 7 concludes this dissertation, where this dissertation's contributions are summarized.

CHAPTER 2

Background & Previous Works

Contents

2.1 Introduction

As stated by International Technology Roadmap for Semiconductor (ITRS), interconnect performance is at the forefront as a critical challenge to achieve overall chip performance [26]. The data transmission on an on-chip interconnect is costly. It requires charging and discharging of the parasitic capacitances associated with the wire and therefore dissipates energy. The aggressive device scaling creates new challenges for interconnects: first, conventional energy and delay models are not precise anymore; and second, the crosstalk effect is becoming increasingly severe in parallel interconnects so that the performance of bus-based interconnects has become the bottleneck to overall system

performance [12].

Interconnects models are primarily employed to develop and evaluate optimization techniques. A high-level accurate estimation model is mainly favorable for high-level designers of bus encoding techniques, who might not have the required circuit-level electrical simulation expertise. A high-level model supports the evaluation of digital circuits early in design stages without using complex circuit simulation. Until recently, conventional delay and energy models could be used as robust models to predict energy and delay of interconnects. However, the variation as a result of technology scaling begins to decrease the conventional models' accuracy.

Many optimization techniques have been proposed in the literature to address the new challenges that interconnect are facing, including physical layer techniques, encoding approaches, error control schemes, and approximate compression.

The physical layer techniques is often used to mitigate the effect of crosstalk and improve the performance. The repeater insertion is an effective traditional technique for driving long interconnects. The uniform repeater insertion's primary objective is to minimize the propagation delay through a long interconnect [27]. The placement of repeaters is becoming increasingly challenging in today's technology nodes, mainly due to the inverse effect of repeaters' energy and performance overhead. Another physical approach to address the crosstalk induced energy and performance problems is to use shielding. The signal lines can be shielded using grounded conductors or alternating ground and V_{dd} lines. The reduced voltage swing is another useful technique to tackle the energy consumption problem. However, this technique results in a decreased noise margin and raises reliability issues when the primary goal is error-free data transmission. As a result of shrinking technology, the physical layer techniques are hardly sufficient to use as a standalone technique. Bus codings are approved as a valuable technique for energy and performance improvement along with physical approaches.

Bus encoding is an effective technique that aims at the delay, power reduction, and reliability improvement. Crosstalk avoidance coding techniques can improve the interconnect to operate faster while consuming lower power by eliminating certain classes of crosstalk patterns [12]. The original data (dataword) is coded using an encoder, and the output of the encoder (codeword) is transferred

through the interconnect. At the receiver side, the codewords are decoded, and the dataword is recovered. Typically, the number of bits of the codewords is higher than that of the datawords. Despite their efficiency, encoding techniques suffer from encoder and decoder (CODEC) complexity, overhead, and redundant wires. To effectively tackle the performance-reliability trade-off, error-control schemes such as forward error correction (FEC), and automatic repeat request (ARQ) [28] has been proposed. An extensive overview of these schemes is covered by [29]. Even though these mechanisms improve on-chip interconnects reliability, they negatively impact area, performance, and power. Error control mechanism introduces an overhead due to the additional logic, which in most cases is not affordable for on-chip interconnects [30, 31]. The languishing benefits of technology scaling in communication have pushed designers to look for more effective approaches.

As variability in communication behavior increases, it is difficult to achieve the deterministic behavior as performance and energy penalties should be paid to ensure reliable data transmission. As stated in the International Technology Roadmap for Semiconductors, "relaxing the requirement of 100% correctness for devices and interconnects may dramatically reduce costs of manufacturing, verification, and test". Approximate computing has recently drawn researchers' attention as a way to achieve a higher reduction in energy consumption and delay by allowing error in some applications. Primarily focused on computation units, approximate computing is currently extended to communication. There are potential benefits associated with approximate communication exploiting applications' inherent error resiliency, which should be exploited.

The rest of this chapter is organized as follows: first, We introduce the interconnect configuration. We explain some physical layer techniques that improve data transmission in buses. Second, we review the most relevant energy and delay models in the literature. Also, we derive the classical energy and delay models of the interconnect. The effect of variation and, in particular, the relative temporal alignment of signals in adjacent wires of the interconnect on model accuracy is studied. Third, the state of the art encoding techniques for interconnects' optimization are reviewed, and it is explained why supplementary or alternative approaches are required. Next, we provide a thorough review of approximate computing and its recent application in on-chip interconnection. Finally, we conclude the chapter.

2.2 Interconnect Configuration

2.2.1 Physical Modeling of Interconnects

Interconnects can be roughly divided into local and global interconnects based on their length and size. Local interconnects are relatively short and generally refer to connections between gates within a functional block. Local narrow interconnects that connecting the neighboring gates are usually routed in lower metal layers. The local interconnect delays can sometimes be significant, but they are relatively easier to deal with through physical layout, floorplanning, circuit re-timing, or logic optimization [12]. On the other hand, global interconnects can be as long as the length of the chip edge. They are used to connect different IP blocks and are generally routed in the higher metal layers. The length of the global interconnects is not scaled with technology, and consequently, they present a significant challenge in modern integrated circuits in terms of delay and power consumption.

There exist different bus types, including serial, parallel, synchronous, and asynchronous. The most common bus type is the synchronous parallel bus, which is employed due to its simplicity, good timing predictability, high throughput [12].

For on-chip buses, important electrical parameters are the resistance R, the inductance L, and the capacitance C. The significance of these parasitic can differ per application and working frequency. The inductive effect only appears in some applications, and it is not very important for most interconnects for data communication [32]. An RC model is usually adequate for low to medium operating frequencies; however, at frequencies exceeding a GHz, an RLC model is often necessary to characterize interconnects [33].

There are two dominant parasitic capacitances: ground capacitance and coupling (inter-wire) capacitance. The ground capacitor is formed between a wire and the substrate and is denoted by C_g. The coupling capacitance is a capacitor between two adjacent wires and is denoted by C_c. The ratio of the coupling capacitance to the ground capacitance is defined a $\kappa = {}^{C_c}/_{C_g}$ which is a function of the bus geometry. For processes with feature size of $1\,\mu$m or greater, $\kappa < 1$, indicating that C_c is negligible compared to the total capacitance. However, in current technology nodes, $\kappa \gg 1$ and C_c is a big

fraction of the total capacitance [12].

A wire's behavior can be approximated using different lumped elements, including L-model, π-model, and T-model. For short local interconnects with an almost negligible interconnect parasitic capacitance compared to load capacitance, a lumped RC structure is sufficient for modeling the interconnect [34]. However, for long global interconnects, a distributed model is used. This is because of the larger propagation delay of the transmission line compared to the gate delay [34]. It is common practice to model long wires with 3 to 5 elements of π-model for simulation [27]. Figure 2.1 shows a K distributed RC circuit of a wire using π-model lumped element. According to this figure, the wire is capacitively coupled to neighboring wires.

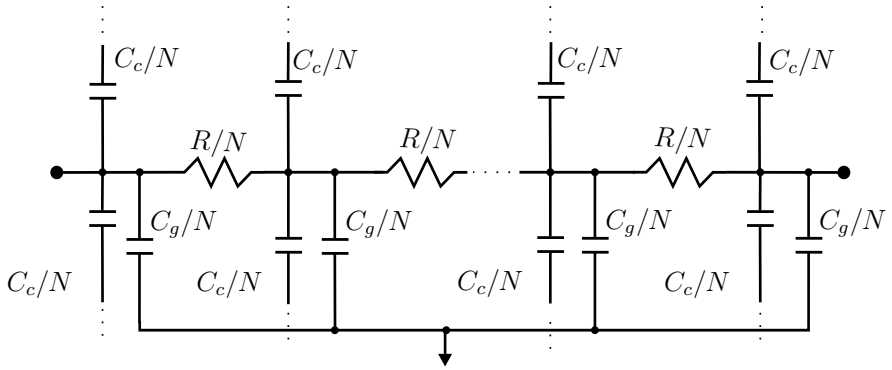


Fig. 2.1: A distributed RC circuit model of a wire with coupling capacitances connected to adjacent wires. The model comprises K π-model lumped elements.

In this thesis, we use synchronous parallel bus modeled using equivalent RC-lumped π-model circuit.

2.2.2 Physical Layer techniques

Floorplanning is the essential step in Very Large-Scale Integration (VLSI) design in which cells are placed on the layout surface. Despite the efforts of floorplanning to position critical communicating units close to one another, long interconnection is inevitable. Interconnects engineering is an essential step to optimize interconnection. There are several physical techniques as well as design choices to engineer wires for their performance metrics. After exploring wire geometry, we explain physical layer design methodologies, including shielding, repeater insertion, and low-swing signaling.

Wire Geometry

The wire geometry can be used to trade-off delay, energy, density, noise, and bandwidth. Figure 2.2 shows a parallel bus. According to this figure, w is the width of the wire and can be adjusted during the physical layout; l is the length and can be determined by the designer; $\mathtt{t}$, is height and depends on the technology node and the metal layer and is constant; s, is spacing between adjacent wires; h, is the distance between bus wires and substrate. The fabrication process determines the minimum width and spacing.

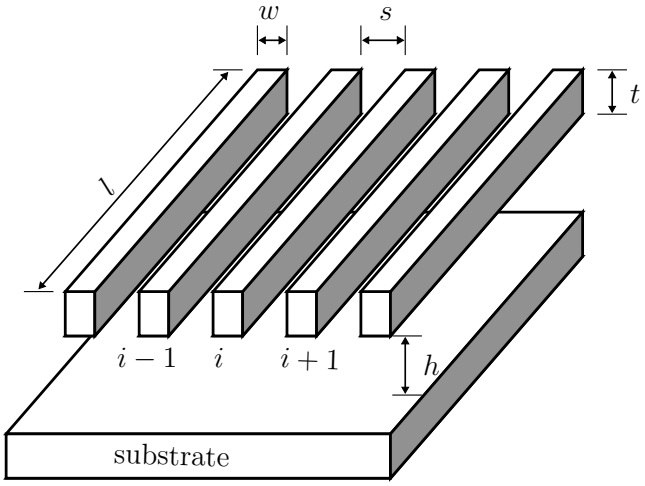


Fig. 2.2: A simplified structure of a parallel on-chip bus with length l, width w, spacing s, height $\mathtt{t}$ and distance between bus wires and substrate h [12].

Given a wire for a technology node, a designer may change the wire width and spacing and choose the metal layer to satisfy the design requirements. To specify the relation between wires geometry and performance metrics, we calculate the resistance of a conducting material, R as a function of wire geometry:

$$R = R_{\square}\frac{l}{w}, \tag{2.1}$$

where $R_{\square} = \rho/\mathtt{t}$, and ρ and $\mathtt{t}$ are respectively, the resistivity and the wire thickness. Similarly, a simple and physically intuitive estimate of capacitance between adjacent wires can be given as [27]:

$$C_{adj} = k\frac{\mathtt{t}}{s}, \tag{2.2}$$

where k is the dielectric constant and s is spacing. Conventionally, the minimum pitch[1] is preferred for best density [27]. Depends on the design requirements, however, designers can reduce the coupling capacitance and thereby reduce the delay and energy consumption by increasing the spacing between adjacent wires or choosing upper layers which are thicker. In the same way, widening the wire can reduce the resistance and delay, however, it can have a negative impact in overall capacitance.

Shielding

Shielding is an established widely used technique that reduces parallel buses' energy and delay by controlling crosstalk. Critical signals are often shielded on both sides such that there is no active aggressor in the neighboring. In this case, the worst-case crosstalk resulting from the opposite transition of signals does not occur, and the effective capacitance of the shielded wire does not exceed $C_g(1 + 2\kappa)$. Crosstalk reduction using shielding can be incorporated into global routing under a broad supply network paradigm [35]. The method utilizes power/ground wires as shields between the signal wires to reduce capacitive coupling while considering the constraints imposed by limited routing and buffering resources. Overall, the performance gains offered by shielding often do not justify the overhead that it imposes. Therefore, alternatively, group shielding of the fraction of the total bit-width is usually suggested.

Repeater Insertion

The delay of an RC interconnect is proportional to the wire length square, i.e., l^2. An effective strategy for reducing the delay of a long RC interconnect is to strategically insert buffers along a line [34]. Splitting the interconnect into N segments with repeaters, the interconnect's total delay is proportional to l^2/N. Furthermore, dividing the interconnect into segments results in the coupling between interconnects, which is also reduced due to the shorter length of coupling between neighboring lines [34].

However, gains associated with repeater inserting do not come without costs and limitations in terms of area, delay, and power. Each repeater adds gate delay to the overall interconnection delay. It is critical to determine the

[1] By definition, pitch is the sum of wire width, w, and wire spacing, s.

sufficient number of repeaters for the wire length l to prevent the dominance of repeaters gate delay over the line delay. If segments are too large, the delay will be dominated by the long wires. If segments are too small, the delay will be dominated by a large number of inverters [27]. Therefore, a trade-off among different design criteria is required for an efficient repeater insertion methodology. The detailed analysis to determine the number and size of repeaters to achieve the minimum delay is presented in [36].

Low-Swing Signaling

Low-swing signaling is achieved by lowering drivers' supply voltage, V_{dd}. Since the energy consumption of an interconnects is proportional to the square of the supply voltage, V_{dd}^2, (see the Equation 2.20 for more details), lowering the supply voltage leads to a quadratic reduction in the energy.

The practice of reducing voltage supply to reduce transmission power, low-swing signaling, as well as the inevitable voltage scaling due to technology node scaling, aggravate the susceptibility to noise sources because of decreased noise margins [33]. The circuit reliability can be improved by using complex circuitry, which in turn impose adverse effects on power consumption and area. For example, differential signaling is attractive low-swing signaling technique. Due to its high common-mode noise rejection, it allows for a further reduction in the signal swing. However, differential signaling has a high overhead due to additional interconnect and its routing [37]. Consequently, designers must trade-off power consumption with reliability.

In the new approximate communication paradigm, however, low-swing signaling can provide low-power communication without the necessity of complex circuitry to guarantee error-free data transmission.

2.3 Delay and Energy Modeling

Interconnect modeling is critical in both the circuit design and verification processes. An efficient and accurate interconnect model can significantly enhance the design and analysis of Integrated Circuit (IC). In this section, we study the energy consumption and propagation delay associated with transmission of patterns. The coupling capacitance between the neighboring bus lines and self-capacitance as well as the distributed nature of the wires

are taken into account. We derive the conventional energy and delay models. Besides, we analyze the conventional models and highlight problems associated with these models as a result of new challenges raised by emerging technology nodes. We finally remark on the accuracy of models using circuit simulation.

2.3.1 Delay Models

In this section, we use the equivalent circuit of a wire segment and a simplified driver to derive delay expressions of a bus.

Let us consider the equivalent circuit model of an interconnect structure and a driver in figure 2.3. A buffer drives the simplified lumped RC equivalent of wire i, and it is capacitively coupled to adjacent wires $i-1$ and $i+1$. The actual non-linear driver is modeled as a voltage source with series resistance, R_d. Drivers' sizes determine the values of drivers' resistance, and it is assumed that identical drivers are used to driving the wires. Furthermore, it is assumed that the pull-up and pull-down resistances are identical. The input signal, b_i, controls the switch, and thereby the wire connects either to the ground or to the supply voltage.

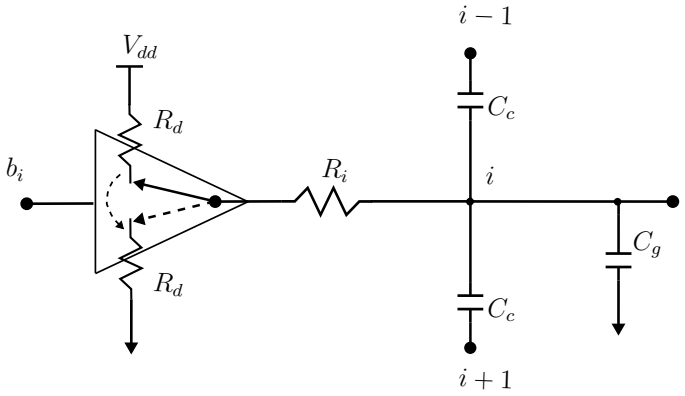


Fig. 2.3: The equivalent circuit model of an interconnect structure and a driver for i^{th} line of the interconnect. The interconnect is coupled to adjacent wires $i-1$ and $i+1$.

To derive the propagation delay, we apply the Kirchhoff's current law on the interconnect node i as follows:

$$\frac{V_{dd}-v_i(t)}{R} = C_g\frac{\partial v_i(t)}{\partial t} + C_c\Big(\frac{\partial(v_i(t)-v_{i-1}(t))}{\partial t} + \frac{\partial(v_i(t)-v_{i+1}(t))}{\partial t}\Big), \quad (2.3)$$

where R is the sum of the equivalent resistance of driver, R_d, and the intercon-

nect resistance, R_i, and $v_i(t)$, $v_{i-1}(t)$ and $v_{i+1}(t)$ are, respectively, the voltage in node i and in neighboring nodes $i-1$ and $i+1$. In this equation, we consider rising transition, i.e. $\Delta b_i = 1$, for derivation of the delay formula, where $\Delta b_i = b_i^+ - b_i^-$ determines the self switching of the metal wire i; b_i^+ and b_i^- are respectively, the logical binary values on the metal-wire after and before the transition. Please note that, the propagation delay is only relevant when the logical value of b_i is toggling compared to the previous clock cycle, i.e. $\Delta b_i^2 = 1$. The value of Δb_i is equal to 1 for a logical 0 to logical 1 transition, -1 for a logical 1 to logical 0 transition, and zero for no transition. In sake of simplicity, we often use arrow notations through out this thesis, a 0 to 1 transition notation (Δb_i=1) is noted as $\uparrow$, a 1 to 0 transition ($\Delta b_i = -1$) as $\downarrow$ and no transition (Δb_i=0) as $\bullet$.

To solve the Eq. (2.3), $v_{i-1}(t)$ and $v_{i+1}(t)$ should be known. The voltage change in neighboring coupled wires can be approximated as $\Delta b_i \Delta b_j \frac{dv_i(t)}{dt}$ where $\Delta b_i \Delta b_j$ is 1 if b_j switches in the same direction as b_i; 0 if b_j is stable; and -1 if b_j toggles in the opposite direction as b_i [12], so:

$$V_{dd} - v_i(t) = \tag{2.4}$$
$$RC_g \frac{\partial v_i(t)}{\partial t} + RC_c\big(\frac{\partial v_i(t)}{\partial t} - \Delta b_i \Delta b_{i-1} \frac{\partial v_i(t)}{dt} + \frac{\partial v_i(t)}{\partial t} - \Delta b_i \Delta b_{i+1} \frac{\partial v_i(t)}{\partial t}\big).$$

The resulting equation can be simplified using the bus factor, κ, as follows:

$$Vdd - v_i(t) = RC_g(1 + 2\kappa - \kappa \Delta b_i \Delta b_{i-1} - \kappa \Delta b_i \Delta b_{i+1}) \frac{\partial v_i(t)}{\partial t}. \tag{2.5}$$

To further simplify the high-level standard delay model expression, we define effective capacitance of the i^{th} wire of the bus:

$$C_{eff,i} = (\Delta b_i^2 + \kappa \delta_{i,i-1} + \kappa \delta_{i,i+1}) C_g, \tag{2.6}$$

where $\delta_{i,j}$ determines the coupling switching between the two direct adjacent metal wires i and j:

$$\delta_{i,j} = \Delta b_i^2 - \Delta b_i \Delta b_j. \tag{2.7}$$

The value of $\delta_{i,j}$ is equal to 2, if simultaneous reverse signal transitions occur on the adjacent wires ($\uparrow\downarrow$ or $\downarrow\uparrow$); if only interconnect i switches ($\uparrow \bullet$ or $\downarrow \bullet$), $\delta_{i,j}$ is equal to 1, for perfectly aligned transitions ($\uparrow\uparrow$ or $\downarrow\downarrow$) and no transition on

wire i, it is 0. Consequently, the effective capacitance of a metal-wire is in the range from $0C_g$ to $(1 + 4\kappa)C_g$. Rewriting the Eq. (2.5) using $C_{\mathit{eff},i}$ results in:

$$Vdd - v_i(t) = RC_{\mathit{eff},i}\frac{\partial v_i(t)}{\partial t}. \tag{2.8}$$

We use the Laplace transform, represented by $\mathcal{L}\{\}$ to solve this differential equation. Transforming the equation into the frequency domain results in:

$$\frac{Vdd}{s} - V_i(s) = RC_{\mathit{eff},i}\{sV_i(s) - v_i(0)\}, \tag{2.9}$$

where $v_i(0)$ is the potential of the node i at $t = 0$. We consider the initial condition $v_i(0) = 0$. Solving the equation for voltage in node i, $V_i(s)$ is given by:

$$V_i(s) = \frac{Vdd}{s\big(1 + sRC_{\mathit{eff},i}\big)}. \tag{2.10}$$

The inverse Laplace transform of this equation results in:

$$v_i(t) = V_{dd}(1 - e^{\frac{-t}{RC_{\mathit{eff},i}}}). \tag{2.11}$$

Finally, the rise time to reach $V_i(t) = \alpha V_{dd}$ is:

$$\tau_i(\alpha) = -RC_{\mathit{eff},i}.\ln(1 - \alpha). \tag{2.12}$$

This equation shows that the delay of the i^{th} wire of the interconnect has a linear relationship with $C_{\mathit{eff},i}$.

Normally, propagation delay is defined as the time required for the output to reach 50% of its final output level when the input changes to 50% of its final input level. Considering this definition, the rising delay in a capacitively coupled interconnect is:

$$\tau_i(\alpha) = 0.69RC_{\mathit{eff},i}. \tag{2.13}$$

Denoting the delay of an ideal cross-talk free wire as, τ_0, the Eq. (2.13) can be rewritten as follows:

$$\tau_{i_{seg}} = \tau_0[T_{t_i} + \kappa T_{e_i}], \tag{2.14}$$

where T_{t_i} and T_{e_i} are respectively, the self-switching and the coupling switching factors given by:

$$T_{t_i} = \Delta b_i^2, \tag{2.15}$$

and

$$T_{e_i} = 2\Delta b_i^2 - \Delta b_{i+1}\Delta b_i - \Delta b_{i-1}\Delta b_i. \tag{2.16}$$

We refer to this model as the standard delay model (STD).

2.3.2 Energy Models

A well-established model for dynamic energy consumption of the interconnect architectures as a function of the input signal switching and the capacitance values is presented in the literature [12, 34]. To derive the dynamic energy model[2], let us consider Figure 2.4. In this figure, we assume three adjacent wires that are capacitively coupled, and the bus is terminated so that the boundary wires are not capacitively coupled to the next wire in the interconnect. The dynamic energy extracted from a driver in the i^{th} line of the bus is given by:

$$E_{e,i} = \int_{t^-}^{t^+} V_{dd}\, I_i(t)dt, \tag{2.17}$$

where V_{dd} is the supply voltage, I_i is the current flow through the driver, t^- denotes the time before the transition, and t^+ denotes the time after transition when the steady state is reached. Ideally, no current flows through V_{dd} in output-high state, the energy extracted from the supply voltage is 0 when the falling transition occurs, i.e. $b_i^+ = 0$.

For the i^{th} wire of a bus with coupling capacitance, C_c, and self capacitance, C_g, and neglecting the leakage, the energy extracted from driver, $E_{e,i}$, can be calculated as follows:

$$E_{e,i} = b_i^+ \int_{t^-}^{t^+} V_{dd}\Big(C_g \frac{\partial v_i(t)}{\partial t} \; + \; C_c\big(\frac{\partial v_{i,i-1}(t)}{\partial t} + \frac{\partial v_{i,i+1}(t)}{\partial t}\big)\Big)dt. \tag{2.18}$$

The voltage drops on the coupling capacitances are defined as $v_{i,i+1} = v_i - v_{i+1}$

[2]Note that this model is neglecting the short-circuit current which is typically a fraction of the dynamic power consumption.

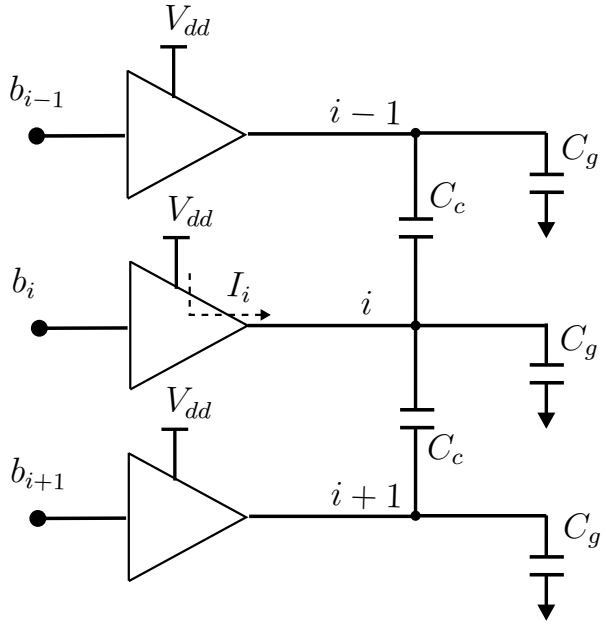


Fig. 2.4: Circuit model of coupled interconnects.

and $v_{i,i-1} = v_i - v_{i-1}$ and $v_i(t)$ is the voltage between the node i and ground. The value of $v_i(t)$ is equal to b_i^+, which is the logical binary values on the metal-wire after the transition is completed, times V_{dd}. So, the energy extracted from supply voltage in i^{th} wire of the bus is:

$$E_{e,i} = C_g V_{dd} b_i^+ \int_{t^-}^{t^+} \partial v_i(t) + C_c V_{dd} b_i^+ \int_{t^-}^{t^+} \partial v_{i,i-1}(t) + C_c V_{dd} b_i^+ \int_{t^-}^{t^+} \partial v_{i,i+1}(t) \tag{2.19}$$

$$= C_g V_{dd} b_i^+ \Delta v_i + C_c V_{dd} b_i^+ \Delta v_{i,i-1} + C_c V_{dd} b_i^+ \Delta v_{i,i+1}$$
$$= C_g V_{dd} b_i^+ \big(V_{dd} \Delta b_i + V_{dd} \kappa (\Delta b_i - \Delta b_{i-1}) + V_{dd} \kappa (\Delta b_i - \Delta b_{i+1}) \big).$$

Equation 2.19 can be rewritten as follows:

$$E_{e,i} = C_g V_{dd}^2 \, b_i^+ \Big(\Delta b_i + \kappa (2\Delta b_i - \Delta b_{i+1} - \Delta b_{i-1}) \Big). \tag{2.20}$$

The energy extracted from drivers in other lines can be similarly calculated. Then, the total extracted energy is the sum of each driver of a line.

The dissipated energy is the difference between the differential stored energy and the extracted energy:

$$E_{diss} = E_e - E_{stored} \tag{2.21}$$

This energy quantity is often reported by different Electronic Design Automation (EDA) tools. Besides, the formulas for energy dissipation (consumption) are less complex. The difference between the energies stored in parasitic capacitances after and before the transition is:

$$E_{stored} = \int_{t^-}^{t^+} C_g \frac{\partial v_{i-1}(t)}{\partial t}.v_{i-1}(t)\partial t + \int_{t^-}^{t^+} C_g \frac{\partial v_{i+1}(t)}{\partial t}.v_{i+1}(t)\partial t \tag{2.22}$$
$$+ \int_{t^-}^{t^+} C_g \frac{\partial v_i(t)}{\partial t}.v_i(t)\partial t + \int_{t^-}^{t^+} C_c \frac{\partial v_{i,i-1}(t)}{\partial t}.v_{i,i-1}(t)\partial t + \int_{t^-}^{t^+} C_c \frac{\partial v_{i,i+1}(t)}{\partial t}.v_{i,i+1}(t)\partial t.$$

Solving the above integrals, we have:

$$\begin{aligned} E_{stored} =& C_g V_{dd}^2 \frac{b_{i-1}^2}{2}\Big|_{t^-}^{t^+} + C_g V_{dd}^2 \frac{b_{i+1}^2}{2}\Big|_{t^-}^{t^+} + C_g V_{dd}^2 \frac{b_i^2}{2}\Big|_{t^-}^{t^+} \\ &+ C_c V_{dd}^2 \frac{b_{i-1}^2}{2}\Big|_{t^-}^{t^+} + C_c V_{dd}^2 \frac{b_{i+1}^2}{2}\Big|_{t^-}^{t^+} \end{aligned} \tag{2.23}$$

Factoring the difference of squares, the stored energy in parasitic capacitances results in:

$$\begin{aligned} E_{stored} = \frac{1}{2} C_g V_{dd}^2 \Big\{ & \big(\Delta b_i (b_i^+ + b_i^-) + \Delta b_{i-1}(b_{i-1}^+ + b_{i-1}^-) + \Delta b_{i+1}(b_{i+1}^+ + b_{i+1}^-)\big) \\ & + \kappa\big((\Delta b_i - \Delta b_{i-1})(b_i^+ - b_{i-1}^+ + b_i^- - b_{i-1}^-) + (\Delta b_i - \Delta b_{i+1})(b_i^+ - b_{i+1}^+ + b_i^- - b_{i+1}^-)\big)\Big\} \end{aligned} \tag{2.24}$$

By substituting, Equations 2.24 and 2.19 in Eq. (2.21), a formula to calculate the energy dissipation is obtained as follows:

$$E_{diss} = \frac{V_{dd}^2}{2}\Big(C_g(\Delta b_i^2 + \Delta b_{i-1}^2 + \Delta b_{i+1}^2) + C_c(\Delta b_i - \Delta b_{i-1})^2 + C_c(\Delta b_i - \Delta b_{i+1})^2\Big) \tag{2.25}$$

Usually, not only the energy consumption for the transmission of a single pattern pair (single transition) over a bus segment is the objective of interest, but rather the overall mean power consumption of the full bus (consisting N-segments), while transmitting one specific data stream. This quantity is expressed as [38]:

$$E_e = V_{dd}^2 N C_g (T_{t,e} + \kappa T_{e,e}), \tag{2.26}$$

where $T_{t,e}$ is the overall self switching probability of the B-bits of the data stream and $T_{e,e}$ is the overall coupling switching probability of the data stream

for the extracted energy. Using the expectation operator, $\mathbb{E}\{\}$, these quantities are calculated by:

$$T_{t,e} = \sum_{i=1}^{B} \mathbb{E}\{b_i^+ \Delta b_i\} \tag{2.27}$$

and

$$T_{e,e} = \sum_{i=1}^{B-1} \mathbb{E}\{(\Delta b_i - \Delta b_{i+1})(b_i^+ - b_{i+1}^+)\}. \tag{2.28}$$

Similarly, the energy consumption can be derived as follows:

$$E_{diss} = \frac{V_{dd}^2}{2} N C_g (T_{t,d} + \kappa T_{e,d}), \tag{2.29}$$

where $T_{t,d}$ is the overall self switching probability of the B-bits of the data stream and $T_{e,d}$ is the overall coupling switching probability of the data stream for dissipated energy. These quantities are calculated by:

$$T_{t,d} = \sum_{i=1}^{B} \mathbb{E}\{\Delta b_i^2\} \tag{2.30}$$

and

$$T_{e,d} = \sum_{i=1}^{B-1} \mathbb{E}\{(\Delta b_i - \Delta b_{i+1})^2\}. \tag{2.31}$$

As we have already mentioned, it is assumed that the bus is terminated so that the boundary wires do not have a neighbor for the derivation of the formulas. If the boundary wires are capacitively coupled with Vdd and Gnd then the starting index and the ending index of the $\sum$ in these equations change to 0 and B respectively. We refer to this model as the standard energy model (STD) in the rest of this report.

2.3.3 Crosstalk-Based Bus Transition Classification

There are mainly two transition classifications in the literature used to develop different high-level models and coding schemes. In the following, we introduce each classification:

3-wire effective capacitance-based classification According to the standard delay model derived in this subsection, the delay is a linear function of effective

capacitance, C_{eff}. Simultaneously, the value of effective capacitance is affected by signals in the wire, i, and its adjacent wires $i+1$ and $i-1$. Thus, the delay is pattern dependent and is a function of the input signals.

For a 3-wire bus, the transition patterns can be classified based on the value of effective capacitance in each wire. Under the assumption of $\kappa \gg 1$, we can ignore the C_g and therefore, the possible values of C_{eff} are $0.C_c$, $1.C_c$, $2.C_c$, $3.C_c$ and $4.C_c$ [12]. The standard paradigm of pattern classification groups patterns into 5 classes: $0C$, $1C$, $2C$, $3C$ and $4C$. The standard pattern classification and sample transition associated with each class is shown in Table 2.1. Given that the signal delay is linear in C_{eff}, the standard classification

Table 2.1: The standard transition pattern classification based on crosstalk [12]

Crosstalk class	$C_{eff,i}$	Sample transition pattern	
		$b_{i-1}(t-1)b_i(t-1)b_{i+1}(t-1)$	$b_{i-1}(t)b_i(t)b_{i+1}(t)$
0C	C_g	000	111
1C	$C_g(1+\kappa)$	000	011
2C	$C_g(1+2\kappa)$	000	010
3C	$C_g(1+3\kappa)$	001	010
4C	$C_g(1+4\kappa)$	101	010

paradigm is the classification system for the speed of individual bus wires [12]. This transition classification has been used mostly for the derivation of coupling avoidance coding techniques.

2-wire coupling switching based classification The coupling switching is computed as it is already discussed (see Eq. (2.7)) based on the correlated switching between two physically adjacent wires. Accordingly, four types of coupling transitions considering two parallel wires can be enumerated [39]: A type I transition occurs when one of the signals switches while the other stays unchanged, e.g., $\uparrow \bullet$. In a type II transition, one bus switches from low to high while the other switches from high to low (reverse transitions), e.g., $\uparrow\downarrow$. In a type III transition, both signals switch simultaneously in the same direction, e.g., $\uparrow\uparrow$. In a type IV transition, both lines do not switch, i.e., $\bullet\bullet$.

Based on this transition classification, the coupling switching activity, T_e, can be rewritten as follows [40]:

$$T_e = k_1 T_1 + k_2 T_2 + k_3 T_3 + k_4 T_4. \tag{2.32}$$

where the T_i, for $i = 1, 2, 3, 4$ are respectively, the average number of transitions for Type I, II, III and IV and k_i are associate weights. By convention, we assume $k_1 = 1$ as a reference for other type weights. The coupling effect in type II is usually twice type I ($\delta_{i,j}$ for $\uparrow\downarrow$ is twice the $\uparrow \bullet$) and in type III and IV no coupling effect can be observed. Therefore k_3 and k_4 are zero.

$$T_e = T_1 + 2T_2. \tag{2.33}$$

This transition classification has been used for derivation of different low-power coding techniques [39, 40, 41].

2.3.4 Standard Models' Problems

The conventional energy and delay models, summarized in the previous subsection, intrinsically assume that the signal switching on each wire of a segment occurs simultaneously (perfect temporal alignment). However, this assumption is not always true [42, 23, 43]. Different sources of misalignment, including intrinsic and extrinsic, should be taken into account while modeling energy and delay.

Intrinsic misalignment of propagated signals is one of the inherent sources of the misalignment. In this project, we mainly focus on intrinsic misalignment.

Let us consider an exemplary 3-bit bus where the size of a ground capacitance of a bus segment is denoted as C and the size of the coupling capacitance is $4C$ ($\kappa = 4$), which is a typical scenario in modern VLSI buses [44]. Furthermore, we assume a power supply voltage of 1 V and a bus, which is made up of two segments. We transmit an exemplary transition of $\bullet \uparrow\downarrow$ through the bus and estimate the delay and energy using the standard models when the signals are perfectly aligned or completely misaligned.

To calculate the energy, we use Eq. (2.29) for an exemplary transmission. The standard energy model estimates the energy consumption of $22C$, which is $11C$ per segment.

For the first segment, we assume that transitions $\uparrow$ and $\downarrow$ on second and third lines are perfectly aligned (they toggle at the same time), and therefore, they consume the energy consumption of $11C$ as the standard energy model estimates it. Substituting the transitions ($\Delta b_0 = 0$, $\Delta b_1 = 1$ and $\Delta b_2 = -1$) into Eq. (2.7) results in a coupling switching of 1 between the first and the

second line and a coupling switching of 2 between the second and the third line. According to Eq. (2.6), the effective capacitance values are 0C, 13C, and 9C for the first, second, and third lines, respectively. The variation in the effective capacitance values leads to the variance among signals' delay. The propagation of the signal on the second line from the input of the first segment to the input of the second segment takes $\approx 13RC$, while it takes $\approx 9RC$ for a signal in the third line to propagate.

For the second segment, simplified, let us assume that the different propagation times in the first segment lead to completely misaligned switchings on the second and third lines of the second segment (signal in the second line starts its transition after the signal in the third line reaches the steady-state). Thus, the transition of $\bullet\uparrow\downarrow$ can be represented by two subsequent sub-transitions: $\bullet\bullet\downarrow$ and $\bullet\uparrow\bullet$. Mathematically, coupling switching in Eq. (2.7) for complete misaligned signals is expressed by:.

$$\delta_{i,j} = \Delta b_i^2 \tag{2.34}$$

Substituting this equation in Eq. (2.6) for the exemplary transitions results in effective capacitance values of 0, 9, and 5, which impose delays of $\approx 9RC$ and $\approx 5RC$ in second and third lines. Neglecting variation and, in particular, the misalignment effect as a result of technology scaling not only can lead to a drastic misprediction of the delay (for our example, misprediction in the second segment is about 40%) but also signals' delay variation leads to a change in the overall energy consumption. According to Eq. (2.29), for completely misaligned signals in the second segment, the overall energy $7C$ which is about 36% less than the standard energy model estimation of $11C$.

To illustrate the effect of intrinsic misalignment on the interconnects' energy, Fig. 2.5 shows the energy extracted from supply voltage for three arbitrary transition patterns of a 5-bit width bus versus the segment numbers. The simulation is carried out on a bus with width and spacing of 0.15 μm and a length of 2 mm using *Cadence*, Spectre. Based on the results, the energy associated with transition pattern $\uparrow\bullet\uparrow\bullet\uparrow$ remains constant for different segments; however, it changes drastically in patterns $\downarrow\uparrow\downarrow\uparrow\downarrow$ and $\uparrow\uparrow\downarrow\downarrow\downarrow$. The extracted energy of the pattern $\downarrow\uparrow\downarrow\uparrow\downarrow$ is approximately halved from segment 1 to segment 8, while the energy of pattern $\uparrow\uparrow\downarrow\downarrow\downarrow$ is approximately doubled. According to this experiment, different transition patterns get unequally misaligned, and

that the misalignment induces drastic changes in energy consumption of the interconnects. Any pattern dependent coding or stochastic approach should be aware of this variation to produce sensible results.

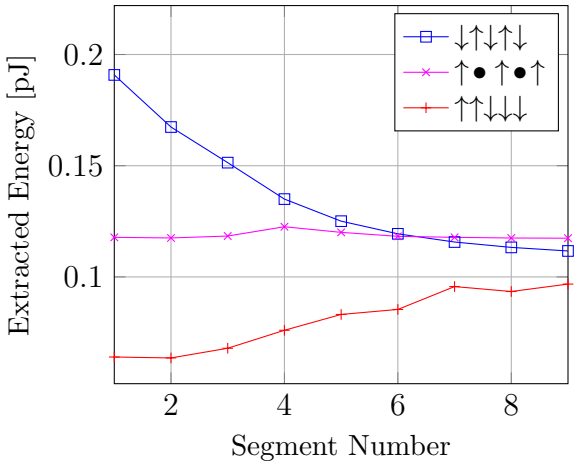


Fig. 2.5: Energy extracted from the supply voltage vs. segment number in three illustrative transitions.

Similarly, the effect of intrinsic misalignment on the interconnects' delay can be observed in Fig. 2.6. This figure shows the maximum delay for three arbitrary transition patterns of a 5-bit width bus versus the segment numbers. As can be seen, different segments impose different delays to the signals. For ↑●↑●↑ transition, the quiet lines practically shield the other signal lines, and therefore the delay variation in different segments cannot be observed. The standard delay model is only reliable for estimating the delay in the first segment of the interconnect where the signals are completely aligned and prone to fundamental error for segmented long interconnects.

In the next chapter, we propose alternative modeling strategies considering the misalignment effect.

2.4 Optimization Techniques

On-chip interconnects do not receive as much benefit from technology scaling as computation units do. There are mainly two unfavorable impacts of technology scaling. First, a smaller feature size means smaller spacing between wires which denotes a higher crosstalk effect between adjacent wires. Second, signals should still traverse a relatively large distance through the interconnects to reach the

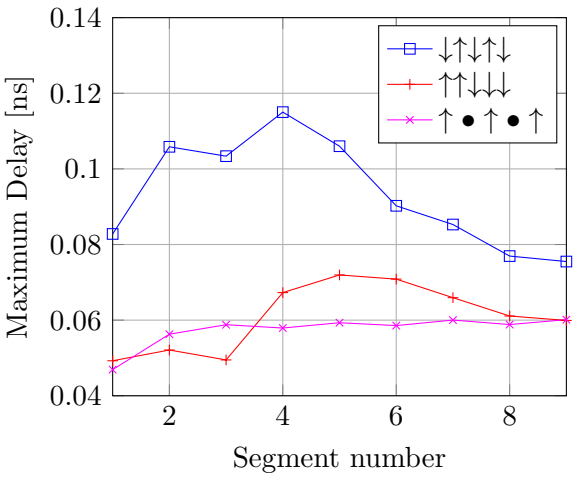


Fig. 2.6: Maximum delay vs. segment number in three illustrative transitions.

destination. This introduces various signal noise and deformation and therefore leads to an error [34]. The low swing signaling reduces the Signal-to-Noise Ratio (SNR) and therefore makes interconnects more sensitive to different sources of noise including power supply noise, crosstalk, etc. The on-chip communication is reaching the fundamental limits of the resources required for a fully reliable operation [45]. To deal with all these challenges, fundamental improvement of on-chip communication is essential.

Bus encoding is a widely used technique to reduce power consumption and crosstalk delay. There are many examples of significant improvements in the communication architecture through coding [12]. These techniques mainly minimize the propagation delay and power consumption of buses as the most critical design objectives in System-on-Chips (SoC), while enforcing determinism in the presence of increasing variability [46]. Nevertheless, a reliable (exact) data transmission is too costly in the appearance of new technologies and more complex systems. Approximate computing is a relatively new design paradigm that leverages applications' resilience to error to improve computing and communication platforms' energy and performance.

In this context, the aim of this section is double: first, we concentrate on the `exact` coding techniques and review the most well-known codings in the literature; and second, we discuss the emerging concept of approximate communication, which accounts for the inevitable variability of communication systems. We also review the different existing approaches of stochastic and approximate communication in the literature.

2.4.1 Review of Exact Coding Techniques

The fundamental problem that the Shannon theory [47] introduces is how to transmit the data reliably over an unreliable channel. Traditionally, it is expected that the received data is equal to the transmitted data. To achieve this goal, designers use coding to turn a noisy channel into a reliable system. The process of transforming transmitted data at the sender end is referred to as encoding. When the transmitted data reaches the receiver, a decoding step is performed to retrieve the actual data (see Figure 2.7). It is essential that the savings obtained using coding are not offset by the power dissipated by the encoding/decoding logic that needs to be added [33].

Fig. 2.7: Encoder and decoder structure for on-chip interconnect optimization.

In general, communication coding can be categorized into two broad categories as follows:

- **Source Coding**: This coding scheme changes the words being transmitted on a channel (interconnect) to adapt to properties of the inputs, for example, by compression of the redundant data. Source coder compresses the input data such that the number of bits required in the transmission is minimized [34].

- **Channel Coding**: This coding scheme changes words being transmitted to adapt to properties of the channel. It protects data against error in a noisy channel, and thereby, redundancy is added for error correction and/or for error detection. In practice, channel coding is employed to reduce the power and improve the performance in the communication channel to achieve the required level of reliability [34].

Chronologically, low-power coding was the first broad use of coding techniques in on-chip communication [8, 48, 49, 9]. Traditionally, these codes reduce the average transition activity as the bus's power dissipation depends on data transition activity. The Bus-Invert (BI) coding [8] is a simple but effective example of a low-power code used for randomly distributed data patterns. In this coding technique, if the current and previous data words' hamming

distance is greater than half of the word bits, the data word is inverted and sent through the channel. An extra signal indicates if the input data is inverted or not. The data is inverted on the receiver side if the `invert` signal is 1. We use this well-known coding technique in the next chapter to evaluate the proposed energy model. Gray code [49] is another low-power technique that can reduce the number of bit transitions and exploit the spatial and temporal locality of the input data for the case of correlated data patterns. This coding encodes the input data so that it ensures consecutive words differ by only one bit. The power consumption reduction of up to 40% is reported using this simple technique. T0 code [9] targets the minimization of the switching activity on address buses. This coding uses a redundant line that is set to value one when the addresses on the bus are sequential. In this case, all other lines on the bus are frozen (remain unchanged). When the addresses are not sequential, the redundant line is set to zero, and the actual address is transferred on the address bus. Techniques to minimize the switching activity based on the prior knowledge of the input stream (application-specific) have been proposed [50, 51]. For example, in [50] statistical information at the word level is used to generate an encoding table that minimizes transition activity.

While low-power codes are useful in self-switching power reduction, their applicability is relatively limited with the technology evolution where crosstalk capacitances and the related timing and noise issues become significant. Many research works have been proposed in literature [39, 52, 40, 41] as the expansion of low-power encoding techniques to account for the coupling switching activity. In the same context as classical BI, the Coupling-driven bus-invert (CBI) method [39] inverts the input data when the inverted signals' coupling effect is less than that of the original signals. Based on the type of transition (see the previous section), a coupling encoder generates codewords that enter the majority voter afterward. The majority voter outputs are high when at least more than half of the inputs are high. It has been shown that CBI outperforms classical BI in total switchings reduction, where total switchings account for both self transitions and coupled switchings. Full-Invert (FI) coding technique is another inversion based technique that inverts the input data word if the invert condition satisfied. According to this technique, if P is the dynamic power consumption of the original input and P' is the power consumption of the inverted input, it is convenient to invert the data before transition when

$P > P'$. We have to note that the FI coder and decoder architecture impose relatively high hardware overhead.

CAC are well-known techniques used to eliminate specific undesirable data patterns (classes) and reduce the delay in the exact communication. In this context, an encoding scheme is used to code a dataword and afterward select and transmit a codeword from a set of possible codewords which is called codebook. There are two types of CACs depending on their memory requirement: memoryless and memory-based codes. A code is memoryless when it uses a fixed codebook. If the codebook changes with time, then the code is memory-based. We mainly focus on memoryless codes.

The delay of a wire depends on transitions on the wire itself and the neighboring wires. As it is discussed before, the delay of a wire has a linear relationship with $C_{eff,i}$. The 3C-free memoryless CACs are the most popular CACs and have been heavily studied in the literature [12, 53, 54, 10]. They ensure that $4C$ and $3C$ crosstalk transitions are eliminated from the bus. Considering a (4,3) memoryless code, a coder encodes a 3-bit dataword as a 4-bit codeword such that the 2^3 4-bit codewords can be sent on the bus with a maximum delay less than or equal to $RC_g(1 + 2\kappa)$ [11]. Two types of the most efficient memory-less $3C$-free codes are Forbidden Pattern Free (FPF) and Forbidden Transition Free (FTF) CACs.

A forbidden pattern free coding avoids codewords with bit patterns 010 and 101. For example, codeword 1001100 is FPF while 1101001 is not a FPF codeword. By eliminating the forbidden patterns, the delay classes $1 + 4\kappa$ and $1 + 3\kappa$ are automatically removed. To generate a complete set of B-bit FPF codewords, usually, automatize codewords generation algorithms instead of exhaustive search approaches is used, especially when B is large. FPF requires redundancy, and encoding all the bits at once is not feasible for wide buses due to the prohibitive size and complexity of the code and decoder hardware [10]. In practice, partial coding is used, in which the bus is divided into multiple narrow buses, which are encoded separately using CACs. Naturally, codewords for narrow buses are combined in such a way as to avoid crosstalk occurrence at their boundaries. The valid codewords of FPF-CAC for a 3-bit datawords are shown in Table 2.2.

A forbidden transition free coding is another efficient 3C-free coding that prohibits two adjacent bits from the transition in the opposite direction, i.e.,

Table 2.2: The valid codewords of FPF and FTF crosstalk avoidance codes for 3-bit input datawords.

FPF-CAC	FTF-CAC
0000	0000
0001	0001
0011	0100
0110	0101
0111	0111
1000	1100
1001	1101
1100	1111
1110	-
1111	-

↑↓ and ↓↑. For example, a transition from codeword 11$\underline{01}$110 to codeword 01$\underline{10}$010 contains an invalid transition. The worst-case delay can be reduced by avoiding transition between codewords that causes adjacent wires to toggle in opposite directions. That means the 01 and 10 patterns cannot coexist in the same set of codewords [12]. Again, redundancy is required (extra line).

Once the valid codewords are generated, it is important to address the mapping scheme and coder and decoder (CODEC) design. The mapping between actual data words and valid codewords is an implementation step. A comprehensive discussion in this regard can be found in [12].

In this subsection, we introduced different coding approaches to improve interconnection in bus-based communication. By incorporating these techniques, the communication energy consumption and delay are reduced. However, in most cases, their application requires redundant wires and coder and decoder implementation and, therefore, imposes a negative impact on area, power, and performance. While effective, the technology scaling and the resulting variations propel the current optimization and coding techniques to consider approximation. It has been shown that preventing error and error control mechanisms in current, and future technology nodes introduce an overhead due to the additional logic and redundant lines, which in most cases is not affordable for on-chip interconnects [30, 31]. To address this problem, approximate

communication methods are suggested. In the next subsection, we concentrate on different approximate techniques for communication.

2.4.2 Approximate Techniques

Most communication systems are designed to prevent errors. However, there are many computing applications, such as image processing, computer vision, and machine learning, which can tolerate error in the results [25]. In this case, a conservative error correction for communication is not worthy for applications that already tolerate errors. Thus, fully reliable data transmission is unnecessary, and it is possible to trade reliability for reduced resource usage.

Hardware approximation mechanisms leverage the application's resilience to an error in order to improve the resource efficiency of various computing system components [25]. The research works in the field of approximate computing have mostly focused on the computational [55, 56] and memory subsystems [57, 58] and only recently on communication subsystems; which can be roughly categorized into the lossy compression/decompression [19, 16, 59], dual-voltage approximation [17, 30], and approximate encoding techniques [20, 60, 18].

There exist potential benefits of approximate computing for communication. In the following, different promising techniques of approximate communication are introduced.

Compression

In principle, data compression is the removal of redundant data. The compression technique can be categorized into lossless and lossy compression. The lossy compression technique is a commonly used approach for approximate data transmission and trades fidelity for a higher compression ratio. Several techniques have been recently proposed to address the communication bottleneck by adopting compression for approximate communication in the context of Network-on-Chips (NoCs), and bus-based interconnects. The main idea is to reduce the data traffic by compressing similar values or discarding the congested packets.

APPROX-NoC [19] proposes to reduce the transmission of approximately similar data between the cache blocks in the NoC to alleviate the data communication bottleneck by delivering approximated versions of precise data (lossy

compression). The APPROX-NoC consists of a *value approximate* module and an encoder/decoder pair for data compression in the network interface. If the cache block is approximable, the data is sent an approximation value compute logic. The error range compute-unit is also included in the approximation logic to guarantee the error within the preset error threshold. In the case of a non-approximable cache block, the approximation module is bypassed, and the data is directly sent to the encoder. The encoder compresses each word in the cache block to be transmitted and sends an encoded index with metadata instead of the whole pattern. They have shown up to 9% latency reduction and 60% throughput improvement compared with state-of-the-art NoC data compression mechanisms.

Approximate multiplane NoC (AMNoC) [21] provides two physically separate subnetworks: lossless-subnet and approx-subnet with no connection between two networks. The regular buffered subnetwork (lossless) guarantees the lossless delivery of non-approximable packets. On the other hand, the low-latency and power-efficient transfer of approximable packets is carried out through a lossy bufferless subnetwork designed based on an approximate switch (ASW) design. In case of two or more flits contend for the same output port for lossy-subnet, the only one can be transferred, and all other conflicting flits are discarded. Since the data in an approximable packet are generally fetched from successive memory blocks, the missing flits can be recovered using a simple linear interpolation. They have shown that AMNoC achieves 48.6% power savings and reduces the area overhead by 53.4%.

Approximate-Based Dynamic Traffic Regulation (ABDTR) dynamically regulates each injected packet's drop rate according to the network congestion state, and its contribution to congestion [16]. Packets that contribute more to network congestion lose more data. A thriller unit drops data within a packet with an interval and packetizes the rest. An appropriate drop-rate to satisfy the output quality is regulated by a controller. When the packets reach the destination node, an approximator uses linear interpolation to predict the missing values. They have reported the average network delay reduction of 30.9% for 10% quality loss.

In the context of bus-based interconnects, AxBA [59] employs the approximate counterpart of the deduplication compression technique. Deduplication is a data compression technique that eliminates duplicates in repeating data, and

it is widely used in secondary storage systems [61]. AxBA stores the sequence of approximately identical data values (relaxing the exact match) as a single value and its count. They take the first element of the data stream as the base element, and a count of how many times that element within an acceptable error margin appears consecutively. They report the average single-core execution time reduction of 19% for different machine learning benchmarks In the following, we explore a few research works available in literature in the area of approximate compression. These works are the most comparable with our proposed techniques in chapter .

In the following, we explain the ABDTR [16], Delta-base and deduplication compression [59] methods in more details. These techniques are the most relevant approaches to what we have proposed in this thesis. They have also shown a promising improvement in performance and energy consumption of communication links producing a tolerable error.

ABDTR

Approximate-based dynamic traffic regulation (ABDTR) [16] proposes a lossy compression approach that dynamically regulates the drop rate based on the network congestion state. An approximator approximates the received data. The buffer filling is used as a congestion measure. Therefore, each router monitors its input buffers, and whenever the number of packets in one of these buffers increased to a threshold value, the router becomes the congestion. The congestion information requires an independent high priority channel to transmit the control signal.

The compressor samples out the data in a given interval of λ. The drop rate of the compressor then calculated as follows:

$$D = \frac{1}{1+\lambda} \tag{2.35}$$

The smaller interval results in a bigger drop rate which cannot exceed 50%. The remaining data is packetized afterward. The linear interpolation is chosen to approximate the receiver's dropped values with low cost and reasonable accuracy. The average of the previous and next values is used as an estimation of the missing value. Please note that *ABDTR* relies on human engineering to decide the threshold value.

Base-Delta Approximation

In Ref.[59], a primarily lossless cash compression technique has been customized to be used as an approximate communication technique for bus-based communication. The values in a data block are represented as a base value and series of deltas, where deltas are the difference between the base and the remaining values in the block. Given an error threshold, delta can subject to error, and therefore the dynamic range of the scheme is extended. The number of bits to represent the delta values is precomputed based on the preferred compression rate and size of the data stream.

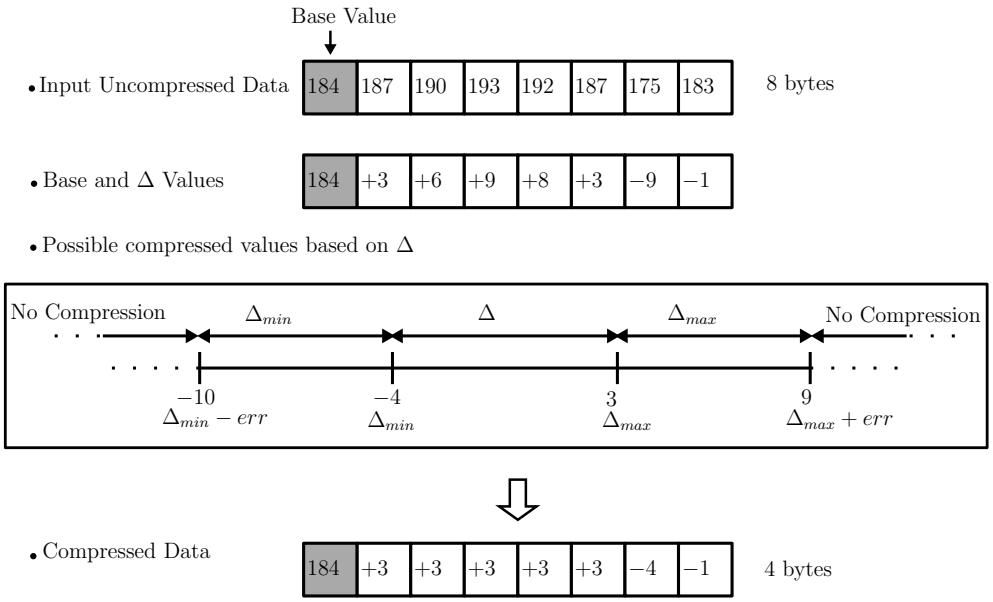


Fig. 2.8: An exemplary delta-base approximate compression scheme for 3 bits delta value and integer value error bound of 6.

Fig. 2.8 shows an example of Base-Delta approach. In this example, an 8-byte data is produced in the IP core, which should be compressed before the data is packetized and sent through the network. Let us assume a 3-bit delta value and the error threshold of 6, i.e., $err = 6$. In this case, the range of valid Δ is between $\Delta_{min} = -2^3$ and $\Delta_{max} = 2^3 - 1$. Each value in the data block can only be compressed if it lies between the range of $\Delta_{max} + err$ and $\Delta_{min} - err$. The data block can only be compressed if all individual bytes in the block can be compressed. To decompress the received values, a decompressor adds up the base and corresponding delta values. Please note that this technique

requires an additional control signal to denote whether the incoming values are compressed or not. In this example, the 8-byte input data block is compressed to a 4-byte compressed data.

Deduplication Approximation for Communication

Deduplication is another data compression technique that eliminates duplicate copies of data mainly in storage elements [62]. In Ref.[59], an approximate version of deduplication is proposed, i.e. *Axdeduplication*. This compression scheme represents a data block as a base value and a count of how many times that element appears consecutively. A value is considered as a repeated instance of base value if it lies within a predefined error bound.

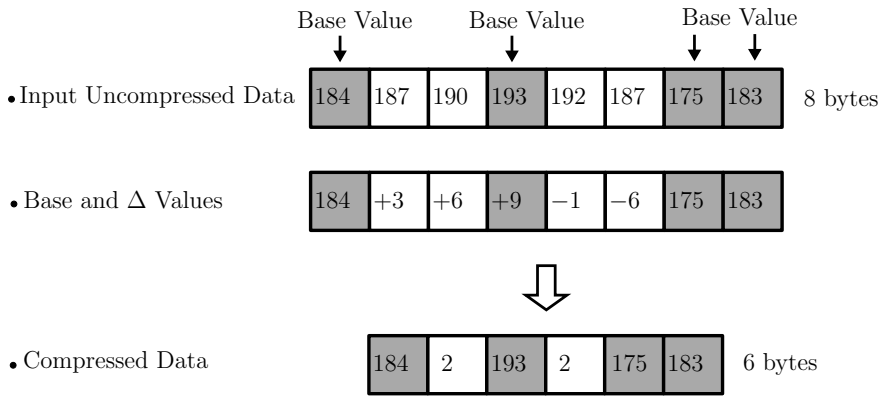


Fig. 2.9: An exemplary Axdeduplication compression scheme for integer value error bound of 6.

An example of Axdeduplication scheme is shown in Fig. 2.9. In this example, we consider the integer error bound of 6. The first value in the data block is considered as the base value. A count of subsequent values within the error bound compressed in a single byte. If the next value falls outside the error bound, it will be the next base value. According to Fig. 2.9, 8 bytes input values are compressed in 6 bytes. A signal determines which value is the count value and which one is the base value. It worth mentioning that the Axdeduplication technique may produce undesirable outputs due to the impact of timing-induced errors.

Coding for Approximate Communication

The coding can also be employed for approximate communication. We refer to coding as approximate coding if it reduces the number of bit errors or the error's magnitude. So far, most research works place focus on coding techniques for approximate serial communication.

Approximate differential encoding for serial communication is one of the recent works on approximate communication, which uses the differential encoding as a baseline scheme and extends it with bounded approximation [63]. The Differential encoding technique's key idea is to transmit the bitwise difference of the consecutive words of the strongly correlated data on the bus. The approximate differential encoding's core idea is to round a certain number of the Least Significant Bit (LSB) to achieve more considerable transition count reduction and, therefore, more immense energy saving. Considering a 3-LSB saturation (rounding of 3 LSBs to 1's or 0's), this technique implicitly accepts a maximum error tolerance of $\frac{7}{256}$ for the 8-bit word data.

The Value-Deviation-Bounded approximate Serial encoding (VDBS) [20] reduces the dynamic power dissipation of serial buses for a tolerable integer distance deviation. Several important applications of computing systems send numeric data. In this case, the effects of errors are best quantized in terms of the magnitude of errors, rather than a Hamming distance (number of bit errors). VDBS works as follows: for an integer value s, find a proximal value, t, such that the reduction in transition count is maximized and the integer value difference is minimized. VDBS encoding scheme is reported to reduce the signal transitions by 55% on average while maintaining Optimal Character Recognition (OCR) accuracy at over 90% for a text-recognition system. An integer-value encoding [64] is employed in a configurable NoC, which can be used in exact or approximate mode. The error-free transmission of the head flit is guaranteed using a CAC coding technique, while the error magnitude of the body flits is reduced using the selective inversion and swap of the signals in the communication links.

There is a category of techniques whose primary goal is to provide the energy and accuracy trade-off. The dynamic power of a circuit consisting Complimentary metal-oxide-semiconductor (CMOS) transistors, P_D can be estimated as:

$$P_D = ACV_{dd}^2F, \tag{2.36}$$

where A is the average switching activity, C is the switched capacitance, V_{dd} is the supply voltage, and F is clock frequency. Reducing the supply voltage will cause a quadratic reduction in dynamic power. Dual V_{dd} and voltage scaling are two well-known low-power techniques. Some approximate methods utilize these techniques to trade-off energy and accuracy.

Considering that not all the communications in an application have the same reliability requirements, AxNoC [17] proposes a router capable of providing per-flit dual-voltage power management for approximate and exact data transfer. Headers and critical flits are transferred correctly at high voltage. The remaining flits are approximated and transferred at a lower voltage. AxNoC shows a power consumption reduction of up to 43% while incurring a small area overhead that does not exceed 6.2%.

With a primary focus on communication links, [30] uses a reliability-aware adaptive voltage swing. It proposes to use a default channel and a low-power channel; the default channel uses the nominal voltage swing while the low-power channel uses a lower voltage swing. The work proposes a link architecture with and without bitline duplication. The method saves up to 43% of energy without impacting the performance of the system substantially. However, in the case of using repeaters, both configurations are expensive in terms of silicon area.

2.5 Conclusion

In this chapter, we have discussed the physical modeling of the interconnect and design choices that the designer should make based on the optimization goals. We also explained some physical layer techniques that are often used to improve the interconnects' performance metrics. Among different techniques, shielding, repeater insertion, and low-swing signaling as the most popular approaches are explored.

We discussed two primary motivations of this thesis: first, we highlighted the challenges presented by technology scaling in development and precise evaluation of the different coding schemes using high-level energy and delay models. We review the existing models in the literature and derived the conventional energy and delay models. The conventional approaches have been assessed for a segmented parallel on-chip interconnect. For example, results show that the conventional energy model mispredicts the energy values by

up to 40%. That is mainly due to the exclusion of the misalignment effect in conventional high-level models. The relative temporal alignment of signals should be taken into account for the development of a precise model.

Second, we surveyed different exact optimization techniques and stated that deterministic behavior is becoming increasingly expensive as performance and energy penalties must be paid to ensure data transmission accuracy. The limitations of exact communication propel approximate communication utilization. We also provided an overview of recent works in the area of approximate communication.

In the next chapter, the misalignment effect is elaborated in more detail. The high-level accurate energy and delay models considering the temporal alignment of signals are derived based on that.

CHAPTER 3

Accurate Interconnect Models

Contents

3.1 Introduction

High-level energy and delay models are crucial for the design of optimization techniques such as coding approaches. The development and efficiency evaluation of a coding technique require accurate energy and delay models.

In the previous chapter, we have studied the effect of temporal misalignment on the neighboring signals for energy and delay of interconnects. We have observed that the signals' intrinsic temporal misalignment results in a drastic variation in the energy consumption and propagation delay of interconnects. Transition patterns respond differently to the misalignment. In principle,

the energy and delay of worst-case transitions tend to decrease due to the misalignment, while best-case transitions tend to increase. We derived the conventional models for delay and energy estimation of interconnects and showed that the conventional models do not consider the effect of signals' alignment.

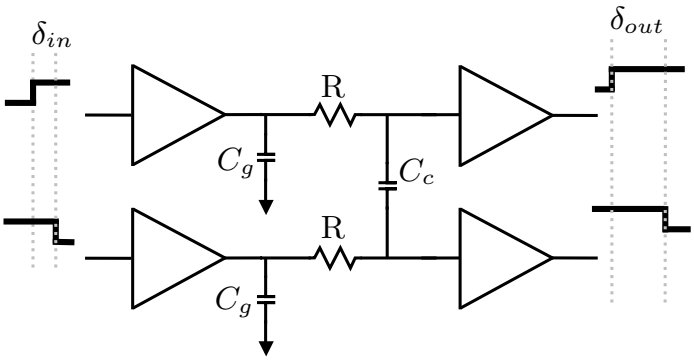
In this chapter, for the first time, we develop high-level analytical delay and energy models for narrow RC-coupled interconnect, Misalignment-Aware energy and delay models (MAA), taking into account the temporal alignment of signals. We focus on the intrinsic misalignment effect, which is conventionally overlooked. Nevertheless, externally triggered misalignment of signals can happen due to the noise with similar consequences. We derive correction factors that provide an accurate reflection of the misalignment effect. We use regression analysis as the most common form of statistical modeling, particularly linear regression. The linear regression is the simplest and thus the most common estimator. We evaluate the proposed models through comprehensive simulations and case studies. Finally, we summarize the chapter.

The contributions in this chapter are published in [43] and [65].

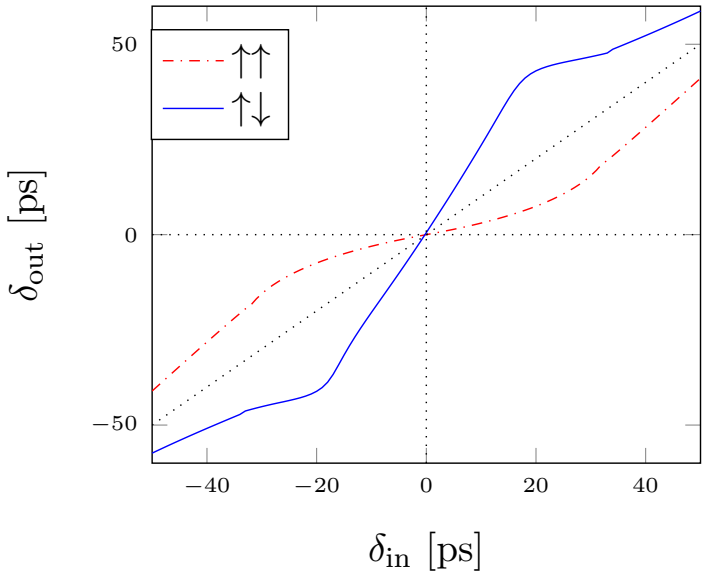
3.2 Alignment Behavior of Neighboring Wires

To develop the misalignment-aware models, we study the relative alignment behavior of two neighboring wires. Let us consider a simple 2-wire bus shown in Fig. 3.1(a). There are three classes based on the standard 2-wire coupling switching classification (see Section 2.3.3 for detailed explanation). In transition patterns with $\delta_{i,j} = \Delta b_i^2 - \Delta b_i \Delta b_j$ equals 1 ($\uparrow \bullet$, $\downarrow \bullet$, $\bullet \uparrow$ or $\bullet \downarrow$), relative changes in the signals alignment cannot lead to a change in the energy and delay in these transitions. The misalignment can only change the energy and delay where simultaneous reverse and forward transitions occur ($\uparrow\downarrow$, $\downarrow\uparrow$ or $\uparrow\uparrow$, $\downarrow\downarrow$). Thus, we only compare these two classes.

To do that, the transition patterns $\uparrow\uparrow$ and $\uparrow\downarrow$ as the representatives of these two classes are simulated. We impose a negative and positive skew to one of the inputs in respect to the other one and compare the relative input delay (δ_{in}) with the relative output delay (δ_{out}). This comparison is presented in Fig. 3.1(b). Simulation is carried out for the bus structure with width and spacing of 0.15 μm in length of 125 μm as an arbitrary bus structure using



(a)



(b)

Fig. 3.1: (a) Illustration of a simple 2-wire, 1-segment bus driven by signals with relative input delay of δ_{in} for the input transition of ↑↓, and (b) comparison of signals' relative input delay and the relative delay of output δ_{out} for ↑↓ and ↑↑ transitions.

Cadence, Spectre.

The results suggest that transitions ↑↓ and ↑↑ respond differently to the input signals misalignment. The misalignment speed in the case where signals in two neighboring wires toggle in different directions (solid blue line) is much higher than where signals toggle in the same direction (red dashed line). In other words, misalignment happens more significantly in the worst-case pattern

↑↓ and less significantly in the best-case pattern ↑↑. When we consider wider buses, the core transitions of ↑↓ and ↑↑ are also affected by their neighbors. Therefore, a logical conclusion is to consider neighboring transitions of the core transitions ↑↓ and ↑↑ to derive our energy and delay models.

3.3 Misalignment-Aware Energy Model

3.3.1 Related Works

Energy efficiency is a primary concern in almost all areas of modern computing. The on-chip communication accounts for a significant fraction of an integrated system energy consumption, and it is increasingly becoming a design bottleneck [33]. The effective development and performance evaluation of different techniques such as low-power coding [38, 66] and stochastic methods [67, 68] requires a precise energy estimation of the interconnects requires thoughtful consideration of signals' alignment.

Traditionally, high-level energy estimation models consider adjacent inter-wire capacitances while assuming signals are synchronized throughout the interconnect [69, 7]. The relative alignment of the signals in adjacent wires is ignored in high-level delay and energy modeling of the interconnects. Recently, an energy model for completely aligned and completely misaligned signals as an alternative for traditional energy models is proposed [70]. However, it does not provide a comprehensive energy estimation model based on the signals' relative alignment.

There are numerous research works studying coupling capacitances and net misalignment [4, 3, 38, 23]. In [3], authors study coupling to determine the switching time that produces the worst-case victim delay considering the misalignment effect. First, they determine the worst alignment of the aggressors' nets relative to each other and produce a composite noise pulse. Then, they determine the worst-case alignment of composite noise pulse with respect to victim transition time. Nevertheless, this approach does not consider the intrinsic misalignment of signals throughout the interconnects that determine the delay and the energy consumption in different segments. In fact, changes in relative aggressors alignment due to the intrinsic misalignment is not investigated in this work. In [23], the crosstalk aggressor alignment induced interconnect delay

is incorporated in statistical static timing analysis. In [38] current challenges regarding non-ideal buses are highlighted; it emphasizes the misalignment as a key challenge for developing robust coding approaches.

In summary, none of these works provides an abstract, high-level model including misalignment effect as a function of input transitions that can be used for data encoding schemes or stochastic approaches.

3.3.2 Definitions

To develop the misalignment-aware energy model, we employ different functions and classes. We define them in the following:

Existence function: The *existence function* specifies the relation between a transition and the existence of that transition; so that the output is 1 when the transition exists and it is 0 when the transition does not exist. We denote the 3 variations of this function, $\uparrow_i$, $\downarrow_i$, $\bullet_i$, as follows:

$$\begin{aligned} \uparrow_i &= \uparrow(\Delta b_i) = \frac{\Delta b_i(\Delta b_i + 1)}{2}; \\ \downarrow_i &= \downarrow(\Delta b_i) = \frac{\Delta b_i(\Delta b_i - 1)}{2}; \\ \bullet_i &= \bullet(\Delta b_i) = 1 - \Delta b_i^2. \end{aligned} \tag{3.1}$$

For instance, the output of the function $\uparrow_i$ is 1 when b_i is a rising transition $\uparrow$, i.e. Δb_i is equal to +1; and it is equal to 0 when b_i is a falling transition $\downarrow$, i.e. Δb_i is equal to -1 or 0.

Patterns class: Transition patterns in a bus have some symmetries, for example, the energy consumptions of the patterns $\uparrow\uparrow\downarrow\bullet$ and $\bullet\downarrow\uparrow\uparrow$ in a 4-bit bus are equal because of the mirror symmetry. We propose to group patterns with symmetries and to form the patterns classes. Each class is a set of different m-bit patterns and it is represented in the form of $[\Delta b_i \Delta b_{i+1} \ldots \Delta b_{i+m-1}]$. The patterns class $[\Delta b_i \Delta b_{i+1} \ldots \Delta b_{i+m-1}]$ consists of $\Delta b_i \Delta b_{i+1} \ldots \Delta b_{i+m-1}$, its negated, $-\Delta b_i\ -\Delta b_{i+1} \ldots -\Delta b_{i+m-1}$, and their mirrors, $\Delta b_{i+m-1} \Delta b_{i+m-2} \ldots \Delta b_i$ and $-\Delta b_{i+m-1}\ -\Delta b_{i+m-2} \ldots -\Delta b_i$. Note that, for some classes the mirrors and the negated patterns are equal. An example of such patterns classes is $[\uparrow\downarrow\uparrow\downarrow]$.

Class function: Using the *existence function* and the *patterns class*, we define *class function*. The class function relates patterns, i.e. input of the function, and logical OR of the multiplication (logical AND) of the logical values corresponding to existence of each pattern, i.e. output of the function. For example the class function $[\uparrow\uparrow\downarrow \bullet](\Delta b_{i-1}, \Delta b_i, \Delta b_{i+1}, \Delta b_{i+2})$ is equal to 1 if for $(i\text{-}1)^{th}$ to $(i\text{+}2)^{th}$ wire of the interconnect any of $\bullet\uparrow\downarrow\downarrow$, $\bullet\downarrow\uparrow\uparrow$, $\uparrow\uparrow\downarrow\bullet$ or $\downarrow\downarrow\uparrow\bullet$ patterns exists. For sake of simplicity, we can rewrite the above mentioned example of the class function as $[\uparrow\uparrow\downarrow\bullet]_i$.

We use these definitions to develop our energy model.

3.3.3 Model development

Based on our observations in Section 3.2 and in order to derive a correction factor, we decompose the bus into groups of four wires ($m = 4$) to form different *patterns classes*, $[\Delta b_i \Delta b_{i+1} \Delta b_{i+2} \Delta b_{i+3}]$. This is in contrast to former standard energy estimation methods which consider a single line bus and its immediate neighbors [71]. First, we consider core transitions (two middle transitions) and choose transitions $\uparrow\uparrow$ and $\uparrow\downarrow$ as representatives of 0 and 2 effective capacitances. Then, adding two neighbors in two sides of the core transitions, there are 18 different pattern classes (3 possible rising, falling and quiet transitions for two neighbors of core transitions: $3 \times 3 \times 2$). Among these classes, 6 of them are already included in other classes and can be ignored (for example the patterns class of $[\uparrow\uparrow\uparrow\bullet]$ is the mirror of the $[\bullet\uparrow\uparrow\uparrow]$ patterns class). The remaining classes can be grouped into two categories depending on their core transitions, namely: the worst-case classes and the best-case classes as follows:

$$\text{Worst-case classes: } \Big\{[\downarrow\uparrow\downarrow \bullet], [\uparrow\uparrow\downarrow \bullet], [\downarrow\uparrow\downarrow\downarrow], [\bullet \uparrow\downarrow \bullet], [\uparrow\uparrow\downarrow\downarrow], [\downarrow\uparrow\downarrow\uparrow]\Big\};$$

$$\text{Best-case classes: } \Big\{[\downarrow\uparrow\uparrow \bullet], [\uparrow\uparrow\uparrow \bullet], [\downarrow\uparrow\uparrow\uparrow], [\bullet \uparrow\uparrow \bullet], [\uparrow\uparrow\uparrow\uparrow], [\downarrow\uparrow\uparrow\downarrow]\Big\}.$$

The two middle transitions do not get equally misaligned for different patterns classes. Among the worst-case classes, $[\downarrow\uparrow\downarrow \bullet]$, $[\uparrow\uparrow\downarrow \bullet]$ and $[\downarrow\uparrow\downarrow\downarrow]$ classes get misaligned which is because of the different effective capacitances seen by core transitions. For example, the effective capacitances of the core transitions $\uparrow\downarrow$ for patterns class $[\downarrow\uparrow\downarrow \bullet]$ are 4 and 3, respectively. In contrast, due to the equal effective capacitances seen by core transitions, in the rest of worst-case classes, a

big misalignment is not expected. The situation is actually more complex. The misalignment effect in other neighboring transitions can propagate to the core transitions. This specially happens when the patterns from the patterns class of [↑↓] exist between the core transition and the source of the misalignment. The [↑↓] class lets the misalignment to propagate, while the transition class of [↑↑] almost prevents the misalignment to propagate. Therefore, we can conclude that the class of [↓↑↓↑] still can get misaligned, however with a smaller extent. Among the best-case classes, only the class [↓↑↑↑] can slightly get misaligned. Basically, the core transition of ↑↑ is reluctant to misalignment and only the biggest asymmetry in effective capacitances can lead to misalignment of the core transitions. As a result, among all classes the followings classes are necessary for developing an accurate energy model:

$$[\downarrow\uparrow\downarrow\bullet]\,,[\uparrow\uparrow\downarrow\bullet]\,,[\downarrow\uparrow\downarrow\downarrow]\,,[\downarrow\uparrow\downarrow\uparrow] \text{ and } [\downarrow\uparrow\uparrow\uparrow] \tag{3.2}$$

To identify the class of transitions that are most affected by the misalignment effect, it is required to assign weights to each class to account for the extent of their contributions to the correction factor. We assign positive weights when misalignment decreases the energy consumption, i.e., ↑↓ core transitions, and negative weights when misalignment increases the energy consumption, i.e., ↑↑ core transitions.

The core transitions of ↑↓ in $[\downarrow\uparrow\downarrow\bullet]$, $[\uparrow\uparrow\downarrow\bullet]$, $[\downarrow\uparrow\downarrow\downarrow]$ classes see different effective capacitances in their neighbors. Therefore, it is expected that they get strongly misaligned and the energy decreases due to the misalignment. We consider a positive weight for them. Based on the experiments, in class [↓↑↓↑], even though the effective capacitances seen in the neighboring transitions are the same, the slight misalignment in the core transitions can decrease the energy. So, a positive and smaller weight is expected for this class. In the class [↓↑↑↑], core transitions see different effective capacitances in their neighbors, they are not likely to get misaligned easily. A small misalignment and therefore an increase in the energy consumption is expected for this patterns class. Thus, yet smaller and negative weight is considered for this class. Finally, we assign different weights to the classes as follows:

$$\alpha_1[\downarrow\uparrow\downarrow\bullet]\,,\alpha_1[\uparrow\uparrow\downarrow\bullet]\,,\alpha_1[\downarrow\uparrow\downarrow\downarrow]\,,\alpha_2[\downarrow\uparrow\downarrow\uparrow] \text{ and } \alpha_3[\downarrow\uparrow\uparrow\uparrow], \tag{3.3}$$

where $\alpha 1 > \alpha 2 > \alpha 3$ and $\alpha 3 < 0$.

3.3.4 Parameter estimation

In this subsection, we use a numerical approach to estimate the effective capacitances to calculate the bus's energy consumption. Numerical analysis methods give an accurate solution for such estimation problems. A large number of procedures have been developed for parameter estimation and inference in linear regression. Ordinary least squares (OLS) is the simplest and thus the most common estimator. It is conceptually simple and computationally straightforward. The OLS method minimizes the sum of squared residuals, and leads to a closed-form expression for the estimated value of the regression coefficient vector β, denoted as $\hat{\beta}$:

$$\begin{aligned} \boldsymbol{y} =& \boldsymbol{X}\beta + \epsilon \\ \hat{\beta} =& (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}, \end{aligned} \tag{3.4}$$

where y is regressand, and ϵ is the error term. $\boldsymbol{X}$ is the matrix of regressors, and $\boldsymbol{X}^T$ is its transpose.

The first naive approach consists of estimating the effective C_g and C_c per segment. Using OLS estimator, we are able to obtain the estimated regression coefficient that corresponds to the changes in the energy consumption in each segment as follows:

$$\begin{aligned} X_{std} =& [T_t\ ,\ T_e\ ]^T \\ C_{std_n} =& [C_{g_n}\ ,\ C_{c_n}\ ] = OLS(E_n\ ,\ X_{std}), \end{aligned} \tag{3.5}$$

where X_{std} is the matrix of regressors for standard energy model which corresponds to the X in Eq. (3.4). C_{std_n} is the estimated value of regression coefficients for n^{th} segment of the interconnect and E_n is the regressand or the measured energy through the simulation in the n^{th} segment. The estimated energy using the regression coefficient C_{std_n} is given by:

$$\hat{E_n} = X_{std}\ \cdot\ C_{std_n} = T_t C_{g_n} + T_e C_{c_n}. \tag{3.6}$$

We refer to this model as linear regression of standard model (LR-STD) for the rest of this paper. The misalignment effect cannot be fully integrated into

LR-STD model.

The second approach for modeling the energy is to define a correction factor that is able to incorporate the effect of the intrinsic misalignment. We propose the correction factor T_{m_i} using the *class function* as follows:

$$\begin{aligned} T_{m_i} = \alpha_1\,[\downarrow\uparrow\downarrow \bullet]_i \;+\; \alpha_1\,[\uparrow\uparrow\downarrow \bullet]_i \;+\; \alpha_1\,[\downarrow\uparrow\downarrow\downarrow]_i + \\ \alpha_2\,[\downarrow\uparrow\downarrow\uparrow]_i \;+ \alpha_3\,[\downarrow\uparrow\uparrow\uparrow]_i. \end{aligned} \tag{3.7}$$

The weights α_1, α_2 and α_3 should be tuned to account for each pattern class extent of contribution in correction factor. We have found the exact value associated with each of them through the experiments ($\alpha_1 = 2$, $\alpha_2 = 1$ and $\alpha_3 = -0.4$). Finally, T_m, that is the sum of the class functions over all wires (B-bit) of the bus with their associated weight, is derived as follows:

$$T_m = \sum_{i=0}^{B-2} \boldsymbol{E}\{T_{m_i}\} \tag{3.8}$$

Now, to have a more accurate model, we improve matrix of regressors by adding the correction factor T_m to it as follows:

$$X_m = [T_t\ ,\ T_e\ ,\ T_m]^T. \tag{3.9}$$

Using this factor, we are able to calculate the regression coefficients using OLS function and estimate the energy in the n^{th} segment of the interconnect:

$$C_n = [C_{g_n}\ ,\ C_{c_n}\ ,\ C_{m_n}] = OLS(E_n\ ,\ X_m) \tag{3.10}$$

$$\hat{E_n} = X_m\,.\,C_n \tag{3.11}$$

In the rest of this report, we refer to this model as the misalignment-aware energy model (MAA).

Based on Eq. (3.10), building the proposed model requires performing regression on data from the electrical simulation, which is not desirable. This model's practical application is inconvenient for high-level designers of bus encoding techniques, who might not have limited expertise in electrical simulations. However, it is possible to characterize the coefficient C_n for a set

of representative high-level parameters (i.e., width, spacing, the metal layer, technology node) in advance. After this technology-characterization process, the C_n values required for a particular bus structure and technology node can be easily obtained using standard interpolation techniques. Therefore, given metal layers and physical characteristics of the interconnection links, a single model is sufficient for each technology node.

The model's coefficients, C_{g_n}, C_{c_n}, and C_{m_n} can be obtained from high-level parameters (i.e. length (l), width (w), spacing (s), metal layer (ml), and driver strength (d)) in the same way that today C_c and C_g are obtained from high-level technology models. We discuss high-level model's coefficients in more details in subsection 3.5.2.

3.4 Misalignment-Aware Delay Model

3.4.1 Related Works

Crosstalk delay is becoming severe in the advent of new technology nodes. Accurate delay models are required to cope with this problem. Current high-level delay estimation models assume that the signals are synchronized throughout the interconnect, which is not a valid assumption, as discussed in the previous chapter.

Many delay models have been proposed in the literature. In particular, there are two categories of delay models, first, numerical [72, 73], and second, analytical [6, 74, 75] approaches.

The most recent delay models in the literature are based on the numerical approaches. Despite the high accuracy, these models suffer from bulky lookup tables, high complexity, and technology dependence that limit their utilization [6].

Traditionally, analytical delay models do not consider the crosstalk between adjacent wires [75]. This exerts a considerable impact on the accuracy of the delay model, particularly for smaller technologies. In [74], the authors proposed a delay model based on a 3-wire bus considering crosstalk capacitance. This model is derived using Elmore delay [76, 77]. In wider buses, however, this model tends to mispredict delay value for different transition classes[1].

[1] details of crosstalk-based bus transition classification can be found in 2.3.3

In [6], a delay model based on a 5-wire bus is proposed. They first obtain differential equations describing a 3-wire bus and solve them using eigenvalues. The resulting equations then solve for 50% signal values. They expand the 3-wire delay model by decomposing 5-wire patterns into the 3-wire patterns. Even though they achieved higher accuracy in comparison with other existing techniques, they ignore victim-aggressor alignment. It has been shown that ignoring the misalignment effect in statistical delay calculation can lead to up to 114.65% mismatch of interconnect delay means with simulation results [23].

3.4.2 Model development

As we discussed in the previous subsection, the intrinsic misalignment effect does not appear in different transition patterns similarly. In other words, misalignment happens more significantly in the worst-case pattern $\uparrow\downarrow$ and less significantly in the best-case pattern $\uparrow\uparrow$.

It is not only the toggling direction of the transitions that determines the misalignment effect. When considering $\uparrow\downarrow$, the absolute difference between T_e of two neighboring wires can also contribute in extent of the misalignment that occurs. Accordingly, we define Misalignment Factor (MF) using the class function as multiplication of $[\uparrow\downarrow](\Delta b_i, \Delta b_{i+1})$ and $\mid T_{e_i} - T_{e_{i+1}} \mid$ as follows:

$$\mathrm{MF}_i = [\uparrow\downarrow]_i \cdot \mid T_{e_i} - T_{e_{i+1}} \mid . \tag{3.12}$$

Combining i^{th} and $(i-1)^{th}$ MF, we also define combined correction factor, CMF_i, as:

$$\mathrm{CMF}_i = \mathrm{MF}_i + \mathrm{MF}_{i-1}. \tag{3.13}$$

Using CMF, it is possible to consider the effect of misalignment on adjacent wires. There are 12 patterns classes where produce unique values of CMF and T_e. For $T_e = 4$, CMF can be equal $0, 1, 2, 3, 4$; for $T_e = 3$, CMF can be equal $0, 1$; for $T_e = 2$, CMF can be equal $0, 1, 2$; and for$T_e = 1$ and $T_e = 0$ CMF is equal to 0. For instance, middle line in transitions $\downarrow\uparrow\downarrow\uparrow\downarrow$ and $\bullet\uparrow\downarrow\uparrow\downarrow$, has a T_e and a CMF equal 4 and 0, and 4 and 1, respectively. However, we have found through the experiments that these two particular patterns impose relatively the same delay in the middle wire. Basically, among all 12 groups, some of the groups are overlapped with each other in the amount of delay they impose. Using, CMF and T_e, and in order to combine some groups, we define

the correction factor T_m as a function of T_e and CMF:

$$T_{m_i} = T_{e_i} - \eta\, \mathbb{F}(T_{e_i}, \mathrm{CMF}_i), \tag{3.14}$$

where η is 0 when signals are aligned in the first segment and is 1 when signals are misaligned in the other segments. $\mathbb{F}(T_{e_i}, CMF_i)$ is a non-linear function of T_e and CMF. We have found the function through the experiments as follows:

$$T_{m_i} = T_{e_i} - \eta \left[\frac{15\mathrm{CMF}_i}{6(T_{e_i}+1) + 2\mathrm{CMF}_i} \right]. \tag{3.15}$$

According to this model, there are five sets of patterns, and they are denoted as kG for $k \in 1, 2, 3, 4, 5$.

3.4.3 Parameter estimation

Similar to development of the energy model, using OLS estimator and the proposed term in previous subsection, T_{m_i}, we are able to attain the estimated values of regression coefficients that correspond to the changes in the delay in each segment of the interconnect as:

$$C_{m_{i,n}} = OLS(\tau_{i,n}, \widehat{T}_{mb_i}), \tag{3.16}$$

where $\widehat{T}_{mb_i}$ is the binary matrix corresponds to the decimal vector of $2^{T_{m_i}}$ as matrix of regressors. $\tau_{i,n}$ is the regressand or measured delay through simulation for i^{th} wire and n^{th} segment of the interconnect. The estimated delay using the regression coefficient, $C_{m_{i,n}}$ is given by:

$$\widetilde{\tau}_{i,n} = \widehat{T}_{mb_i} \,.\, C_{m_{i,n}}, \tag{3.17}$$

where $\widetilde{\tau}_{i,n}$ is the estimated delay in the wire i and segment n. In the rest of this paper, we refer to this model as Misalignment aware delay model (MAA).

3.5 Evaluations

This section assesses the accuracy of the proposed misalignment-aware energy and delay models in multi-segment interconnects. We perform case studies to show how the previous standard models' miss-prediction can lead to overesti-

mation or underestimation of the different low-power and crosstalk avoidance techniques' effectiveness.

3.5.1 Evaluation Setup

Bus Configuration

In this thesis, we focus on an equally separated, multi-segment, parallel global interconnect with group shielding as depicted in Fig. 3.2. The parasitics can be modeled as a distributed RC-lumped model with resistance R, self-capacitance C_g, and the coupling capacitance C_c. The first-order coupling capacitances almost entirely shield the second-order coupling capacitances. Thus, each wire in the interconnect is only capacitively coupled to its adjacent wires.

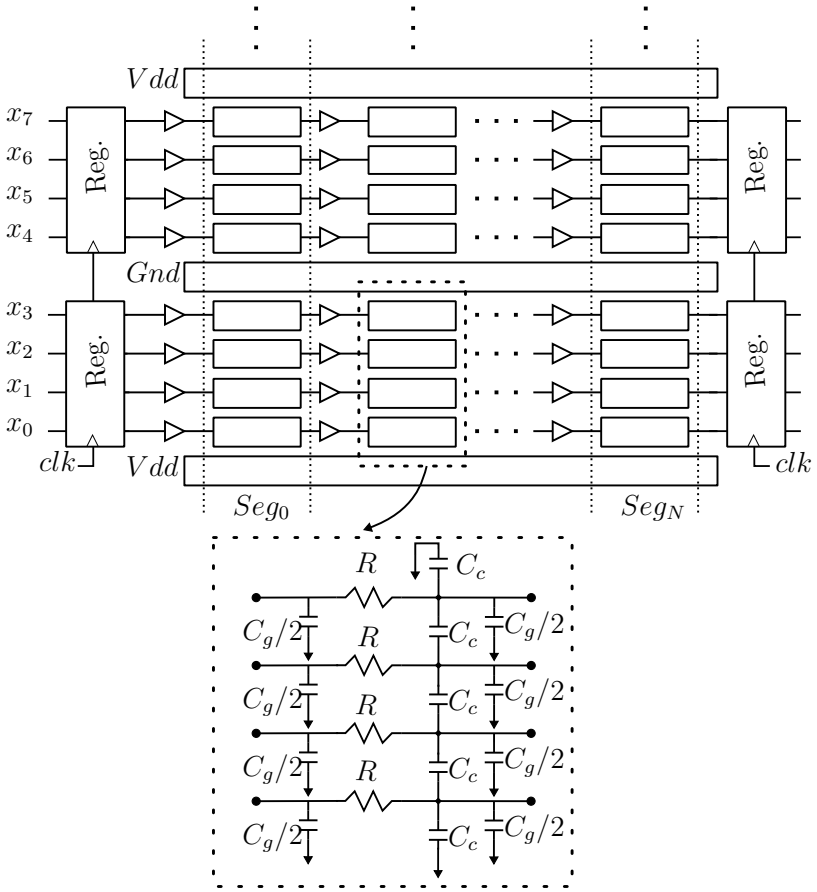


Fig. 3.2: An N-segment bus configuration with repeater insertion and group shielding of 4 wires. The physical RC model of each segment is presented in the dashed box.

As it is discussed earlier in this chapter, optimal repeater insertion is an

effective solution for delay reduction in long interconnects [27]. Both resistance, R, and capacitance, C, increase with wire length, l, so the propagation delay, $t_{pd} = RC$, of a wire increases with l^2. The propagation delay is reduced by splitting the wire into N segments and inserting repeaters to drive the wire actively. The overall delay of the segmented wire can be reduced from approximately l^2 to l^2/N. If the number of segments is proportional to the length, the overall delay increases only linearly with l. Using the repeater in this project, we efficiently split the into N segments driven with identical drivers.

For wires with an aspect ratio (thickness/width) greater than 2, the coupling capacitance can account for 2/3 to 3/4 of the total capacitance [27]. Shielding is used for reducing and controlling crosstalk in this case. One shielding topology which avoids the impact of coupling is to shield every single wire on both sides. However, the considerable increase in the area exceeds the gains of this shielding topology. An alternative, expedient shielding topology would be to shield every group of wires in a bus. We choose to shield groups of 4 wires by paralleled Vdd and Gnd wires. This typical group shielding topology fits perfectly to the bit-width of the real-world signals, usually transferred through the bus (for example, an 8-bit or 16-bit image, which is a factor of 4).

The circuit structure in Fig. 3.2 is used for the experiments in this section. A 2 mm, 5-bit-width global interconnect is divided into 9 segments and each segment is driven using a CMOS driver. The driver, which is an inverter, has the drive strength of α times of the minimum sized inverter (e.g., a $\times 3$ driver uses n-channel transistor width, which is three times minimum channel width and p-channel transistor width, which is six times minimum channel width). We assume that the identical wires are driven and loaded by equally sized inverters. The experiments are carried out on an interconnect in metal layer 4 for bus structure with different widths and spacings. The designers usually select the wire width, spacing, and layer usage to trade off delay, bandwidth, energy, and noise. We have chosen the standard values. The simulation results are obtained using a commercial 65 nm technology node with the supply voltage of 1.2 V and $\times 3$ and $\times 2$ sized drivers. The SPICE-level simulation is carried out using *Cadence* Spectre circuit simulator.

Automatic Simulation Framework

This subsection briefly explains the simulation framework that we use to automatize delay and energy acquisition. Using this framework, we can automatize the simulation transparent to technology.

First, we should extract the electrical parameters. To precisely model the behavior of on-chip interconnects, it is necessary to determine the value of parameters, including the distributed capacitance, resistance, and in some cases, inductance. Generally, there are a couple of methods to acquire these parameters for a given interconnect configuration. The information obtained from the design manual can be used to construct an accurate model in a 3D EM-field solver [78]. It is also possible to use general interconnect data provided by ITRS [26]. However, the data is not necessarily accurate for the specific process. To get analytical expressions for the distributed parameters as a function of material properties and dimensions, one can also assume some simplifications in the configuration and use the Maxwell equations. Commercial high-level technology models are also developed and available for a specific technology node. These high-level technology models function based on Maxwell equations. In this thesis, we have used the commercial high-level technology models to obtain the distributed parameters.

Second, the accurate measurement of energy, delay, and output values are carried out using components written in Verilog AMS hardware description language. Using these modules results in more accurate quantification of energy, delay, and output signals in comparison with the build-in calculator of the Cadence Virtuoso software.

Automatic execution of the simulation is conducted using shell script and run by the Unix shell. We assign different netlists, stimuli, and circuit specifications such as supply voltage, driver strength, and the number of noise run in the shell script. The simulation for a new technology node can be performed by changing the netlist (more specifically, transistors and the supply voltage).

3.5.2 Energy Model Evaluation

We evaluate the proposed energy model (MAA) by comparing it with the standard energy model (STD) and the linear regression of the standard model (LR-STD). Fig. 3.3 illustrates the Mean Square Error (MSE) versus the segment

number for two different edge effect scenarios explained in Section 2. The results are obtained for 3× drivers and bus width and spacing of 0.15 μm using all possible input patterns. According to the results, the naive STD model is inaccurate in estimating the multi-segment interconnects' energy consumption. The LR-STD model, as an intermediate solution, can improve the accuracy of the standard model. For example, in 8^{th} segment of the interconnect, MSE of LR-STD ($\approx 0.97e^{-28}$) is about 1.8 times smaller than that of the STD ($\approx 1.77e^{-28}$). Even though LR-STD improves the accuracy of energy estimation considerably in comparison with STD, MAA model provides an even better accuracy improvement; MSE of MAA model ($\approx 0.063e^{-28}$) is about 28 times smaller than that of the STD in 8^{th} segment of the interconnect. The limitation of the STD model to incorporate the misalignment effect renders it inappropriate for energy estimation of segmented parallel interconnects.

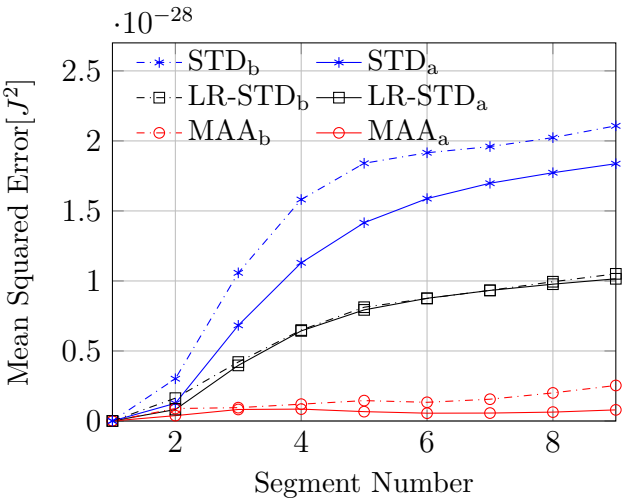


Fig. 3.3: Comparison of the accuracy provided by the proposed energy model (MAA), traditional standard model (STD) and the linear regression of the standard model (LR-STD). Subscribes a and b denote V_{dd}-*shielded* and *non-shielded* edge effect scenarios, respectively.

To evaluate the proposed model's effectiveness for different bus characteristics, we compare MAA and STD for different bus widths, spacings, and drive strengths. The bus is structured using *non-shielded* edge effect scenario. We use normalized mean square error and normalized maximum absolute error as two accuracy comparison metrics. The maximum error that indicates the extreme value of the error is an important metric for developing the low-power techniques, where the peak energy consumption is a constraint. Results are presented in three

different segments of the interconnect in Table 3.1. In the first segment, there is no significant misalignment. Thus no improvement in the accuracy of the MAA compared to the STD is expected. However, according to our simulation results, the MAA outperforms the STD in both comparison metrics in other segments and for different bus structures. For instance, for the typical structure of a bus with bus width of 0.15 μm, spacing of 0.15 μm and drive strength of 3×, the normalized mean square error of MAA, 0.99%, is more than 8 times better than that of STD, 8.29%. In today's drastic downscaling of layout geometries, it is a realistic scenario to consider the minimum bus width and spacing. In small bus geometries, the error of the STD is much higher than the MAA, and the biggest accuracy improvement of MAA happens in these bus structures.

According to the proposed methodology, the measured energy through the simulation, E_n, is required for calculation of the regression coefficient in Eq.3.10. Once the simulation results for a specific bus structure is obtained, the estimated regression coefficient, C_n, should work perfectly for any other input patterns and bus bit-widths. Therefore, to characterize the effectiveness of the proposed energy model, we evaluate the accuracy of the MAA in three different scenarios as follow:

(A) In this scenario, we estimate the energy consumption of the interconnect so that the regressors, X_m, and regression coefficient C_n in Eq.3.10 are obtained using the same B-bit random input transitions.

(B) The energy is estimated by exploiting X_m with a B-bit random input stream, while C_n is obtained using a different set of B-bit random input stream.

(C) X_m is exploited with a random B-bit input stream while C_n is obtained with a random N-bit input stream, where $N \neq B$.

In a 3-wire bus, scenario (C) is not applicable since there is no lower bit-width bus available. A 4-wire and 5-wire bus, scenario (C) is analyzed while C_n is obtained using a 3-bit random input stream. Furthermore, in the case of a 6-wire bus, scenario (C) is analyzed while C_n is obtained using a 5-bit random input stream. We perform the analysis for a bus width of 0.15 μm, spacing of 0.15 μm and drive strength of 3×. The bus is structured using *non-shielded* edge effect scenario in this simulation. This analysis also determines whether the proposed model is valid for any bit-width or not.

Table 3.1: Normalized mean square error in % and normalized maximum absolute error in % in different segments considering different bus characterizations: traditional standard energy model (STD) vs. misalignment-aware energy model (*MAA*).

Bus Characteristics			**Segment**	**Normalized Mean Squared Error**		**Normalized Maximum Absolute Error**	
Width (um)	**Spacing (um)**	**Drive Strength**		**STD**	**MAA**	**STD**	**MAA**
0.15	0.15	2x	1	0.001	0.001	0.319	0.319
			5	8.305	0.621	51.019	16.985
			9	9.194	1.108	62.646	21.168
		3x	1	0.001	0.001	0.534	0.534
			5	7.484	0.592	47.319	17.524
			9	8.293	0.998	58.627	17.197
	0.45	2x	1	0.001	0.001	0.484	0.484
			5	4.233	0.228	34.126	9.374
			9	5.058	0.417	46.505	18.104
		3x	1	0.002	0.002	0.800	0.800
			5	3.509	0.182	29.902	8.289
			9	4.389	0.336	43.022	17.444
0.45	0.15	2x	1	0.001	0.001	0.271	0.271
			5	7.780	0.615	48.415	17.540
			9	8.575	0.962	61.023	17.785
		3x	1	0.001	0.001	0.458	0.458
			5	7.028	0.527	47.530	15.755
			9	7.847	0.910	54.607	16.523
	0.45	2x	1	0.180	0.180	0.402	0.402
			5	3.988	0.208	32.874	9.129
			9	4.882	0.385	44.053	17.296
		3x	1	0.002	0.002	0.672	0.672
			5	3.378	0.169	29.225	8.122
			9	4.315	0.311	41.397	16.825

Table 3.2 compares the accuracy of the MAA and the STD for different simulation scenarios using normalized maximum absolute error and normalized mean squared error. According to the results, the MAA preserves its accuracy for different simulation scenarios, and the discrepancy of the results is negligible. For instance, for the 6-wire bus and in segment 9, the normalized mean squared error changes only by 0.009% comparing results in scenarios (A) and (C). Similar results are obtained for the case of a 4-wire bus. Please note that we have shown the results for 3, 4, 5, and 6-wire buses as exemplary cases.

Table 3.2: Normalized mean square error in % and Normalized maximum absolute error in % in different segments considering different bus bit-width for three regression coefficient scenario: Standard Energy model (STD) vs. Proposed Misalignment Aware Energy model (MAA).

Number of Wires		Regression Coefficient Estimation Scenario	Normalized Mean Squared Error			Normalized Maximum Absolute Error		
			Seg.1	Seg.5	Seg.9	Seg.1	Seg.5	Seg.9
3	MAA	A	0.002	0.472	0.819	0.615	12.339	14.489
		B	0.002	0.472	0.819	0.615	12.339	14.489
	STD	A , B , C	0.002	16.556	17.971	0.615	56.930	57.324
4	MAA	A	0.001	0.868	1.536	0.581	17.297	29.778
		B	0.001	0.867	1.534	0.581	17.281	29.761
		C	0.001	1.052	1.810	0.581	15.534	31.643
	STD	A , B , C	0.001	9.249	10.333	0.581	40.068	40.383
5	MAA	A	0.001	0.593	1.066	0.400	17.706	17.279
		B	0.001	0.600	1.070	0.400	17.197	17.031
		C	0.001	0.778	1.291	0.400	18.017	17.985
	STD	A , B , C	0.001	6.957	7.726	0.400	47.304	58.611
6	MAA	A	0.001	0.437	0.909	0.437	16.076	20.681
		B	0.001	0.438	0.914	0.437	15.896	20.308
		C	0.001	0.447	0.918	0.437	15.436	19.866
	STD	A , B , C	0.001	5.465	6.308	0.437	36.038	52.486

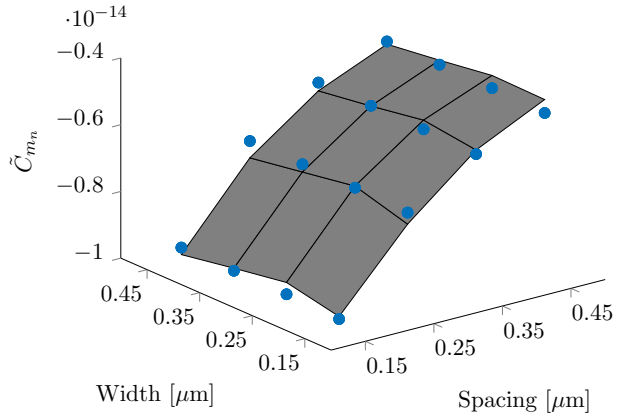
Moreover, we focus mainly on narrow on-chip interconnects, which implies buses with bit-width less than 8 bits. Considering shielding as an established widely used technique for energy and delay reduction of parallel buses, our model for a narrow bus can be employed for wider buses. In this regard, the proposed energy model can be applied to most interconnection design scenarios.

As it is discussed earlier in this section, a high-level model to estimate the energy model's coefficients for a given process technology is favorable for designers of bus encoding techniques. For a given technology and driver strength, the model's coefficients can be characterized using the standard regression technique. For example, a high-level model to estimate the coefficient C_{m_n} can be obtained using Ordinary Least Square (OLS) as follows:

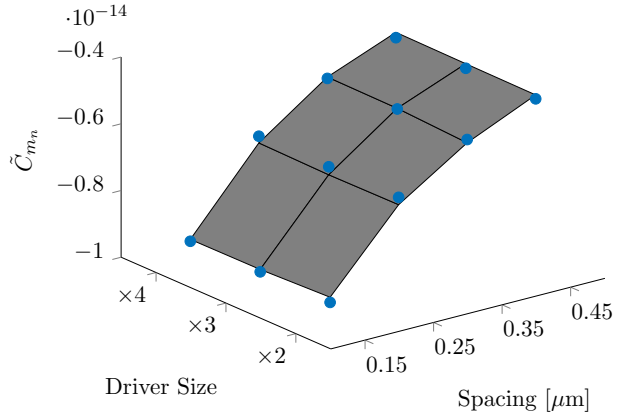
$$\tilde{C}_{m_n} = OLS(C_{m_n}\ ,\ [\alpha, \beta]), \tag{3.18}$$

where $\tilde{C}_{m_n}$ is the estimated coefficient, and α and β are nonlinear functions of interconnects' width and spacing. We compare C_{m_n} and its estimation, i.e. $\tilde{C}_{m_n}$, for a 2 mm, 5-bit-width bus in Fig.3.4. The surface plot represents C_{m_n}

and the blue dot represent $\tilde{C}_{m_n}$. The results are illustrated for different widths and spacings for the driver size of $\times 4$ in Fig.3.4-(a) and for different driver sizes and spacings for the bus width of $0.25\,\mu$m in Fig.3.4-(b). According to the results, the discrepancy between the estimated value, $\tilde{C}_{m_n}$, and exact value, C_{m_n}, is negligible.



(a)



(b)

Fig. 3.4: The comparison of the estimated value ($\tilde{C}_{m_n}$) and the exact value (C_{m_n}) for a 2 mm bus. The results are illustrated for a fixed value of the driver size ($\times 4$) (a), and for a fixed value of the width ($0.25\,\mu$m) (b) as illustrative examples.

To validate the high-level model, we calculate the energy using $\tilde{C}_{m_n}$ and C_{m_n} coefficients. We compare the mean squared error for a 5-bit bus with width and spacing of $0.15\,\mu$m and drive strength of $4\times$. According to the results, using $\tilde{C}_{m_n}$, the MSE is $2.849e^{-29}$, while using C_{m_n} it is $2.825e^{-29}$. This variation in the MSE is negligible especially when it is compared with the large MSE of

the standard energy model which is $2.138e^{-28}$.

The efficiency characterization of the low-power techniques depends on the accuracy of the energy model, and therefore the accuracy improvement of the proposed energy model is relevant, especially in smaller technology nodes.

Please note that the design space exploration to find the best physical topology of the interconnection medium based on the given data is not discussed in this thesis.

Case-study

This section's purpose is twofold: first, we want to show how relevant the proposed model improvement in accuracy of the energy estimation in the on-chip interconnects is; and second, we discuss possible future works for the development of new low-power coding frameworks based on the proposed energy model.

We estimate the energy reduction of two well-known, low-power codings presented in the literature, classical bus invert (CBI) [8] and full invert (FI) [40] coding. The simulation is carried out using the minimum bus geometries, i.e. width of 0.15 μm, spacing of 0.15 μm and drive strength of 3×. Like the earlier analysis, the bus boundary wires are structured using *non-shielded* edge effect. The estimated energy reduction of the low-power coding techniques using simulation is compared to that of MAA and STD models in Fig. 3.5. As is expected, the energy estimation of the STD model for different segments of the interconnect is constant. Although STD estimation is accurate for the first segment, it has up to 45% misprediction for the other segments of the interconnect. On the other hand, MAA has a minor inaccuracy compared to the simulation. These results suggest that an accurate energy model is imperative for a reasonable comparison of different low-power schemes.

So far, most low-power coding schemes like CAC aim to avoid the worst-case transitions to reduce the coupling effect. Even though it is true, due to the dominance of coupling capacitances over ground capacitances, according to evaluations in this paper, the worst-case pattern importance has been diminished due to the intrinsic misalignment effect. The misalignment effect in the worst-case transition reduces the effective coupling capacitance associated with the transitions in different interconnect segments. This fact suggests that the improvement of coding schemes that only focus on avoiding worst-case patterns

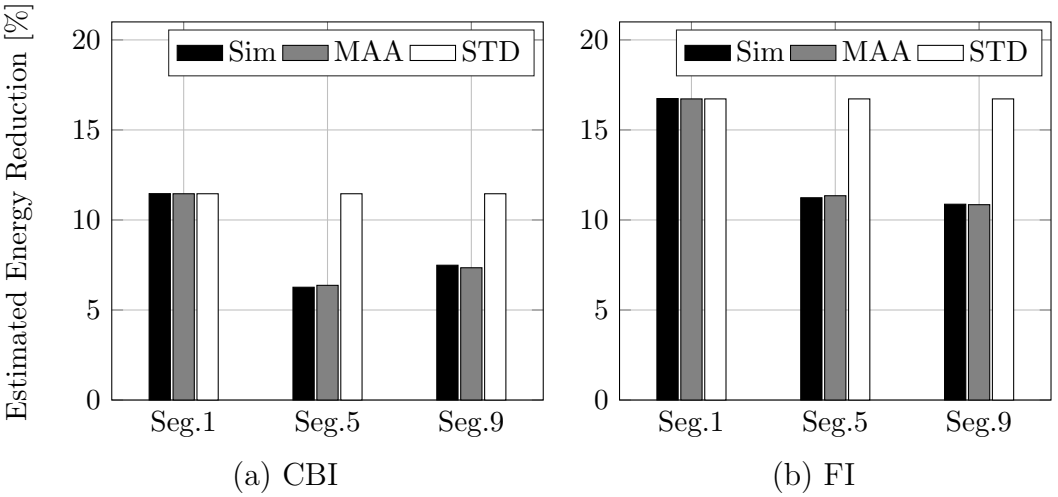


Fig. 3.5: Estimation of the energy reduction (in %) provided by reference simulation results (Sim), traditional standard energy model (STD), and the proposed energy model (MAA). The results are provided for two low-power coding schemes Classical Bus Invert(CBI) and Full Invert(FI) in different segments of the interconnect.

is not as significant as is expected in the literature, and further investigation is required. The standard model misprediction prevents the precise evaluation and modification of low-power codings. The proposed model can be used for developing a more effective low-power scheme.

3.5.3 Delay Model Evaluation

We evaluate the performance of the Misalignment Aware delay model (MAA) in multi-segment interconnects in two different scenarios. First, we evaluate the accuracy of our proposed delay model in comparison with the standard delay model (STD)[2] and Ref. [6]. Second, we perform a case study to show how the previous delay models' inaccurate prediction can lead to overestimation or underestimation of the developed CAC codings.

The simulation results are obtained for a commercial 65 nm technology using the SPICE-like circuit simulator, Cadence SPECTRE for a 5-wire bus in 9 segments as an arbitrary narrow bus structure. We assume that the identical wires are driven and loaded by equally sized inverters with a drive strength of 6×. We carry out the simulation on a 1 mm bus in metal 4 with ×3 drivers.

[2]For a detailed derivation of the standard delay model see Subsection 2.3.1

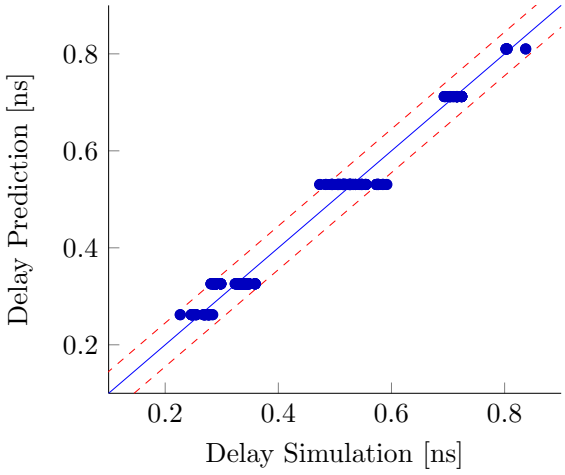
The delay estimation and misprediction of the worst-case transition for different delay models are tabulated in Table 3.3. Results are obtained for the worst-case delay in wire 3. According to the results, MAA has a significantly better performance in all segments. It shows its best performance where the signals are completely misaligned in segment 9. Mis-prediction of MAA in this segment is 3.2%. According to this table, the proposed delay model preserves its accuracy in different interconnect segments and outperforms other delay models.

Table 3.3: Simulation delay and delay estimation for worst-case pattern in [ns] and mis-prediction of worst-case transitions in [%] according to MAA, STD and [6] in different segments of the interconnect

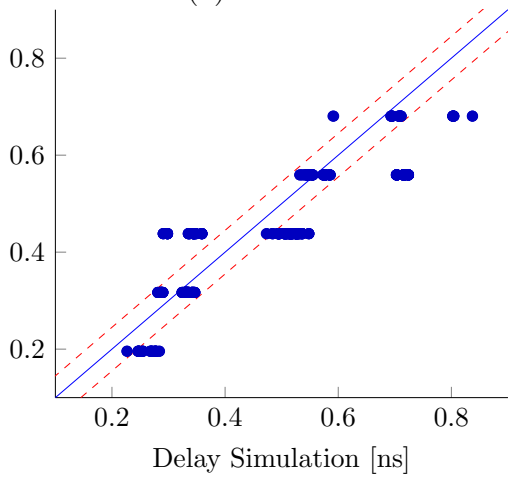
	Delay of worst-case transition in wire 3 in [ns]				Mis-prediction of worst-case transition delay in [%]		
segment	sim.	MAA	STD	[6]	MAA	STD	[6]
2	0.191	0.185	0.151	0.167	3.49	21.10	12.61
5	0.51	0.479	0.378	0.418	6.9	26.5	18.6
9	0.84	0.810	0.680	0.754	3.2	18.7	10.0

Delay prediction of STD and MAA models in comparison with simulation results in wire 3 and segment 9 for all possible input transitions are illustrated in Fig. 3.6. Dashed lines in this figure show $\pm 5\%$ deviation of delay prediction from the exact delay. According to the Fig. 3.6-(a), almost all the transitions lie within the $\pm 5\%$ boundaries. However, according to Fig. 3.6-(b), the STD model delay prediction deviation from the simulation results in many transition patterns is outside of the dashed line boundaries. Results infer that MAA has the misprediction of less than 5% for all transition patterns in segment 9; without requiring more groups than previous models. This is mainly because MAA is developed to incorporate variations, including intrinsic and extrinsic misalignments.

The variability awareness of a delay model can be assessed while the interconnect is affected by noise or other variations. To analyze this effect, we have done simulations taking into account the noise. We compare the results for MAA and STD models for wire 3 and segment 9 in Fig. 3.7. In Fig. 3.7-(b), we use the standard paradigm of pattern classification, iC for $i \in 1, 2, 3, 4, 5$, to split transition patterns into different classes and group them into different colors. As it is illustrated in this figure, standard model estimation of different



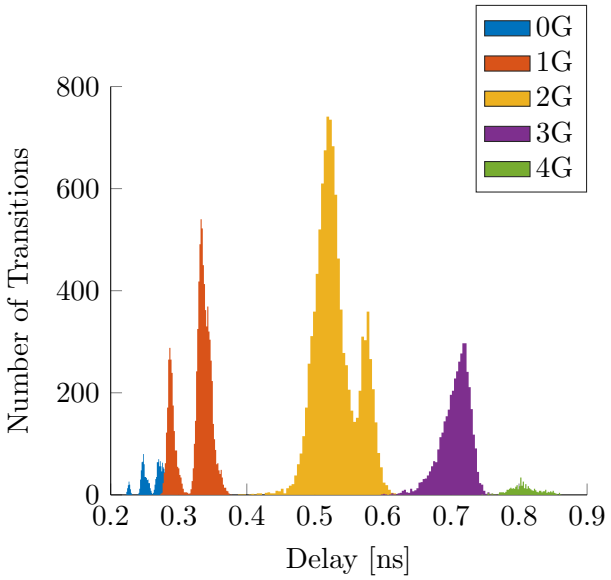
(a) MAA



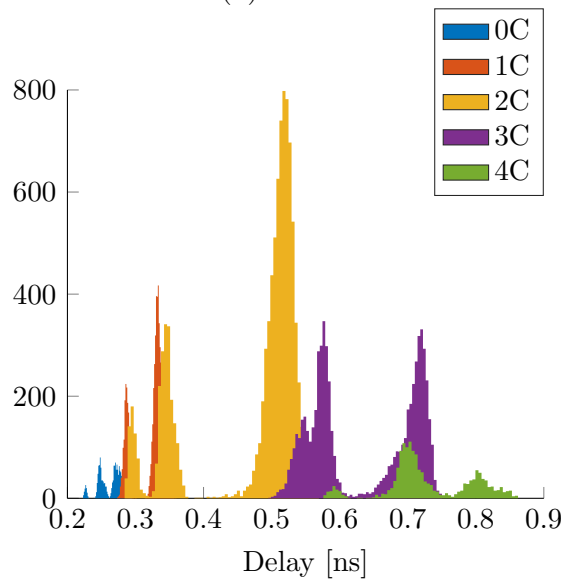
(b) STD

Fig. 3.6: The delay prediction of MAA and STD models versus the simulation in wire 3 and segment 9 of a 5-wire interconnect. the dashed lines are ±5% delay mis-prediction boundaries

transition classes are overlapped with each other. For instance, class 3C is overlapped by 4C, and 3C by itself extends over 2C, and so on. This issue can constrain extra unnecessary overhead on CAC coding schemes. In Fig. 3.7-(a), we use the proposed paradigm of pattern classification, iG for $i \in 1, 2, 3, 4, 5$, to split transition patterns into different classes. According to this figure, each class has a neat edge that there is almost no overlapping. Results show the effectiveness of MAA in modeling the delay of the interconnects in the presence of variations.



(a) MAA



(b) STD

Fig. 3.7: Classified delay distribution of all possible patterns affected by variations (noise) using STD and MAA models for wire 3 and segment 9 of a 5-wire interconnect

Case-study

To illustrate the relevance of the improvements in accuracy of the delay estimation of MAA in the on-chip interconnects, we characterized the delay reduction

of a well-known CAC code presented in the literature: FTF[3] codings [54].

We measure the reduction in the delay achieved by FTF code in 3^{rd} wire of a 5-wire, 9-segment interconnect. The overall delay is measured in 3 different segments of the interconnect to compare the error of different models in the estimation of delay reduction of FTF. The results reported in Table 3.4 show the accuracy of the estimation for the different models. The misprediction of STD is exceptionally high in all the segments. The expected delay reductions of MAA are 36.6%, and that of simulation is 35.9%. While the errors of [6] and STD are significant, they expect 51.1% and 47.67% delay reduction respectively in 9^{th} segment of the interconnect.

Table 3.4: Estimated delay improvement of FTF coding by different delay models, MAA, STD and [6] and the error in estimation of the improvement for each model in [%] for different segments of the interconnect

	Estimated performance improvement in [%]				Mis-prediction of FTF delay improvement in [%]		
segment	sim.	MAA	STD	[6]	MAA	STD	[6]
2	42.03	40.88	49.20	52.55	1.15	6.9	10.52
5	42.99	42.77	52.70	55.81	0.22	9.71	12.82
9	35.95	36.63	47.69	51.13	0.68	11.74	15.18

These results suggest that an accurate delay model is imperative for a sensible quantification of different coupling avoidance coding schemes early in the design flow. It is necessary to model the misalignment effect with high-level models, usable during the construction and quantification of coding approaches.

3.6 Conclusion

The temporal alignment of signals drastically influences the energy and propagation delay of interconnects. The misalignment effect is originated intrinsically in unequal effective capacitances seen by each driver and extrinsically in the technology variability. Precise high-level energy and delay models are essential for the development and characterization of interconnects encoding techniques. Besides, design space exploration to find the best physical topology of the interconnection medium requires accurate models.

[3]detailed explanation of different optimization techniques can be found in Section 2.4

This chapter analyzes the neighboring wires' alignment behavior to identify the extent of misalignment for worst- and best-case transitions. Based on our observation, we have developed novel energy and delay models considering the misalignment effect.

Fo energy model development is necessary to analyze the misalignment in a group of 4 wires to incorporate this effect into the energy model. We have introduced a correction factor that accounts for the misalignment effect, and it is used to develop a misalignment-aware energy model. Moreover, we evaluate the proposed model's accuracy, MAA, for different bus characteristics and bit-widths. The results demonstrate a significant improvement in accuracy estimation using the proposed model compared to the standard energy model. For instance, in bus structure with the bus width of 0.15 μm, spacing of 0.15 μm and drive strength of $\times 3$, the normalized mean squared error of 8.29% in STD model is reduced to 0.99% using MAA. The proposed energy model can be used for precise evaluation and fair comparison of different low-power techniques like coding and stochastic schemes. While the STD model estimates the energy reduction of more than 10% for classical bus invert coding in 5^{th} segment of the interconnect, the simulation results and the proposed energy model estimations show almost half of this value.

Similarly, we proposed a delay model to account for the misalignment effect, statistical variations, and noise. Our proposed model establishes the connection between the delay of the coupled interconnects and transmitted signal patterns; thus, it enables a reliable efficiency characterization of CACs and can be used to study precisely stochastic approaches. The results of delay model evaluation show significant improvement in accuracy compared with previous models. For example, MAA can improve the delay prediction by 74% and 63%, respectively, in comparison with STD and [6] models. Furthermore, as reported in the case-study, ignoring the intrinsic misalignment can result in more than 15% misprediction in the efficiency evaluation of CAC codes.

CHAPTER 4

Approximate On-Chip Communication

Contents

4.1 Introduction

In chapter 2, we have studied various encoding techniques proposed in the literature. Even though they are effective techniques to improve interconnection, almost all proposed techniques greatly suffer from noise and the following bounds required for a reliable utilization. Approximate computing approaches relax some reliability requirements of the designs to gain speed-up or/and energy efficiency in specific applications [79]. Even though approximate computing is extensively explored for computing systems, only a little attention has been given to its potentials for communication improvements.

In this chapter, we explore the benefits of the approximation for communication. In particular, we tackle the communication bottleneck by proposing four promising approximation techniques.

First, we propose an integer-value coding technique based on Swap (swizzling) and Inversion of the input signals. The proposed technique is designed to target the area-constrained applications. The optimal combination of Swap and Inversion coding is found using an exhaustive search framework. This coding technique aims to reduce the error's magnitude, so that error values fall within the tolerable error margins for a given application. Therefore, the main target communication cost (either power or delay) can be safely reduced by the coding.

Second, intended for applications with a more relaxed area constraints, we propose a Crosstalk Avoidance-based approximate coding. CACs use a fixed codebook to generate codewords based on the input data. Even though these coding techniques are very useful in delay reduction, they are susceptible to noise when used for approximate communication. The optimal mapping of the data words and codewords leads to the minimum error magnitude. We use a primary decoder to map non-codeword received values to the associated valid codewords. Moreover, we employ the Inversion of selective signals to leverage the input data's spatial value locality.

Third, we present a novel stochastic on-chip communication approach combining the classical wave-pipelining and the stochastic/approximate communication. On-chip communication performance can be dramatically improved using wave-pipelining, which is very power efficient since it avoids additional pipeline registers. However, the error-free operation is getting challenging as technology scales down. Considering the challenges of classical wave-pipelining, we propose the concept of stochastic wave-pipelining communication seeking two specific benefits: First, it relaxes the stringent constraints of the classical wave-pipelining, which in some cases prevents the effective practical use of this technique. Second, it provides an additional improvement in the on-chip interconnection performance depending on the applications' error-tolerance.

Finally, a low-power alternating bit truncation techniques is proposed. Unlike the simple bit truncation approach, we suggest setting the predefined number of LSBs to 0 to inactivate the wires. Swizzling of the signals, we utilize the grounded wires as virtual shields between data lines. This bit truncation technique has a great potential in energy and error reduction for approximate

communication.

The remainder of this chapter is structured as follows. In Section 4.2, we introduce encoding schemes with a very light overhead for stochastic communication targeting area-constrained applications. Also, a Crosstalk-Avoidance-based Integer-Value coding is presented targeting applications with looser area constraints. In Section 4.3, we present stochastic Wave-pipelining, a novel stochastic communication technique whose primary aim is to increase the performance for an acceptable error. Finally, in Section 4.4, we propose Alternating bit-truncation. The contributions in this chapter are published in [60] and [80].

4.2 Integer-Value Encoding

This section exploits the integer-value representation of transmitted signals in the approximate on-chip parallel buses and minimizes timing errors in term of the integer distance using a dedicated family of codings, called IVC. The integer-value deviation, as opposed to Hamming distance error, mainly focuses on integer distance rather than the number of bit errors, bit error rate (BER). The integer-value deviation error is defined as the difference between the integer value of the possibly erroneous received output, $\hat{X}$, and the integer value of the input, X, i.e., $\varepsilon = \hat{X} - X$.

First, we apply the simplest codings with minimum area overhead for area-constrained applications.

Restricted to the minimum overhead, we explore two classes of simple, memoryless integer-value encodings (*Swap-* and *Inversion-Codings*) theoretically. These coding techniques are exploited to reduce the integer value error for uniform and highly correlated data, respectively. The *Swap-Coding* changes the assignment of the input signals to wires while the *Inversion-Coding* inverts selective signals in order to minimize the magnitude of error. We propose a combination of *Swap-* and *Inversion-Codings* (we refer to it as Combined Integer-Value (CIV) encoding) to address the real scenarios. Fig. 4.1 shows an arbitrary example of combined *Swap-* and *Inversion-Codings*. We refer to coding with the weight of the bit being transmitted, w_x; a $\bar{w}_x$ denotes an inversion. Thus, we refer to the conventional data transmission without coding as (8,4,2,1), and the nomenclature for the combined coding of Fig. 4.1 will be ($\bar{8}$,$\bar{2}$,1,4). The numbers in this nomenclature represent the weights associated

to each signal transmitted through the bus, i.e., $(w_{x_3}, w_{x_2}, w_{x_1}, w_{x_0})$[1].

For a more relaxed area-constrained applications, we propose a Crosstalk-Avoidance-based Integer-Value (CA-IV) coding technique. We propose to find the mapping of the input data words and the valid code words that minimize the error magnitude. The valid codewords are the codewords that guarantee the error-free data transmission in different CAC encoding techniques, e.g. 3C-free forbidden pattern free coding (3C-FPF) [12]. The selective inversion of the input data words (*Inversion-Coding*) can provide an additional improvement to the results. This coding technique is capable of error magnitude reduction in higher frequencies at the expense of a small area overhead.

The input X can be characterized by the joint probability density function of the current input value, X^+, and the previous value, X^-. The transmission characteristics can be represented by the probability of receiving $\hat{Y}^+$ when the current coded value, Y^+, and the previous coded value, Y^-, are transfered through the interconnect. In summary, we define the *stochastic communication problem* as follows: Given a narrow bus structure, characterized by a $P_Y[\hat{Y}^+|Y^-, Y^+]$, and an input data source X, characterized by a $P_X[X^-, X^+]$, obtaining a memoryless coder, $Y = \mathbb{C}(X)$, and related decoder, $\hat{X} = \mathbb{D}(\hat{Y})$, that minimizes the mean square integer value error, $MSE = \mathbb{E}\{(\hat{X} - X)^2\}$.

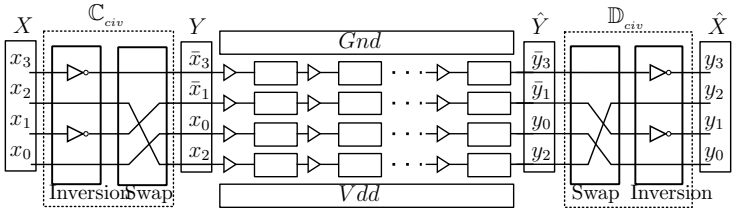


Fig. 4.1: An arbitrary coder and decoder structure for 4 bit-width bus. ($\bar{8}$,$\bar{2}$,1,4) coding combines the *Swap*- and *Inversion-coding*s. X, Y, $\hat{Y}$ and $\hat{X}$ are input, input coded, received and received decoded integer values, respectively.

4.2.1 Swap-Inversion-based Integer-Value Encoding

Given a transition pattern, the receiver may receive different possible erroneous values with frequency scaling. The received values depend on the working frequency and the delay imposed on each signal on the bus. For a 4-bit bus,

[1]For example, number 8 in this nomenclature is the weight of MSB and 1 is the weight of LSB.

the transition patterns ↓↑↓↑ and ↑↓↑↓ are the worst-case transitions when the misalignment effect is considered.

To simplify the problem, we define two timing regions: R_0 and R_1. Sampling the received signals in the region R_0, only the worst-case transitions produce errors. In other words, the clock period, T, belongs to R_0 if only the worst-case transitions of ↓↑↓↑ and ↑↓↑↓ are erroneous. Nevertheless, the receiver in this timing region may receive different erroneous outputs depending on the working frequency, supply voltage, noise, and bus specifications. The timing region R_1 in which the clock periods are smaller than region R_0, not only the aforementioned worst-case transitions produce the error, but also other worst-case transitions may produce the error.

Let us consider the transmission of, $X^+ = 0101$, when the previous value in the bus is $X^- = 1010$, which is the ↑↓↑↓ pattern in timing region R_0. According to the Eq. (2.7), wires have the effective coupling capacitance of 3, 4, 4, and 3, respectively. Based on the conventional expectations, the two middle wires, which have the higher effective capacitances, are slower and are the first to incur an error. Subsequently, the receiver is expected to receive 0011 instead of 0101. However, in reality, a small misalignment triggered by a noise or transistor mismatch can result in a discrepancy between these wires' delay. Thus, we can reasonably infer that the two middle wires do not necessarily violate the timing of the bus at the same time when the bus starts to produce the error.

The possibly received values when X^- and X^+ transferred through the bus, i.e. $\hat{Y}^+|X^-, X^+$, and their associated probabilities, $Pr.$, in the region R_0 for $X^- = 1010$ and $X^+ = 0101$ can be summarized as follows:

$$\hat{Y}^+|1010, 0101 = \begin{cases} 0101\ , \text{ with Pr.} = 1 - p_e \\ 0111\ , \text{ with Pr.} = \gamma p_s \\ 0001\ , \text{ with Pr.} = (1-\gamma) p_s \\ 0011\ , \text{ with Pr.} = p_e - p_s \end{cases}, \tag{4.1}$$

where p_e is the total probability of erroneous outputs, p_s the probability of error in a single wire, and $p_e - p_s$ is the probability of error in both middle wires simultaneously. γ is constant between 0 and 1 which depends on the tendency of having rising or falling errors when a single error occurs. A similar equation holds for $\hat{Y}^+|0101, 1010$.

Operating in smaller clock periods, T, in timing region R_1, (With the faster transmission of the signals) $\hat{Y}^+|1010, 0101 = 0011$. However, the additional worst-case patterns' occurrence makes it too complex to address region R_1 analytically.

In the following subsections, we first investigate the class of codings that address the uniformly distributed input signals. Next, we propose a coding scheme that works effectively for the correlated input signals.

Swap-Coding: A coding scheme considering uniformly distributed signals

To propose a coding scheme that works effectively for the uniformly distributed signals, in this section, we introduce the *Swap-coding*. This coding scheme leverages the physical properties of the CMOS interconnects and changes the assignment of signals to the wires to reduce the integer-value deviation. For a 4-wire bus, there are 4! possibilities for assignment of the signals to the wires. A careful swap of the signals to the wires can reduce the integer value error, ε, and therefore, the MSE. Note that bit inversions do not affect uniformly distributed signals due to the equal probability of transitions' occurrence.

In order to show the functionality of *Swap-Coding*, let us consider a *Swap-code*, $(w_{x_3}, w_{x_2}, w_{x_1}, w_{x_0})$. The decoded received integer values, $\hat{X}^+$, and their associated probabilities for a transition, where $X^+ = 0101 = 5$ and $X^- = 1010 = 10$, in R_0 can be given by:

$$\hat{X}^+|_{10,5} = \begin{cases} w_{x_2} + w_{x_0}, & \text{with Pr.} = 1 - p_e \\ w_{x_2} + w_{x_1} + w_{x_0}, & \text{with Pr.} = \gamma p_s \\ w_{x_0}, & \text{with Pr.} = (1-\gamma) p_s \\ w_{x_1} + w_{x_0}, & \text{with Pr.} = p_e - p_s \end{cases}, \tag{4.2}$$

where w_{x_3}, w_{x_2}, w_{x_1} and w_{x_0} are the associate weights for each signal transmitted through the bus. In order to simplify the problem, we make the assumption that the probabilities of the worst-case transitions of $\downarrow\uparrow\downarrow\uparrow$ and $\uparrow\downarrow\uparrow\downarrow$ are equal, i.e., $P_X[1010, 0101] = P_X[0101, 1010] = p_{X_\alpha}$ (uniform distribution is considered). Then, the mean squared of the integer value error as a function of the signals weights for R_0 is as follows:

$$MSE(w_{x_3}, w_{x_2}, w_{x_1}, w_{x_0}) = \mathbb{E}[(\hat{X}^+ - X)^2]$$

$$\approx p_{X_\alpha}\{2(p_e - p_s)(w_{x_1} - w_{x_2})^2 + p_s(w_{x_1}^2 + w_{x_2}^2)\}. \tag{4.3}$$

If no coding is used, i.e., $(w_{x_3}, w_{x_2}, w_{x_1}, w_{x_0})$=(8,4,2,1), then, the received values for transition from 10 to 5 and MSE in timing region R_0 can be given as follows:

$$\hat{X}^+|_{10,5} = \begin{cases} 5, & \text{with Pr.} = 1 - p_e \\ 7, & \text{with Pr.} = \gamma p_s \\ 1, & \text{with Pr.} = (1-\gamma)p_s \\ 3, & \text{with Pr.} = p_e - p_s \end{cases}, \tag{4.4}$$

$$MSE(8,4,2,1) \approx p_{X_\alpha}\{8(p_e - p_s) + 20p_s\}. \tag{4.5}$$

According to Eq. (4.3), MSE in region R_0 depends on w_{x_1} and w_{x_2}. The assignment of the signals to the wires so that x_1 and x_2 are assigned to the wires with minimum weights will minimize the integer value error. The (8,1,2,4) and (4,1,2,8) *Swap-Codings* are optimal solutions for minimizing the MSE for the uniform distributed signals in region R_0. Using the (8,1,2,4) *Swap-Coding*, we obtain:

$$MSE(8,1,2,4) \approx p_{X_\alpha}\{2(p_e - p_s) + 5p_s\}, \tag{4.6}$$

which is 4 times smaller in comparison with the case of no coding.

Fig. 4.2 illustrates the circuit structure of the (8-1-2-4) Swap coding. The coder and decoder make a straight swap between x_0 and x_2. In this case, the error appears in signal x_0 instead of x_2. The reduction in the integer-value deviation achieved together with **no** coder/decoder overhead. Note that, the proposed (8-1-2-4) and (4-1-2-8) *Swap-Coding* are the optimal coders only for timing region R_0.

Inversion-Coding: A coding scheme for correlated input signals In uniform distributed signals, all the transitions' probabilities are equal. However, many real-world applications such as image/video processing, wireless sensor networks, and voice processing in which correlated signals are transmitted and processed. In this case, the probability of transitions differs. *Inversion-Coding*

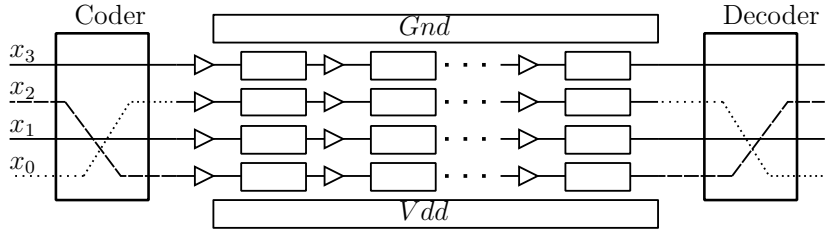


Fig. 4.2: The circuit structure of (8-1-2-4) *Swap-Coding*.

leverages the spatial value locality of the real-world data to exchange the probability of the worst-case transitions with the lowest probability transitions.

The input vector $\boldsymbol{X}=\left(\begin{smallmatrix} X^- \\ X^+ \end{smallmatrix}\right)$ is said to have a bivariate Gaussian distribution with the mean vector of $\boldsymbol{\mu}=\mathbb{E}(\boldsymbol{X})$ and the covariance matrix of $\boldsymbol{\Sigma}=\sigma^2 \left(\begin{smallmatrix} 1 & \rho \\ \rho & 1 \end{smallmatrix}\right)$ if its probability density function is given by:

$$P(\boldsymbol{X}) = \frac{1}{2\pi. \mid \boldsymbol{\Sigma} \mid^{\frac{1}{2}}} \exp\Big\{ -\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{X}-\boldsymbol{\mu}) \Big\}, \tag{4.7}$$

where ρ is the correlation coefficient and σ^2 is the variance.

According to Eq. (4.3), in region R_0, MSE depends on p_{X_α} as well as w_{x_1} and w_{x_2}. The selective inversion of wires, such that the erroneous transitions with high probabilities, i.e. $\downarrow\uparrow\downarrow\uparrow$ and $\uparrow\downarrow\uparrow\downarrow$, are exchanged with the transitions with lowest probabilities, will reduce the integer value error in correlated input signals. In other words, a coding technique is required, so that $P_X[\mathbb{D}(1010), \mathbb{D}(0101)]$ and $P_X[\mathbb{D}(0101), \mathbb{D}(1010)]$ are minimal.

For the unsigned positive correlated input signals, i.e., $\boldsymbol{\mu}=\left(\begin{smallmatrix} 7.5 \\ 7.5 \end{smallmatrix}\right)$ and $0<\rho<1$, $P_X[0000, 1111] = P_X[1111, 0000] = p_{X_\beta}$ are the minimum probabilities. Therefore, inversion of even or odd wires of the bus, i.e., (8,$\bar{4}$,2,$\bar{1}$) and ($\bar{8}$,4,$\bar{2}$,1), are the optimal *Inversion-Coding*s which exchange the $\downarrow\uparrow\downarrow\uparrow$ and $\uparrow\downarrow\uparrow\downarrow$ patterns with the $\downarrow\downarrow\downarrow\downarrow$ and $\uparrow\uparrow\uparrow\uparrow$.

Fig. 4.3 shows the exchange of probabilities using (8,$\bar{4}$,2,$\bar{1}$) *Inversion-Coding* for a exemplary positively correlated unsigned input signal. By applying this simple coding technique, we gain a reduction in the integer value of error in region R_0 by the factor of p_{X_α}/p_{X_β}. The achieved reduction is remarkable when $p_{X_\beta} \approx 0$ which is quite common in highly correlated signals.

For *Inversion-Coding* to be generally applicable for signed/unsigned positive/negative correlated integer values, we have proposed a logic circuit that converts every input signal correlation to an unsigned positive correlated signal.

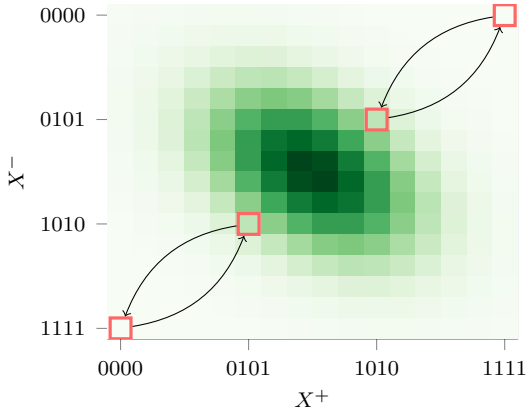
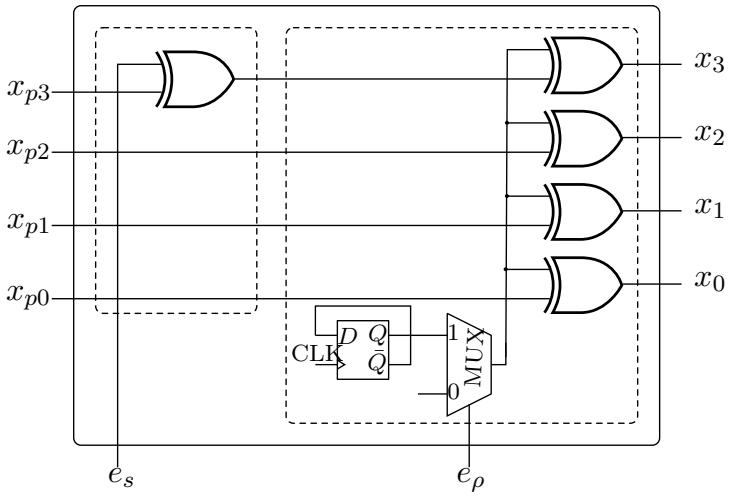


Fig. 4.3: The probability density plot of an illustrative randomly generated Gaussian distribution, $\boldsymbol{\mu}=\left(\begin{smallmatrix}7.5\\7.5\end{smallmatrix}\right)$ and $\boldsymbol{\Sigma}=8\left(\begin{smallmatrix}1 & 0.4\\0.4 & 1\end{smallmatrix}\right)$.



X_p distribution scenarios	e_s	e_ρ
Unsigned Positively correlated	0	0
Unsigned negatively correlated	0	1
Signed negatively correlated	1	0
Signed positively correlated	1	1

Fig. 4.4: The proposed reconfigurable architecture to convert unsigned negative, signed positive and negative correlated integer values to unsigned positive correlated integer values.

Fig. 4.4 shows the circuit structure for distribution conversion. The enable signals e_s and e_ρ are used to configure the circuit based on the primary input signal distribution, X_p. The enable signal e_s should set 1 when the input signal is signed; e_ρ should set 1 when the input signal is negatively correlated, and 0

otherwise. Therefore, any input distribution can be supported by the proposed coding scheme.

Fig. 4.5 illustrates the circuit structure of (8,$\bar{4}$,2,$\bar{1}$) *Inversion-Coding*. As shown in this figure, input signals are coded using two inverters and are transferred through the bus. Correspondingly at the destination, the received signals are decoded using two inverters.

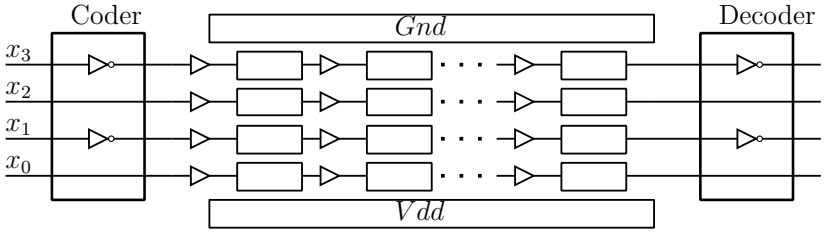


Fig. 4.5: Circuit structure of ($\bar{8}$-4-$\bar{2}$-1) Inversion coding.

Note that, the optimal coding for the correlated input signals however, is to use the optimal *Inversion-Coding* in combination with optimal *Swap-coding* in region R_0, i.e. (8,$\bar{1}$,2,$\bar{4}$). In this way, the overall gain is $4^{p_{X_\alpha}}/_{p_{X_\beta}}$.

Optimal combined Swap-Inversion Integer-Value encoding for given constraints In this subsection, we propose to combine the *Inversion-* and *Swap-Coding* schemes and introduce CIV coders. Selecting the best approach among the combined 4! different *Swap-Codings* with 2^4 different *Inversion-Codings* (which is 384 CIV coders), every possible input signal distribution can be encoded effectively. For the given constraints such as operating frequency, voltage swing, input signal characteristics, and bus specifications, there exists one optimal coder which can minimize the integer value deviation of the received signal[2].

In practice, the amount by which CIV coders can reduce the integer deviation depends on the distribution of transition values. In many applications, such as digital signal and image processing, input signals have a shared distribution characteristic. Therefore, an optimized CIV coder can reduce the integer value deviation of such input signals effectively.

Fig. 4.6 illustrates the probability distribution of transition values for a set of 13 8-bit grayscale images provided by the USC-SIPI image database [81]. Since we focus on 4-bit-width buses, the resulting distribution is shown for 4 LSBs and 4 Most Significant Bit (MSB) of the 8-bit images. Fig. 4.6(a) is the distribution

[2]Some architectures are universal as discussed experimental results later in Subsection 5.3.1

of transitions associated with 4 MSBs. It shows a highly correlated signal distribution that resembles the Gaussian distribution. Conversely, Fig. 4.6(b), which is a distribution of transitions associated with 4 LSBs, somewhat resembles the uniform distribution. For each of these distributions, there exists an optimal combined coder. Since the number of available coders is not so large, we can search among all combined coders to find the optimal coder for each distribution. We have developed a framework that characterizes the bus's error and determines the optimal coding using a brute-force search.

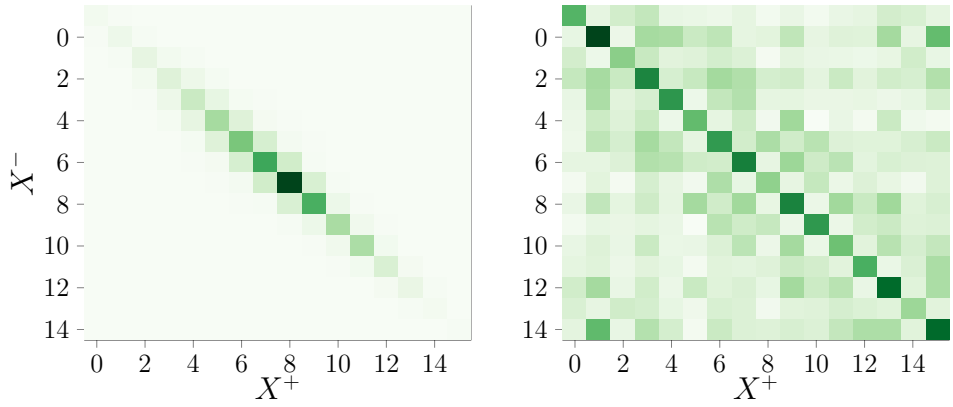


Fig. 4.6: Distribution of transitions for 4 MSB wires (a) and distribution of 4 LSB wires (b) for a set of 13 8-bit gray scale images from USC-SIPI image database [81].

4.2.2 Crosstalk-Avoidance-based Integer-Value encoding

Traditionally, CAC is used to eliminate certain undesirable data patterns and thereby reduces the delay in the exact communication. The memoryless coding approaches use a fixed codebook to generate codewords based on the current input data. Different types of the memoryless CACs such as FPF and FTF have been proposed which offer delay reductions ranging from 44% to 65.5% [12]. However, CACs are very sensitive to errors and result in a high magnitude of error when used for approximate communication.

The previous section has developed a coding scheme for approximate communication by focusing on the minimum coder/decoder overhead. In the case of less stringent area constraints, a memoryless CA-IV scheme can be employed to reduce the magnitude of the integer-value error in higher frequencies for

approximate communication.

The general scheme of the proposed coding technique for a 4-bit bus structure is shown in Fig. 4.7.

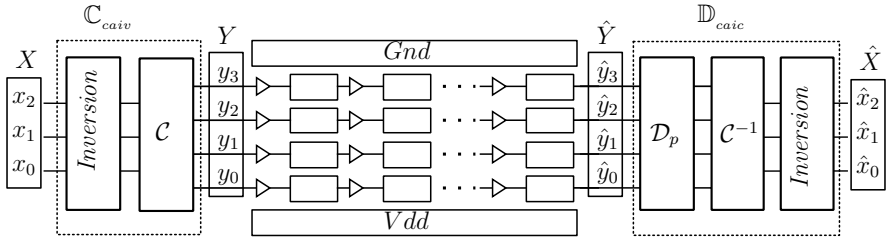


Fig. 4.7: The general scheme of the CA-based coding structure for 4 bit-width bus. The proposed scheme uses the combination of the inversion coding and an optimal 1-to-1 CAC mapping. Using $\mathcal{D}_p$ decoder further reduction of the error magnitude is possible.

As shown in this figure, given a 4-bit narrow bus structure, we employ a CA-IV coder, $\mathbb{C}_{CA-IV}$, to code a 3-bit input data, $X = x_2x_1x_0$. $\mathbb{C}_{CA-IV}$ consists of an *Inversion* coder and a CA-based coder, i.e. $\mathcal{C}$. Afterward, the received value, $\hat{Y}$, is decoded using a CA-IV decoder, $\mathbb{D}_{CA-IV}$. It consists of a primary decoding stage, i.e. $\mathcal{D}_p$, a CA-based decoder, i.e. $\mathcal{C}^{-1}$, and the *Inversion*. The primary decoding stage is used to map the received 4-bit values to a 4-bit data containing only valid codewords. Later on, $\mathcal{C}^{-1}$ maps the output data of the primary decoder to a 3-bit data; finally, the corresponding *Inversion* is used. It is worth mentioning that the typical group shielding topology of 4 wires allows coder/decoder design with smaller overhead.

$\mathcal{C}$ and $\mathcal{C}^{-1}$ optimal design Let S be the set of valid CAC codewords. Accordingly, for the set of all possible permutations of valid codewords combinations, there exists an optimal mapping such that the error magnitude is minimized. To achieve the optimum, the best mapping can be determined using a brute-force search. The coder, $\mathcal{C}$, and decoder, $\mathcal{C}^{-1}$, can then be realized using a synthesis tool. Of course, the optimal coder/decoder depends on the working frequency, the bus structure as well as the input data distribution. Considering a set of input data, e.g., images, an optimal mapping for the combinatorial distribution of a set can perfectly reduce the error magnitude for individual data in the set. This will be discussed further in experimental results.

$\mathcal{D}_p$ code design Let S_{opt} be the set of 2^3 (3-bit input data) 4-bit valid codewords for optimal mapping and Y_j^+ be a member of this set:

$$S_{opt} = \{Y_0^+, ..., Y_8^+\}.$$

As a result of variations (including noise), the receiver may receive different possible erroneous values by sending a codeword, Y^+, through the interconnect. The received value, $\hat{Y}^+$, may not necessarily result in a valid CAC codeword. Here, $p_{Y_j^+} = P[Y_j^+|\hat{Y}^+]$ is the probability of sending Y_j^+ when $\hat{Y}^+$ is received. To minimize the error magnitude, the primary decoder, $\mathcal{D}_p$, maps $\hat{Y}^+$ to Y_j^+ if $\max(\{p_{Y_0^+}, ..., p_{Y_8^+}\})$ is $p_{Y_j^+}$.

For an arbitrary 3C-FPF codeword set:

$$S_{opt} = \{0000, 0001, 0011, 0110, 0111, 1100, 1110, 1111\},$$

Table 4.1 shows $P[Y_j^+|\hat{Y}^+]$ for every Y^+ at 1.2 V and clock period of 0.55 ns and drive strength of ×16. Maximum probabilities are highlighted. According to this table, for example, when the receiver receives the value of 1100, i.e. $\hat{Y}^+ = 1100$, then the value which has been sent through the interconnect, Y^+, is: 0110 by the probability of 0.19, 1100 by the probability of 0.63, and 1110 by the probability of 0.18.

Table 4.1: Probability of the received values $\hat{Y}^+$ for every encoded value Y^+ at 1.2 V and clock period of 0.55 ns and drive strength of ×16. Highlighted probabilities show the optimal mapping between $\hat{Y}^+$ and Y^+.

		$\hat{Y}^+$													
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1010	1100	1101	1110	1111
Y^+	0000	0.63	0.18	-	-	-	-	-	-	-	-	-	-	-	-
	0001	0.37	0.57	-	0.15	-	-	-	-	-	-	-	-	-	-
	0011	-	0.25	1	0.54	-	1	0.14	0.16	-	-	-	-	-	-
	0110	-	-	-	0.16	-	-	0.3	0.24	-	-	0.19	-	0.15	0.09
	0111	-	-	-	0.15	-	-	0.14	0.36	-	-	-	-	0.15	0.18
	1100	-	-	-	-	1	-	0.14	-	1	1	0.63	1	0.22	-
	1110	-	-	-	-	-	-	0.14	0.12	-	-	0.18	-	0.33	0.28
	1111	-	-	-	-	-	-	0.14	0.12	-	-	-	-	0.15	0.45

Inversion-Coding As it is discussed earlier in Sec.4.2.1, *Inversion-Coding* uses the presence of spatial value locality in input data set. Therefore, the *Inversion-Coding* can be used to achieve an extra improvement for the CA-based coding technique. Please note that *Swap-Coding* is not applicable in the CA-based encoding context because it may distort the valid codewords for CA-based coding.

4.3 Stochastic Wave-Pipelining

The majority of the approximate communication techniques in literature are limited to compression. Despite their quite competent performance, they suffer from compressor/decompressor overheads. This section proposes a novel lightweight technique, SWP, based on the classical wave-pipelining.

Considering the restrictions of the classical wave-pipelining, in this paper, we revive the idea of wave-pipelining in the context of stochastic transmission. The proposed technique brings the considerable benefits of stochastic communication to the classical wave-pipelining while dramatically alleviating the classical wave-pipelined interconnects' timing constraints. Correspondingly, the receiver may receive erroneous values for some transition patterns depending on the application's error resiliency, as can been seen in Fig. 4.8.

4.3.1 Classical Wave-Pipelining

Ordinary pipelined systems break a long wire into shorter pipeline stages by inserting flip-flops to improve the performance of the interconnects [82]. Flip-flops should drive large loads in long interconnects, and therefore, they should be large. Then, they impose significant power/area penalties to interconnects. Even though the standard pipelining is an effective technique to increase the frequency and throughput rate of the global interconnects, its high overhead outweighs the advantages. In contrast, wave-pipelining does not need extra flip-flops and exhibits great potential as an alternative to ordinary pipelining.

For a wave-pipelined system to operate correctly, the output data should be sampled after the latest data has arrived at the outputs (worst-case) and before the earliest data from the next clock cycle (best-case) arrives at the outputs [83]. The sampling point, t_r, is defined as a time at which the output signals

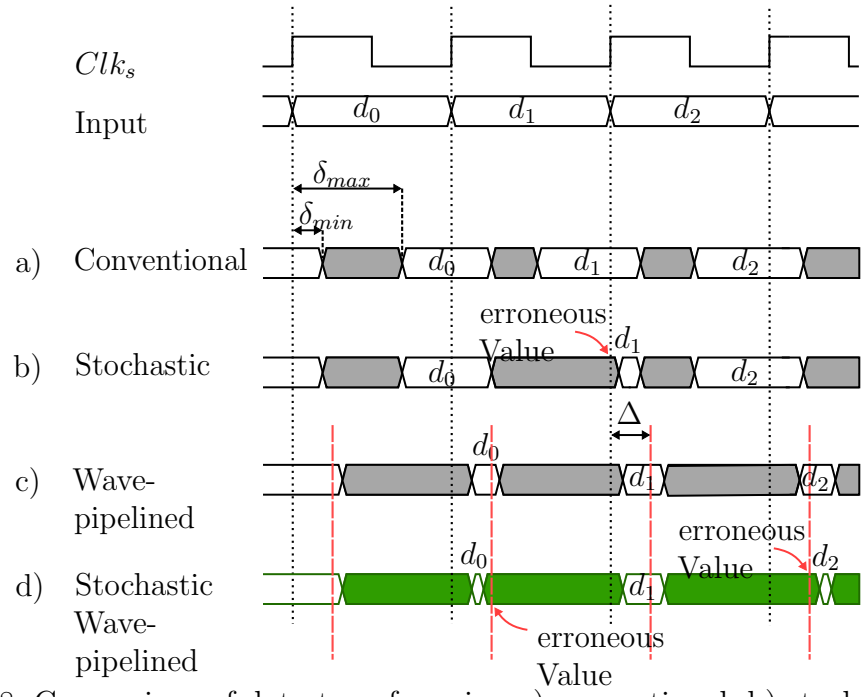


Fig. 4.8: Comparison of data transfer using a) conventional, b) stochastic, c) wave-pipelined and d) the proposed stochastic wave-pipelined approaches.

are sampled. Therefore, the simplified lower bound on t_r is given by:

$$t_r > \delta_{max}, \tag{4.8}$$

where δ_{max} is the worst-case delay among all possible transitions in a B-bit width interconnect. The upper bound of the t_r is constrained by the earliest arrival of the next clock cycle. As a result, the t_r is bounded above as follows:

$$t_r < T_{clk} + \delta_{min}, \tag{4.9}$$

where δ_{min} is the best-case (minimum) delay and T_{clk} is the clock period. Combining the lower and upper bound constraints, the simplified maximum rate pipelining condition is derived as follows:

$$T_{clk} > \delta_{max} - \delta_{min}. \tag{4.10}$$

Satisfying the above condition for error-free data transmission in the presence of noise and uncertainty is quite challenging. Correspondingly, it keeps

wave-pipelining from being a practically useful communication technique. An alternative approach can be utilizing the wave-pipelining for stochastic communication.

4.3.2 A Stochastic Solution: Stochastic Wave-Pipelining

Approximate computing provides a trade-off between the level of accuracy required by the applications and the achieved optimizations. In this regard, relaxing the constraints for designing a wave-pipelined on-chip interconnection to allow erroneous outputs to occur has a twofold benefit: First, it makes the selection of sampling time for the wave-pipelined communication structure more straightforward. Second, extra performance improvement can be achieved by combining wave-pipelining and stochastic/approximate techniques. Different techniques can be potentially combined with stochastic wave-pipelining to improve performance. For example, the approximate coding technique using Swap and Inversion of input signals [80] (introduced in the previous section) reduces the magnitude of the error and thereby enables the utilization of higher frequencies with smaller errors for a given application.

An illustrative example of performance improvement using the stochastic wave-pipelining is shown in Fig. 4.9. In this figure, the delay distribution for a 5-bit bus in segment 9 is illustrated. We have done simulations taking into account the noise; and further on, the standard pattern classification paradigm [74] is used, iC for $i \in 1, 2, 3, 4, 5$, to split transition patterns into different classes and group them into different colors. We have compared the minimum clock period for conventional, wave-pipelined, stochastic, and stochastic wave-pipelined communications. Using the conventional error-free data transmission, the clock period is slower than the worst-case delay, $T_{clk} > \delta_{max}$. However, using the wave-pipelining the clock period is $T_{clk} > \delta_{max} - \delta_{min}$, since it depends on both the worst-case and best-case propagation delays. Now let us consider an application that can produce acceptable results with errors. In this case, the stochastic communication can improve the performance by operating in clock periods $T_{clk} > \delta_{max_{stoch}}$ when it compares with conventional data transmission. $\delta_{max_{stoch}}$ is the maximum delay value which a specific application can tolerate. Finally, using stochastic wave-pipelining can provide an extra improvement on top. Assuming the maximum and minimum tolerable stochastic delays of $\delta_{max_{stoch}}$ and $\delta_{min_{stoch}}$, respectively, data transmission can be accomplished by

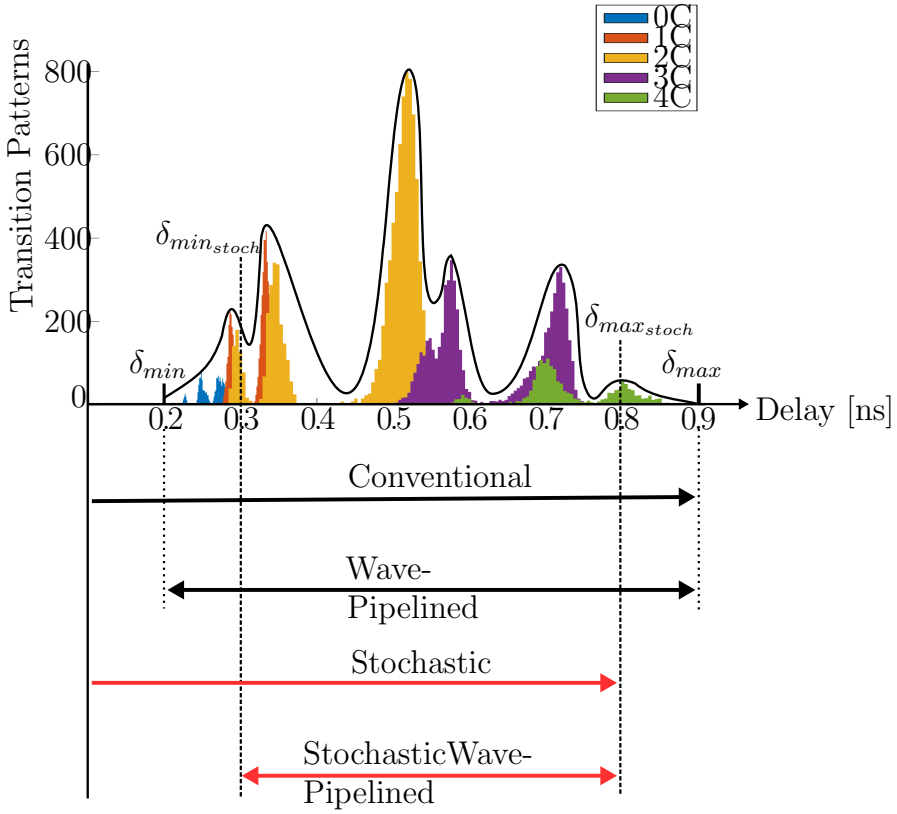


Fig. 4.9: An illustrative example of potential performance improvement using stochastic wave-pipelining. The minimum clock periods T_{clk} are shown for conventional, wave-pipelined, stochastic and stochastic wave-pipelined communications for an exemplary delay distribution of a 5-wire interconnect.

$T_{clk} > \delta_{max_{stoch}} - \delta_{min_{stoch}}$.

According to the exemplary data distribution in Fig. 4.9, using the conventional data transmission the minimum clock period is approximately 0.9 ns. On the other hand, using the wave-pipelining the minimum clock period can be reduced to approximately 0.7 ns which is 22% faster in comparison with conventional data transmission. Assuming $\delta_{max_{stoch}} = 0.8$ ns, the stochastic communication can improve the performance by 11% in comparison with conventional data transmission. Considering $\delta_{min_{stoch}} = 0.3$ ns, data transmission using stochastic wave-pipelining can be accomplished by $T_{clk} \approx 0.5$ ns which is 44% faster than the conventional data transmission.

4.4 Alternating Bit-Truncation

In this section, we propose an approximate communication technique inspired by concepts of bit truncation and shielding.

Bit truncation is a widespread technique, especially for image compression, due to its straightforward implementation [84]. Bit truncation is a typical example of the quantization process. One of the first contributions to the quantization theory is made by Bennett [85]. According to that, a uniform quantizer with the quantization step size, Δ, has the average square of error is $\approx {}^{\Delta^2}/_{12}$. Based on this formula, for a bus with the number of truncated LSBs of K_T, MSE can be calculated as follows:

$$MSE = \frac{(2^{K_T})^2}{12} \tag{4.11}$$

In the figure below, we compare the $log_2(MSE)$ for the different number of truncated LSBs, K_T, using the Eq. (4.11) and the simulation. For the simulation, we transmit a Lena image through a bus using a nominal voltage and frequency. On the receiver side, the truncated bits are approximated by median value based on the K_T. The details regarding the simulation setup can be found in the next chapter.

According to Fig. 4.10, the error increases linearly with the number of truncated LSBs, and the formula can estimate the error with high precision. Please note that no timing error due to the voltage or frequency overscaling is considered. This technique is only applicable for applications that can tolerate the minimum inherent error of truncation calculated by Eq. (4.11).

Images show a general property of inter-pixel correlation in which the least significant bits do not contribute significantly to the image quality. In this regard, bit truncation can be utilized for approximate communication within two frameworks: First, ignore a certain number of LSBs and remove the associated wires to increase the area and energy efficiency for an increased error. Second, setting a certain number of LSBs to zero without removing the associated physical wires. In this case, the truncated wires can be used as virtual shields to gain the energy and performance improvements within an acceptable error bound.

Traditionally, to cope with increasing interconnects delay and control the crosstalk noise, physical shielding is used [86]. Physical shielding inserts

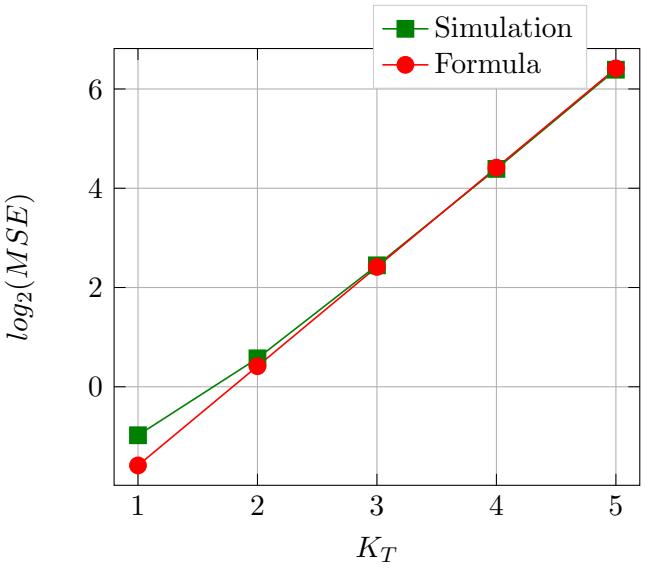


Fig. 4.10: $log_2(MSE)$ versus number of truncated LSBs, K_T, using simulation and formula. The results of simulation are obtained for Lena image across a bus. More details regarding the simulation setup are provided in chapter 5.

grounded wires between adjacent signal wires to eliminate coupling. Even though shielding is a widely used effective technique, it imposes an undesirable area overhead on the overall system.

In this section, we propose an Alternating Bit-Truncation (ABT) technique. ABT performs truncation by setting a predefined number of LSBs to zero. The zero-valued LSBs have twofold benefits: (1) They do not contribute to any logic transition and therefore reduce the power dissipation significantly. (2) The LSB wires can function as shielding lines without imposing extra overhead, therefore, reduce the effect of coupling in remaining active wires.

Depending on the bus width and the tolerable error threshold of an application, the designer chooses the number of truncated LSBs. This technique then changes the assignment of signals to the wires to leverage the physical properties of the CMOS interconnect to reduce the integer-value error and provide virtual shields between neighboring wires. A careful swap of the wires can drastically improve data transmission in comparison to the naive truncation method.

Fig. 4.11 shows an example of using ABT technique for an 8-bit input signal. According to this figure, 3 LSBs of the input data, X, is truncated and set

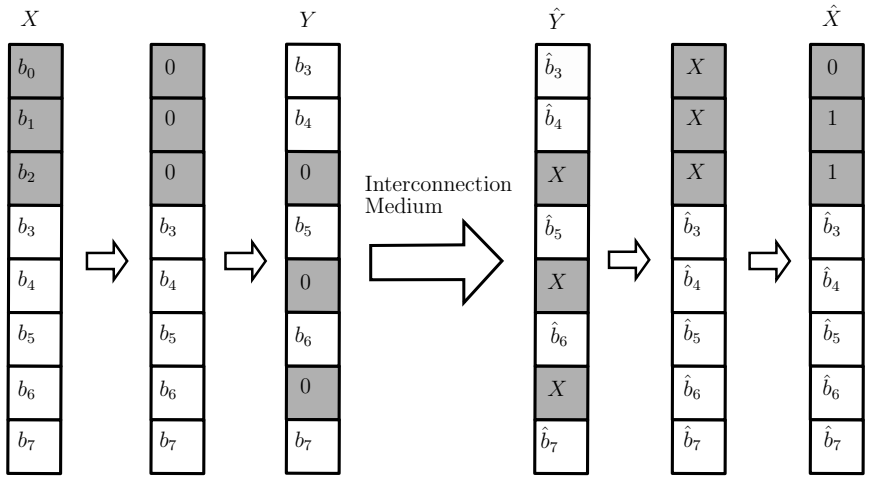


Fig. 4.11: An example of Alternative Truncation technique for an 8-bit bus.

to zero. These grounded wires are used as virtual shields between the other signal lines. In the final stage of the coding process, the wires are swapped to minimize the coupling effect. The truncated wires' values can be approximated as the median integer value of the 3-bit truncated wires on the receiver side. We choose the integer value of 3 as the median value (the value at the center of possible integer values.). In this example, opposite worst-case transitions can only happen between b_3 and b_4 signals that can drastically reduce the crosstalk noise and subsequently delay without the adverse effect on the area. In addition to the energy and performance benefits of eliminating the worst-case transitions, power saving can be achieved due to the inactive lines. However, the advantages mentioned above come with accuracy loss, which is tolerable for many applications.

4.5 Conclusion

The approximate computing paradigm provides an unprecedented opportunity for optimizing power consumption, performance, and area of error-resilient circuits and applications. Until recently, the domain of approximate communication is mainly overlooked. The considerable potential of approximate communication in overall system energy and performance improvement is neglected. In this chapter, we proposed various techniques for approximate communication.

First, we presented two classes of encoding techniques to reduce the integer-

value error of the transmitted signals. (1) Targeting the area-constrained applications, we propose an integer-value coding based on the Swap and Inversion of the input signals. The optimal combination of Swap and inversion coding is found using the proposed exhaustive search framework. (2) Intended for applications with looser area constraints, we also propose a CA-based coding. The optimal mapping of the data words and codewords leads to the minimum error magnitude. We use a primary decoder to map non-codeword received values to the associated valid codewords. Moreover, we employ the inversion of the selective signals to leverage the input data's spatial value locality.

Second, we have presented a novel stochastic on-chip communication approach combining the classical wave-pipelining and stochastic/approximate communication. We have shown that stochastic wave-pipelining can provide two specific benefits: (1) It relaxes the stringent constraints of the classical wave-pipelining, which in some cases prevents the effective practical use of classical wave pipelining. (2) It provides an additional improvement in the performance of the on-chip interconnection depending on the error-tolerance of the applications. SWP, unlike lossy compression and approximate coding approaches, does not change the input signals' characteristics; thus, it can be used as a complementary technique along with other techniques.

Finally, we developed an effective bit-truncation technique. Our proposed technique sets the predefined number of LSBs to zero. The truncated lines have twofold benefits: (1) The inactive lines do not contribute to switching energy consumption, and consequently, drastic energy consumption saving is expected. (2) Truncated lines can be used as virtual shield lines and reduce the crosstalk noise, which can improve the energy and timing as well. A careful swap of the signals leverages the physical properties of the CMOS interconnects and can lead to integer-value reduction as in *Swap-Coding*.

A thorough evaluation of the proposed techniques is carried out in the next chapter.

CHAPTER 5

Evaluation of Approximate Communication Techniques

Contents

5.1 Introduction

In the previous chapter, we proposed multiple approximate/stochastic communication techniques. In this chapter, we quantify the benefits of proposed techniques through comprehensive simulations. The experiments are carried out using synthetic data and sending images. We validate the proposed methods using different case studies such as Sobel edge detection, Optical Character Recognizer (OCR) and motion estimation algorithms for real-world scenarios.

The remainder of this chapter is organized as follows: First, we introduce the experimental environment. Second, we empirically show the effectiveness

of various Integer-Value coders (IVCs) in the value deviation reduction of data communication. We thoroughly assess the effectiveness of the CIV coding technique as a light-weight memoryless coder using different synthetic and real data scenarios. Then, we evaluate CA-IV coding as an alternative to CIV. We also evaluate the proposed IVCs in real scenarios. Third, we validate SWP using synthetic and real data as well as a case study, OCR, in a similar fashion as IVC validation. Finally, we assess ABT technique by comparing it with *Conventional* data transmission, CIV, and the naive bit truncation approach.

5.2 Experimental Environment

The circuit structure in Fig. 3.2 is used for the experiments. A 3 mm 8-bit-width global interconnect is divided into 8 segments and each segment is driven using a CMOS driver. The groups of 4 wires are shielded by paralleled VDD and GND wires. Thus, each 4-bit input signal is coded and decoded separately. The experiments are carried out on an interconnect in metal layer 4 with the width and spacing of 0.15 μm. As mentioned earlier, the selection of the wire width, spacing, and layer usage is usually made by the designers in order to trade off delay, bandwidth, energy, and noise. We have chosen the standard values. The driver, which is an inverter, has the drive strength of α times of the minimum sized inverter (e.g., a $\times 6$ driver uses an n-channel transistor width which is six times minimum channel width and a p-channel transistor width, which is 12 times minimum channel width).

The simulation results are obtained using a commercial 65 nm technology node with the supply voltage ranging from default, 1.2 V, to low-power, 0.8 V, and $\times 6$, $\times 16$ and $\times 32$ sized drivers. The SPICE-level simulation is carried out using *Cadence Spectre* circuit simulator. To account for noise effects, we run *transient-noise* simulation for 100 noise-runs and different clock frequencies. The signals are expected to traverse through the interconnect in one clock cycle [1]. The results are reported for the 8^{th} segment of the interconnect.

[1] We refer to error as the timing error in the presence of transient noise when transmitting signals in one clock cycle through the bus structure in Fig. 3.2.

5.3 Integer-Value Coding Validation

5.3.1 CIV Efficiency Evaluation

Experimental evaluation using synthetic data

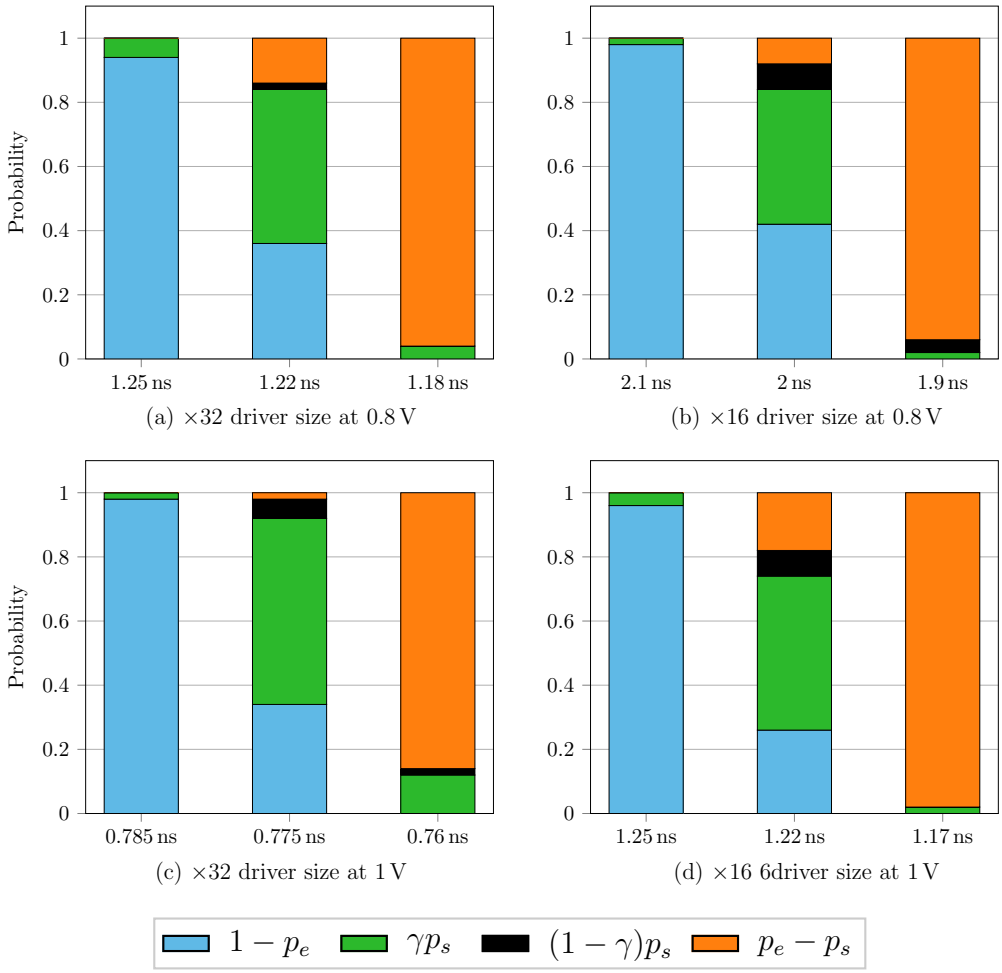


Fig. 5.1: The probability of the received values for worst-case transition of ↓↑↓↑ at 0.8 V and 1 V supply voltages and ×16 and ×32 driver sizes in timing region R_0. $1-p_e$, γp_s, $(1-\gamma)p_s$ and p_e-p_s are the probability of the exact (error free) received values and inexact received values when a single wire, wire 1, wire 2 and both wires 1 and 2 simultaneously are erroneous, respectively.

The proposed encoding scheme is addressed in timing region R_0 based on two assumptions: First, two middle wires of worst-case transitions of ↓↑↓↑ and ↑↓↑↓ limit the performance of the interconnect, and second, the two middle wires do

not necessarily violate the timing of the bus simultaneously. To validate the proposed arguments, Fig. 5.1 illustrates the probability of the received values for ↓↑↓↑ transition (10 to 5 transition) for 0.8 V and 1 V supply voltages, and ×16 and ×32 driver sizes in timing region R_0.

According to the results, the receiver receives either the exact value (in blue) or the erroneous values, which error occurs in one of the wires of 1 (in green) and 2 (in black) or both of them (in orange). For example, let us consider Fig. 5.1(c) at a clock period of 0.785 ns, when the bus starts to produce the error. According to the results, the probability of happening error is minimal. Consequently, the receiver either receives the exact integer value 5 or the erroneous value of 1 (error occurs in one of the middle wires, γp_s). With a smaller clock period (higher frequency), the timing error can occur in both middle wires with a probability of $p_e - p_s$, as well as one of the middle wires with the probability of p_s. Therefore, the receiver may receive 5, 1, or 7. Finally, at 0.76 ns, the probability of receiving the exact value is zero, and the receiver may receive the erroneous values of 1, 7, or 3 with different probabilities. The systematic changes in the received values by clock period can be observed for different design choices. This systematic behavior guarantees the efficiency of the proposed coding techniques regardless of the design choices. Note that the large value of γp_s or $(1-\gamma)p_s$ in each case is due to the different rising and falling delays and the following misalignment effect. Furthermore, it can be noticed that the error distribution, e.g. at 1.22 ns Fig. 5.1(a), cannot be explained using the traditional model; intrinsic misalignment of signals is responsible for that.

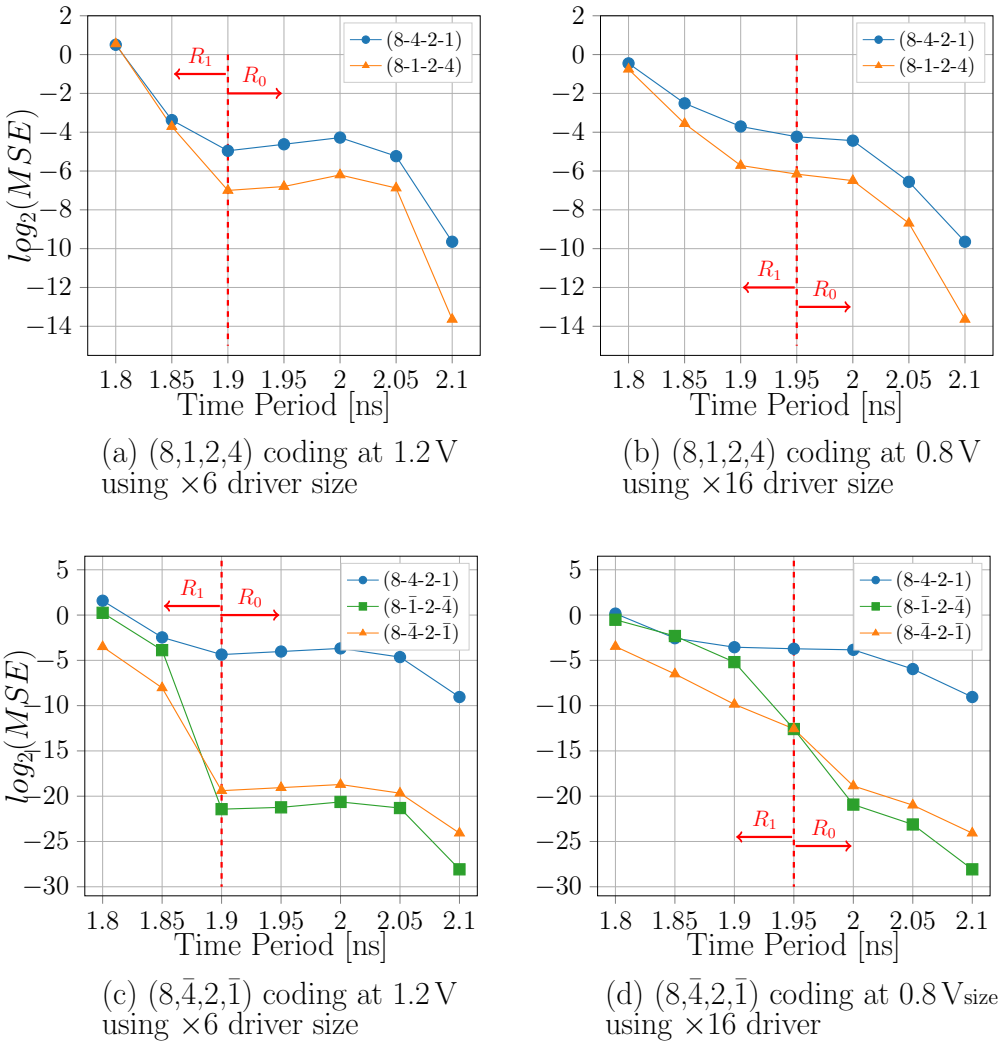
We evaluate the proposed *Swap-Coding* and *Inversion-Coding* across all possible transitions for a 4-bit-width bus for uniform and correlated input signals.

Uniform distribution: We compare the conventional no-coding transfer of the signals with (8,1,2,4) *Swap-Coding*. As is already mentioned in previous sections, we use (8,4,2,1) nomenclature to refer to the conventional transfer of the signals with no coding. Note that the proposed coders have been developed for the timing region R_0 and are optimal for this region. Thus, we focus on the analysis of the results in this timing region. The dashed lines in these figures separate the timing regions R_0 and R_1 approximately. Fig. 5.2(a) and

(b) show the mean square of error in logarithmic scale, $log_2(MSE)$, vs. clock period for different supply voltages and driver sizes. It is expected that the proposed coders maintain their effectiveness regardless of the design choices. According to the results in Fig. 5.2(a), (8,1,2,4) *Swap-Coding* can reduce the MSE by factor of $2^{1.65}$ in clock period 2.05 ns and 2^4 in clock period 2.1 ns when compared with (8,4,2,1). Similarly, at 0.8 V, the proposed coder can reduce the MSE by factor of $2^{2.13}$ in clock period 2.05 ns in Fig. 5.2(b). Please note that MSE changes with the clock period, because as could be expected, the number of errors and the magnitude of the error are increased by faster data transmission.

Gaussian distribution: We evaluate the proposed *Inversion-Coding* and the combined (8,$\bar{1}$,2,$\bar{4}$) coding across all possible transitions for a 4-bit-width bus. Fig. 5.2(c) and (d) illustrate the comparison for a randomly generated Gaussian distributed input signal. We use the sample randomly generated Gaussian distribution in Fig. 4.3 as an input to evaluate the proposed *Inversion-Coding*. According to the results in Fig. 5.2(c) a drastic reduction of the MSE using the proposed (8,$\bar{4}$,2,$\bar{1}$) *Inversion-Coding* and CIV coding of (8,$\bar{1}$,2,$\bar{4}$) at typical voltage swing of 1.2 V can be achieved. In region R_0, the optimal (8,$\bar{1}$,2,$\bar{4}$) coding, outperforms other transmission methods. For example, in 2.1 ns data transfer can be accomplished with the MSE of about 2^{-9}, 2^{-24} and 2^{-28} using (8,4,2,1), (8,$\bar{4}$,2,$\bar{1}$) and (8,$\bar{1}$,2,$\bar{4}$), respectively. Restricted to a given constraint, e.g., $MSE < 2^{-5}$, the data transmission can be accelerated by about 13%, operating in clock period of 1.85 ns, using the *Inversion-Coding* instead of conventional transmission of the signals in 2.1 ns in Fig. 5.2(c). Obviously, the extent of the improvement depends on the applications constraints and probability distribution of the transmitted data. The results at low-power voltage swing of 0.8 V for different clock periods using $\times$16 driver size in Fig. 5.2(d) shows the superiority of the proposed techniques in comparison with conventional data transmission.

Since the proposed coders are developed considering the worst-case transitions, the extent of the improvement is lower in the case of smaller clock periods. Other transitions start to produce the timing errors in higher frequencies, and therefore an optimal CIV coder based on the working frequency should be employed.



(a) (8,1,2,4) coding at 1.2 V using ×6 driver size

(b) (8,1,2,4) coding at 0.8 V using ×16 driver size

(c) (8,4̄,2,1̄) coding at 1.2 V using ×6 driver size

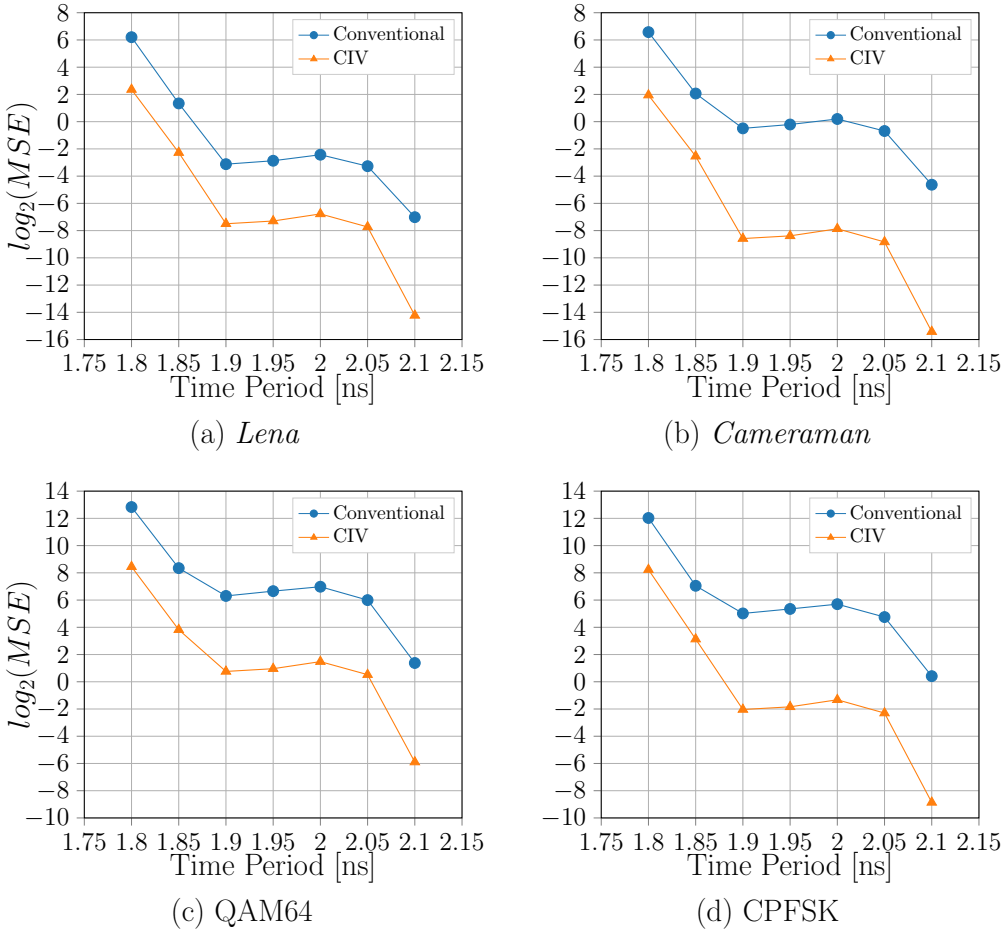
(d) (8,4̄,2,1̄) coding at 0.8 V_{size} using ×16 driver

Fig. 5.2: $\log_2(MSE)$ versus clock period for conventional data transmission, (8,4,2,1), and (8,1,2,4) *Swap-Coding*, (8,4̄,2,1̄) *Inversion-Coding* and the combined coding of (8,1̄,2,4̄). The results are obtained across all possible values for a 4 bit-width bus with uniformly distributed input signals (a) and (b) and a sample Gaussian distribution (c) and (d).

Experimental evaluation sending real-world data

To evaluate the proposed coding techniques, we transfer two types of data through the 8-bit bus: images and sampled radio communication signals with different modulations. Each input data splits up into groups of 4-bit data. Each 4-bit data is coded using an optimal CIV encoder. The optimal coder is found employing our proposed framework based on the given constraints.

Fig. 5.3 illustrates $log_2(MSE)$ versus the clock period for two 8-bit images, *Lena* and *Cameraman* from scikit-image database [87] and two 16-bit radio communication signals with two widely used modulations, CPFSK and QAM64 [88]. The results are obtained using ×6 driver and 1.2 V supply voltage. According to the results in Fig. 5.3(a)-(b), our proposed coding scheme improves



(a) *Lena* (b) *Cameraman*

(c) QAM64 (d) CPFSK

Fig. 5.3: $log_2(MSE)$ versus clock period for conventional data transmission and optimal CIV coder for two 8-bit images (*Lena* and *Cameraman*) and and two 16-bit radio communication signals with QAM64 and CPFSK modulations.

the quality of the received images regardless of the input image. For example, the proposed CIV encoder reduces the MSE by factor of $2^{4.4}$ and 2^8 for a period of 1.9 ns for *Lena* and *Cameraman*, respectively. The extent of the accuracy improvement depends on the probability distribution of the input images. The CIV coder can effectively code the input signals in timing region R_1 as well as R_0.

The effectiveness of the CIV coding in the quality improvement of the non-image workloads, as well as higher bit-width data, is evaluated in Fig. 5.3(c) and (d). A similar quality improvement is observed as they compare with Fig. 5.3(a) and (b). For example, the proposed CIV encoder reduces the MSE by factor of $2^{5.5}$ and 2^{7} for a period of 1.9 ns for QAM64 and CPFSK modulations, respectively.

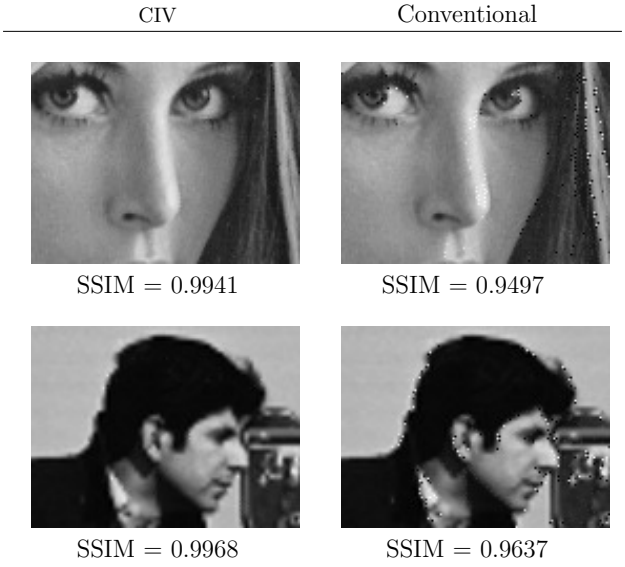


Fig. 5.4: Comparison of the received images quality for CIV and Conventional transmission of signals. The results are obtained at 1.2 V in 1.8 ns.

We compare the quality of the received images using CIV coding and the conventional transmission of signals at 1.2 V supply voltage and in 1.8 ns clock period. A 96×128 fraction of the original images is transferred through the bus. We use the Structural Similarity Index (SSIM) for measuring the image quality. The SSIM is a well-establish metric to quantify the visibility of errors between the distorted image and reference image. This metric is a complement to the traditional metrics like MSE. Fig. 5.4 illustrates the received images and the corresponding SSIMs. As it is shown the quality of the received CIV encoding images is remarkably higher than the received images using conventional data transmission.

Analysis of CIV for energy-constrained applications

In applications where the main concern is the energy-saving, supply voltage scaling is often used. The proposed coding scheme should maintain its efficiency

for the low-power application using voltage scaling and different driver sizes. Fig. 5.5 illustrates the $log_2(MSE)$ versus clock period for *Lena* image at 0.8 V and 1 V supply voltages using ×16 and ×32 drivers.

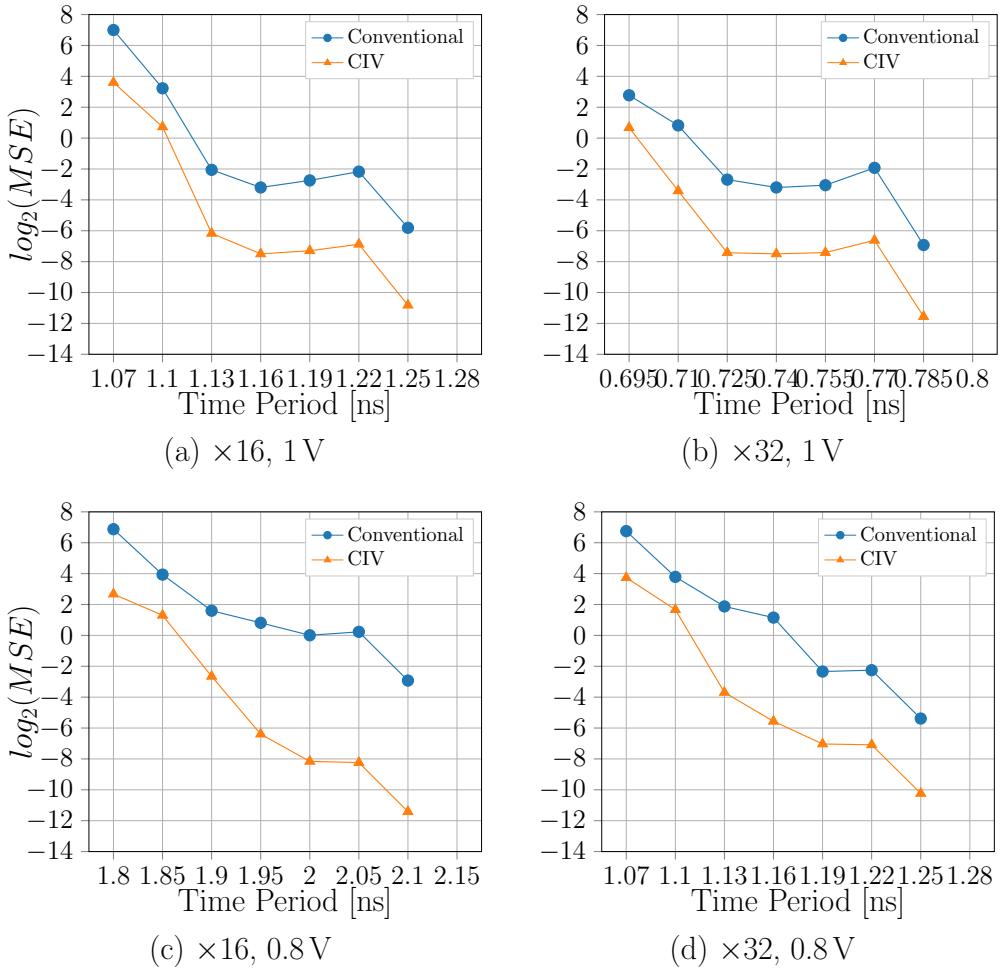


Fig. 5.5: $log_2(MSE)$ of a 8-bit *Lena* transferred through a 8 bit-width bus versus clock period, for voltage swings of 0.8 V and 1 V using ×16, ×32 drivers. Comparison is made between (8-4-2-1), i.e., no coding, and CIV coding.

According to this figure, the proposed encoding technique is able to reduce the error for different design choices. Let us consider an example of a specific application that can tolerate the $MSE < 2^{-6}$ while transferring the *Lena* image through the bus with ×16 drivers. According to the results in Fig. 5.5, the designer can trade the energy and/or performance for the tolerable deviation. Thus, one of the following alternatives using CIV coding can be chosen: (i) Clock period of 1.13 ns at voltage swing of 1 V, (ii) Clock period of 1.95 ns at voltage

swing of 0.8 V. Considering the given constraint, the choices in which the bus can operate using the traditional approach is as follows: (i) Clock period 1.28 ns at voltage swing of 1 V, (ii) Clock period 2.15 ns at voltage swing of 0.8 V. Depending on the efficiency goal of the design, an alternative among the aforementioned choices can be picked. For example, using the voltage swing of 1 V, CIV encoding is 11.7% faster by operating in clock period of 1.13 ns, in comparison with (8,4,2,1), i.e., no coding, operating at 1.28 ns. Similarly, using 0.8 V voltage swing, CIV encoding is 9.3% faster by operating in a clock period of 1.95 ns, in comparison with (8,4,2,1) operating at 2.15 ns. In addition, using a voltage swing of 0.8 V results in 36% energy consumption reduction in comparison with a 1 V voltage swing.

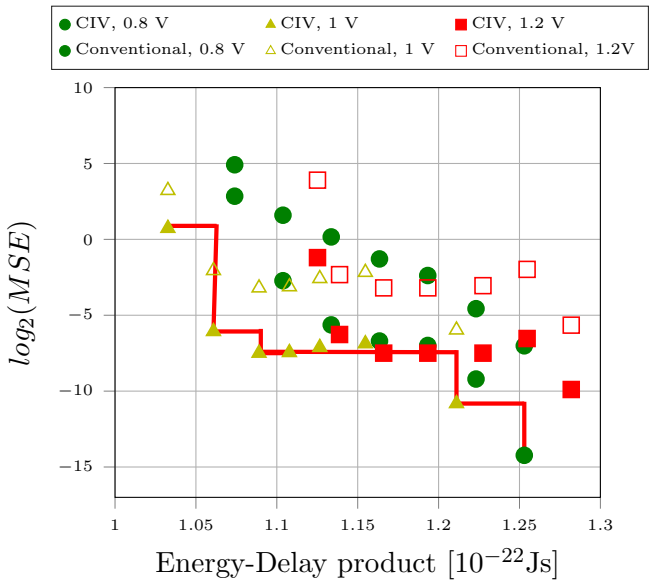


Fig. 5.6: $log_2(MSE)$ versus the energy-delay product for proposed CIV coding and with no coding in three different voltage swings for ×16 driver strength transmuting the *Lena* image. The line shows the pareto-optimal.

To find a suitable trade-off between the energy savings and the performance degradation, Energy-Delay Product (EDP) should be studied. Fig. 5.6 shows $log_2(MSE)$ versus the EDP at different voltage swings for *lena* image for ×16 driver size. The proposed coding technique preserves its improvement for different voltage swings. For the desired EDP and the target application's quality, the design can be configured to work in a certain voltage swing. The filled and empty marks correspond to CIV and conventional data transmission,

respectively. The line shows the Pareto-set. For example, a specific application which is restricted to $MSE < 2^{-10}$ can solely operate using CIV coder at either 1 V or 0.8 V supply voltages.

CIV universal coder evaluation

Based on the Integer-Value encoding framework discussed in Section 4.2.1, for the given constraints, such as supply voltage, frequency, and the input signal probability distribution, an optimal CIV encoder. The dependence of the optimal encoder in the input signal characteristics is undesirable. Thus, in order to assess the robustness of the proposed encoding technique, we study to employ a universal CIV encoder to code two different random images (*Couple* and *Aerial*) among a set of 13 8-bit grayscale images provided by USC-SIPI image database [81]. Fig. 4.6 illustrates the probability distribution of the transition values for this set of images. Using the proposed framework, we find an optimal CIV coder for each of the 4-bit input signal distributions of Fig. 4.6 (optimal universal coders). We apply the universal coders to two random images from the image set. Then, we compare the quality of the received images for the universal coder and optimal CIV coder for each image.

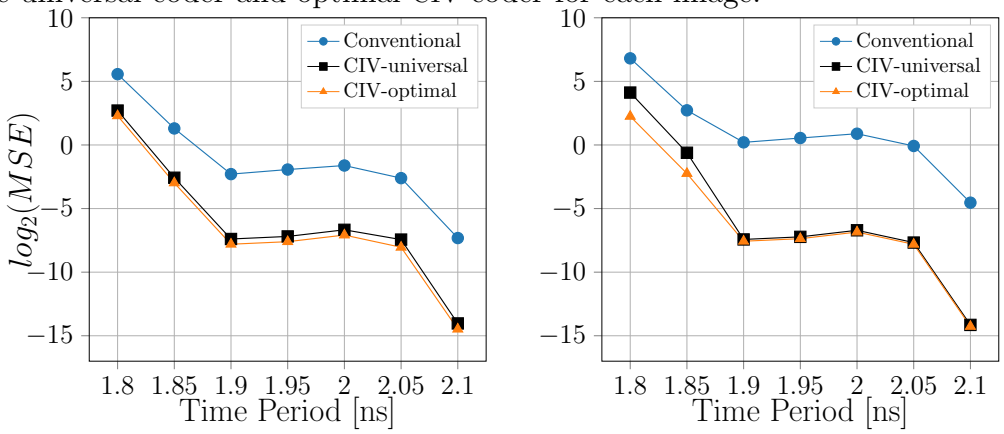


Fig. 5.7: $log_2(MSE)$ versus clock period for 3 different images from a set of 13 8-bit grayscale images provided by USC-SIPI image database [81]: *Couple* (a), *Aerial* (b) at 1.2 V and using ×6 driver size.

Fig. 5.7 illustrates the $log_2(MSE)$ versus the clock period for *Couple* and *Aerial* images at 1.2 V and using ×6 driver size. The results show a very small discrepancy between the universal coders in comparison with the optimal coders

for each image, which is negligible in region R_0. For example, in clock period 1.9 ns, using the universal coders (4 MSBs: ($\bar{4}$,1,2,8) and 4 LSBs: (4,1,2,8)) and CIV-optimal coders for each image result in MSEs which differ only by factor of $2^{0.41}$, $2^{0.14}$, $2^{0.01}$ and 2^0 for *Couple* and *Aerial* images, respectively. The results show the robustness of the proposed coding approach to variation in the Probability Density Function (PDF) of the input signals.

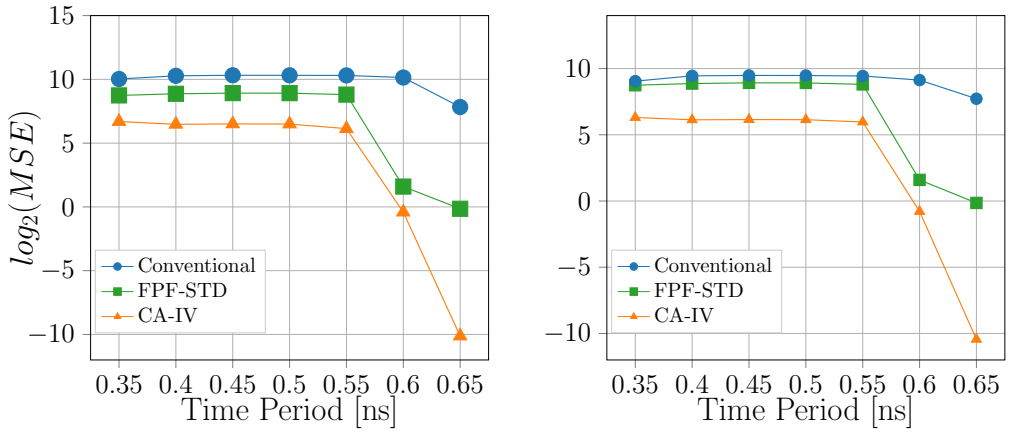
5.3.2 CA-IV Efficiency Evaluation

In this subsection, we evaluate the proposed CA-IV by sending images. In this experiment, we develop the CA-IV coding using the FPF valid codewords. We compare CA-IV with the conventional transmission of signals as well as the area-optimized forbidden pattern free coder based on Fibonacci numbering [12]. We refer to this FPF mapping as FPF-STD. The codebook for FPF-STD is as follows:

Input Data Word (X)	Optimal Codeword (Y)
000	0000
001	0001
010	0011
011	0110
100	0111
101	1100
110	1110
111	1111

We transfer 8-bit *Lena* and *Cameraman* images through the interconnect. The input data splits up to groups of 3-bit data (of course the last group is only 2-bit data). Each group is coded separately using the optimal CA-IV coder. The simulation is carried out at 1.2 V supply voltage using a $\times 16$ sized driver.

Fig. 5.8 illustrates $log2(MSE)$ versus clock period using CA-IV, FPF-STD and Conventional techniques. According to this figure, the proposed CA-IV coding outperforms Conventional and FPF-STD techniques. For example, in Fig. 5.8(a) and in a clock period of 0.65 ns, CA-IV can improve the data transmission accuracy by about 2^{18} and 2^{10} times, respectively, as compared with conventional and FPF-STD techniques. Similarly, the CA-IV coding can improve the MSE in the range of $2^{2.4}$ to 2^{10} for different clock periods compared with FPF-STD when sending Cameraman through the interconnect.



(a) *Lena* (b) *Cameraman*

Fig. 5.8: $log_2(MSE)$ versus clock period for *Lena* and *Cameraman* images at 1.2 V and using ×16 driver size. CA-IV is compared with the Conventional and FPF-STD techniques.

To ensure the comprehensive analysis of the proposed coding technique for approximate communication, we send the cameraman and *Lena* images (a 96×128 fraction of the original images is used) through the interconnect and compare the quality of the received images. To compare the resultant images, we use the structural similarity index (SSIM). Fig. 5.9 illustrates the received images using different techniques and the associated SSIM. As shown, the quality of the received images using the proposed coding is much higher than the other techniques. For example, SSIM is about 10% and 40% higher, respectively, when using the CA-IV coder compared with the FPF-STD for *Cameraman* and *Lena*.

CA-IV universal coder evaluation

The optimal CA-IV encoder may not be the same for the given constraints, such as supply voltage, frequency, and the input signal probability distribution. As discussed earlier, the optimal encoder dependency, particularly in the input signals characteristics, is unfavorable. A universal coder should be able to code the set of input data effectively. A universal coder for the combination of 13 image distributions from the USC-SIPI image database [81] is used. We compare the results using the universal coder, CA-IV-universal, and the optimal coders, CA-IV-optimum, for *Couple* and *Aerial* images from the dataset.

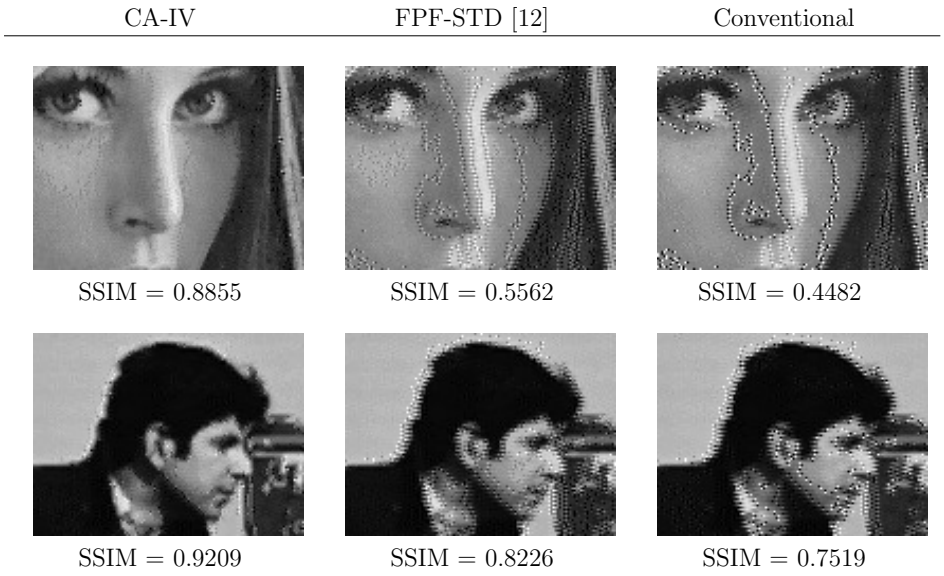


Fig. 5.9: Comparison of a received Lena and Cameraman image at 1.2 V and using ×16 deriver size. The image is transferred using CA-IV encoding and with no coding as well as FPF-STD. The results are tabulated in clock period of 0.55 ns.

Fig. 5.10 shows the results for different clock periods. According to the results, there is a negligible discrepancy between the CA-IV-universal coder and the CA-IV-optimum. The discrepancy is, at most, no higher than $2^{0.02}$ at 0.6 ns of *couple* image.

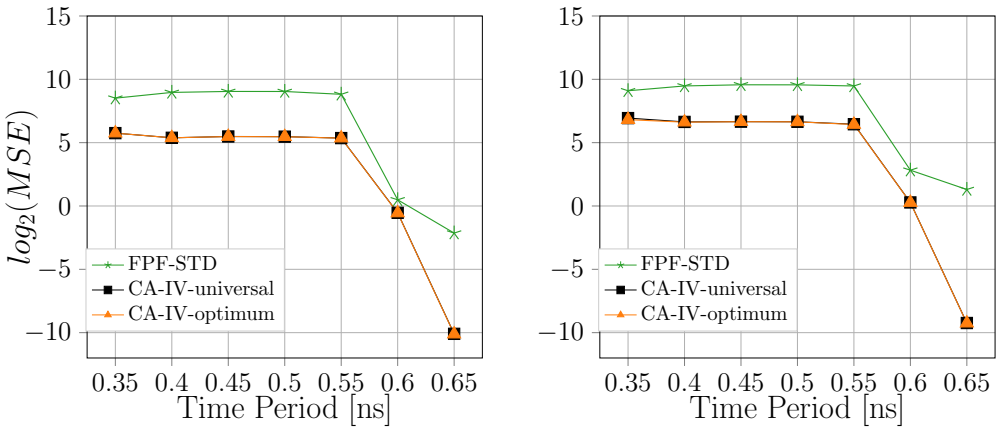


Fig. 5.10: $log_2(MSE)$ versus clock period for 3 different images from a set of 13 8-bit gray scale images provided by USC-SIPI image database [81]: *Couple* (a) and *Aerial* (b) at 1.2 V and using ×16 driver size.

5.3.3 Case-Study

Motion estimation application

Egomotion estimation is a widely used technique that holds great significance for computer vision applications, robotics, augmented reality and visual simultaneous localization and mapping. In this case study, Scale-Invariant Feature Transform (SIFT)-based *Egomotion estimation* has been implemented as a reference application for interpreting scenes based on the extraction of distinctive image features.

The purpose of SIFT is to detect distinctive image keypoints and to generate a unique description, which can be used to identify features and objects in images independent of their size and orientation, i.e., scale- and rotation-invariant. The SIFT algorithm creates a scale space of the original image, which is the consecutive Gaussian convolution of the original image with increasing standard deviation and downsampling of the resulting images. After the scale space is constructed, a Difference of Gaussian (DoG) is formed by subtracting adjacent images. It determines the approximate location and scale of the keypoints. In the next step, the algorithm refines the keypoints' location and assigns orientation to them. Finally, keypoints descriptor as a final representation of the image is generated. Using this representation, the features can be identified, and keypoints of different images can be matched. A detailed description of the SIFT algorithm can be found in [89].

In the next step, the keypoints are used by the *Egomotion estimation*. Egomotion provides a method to reconstruct the 3D scene and measure the movement of a stereo camera system from a sequence of images [90]. This algorithm is based on keypoint and feature matching. Brute-Force Matching is the basic *keypoint matching* method to find similar keypoints in different images. The feature descriptors extracted by SIFT from two images are compared one by one. Because the descriptor is rotation-invariant and normalized, further processing is not needed. The distance of the two features indicates the similarity between the two keypoints. Two feature vectors with the smallest distance are accepted as the most similar. To reduce false matchings, a position check is used to filter out impossible keypoint pairs, whose vertical differences are too large to be reasonable for a stereo camera scene.

The goal of the case study is to assess the applicability of the proposed coding

scheme in a real application. We transmit images of the stereo camera system from the KITTI dataset [91] to a horizontal Single instruction, multiple data (SIMD) vector processor to execute the *Egomotion estimation*. To decouple the performance of the SIMD coprocessor from the main Microprocessor without Interlocked Pipelined Stages (MIPS) CPU, the coprocessor resides in an own clock domain. The images are transferred at a supply voltage of 1.2 V and using: (i) CIV coding technique at clock periods of 1.7 ns, 1.8 ns and 1.9 ns, (ii) Conventional data transmission (No-Code) at clock periods of 1.7 ns, 1.8 ns and 1.9 ns, (iii) and Exact/Ideal data transmission at clock period of 2.2 ns. After applying the Egomotion estimation on the received images, we compare the results using the quantitative quality metrics of the *rotation angle mean absolute error*, R_{MAE}, and the *translation velocity mean relative error*, T_{MRE}:

$$R_{MAE} = \frac{1}{N}\sum_{i=1}^{N} |\vec{r}_{reference}(i) - \vec{r}_{estimated}(i)|, \tag{5.1}$$

$$T_{MRE} = \frac{1}{N}\sum_{i=1}^{N} |\frac{\vec{v}_{reference}(i) - \vec{v}_{estimated}(i)}{\vec{v}_{reference}(i)}|, \tag{5.2}$$

where N is the number of frames of the evaluated video sequence, $\vec{r} = (r_x, r_y, r_z)$ the angular rotation rate per frame around the vehicle axes, and $\vec{v} = (v_x, v_y, v_z)$ the translation velocity along the vehicle axes. Although these metrics do not necessarily reflect subjective perception, quantitative measures are required for a quick and automated evaluation of the impact of approximate and stochastic mechanisms on the application. we calculate the results using the corresponding Global Positioning System (GPS) data as reference. In this experiment, for each clock period, the optimal CIV coder is deployed.

According to the results in Table 5.1, the proposed encoding scheme reduces the error in both metrics. For example, R_{MAE} is improved by more than 20% in a clock period of 1.7 ns using CIV coding in comparison with conventional data transmission, which is a considerable improvement in the performance of the *Egomotion estimation*. Similar improvements are observed using CIV coding while T_{MRE} is considered. Besides, the conventional data transmission results in invalid frames for which keypoint matching is not possible, for clock periods of 1.7 ns and 1.8 ns. Hence, it significantly reduces the estimation accuracy.

Table 5.1: SIFT-based motion estimation quality for stochastic data transmission using CIV and no-code in comparison with the exact/ideal data transmission.

	Clock Period [ns]	R_{MAE}	T_{MRE} [%]	Valid/Invalid frames
ideal	2.2	8.05e-04	4.19	153 / 0
Conventional (No-Code)	1.9	8.16e-04	4.14	153 / 0
	1.8	9.33e-04	4.96	153 / 1
	1.7	11.2e-04	5.59	153 / 2
CIV	1.9	7.95e-04	4.13	153 / 0
	1.8	8.33e-04	4.8	153 / 0
	1.7	8.78e-04	4.52	153 / 0

Table 5.2: Comparison of the filtered images received using CAIV, FPF-STD and Conventional techniques. Structural similarity index is used to quantify the quality of the filtered images in comparison with the original filtered image.

	Lena			*Cameraman*		
Clock Period [ns]	CAIV	FPF-STD [12]	Conventional	CAIV	FPF-STD [12]	Conventional
0.65	1	1	0.9460	1	1	0.9681
0.6	0.9910	0.9945	0.5556	0.9950	0.9987	0.8545
0.55	0.9162	0.5538	0.5007	0.9503	0.8519	0.8086
0.5	0.9311	0.5517	0.4984	0.9428	0.8506	0.8061

Sobel edge detection algorithm

In this case study, we use a two-step image processing algorithm. In the first step, the input images are filtered with the average filtering using a 3×3 square kernels. In the second step, the average filtering result is fed to the Sobel filter, an edge detection algorithm. Table 5.2 compares the quality of the filtered images using SSIM for CA-IV, FPF-STD, and Conventional methods. The results are tabulated for different clock periods for *Lena* and *Cameraman* images. As can be seen, CA-IV outperforms the other techniques and maintains the quality of the received images by over 91% in all the clock periods. CA-IV degrades the quality compared with FPF-STD by only 0.3% in a clock period

of 0.6 ns.

Fig. 5.11 depicts the comparison of the original filtered *Cameraman* and *Lena* images with the CA-IV, FPF-STD and conventional technique in a period of 0.55 ns at 1.2 V supply voltage. As it is shown in this figure, the edge detection operator is able to detect the edges better using CA-IV in comparison with the other methods.

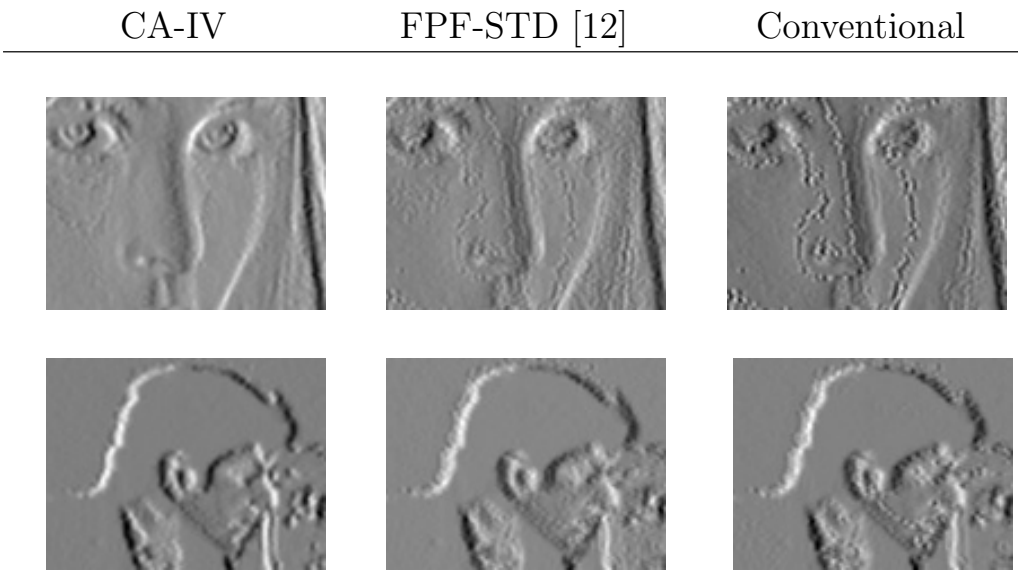


Fig. 5.11: The received images filtered using Sobel edge detection operator. The received images are obtained using CA-IV, FPF-STD and Conventional methods for a period of 0.55 ns at 1.2 V supply voltage.

5.4 Stochastic Wave-Pipeling Validation

The proposed stochastic wave-pipelined communication approach is evaluated in this subsection. We present the simulation results using synthetic and real data (images). We also validate our work using two case studies. For simulations in this subsection, we use inverters with the drive strength of 6 times of the minimum sized inverter. The simulation results are obtained using a commercial 65 nm technology node with the supply voltage ranging from default, 1.2 V, to low-power, 1 V.

5.4.1 Evaluation using Synthetic Data

The proposed communication approach is evaluated across all possible transitions for a 4-bit-width bus for uniformly distributed signals[2]. Fig. 5.12 illustrates the mean square error in logarithmic scale, $log_2(MSE)$, versus different offsets for three time periods. The conventional transmission of signals is obtained at

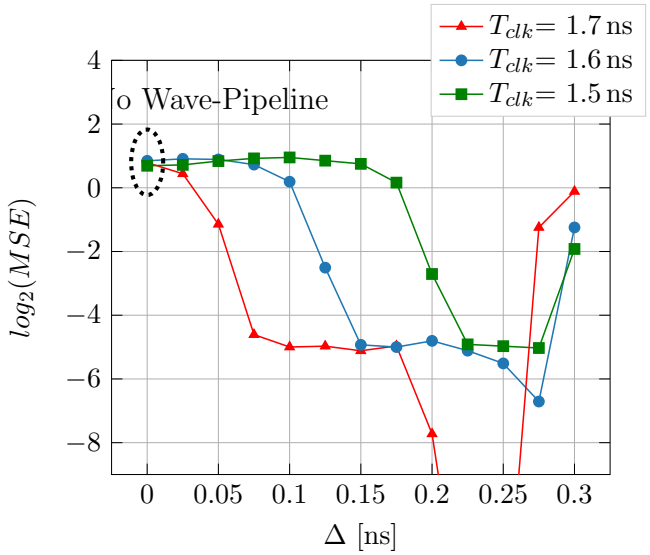


Fig. 5.12: $log_2(MSE)$ versus offsets (Δ) for different time periods. The results are obtained for Uniform data distribution of all possible input transitions for a 4-bit bus at supply voltage of 1.2 V.

$\Delta = 0$, which imposes relatively high errors. By increasing the offset using a stochastic wave-pipelining approach, the error can be reduced drastically. For example, the Δ values between 0.075 and 0.25 ns can guarantee $log_2(MSE)$ smaller than -4 in time period of 1.7 ns. This is $2^{5.3}$ times smaller than the conventional data transmission approach. In addition to that, the large range of the Δ allows a reliable design of the stochastic wave-pipelining. Even though wave-pipelining can provide error-free transmission of data at Δ values between 0.225 and 0.25, a drastic error increment with a small change in Δ makes the selection of this operation point challenging. However, the selection of the operating point is more robust using the stochastic approach.

Considering the variation of transistors, while the fastest clock period in which conventional data transmission can be accomplished is 2.29 ns, using the classical wave-pipelined approach, data transmission can be improved to reach

[2]Similar results are obtained for correlated input signals.

the clock period of 1.81 ns. Using stochastic wave-pipelining, even faster data transmission is possible. According to the Monte Carlo analysis results, data can be typically transmitted at a clock period of 1.36 ns that is about 41% and 25% faster compared to conventional and classical wave-pipelined approaches, respectively.

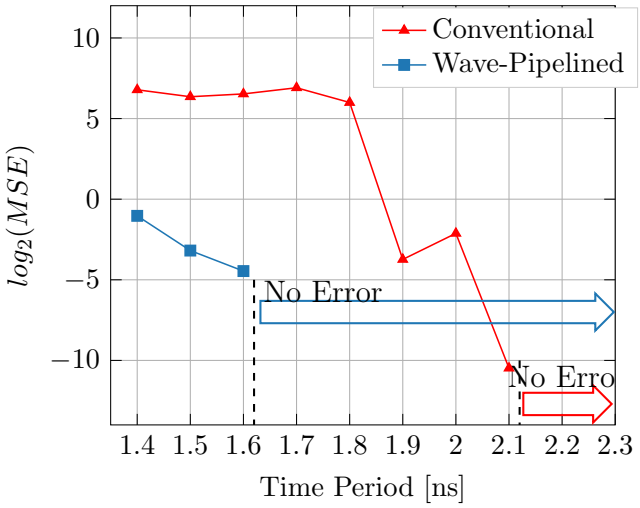


Fig. 5.13: $log_2(MSE)$ versus time period using optimal offset for each time period. The results are obtained for the 8-bit *Lena* image at supply voltage of 1.2 V.

5.4.2 Evaluation Sending Images

To evaluate the stochastic wave-pipelining in a practical scenario, we transfer the *Lena* image through the interconnect. Fig. 5.13 shows $log_2(MSE)$ versus the time period for the conventional and wave-pipelined structures. The optimal Δ is used for each time period. According to the results, the error-free data transmission requires a time period of 2.2 ns for conventional and 1.7 ns for wave-pipelining. The stochastic wave-pipelining provides an extra improvement which depends on the tolerable deviation of the target application. For example, Considering an acceptable $MSE < 2^0$, data can be transmitted at a clock period of 1.4 ns that is is 18% faster than the classical wave-pipelined approach.

As an illustration, Fig. 5.14 shows the received images using conventional and stochastic wave-pipelined approaches. A drastic improvement in the quality of the received image using the stochastic wave-pipelining can be observed compared to conventional communication. The quality of images can be

quantified using the Structural Similarity Index (SSIM). SSIM quantifies the quality degradation of a processed image in comparison with the reference image. The SSIM for the conventional received image is 0.901, and it is 0.997 for stochastic wave-pipelined received image. The results support the arguments in this paper.

(a) Conventional (b) Stochastic Wave-pipelined

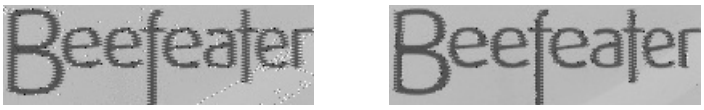
Fig. 5.14: Comparison of the received Lena images at 1.2 V and time period of 1.6 ns. The image is transferred using *conventional* as well as *stochastic wave-pipelined* data transmission.

5.4.3 Case-Study

Optical Character Recognizer

To assess the applicability of the proposed scheme in a real application, the performance enhancement of the stochastic wave-pipelining is evaluated using an Optical Character Recognizer (OCR). OCR is a robust to noise image processing application that can tolerate a certain amount of quality loss in the images. In this case study, a random image from the ICDAR text image dataset [92] has been transferred through the bus at reduced voltage swing of 1 V and using $\times 6$ driver size. *Tesseract* [93], an open-source optical character recognizer engine, is used to recognize the text of the received word images. Fig. 5.15 illustrates the received images and their corresponding recognized text for a time period of 1.55 ns for conventional transmission of the signals and stochastic wave-pipelining. The quality enhancement of the received image using wave-pipelining can be observed compared to the conventional transmission of the images. According to the results, *Tesseract* can recognize the text in received images using wave-pipelining correctly. However, quality deterioration of the received images using conventional transmission has led to

the incorrect recognition of the text by *Tesseract*.



Received text: 'geefea'feg Received text: Beefeater

(a) Conventional (b) Stochastic Wave-pipelined

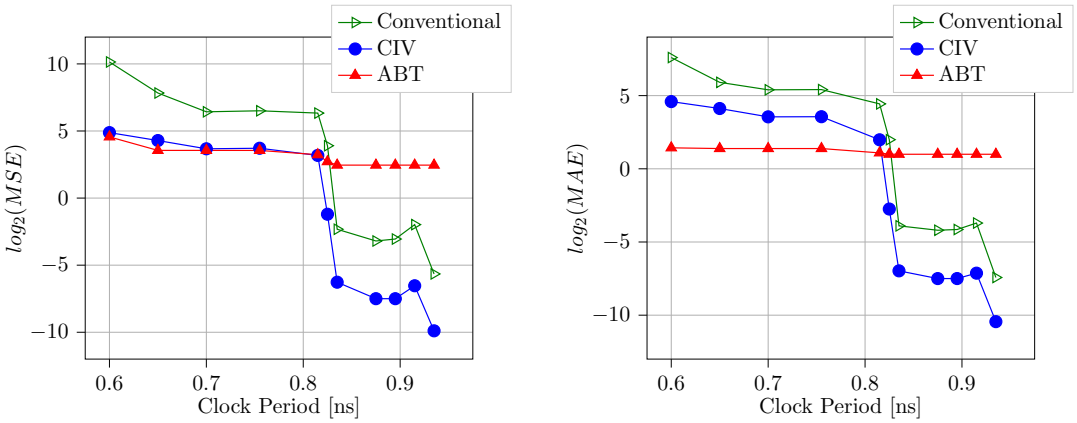
Fig. 5.15: Comparison of a received sample word image and the OCR text recognized by Tesseract OCR engine. The resulting images are obtained using *conventional* and *stochastic wave-pipelining*.

5.5 Alternative Bit-Truncation Validation

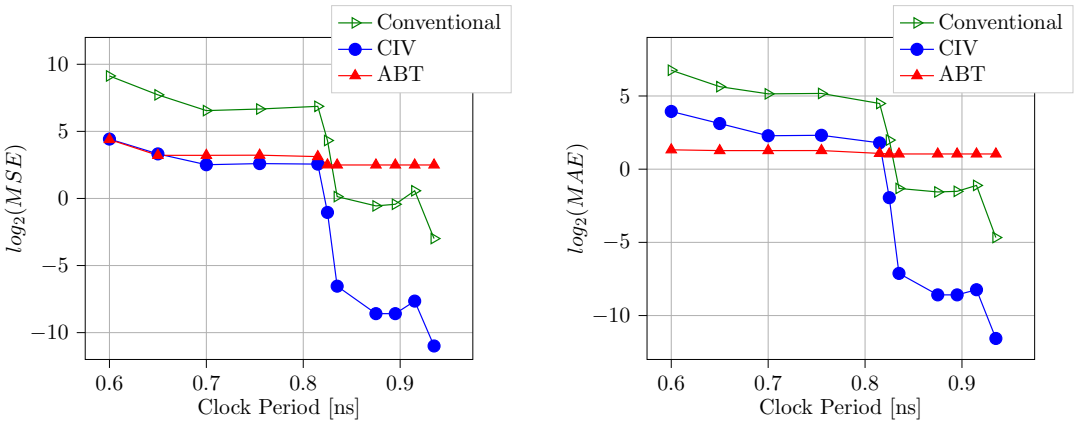
In this subsection, we evaluate ABT technique transferring *Lena* and *Cameraman* images through a 3-mm bus in Fig. 3.2. We compare Alternative Bit-Truncation technique with the *Integer-Value* coding technique proposed in subsection 4.2 and conventional data transmission at voltage supply of 1.2 V using drivers with driver strength of 16 (which means they are 16 times bigger than the minimum sized drivers).

Fig. 5.16 shows $log_2(MSE)$ and $log_2(MAE)$ for Conventional, CIV and ABT techniques for different clock periods. The results are obtained for *Lena* and *Cameraman* images. According to the results, Conventional and CIV approaches outperform the ABT technique in lower frequencies. In principle, it is due to the ABT's inherent frequent errors of the truncated LSBs that occur independent of the timing error. However, by frequency scaling and occurrence of the timing error, ABT can avoid the surge of the integer-value error as opposed to CIV and conventional techniques, and outperforms both in terms of MSE and MAE. For example, transferring *Lena* image ABT can reduce the MAE by 81% and 68% in respect to Conventional and CIV, respectively. It is a remarkable improvement in accuracy considering the concurrent performance and energy improvements of ABT technique.

To evaluate the energy-performance benefits of ABT, we study the mean square of error versus the energy-delay product (EDP) for *Lena* and *Cameraman* images in Fig. 5.17. According to this figure, ABT exhibits remarkable results and outperforms both CIV and conventional techniques. Let us consider an exemplary application with tolerable error threshold of $log_2(MSE) < 3$. The



(a) *Lena*

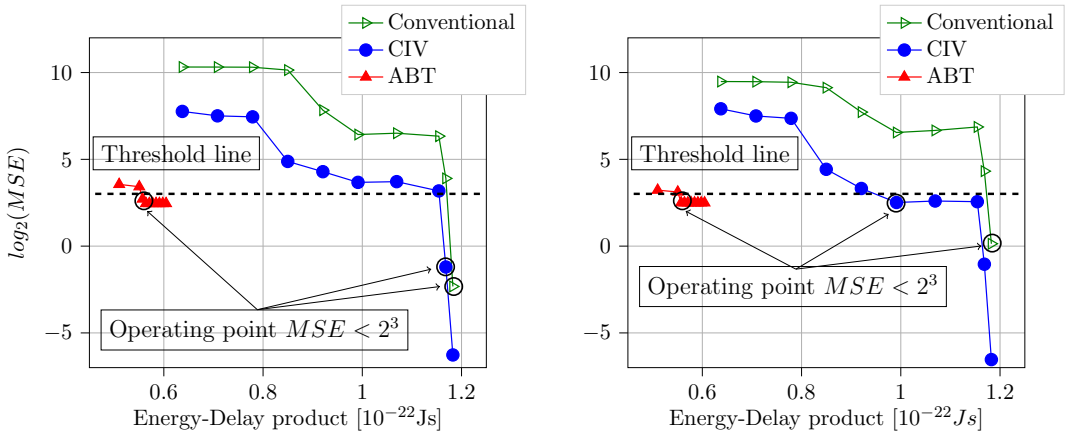


(b) *Cameraman*

Fig. 5.16: $log_2(MSE)$ versus clock period and $log_2(MAE)$ versus clock period for transmission of Lena through an interconnect.

dashed lines in Fig. 5.17 represent the exemplary threshold lines. For the given application, the EDP-based optimized operating points for Fig. 5.17-(b) are respectively, 0.557×10^{-22}, 0.991×10^{-22}, and 1.182×10^{-22} for ABT, CIV, and Conventional techniques. The results imply ABT EDP improvement of 52% and 43% in respect to Conventional and CIV approaches, respectively. The similar results can be observed in Fig. 5.17-(a). Thus, given an error resilient application, considerable energy and performance benefits using ABT can be achieved.

As we have already discussed in Section 4.4, ABT changes the assignment of signals to wires. For an 8-wire bus, there are 8! possibilities. Given the constraints of the target application, the specification of the interconnection



(a) *Lena* (b) *Cameraman*

Fig. 5.17: $log_2(MSE)$ versus Energy-Delay products for transmission of Lena and Cameraman images through an interconnect. The results are obtained for Conventional, CIV and ABT techniques.

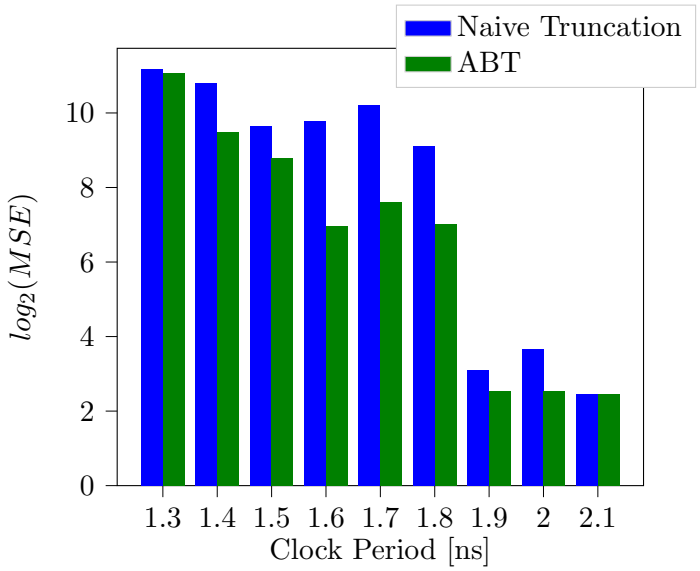


Fig. 5.18: $log_2(MSE)$ versus different clock periods comparing ABT and the Naive truncation technique.

medium, and the input signal, an optimized coder can minimize the integer-value error.

In Fig. 5.18, we compare the optimized alternative truncation technique with naive truncation approach for different clock period sending *Lena* through a network-on-chip (NoC). The image is sent from router A to router B through two physical links at 1.2 V supply voltage using $\times 6$ drivers. More information

regarding the setup of NoC can be found in Chapter 6.

The Naive Truncation approach results are obtained by setting the 3 LSBS of an 8-bit link to zero. Then, the receiver approximates the truncated bits as a median integer value. In this case, Naive Truncation does not enjoy the potential advantages of swapping the wires. According to the results, ABT outperforms the Naive Truncation in almost all clock periods as timing errors surge.

5.6 Conclusion

We carry out a thorough experimental evaluation to show the performance of the proposed approximate methods. The proposed CIV coder demonstrates great capability in reducing the mean square of the error magnitude for different workloads and bus structures. For example, when transmitting the *Lena* image at a supply voltage of 1.2 V and a driver strength of $\times 6$, the CIV coding reduces MSE in a range from 24 to 212 times compared to the conventional transmission for different frequencies. Similarly, proposed CA-IV coding reduces the MSE by up to 210 times compared to conventional data transmission. Two case studies using motion estimation and Sobel edge detection algorithms demonstrate the effectiveness of the proposed Integer-Value Codings for real applications. SWP method is validated using synthetic and image data. Drastic performance improvements can be achieved using SWP. For example, considering a tolerable error of 2^0, SWP can transmit the *Lena* image at supply voltage of 1.2 V and driver strength of $\times 6$ 30% faster than the Conventional transmission of data. The efficiency of SWP is also validated using an optical character recognizer application. Finally, we assessed the Alternative Bit Truncation method. With initial inherent erroneous output due to the bit truncation, it is applicable to applications with a certain error tolerance. Accepting that, ABT exhibits an outstanding EDP in comparison with other techniques. For a given application with tolerable MSE of 2^3, ABT can improve EDP by 52% and 43% in comparison with Conventional and CIV approaches.

In the next chapter, we put the proposed techniques into practice by applying them to a NoC communication medium. We will compare the proposed techniques with a state of the art compression technique in the literature and propose frameworks to integrate the different techniques into NoCs.

CHAPTER 6

Case-Study: Approximate Network-On-Chip

Contents

6.1 Introduction

System-on-chip enables the implementation of complete computing systems including memories, interconnection, processors, and I/O interfaces on a single chip. Many SoCs have more than one Processing Element (PE), effectively being MPSoC. Thanks to massive parallelism, which results in significant system performance improvements, current multi-core SoCs typically have hundreds of cores, and it is expected the count reach thousands of cores soon [94, 95].

Networks-on-Chips (NoCs) have become a prevailing communication solution to connect cores and various on-chip components, including accelerators, caches, and memory controllers. NoCs provide better scalability, and higher bandwidth

than traditional bus-based interconnections [96]. The state-of-the-art NoC design, however, is becoming a bottleneck with historically unprecedented heavy NoC communication loads.

The power consumption of the NoCs accounts for a significant fraction of the overall system power. NoC power consumption can reach up to 40 percent of the overall chip power [97, 22, 98]. For instance, In the Intel TeraFLOPS possessor, the communication power consumption accounts for 28% of the overall network tile power [99]. NoCs consume a disproportionate amount of the overall system energy, and therefore, an improvement in NoCs is essential.

To reduce NoCs power consumption, researchers have proposed different methods such as dynamic voltage and frequency scaling and power gating [100, 101]. They typically exploit the performance-energy trade-off in order to compromise one optimization goal for another, e.g., power supply reduction reduces the power consumption and can adversely affect the performance. Conventionally, communication fabrics are designed so that they enable reliable and deterministic communication.

As we have discussed earlier in Chapter 4, error-free data transfer limits the hardware optimization. It has been shown that several computing applications, such as image processing, pattern recognition, and machine learning, are able to tolerate errors in the computation results. In contrast to conventional performance-energy trade-off, hardware approximation mechanisms propose to trade-off between the level of accuracy required by applications and high performance and/or energy efficiency [14]. Thus, exploring this new design space can leverage applications error resilience to achieve better network efficiency.

Following this concept, some approximate NoC techniques recently focus on providing gains in efficiency metrics accepting an accuracy loss. A brief overview of these techniques is presented in Chapter 2.

This chapter aims to explore and compare the proposed approximate communication techniques, *SWP*, *ABT*, and the combination of these techniques, with *Conventional* data transmission and the state-of-the-art *ABDTR* compression [16] in the context of NoC communication. First, we elaborate on different characteristics that define an NoC. NoC topology, switching strategy, and clocking scheme are among these characteristics. After defining these attributes, we introduce the NoC architecture used to compare the approximate techniques. Finally, after a detailed overview of different existing compression techniques,

we evaluate approximate communication techniques from the following perspectives: performance, power consumption and result quality. We also implement an image processing algorithm on the destination node of the NoC and evaluate the quality of the received images, and the received filtered images using various techniques.

6.2 Preliminaries to Networks-On-Chip

The elements of a network are processing and storage units, so-called *nodes*, which are connected through network fabric that consists of Network Interface (NI), switches (routers), and physical links (wires).

NoC architectures can be differentiated based on specific attributes such as the network topology and protocols that mainly determine switching, routing, and control flow. Designers usually carefully choose these characteristics to meet specific design goals, including energy consumption, performance, and scalability.

In this section, we explore different NoC characteristics. Specifically, we introduce different NoC topologies, switching strategies, routing algorithms, and flow controls. These are the main design decisions that significantly impact the cost, performance, and power of the NoC.

6.2.1 Network Topology

Generally, there are three broad categories of network topology, direct, indirect, and hybrid. Each category is explained below:

Direct Networks: In this network, each node's router is directly connected to the subset of other nodes' routers through channels. As the number of nodes increases, the total communication bandwidth also increases [33]. Therefore, the designers' task is to find the best trade-off between the performance and higher area and energy consumption based on the design goal. A fully connected (point-to-point) topology, where each node is connected to all other nodes, is expensive as it imposes high area and energy costs. In practice, direct networks adopt orthogonal topology, requiring messages to traverse through intermediate nodes before reaching the destination. A simple and famous example of direct orthogonal architecture is n-dimensional mesh.

Indirect Topology: In this topology, each node is connected to external switches through the NI and switches have a point-to-point link to other switches. Switches set up the communication path. Crossbar is the simplest indirect topology example.

Hybrid Networks: This topology, also known as the irregular network, is usually the combination of other network topologies. The goal of combining the shared-bus, direct and indirect networks is to increase the bandwidth while reducing the distance between the nodes. A cluster-based hybrid network is an example of this topology.

6.2.2 Switching Strategies

A message is usually partitioned into several *packets* and then is sent from one node to another node. A packet is further divided into multiple smaller pieces known as flow control units (*flits*). The flow control operations are performed on flits. There are three different types of flits: header, payload, and tail. The first flit of the packet is the Header flit that contains information about the address of source and destination processing elements, the number of flits in the packet, etc. The rest of the flits in a packet are in the payload flits, and finally, the last flit of a packet is the tail flit that indicates the packet end. A flit is made up of one or more physical units (*phits*). Phit is transferred through the physical link in a single clock cycle and usually has the same bit-width as the communication link. Fig. 6.1 shows the structure of a message.

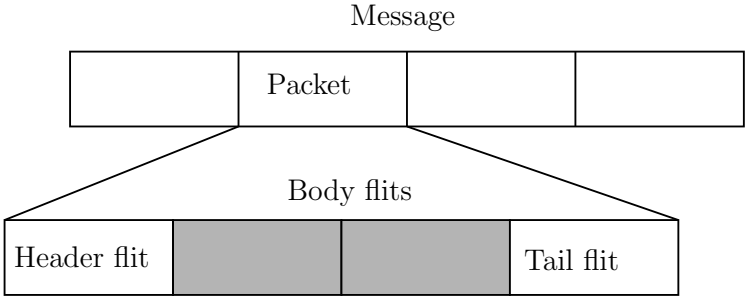


Fig. 6.1: NoC message structure. The message consists of multiple packets. Each packet contains different flits including Header flit, body flit and tail flit.

There are two main switching strategies: Circuit switching and packet

switching:

Circuit Switching: This switching strategy reserves the physical path between the source and the destination for transmission of data. The header flit traverses from source to destination and reserve the path along the way. When the header arrives in the sink core, it sends an acknowledgment to the source, and the rest of the flits pass through the reserved path. If the header can not arrive at the destination (e.g., one of the links is busy for another communication), the header will wait until the link becomes free again. The reservation stays valid until the end of the communication, and with the tail flit, all the links in the path will be set free for the usage of other messages [34]. A drawback of this switching strategy is its poor scalability. In larger networks, several links are occupied for a long time as data transmitted through the links.

Packet Switching: In this switching strategy, packets can be transmitted through the links independently, possibly traverse within different routers, and therefore they have different delays. Unlike circuit switching, packet switching does not suffer from start-up waiting time delay. However, without link reservation, it is harder to guarantee the Quality of Service (QoS) than the circuit switching. There are three popular packet switching techniques:

- Store And Forward switching (SAF): The packet is sent from one router to the next only if there is space in the buffer of the receiver router for the entire packet. Predictably, this technique requires a large buffer size.

- Virtual cut through (VAT): Instead of waiting for space for the entire packet, portions of a packet may be forwarded (cut-through) to the next switch before the entire packet is stored at the current switch. VCT technique, like SAF, needs large buffers, but its latency penalty is less than SAF.

- Wormhole switching (WH): In this switching technique, a flit can be forwarded to the next router if there is at least one flit space in the receiving router. In this case, a message can distribute in multiple routers, which increases the congestion probability. Nevertheless, WH is the most common switching strategy.

6.2.3 Routing Algorithms

A routing algorithm determines a path that a packet takes to reach the destination. These algorithms can be classified as static or dynamic, distributed or source, and minimal or non-minimal.

Unlike a dynamic algorithm that makes routing decisions considering the network's current state, static routing uses fixed paths between a source and a destination regardless of the load on routers and links. Although the adaptive behavior of dynamic routing results in better distribution of traffic, it costs additional resources. On the other hand, static algorithms are easy to implement, but they can not adopt the routing path. XY [38] and fully adaptive [37] are examples of deterministic and dynamic

Researchers usually group algorithms as Source and Distributed routing by the place where the routing information is stored. In the Distributed algorithm, the routing decision is made in each router. When receiving the packet, each intermediate router, by looking at the receiver core's address, makes the routing decision and sends the packet to the next router. While source algorithm calculates and stores the routing paths in the NI. NI puts the routing information into the header flit when a packet is generated, and the routers, by reading this information, direct the packet to the destination. Source routing requires additional information in the header packet, and the number of required bits increased for longer paths and larger networks.

Finally, routing algorithms can be classified based on the routing distance. The minimal routing sends a packet only if the shortest path from the source node to the destination node exists. In contrast, non-minimal routing algorithms allow a packet to be sent even if the shortest path does not exist. In comparison to minimal routing, a non-minimal routing scheme consumes additional power while it is useful to avoid congestion.

In general, routing algorithms should have three characteristics: deadlock, livelock, and starvation free. A deadlock occurs when one or more packets in a network become blocked and remain blocked for an indefinite amount of time [34]. Similarly, livelock occurs when a number of packets experience an unending cycle without any progress. Finally, starvation refers to a scenario where certain low priority packets never access the link o reach the destination.

6.2.4 Flow Control

Flow control techniques ensure that a sender does not send more data packet to the NoC than can be accepted by the receiver router or its NI [34]. Credit-Based and handshake are two techniques that are used for flow control. In a handshake, the sender router waits for a signal from the receiver router data to ensure that the receiver has been stored the flit and is ready to receive a new flit. The sender router knows how many empty spaces (credits) are available in the receiver router in the credit-based technique. So, the sender puts the valid data on the communication link, sets the valid signal to 1, and decrees the credit. When the sender receives the ready signal, the credit is incremented. The credit-based technique has a higher flow rate than handshaking because it does not wait for acknowledge signal, but its overhead is higher because of using the registers to store the credits and additional logic as credit counter.

6.3 Approximate NoCs Schemes

In this section, we present frameworks by which the approximate NoC is realized. We implement and compare the proposed approximate approaches with similar approaches in the literature in the context of NoC.

Previous research works can be classified based on their fundamental approach in trading energy/performance for accuracy as well as the framework in which they are implemented:

Compression or source coding techniques enjoy the performance benefits primarily due to the reduced volume of data transferred through the network [16, 59]. Due to the data manipulation characteristics of compression techniques, they should be implemented before the Network interface (NI).

Coding techniques mainly focus on manipulating the input data to adapt to the channel specifications. In this case, the primary goal is to reduce either bit-error-rate or the magnitude of the error while achieving higher performance or reduced energy consumption with voltage and frequency scaling [20, 60]. Since data packetization changes the original data sequence, a different framework to implement coding techniques are required after the NI. In the context of Noc, we categorize Alternating Truncation (AT) technique (see section 4.4) and Stochastic wave-pipelining approach [80] (see section 4.3) as coding approaches

and implement them using the similar framework.

Dual-Vdd is one of the fundamental techniques to reduce dynamic power. There exist some research works that use this approach to improve energy efficiency for reduced reliability [30, 102, 64, 17].

6.3.1 Compression

One common technique for approximate NoCs is to use lossy compression. The premise of data compression is the removal of redundant data. In the context of NoC, standard practice is to apply compression before the packet injection. Thus, the data is compressed before it is fed into NI and is decompressed after it is received from NI in each IP core. The data is packetized in NI and transferred through the network. Fig. 6.2 shows the general compression framework for NoCs.

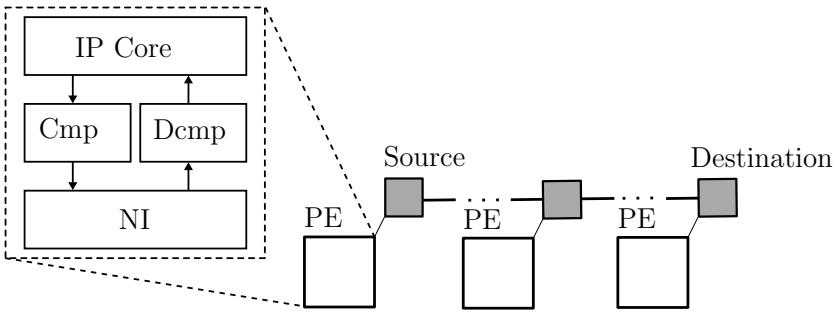


Fig. 6.2: NoC general compression framework. The data is compressed before it is injected to the network and decompressed as it is received in Network Interface.

6.3.2 Coding Techniques

Channel coding is a widely used approach to reduce the Bit-Error-Rate (BER) by changing the data's characteristics being transmitted. Unlike conventional channel coding strategies, the coding can be employed in approximate communication to reduce the integer-value of the error. We have introduced different approximate communication strategies, including integer-value coding, stochastic wave-pipelining, and alternative truncation earlier in Chapter 4.

The general framework for implementation of the coding strategies in the context of NoC is shown in Fig. 6.3. According to this figure, the data, first,

is packetized in the NI of each PE and then the packetized data is coded and injected into the network. The received data is decoded before it is fed to the IP core. The main challenge associated with this framework is the error-free transmission of the head flit.

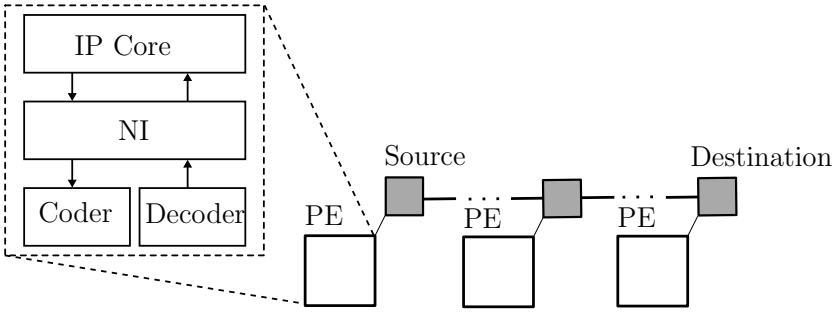


Fig. 6.3: NoC general coding framework

In general, we propose the following strategies to tackle the aforementioned problem:

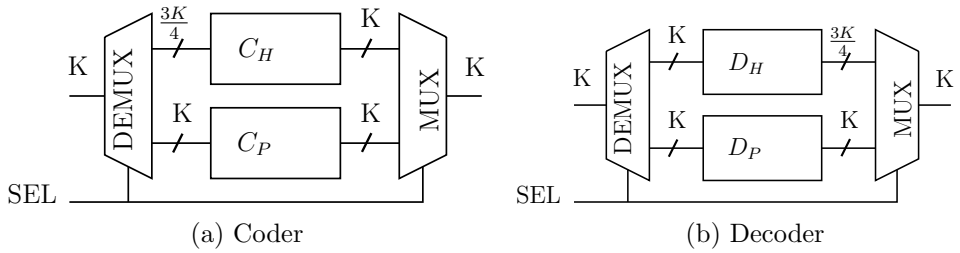
Independent coding of head and body flits

To utilize the coding framework in Fig. 6.3, independent coding techniques can be applied to head and body flits. The main idea is to ensure the error-free transmission of the head flits while minimizing the body flits' integer-value error in the presence of timing errors (due to voltage or frequency scaling).

The general scheme of the proposed framework is depicted in Fig. 6.4. It is assumed that K-bit data transmits through the NoC. A demultiplexer (DEMUX) is used to redirect the data to the dedicated encoder/decoder for head and body flits (head flit encoder and decoder: $\mathbb{C}_H$ and $\mathbb{D}_H$, body flits encoder: $\mathbb{C}_P$ and $\mathbb{D}_P$) based on an SEL signal. The SEL signal determines if the input flit is a head flit or a body flit (0: body flit and 1: head flit). The respective head flit decoder, $\mathbb{D}_H$, should be implemented in each router if the header information is needed to be retrieved.

We employ a coupling avoidance code (CAC) to guarantee the exact transmission of head flits. A thorough review of CACs can be found in subsection 2.4.1.

The $3C$-free forbidden pattern coding is one of the most popular CACs, ensuring $4C$ and $3C$ patterns are eliminated by adding redundant bits to the



(a) Coder (b) Decoder

Fig. 6.4: Proposed approximate coding framework for NoCs. Header flit and body flit coders, $\mathbb{C}_H$ and $\mathbb{C}_P$, are used to code the data in the source. The coded head flit data may decoded in intermediate routers to retrieve the address. Finally, coded header and body flit data are decoded using their respective decoders, $\mathbb{D}_H$ and $\mathbb{D}_P$.

link. However, there are two restrictions associated with FPF-CAC coding. First, the misalignment effect prevents conventional CACs to guarantee a certain performance improvement. Second, in the context of the proposed approximate NoC framework, body and head flits are coded separately using different coders. Unlike the head flit coded using a CAC and transmitted through the link, the previous body flit in the link is not encoded using CAC.

To illustrate the first restriction empirically, we use the standard pattern classification, iC for $i \in 1, 2, 3, 4, 5$, to split transition patterns into different classes and group them into different colors in Fig. 4.9. As it is shown in this figure, different transition classes are overlapped with each other; class $4C$ is overlapped by $3C$, class $3C$ is overlapped by $2C$, and so on. Even though the FPF-CAC can drastically improve performance, it cannot guarantee the head flit's exact transmission.

In the conventional bus structure, CAC codings are applied to all data words. However, in the proposed framework, head and body flits are coded separately. Transmitting the head flit on the link, the previous flit is a body flit. Based on the framework, body flits can be coded using any approximate coding technique, and the previous value on the link can be any value between 0000 to 1111 in a 4-bit link. Let us consider the transmission of a head flit, 0111, while the previous data (body flit) in the link is 1010. Even though 0111 is a valid 3C-free FPF codeword, the link's previous value is not a valid codeword. In this transition, $\downarrow\uparrow \bullet \uparrow$, the wires have the effective capacitance of 3, 3, 0, and 2. Therefore, this transition has a delay proportional to the 3C class and produces a timing error. An alternative approach is required.

Typically less than 3/4 of the bits in the head flit are used. For example, for a 4×4 2-D mesh NoC and the flit size of 32-bits, a minimum of 8 bits are required to represent the node destination and source addresses in the network. In the packet header, 4 bits carry the packet sequence number. The remaining bits are unused, and the packet's reserved field can be used for application specific requirements such as packet type and timestamp. We propose to employ the unused bits by a CAC coding approach to mitigate the redundancy problem of CACs.

The coding should execute a 1-to-1 mapping of data words and valid codewords so that the head flit is transmitted with no timing error. The valid codewords are the codewords that prevent erroneous outputs for a given link structure and a maximum frequency. In summary, given a 4-bit narrow link structure, and a 3-bit input data $X = x_0x_1x_2$ (3/4 used head flit data), obtaining the a CAC coder, $\mathbb{C}_H$, the input data is mapped to an error-free data of $Y = y_0y_1y_2y_3$. The associated decoder, $\mathbb{D}_H$, decodes the received data, $\hat{Y} = \hat{y}_0\hat{y}_1\hat{y}_2\hat{y}_3$, at destination. The typical group shielding topology of 4 wires allows the design of a coder/decoder with negligible overhead.

The valid codewords for a 4-bit link vary with the link structure (link width, spacing, length. driver strength), frequency, voltage swing, and the application's tolerable deviation. Taking into account that all data patterns are allowed in previous body flit in the link, the valid head flit codewords should be obtained by a realistic delay analysis of link. Table 6.1 compares the valid codewords words of FPF-CACs and two different bus structures for a 4-bit link. As it can be observed, the valid codewords of the proposed CAC differ from the FPF-CAC.

Based on the valid codewords listed in Table 6.1, the proposed CAC coding technique's circuit structure can be realized by generating its VHDL description. Depending on the optimization goal, different circuit architectures of the coder and decoder can be realized. For example, Fig. 6.5 illustrates the hardware implementation of the CAC coder and decoder for the link structure with $\times 16$ driver size and minimum time period of 1.85 ns. The proposed coder has a negligible overhead, while the decoder can be realized with no overhead. It is especially important to have a minimum overhead decoder since it should be implemented in each router of the NoC.

Table 6.1: Valid codewords for a 4-bit link of FPF-CAC and the proposed CAC code. The valid codewords of the proposed CAC are obtained from a 4-bit link at 0.8 V voltage swing and 8^{th} segment of the link. Strc1: $\times$16 driver with minimum time period of 1.85 ns, Strc2: $\times$32 driver strength with minimum time period of 1.13 ns.

FPF-CAC	Proposed CAC	
	Strc-1	Strc-2
0000	0000	0000
0001	0001	0001
0011	0011	0011
0110	0110	-
0111	-	0111
1000	1000	1000
1001	1001	1001
1100	-	1100
1110	1110	1110
1111	1111	1111

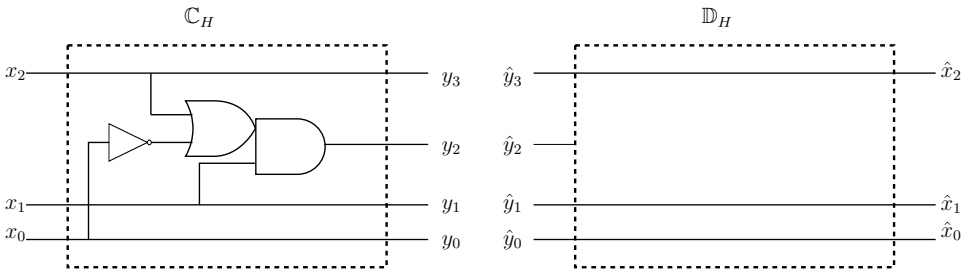
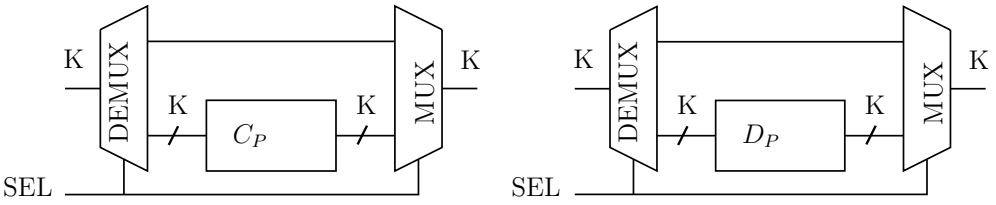


Fig. 6.5: The proposed CAC coder and decoder implemented for bus structure with drive strength of $\times$16

Varying cycle transmission of head and body flits

To utilize the coding framework in Fig. 6.3, varying clock cycle transmission of head and body flits can be considered. It is a practical approach, especially when the system cannot bear the extra area and energy overhead of the exact head flit's coder and decoders. According to the Fig. 6.6, head flits are transmitted with no coding while the body flits coded and decoded using $\mathbb{C}_P$ and $\mathbb{D}_P$ coder and decoder. In the context of bus-based communication, this approach is utilized for retransmission of the erroneous data for AMBA bus standard in [103]. A similar approach can be used for NoC communication.



(a) Coder (b) Decoder

Fig. 6.6: Varying clock cycle approach for head flit and approximate coding for body flits of an NoC. The body flit coder, $\mathbb{C}_P$, is used to code the data in the source while the head flit traverses through the link in more than one clock cycle. The coded body flit data is decoded using their respective decoders, $\mathbb{D}_P$.

6.3.3 Dual-V_{dd}

A reliability-aware adaptive voltage swing approach utilizes a default channel, and a low-power channel [30]. The error-free data transmission is accomplished using the default channel, while lenient accuracy data transmission is carried out in a low-power channel.

There are two elaborations of the Dual-V_{dd} technique. In the first implementation, default and low-power channels are mapped into two different physical lines. The default channel uses the nominal voltage swing, and the low-power channel uses a lower voltage swing. The data flits as well as head flit traverse on the default channel as usual. However, the packets' flits that do not have stringent reliability requirements use the low-power channel with a consequent energy consumption reduction and error increment. The decision-making process is carried out using a single bit in the head flit. This approach suffers a huge area overhead due to line duplication.

In the second implementation, the bitline is preceded by a demultiplexer, two tapered buffers as line drivers, and two tristate buffers [30]. Based on a select signal, the full or low swing path is activated, while the other swing path is disconnected by the high impedance state introduced by the tristate buffer. The level restorer circuit restores the signal at full swing if the signal on the line is set to low swing or maintains the original swing if the signal is in full swing mode.

6.4 Evaluation

In this section, we evaluate different interconnect optimization techniques introduced in the previous sections in the context of NoC. We introduce the NoC architecture and design choices that we have made to implement different techniques. Afterward, we explain the evaluation methodology of the implemented NoC. The individual physical link as the bottleneck of every communication network is assessed. Finally, the implemented NoC is evaluated using detailed simulations sending images from an IP core to another. The results are obtained for different technology nodes to provide a comprehensive evaluation of proposed techniques.

6.4.1 NoC Architecture

An NoC architecture defines attributes of an NoC such as topology, switching, routing, and flow control. NoC architecture's choice is determined by the design requirement, including scalability, cost, latency and power consumption, etc. Here, we choose some typical characteristics to set up an NoC architecture to evaluate different approximate communication techniques.

Among different topologies, the mesh is a direct topology that is scalable and eases the physical design efforts mainly because of identical channel length. Furthermore, the network area increases linearly with the number of the IP cores, and it is easy to achieve deadlock freedom [34]. Since the mesh has a grid structure, IP cores' addresses are their position on the X and Y-axis. Fig. 6.7 depicts a 4×4 2D-mesh NoC and the addresses associated with each IP or node. Topology determines the number of ports in each router. A fully connected 2-D router has five ports (east, west, north, south, and local).

Flit size has a significant impact on the overall network cost and performance. The physical unit (Phit) that determines the communication link's size is usually equal to the flit size. We choose a typical 32-bit flit size.

Wormhole packet switching is a widely used switching strategy. It offers several advantages over other strategies like a smaller buffer size and zero start-up time. In our NoC architecture, we use WH switching.

The XY routing algorithm is a simple distributed static routing that is a deadlock, and livelock free [104]. According to this algorithm, first, the packet will move in X-direction and then in Y-direction. Packets cannot use alternative

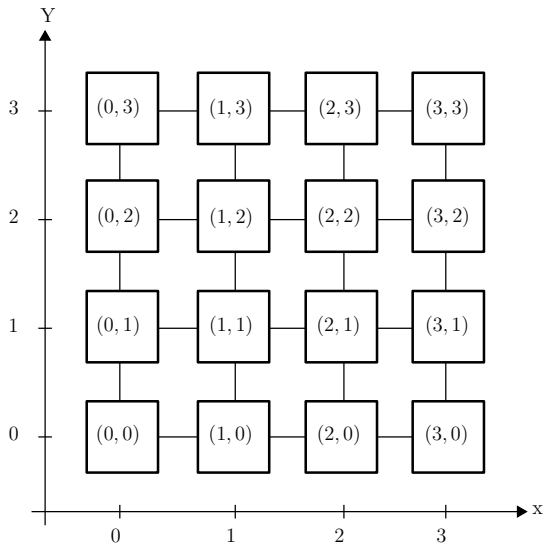


Fig. 6.7: A 4×4 mesh topology and typical addressing format of IPs

routes to bypass the blocked routes. The (x, y) coordinate of the current router is compared to the (x, y) coordinate of the destination router to compute the path.

The header flit contains specific information base on the application and architecture characteristics of the NoC. The header flit must include information regarding the address of source and destination based on the routing algorithm. The length of the address can be calculated as follows:

$$N = \lceil log_2 x \rceil + \lceil log_2 y \rceil \tag{6.1}$$

In this case, for a 4×4 2D-mesh network, a minimum of 8 bits are required to represent the source and destination addresses. Packet ID needs 6 bits to rebuild the message at the destination. To determine the end of the packet, packet length requires 4 bits. The remaining unused bits can be employed to describe packet type, priority, etc.

Due to the higher flow rate in comparison with handshaking, the credit-Based technique is the selected method for the implementation of flow control. In this design, to reduce packet congestion, virtual channels in the input port's buffer are used.

6.4.2 Evaluation Methodology

We use the NoC architecture explained in the previous subsection and send *Lena* and *Cameraman* images on the network. The images are sent from source node, $(1, 2)$, traversed through intermediate note of $(1, 1)$ and received in destination node, $(1, 0)$, as it is shown in Fig. 6.8. This is an exemplary route consisting of three routers and two physical links to validate the proposed techniques.

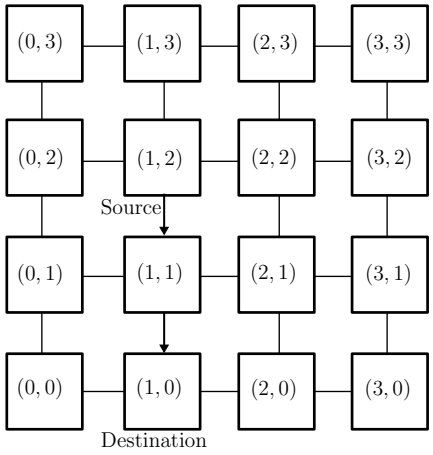


Fig. 6.8: An exemplary route on a 4×4 mesh topology. The data is send from source node $(1, 2)$ and received in destination node $(1, 0)$.

In this experiment, we assume that the packets traverse through each link in one clock cycle. Fig. 6.9 shows the link structure that we use to connect two routers in an NoC.

We use 3 mm, 32-bit-width bus as the physical link divided into eight segments, and each segment is driven using a CMOS driver. The long interconnection link is used to consider the worst-case scenario. NoC design inherently can alleviate the long interconnects' delay. However, the increasingly dominant wire delay in smaller technology nodes justifies repeaters' utilization in NoC fabric. The groups of 4 wires are shielded by paralleled VDD and GND wires. The experiments are carried out on an interconnect on metal layer 4.

The simulation results are obtained using commercial 65 nm and 45 nm technology nodes with the supply voltages of 1.2 V and 1.1 V, respectively. The wires' width and spacing are 0.15 μm for 65 nm technology and 0.12 μm for 45 nm technology node. The designers usually select the wire width, spacing, and layer usage to trade off delay, bandwidth, energy, and noise. We have chosen the standard values. The driver, which is an inverter, has the drive

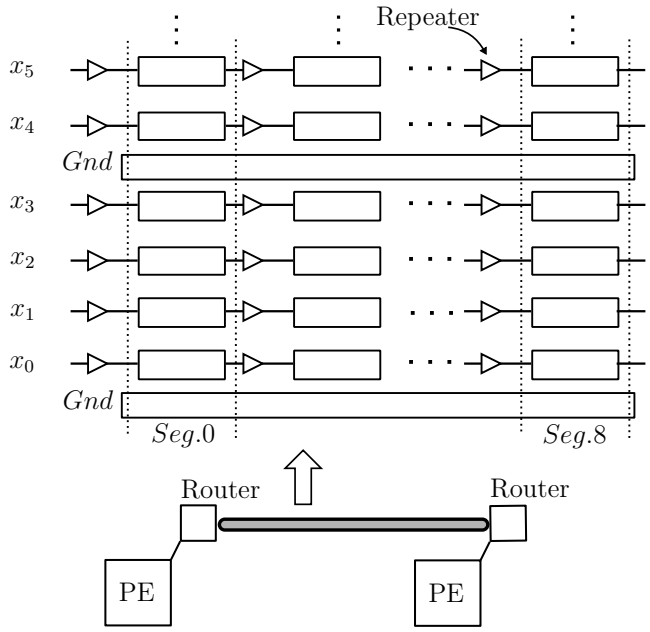


Fig. 6.9: Switch-to-switch link structure used for simulation.

strength of α times of the minimum sized inverter. We choose $\times 6$ sized drivers for this experiment.

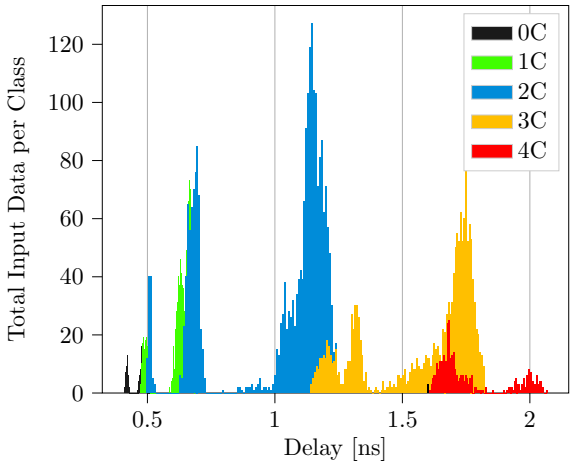
The NoC components are implemented in Register-Transfer Level (RTL). We carry out the SPICE-level simulation using *Cadence Spectre* circuit simulator to reflect the realistic noise and timing error effects of physical links. To account for noise effects, we run *transient-noise* simulation for 100 noise-runs. Then, the overall functionality of the network is simulated using the ModelSim simulator by Mentor Graphics.

6.4.3 Physical Link Evaluation

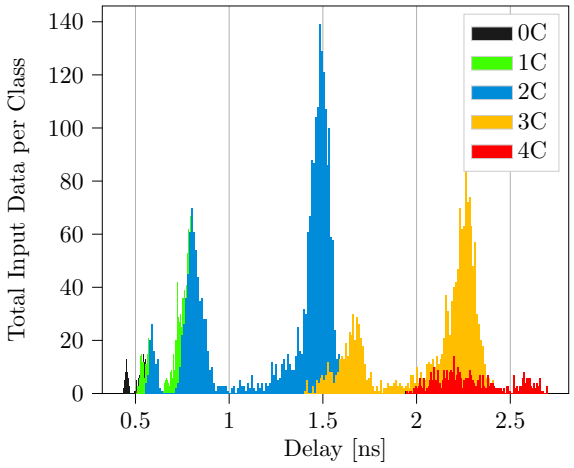
As we have discussed earlier, the links are the cornerstone of the NoC communication framework. This subsection focuses on the individual physical link that we use in the NoC and evaluates its performance metrics.

The simulation results are obtained for 8^{th} segment of the bus using commercial 65 nm and 45 nm technologies, respectively, with supply voltages of 1.2 V and 1.1 V, and $\times 6$ sized drivers as repeaters. We evaluate the links using synthetic data of all possible transitions and observe:

1. The delay of different input transitions classified based on the effective



(a)



(b)

Fig. 6.10: Effective capacitance based classified delay for all possible transitions of a 4-wire bus using 65 nm (a) 45 nm (b) technologies.

capacitances[1]. The 50% delay is defined as the time difference between the input signal and the corresponding output signal crossing 50% of the supply voltage.

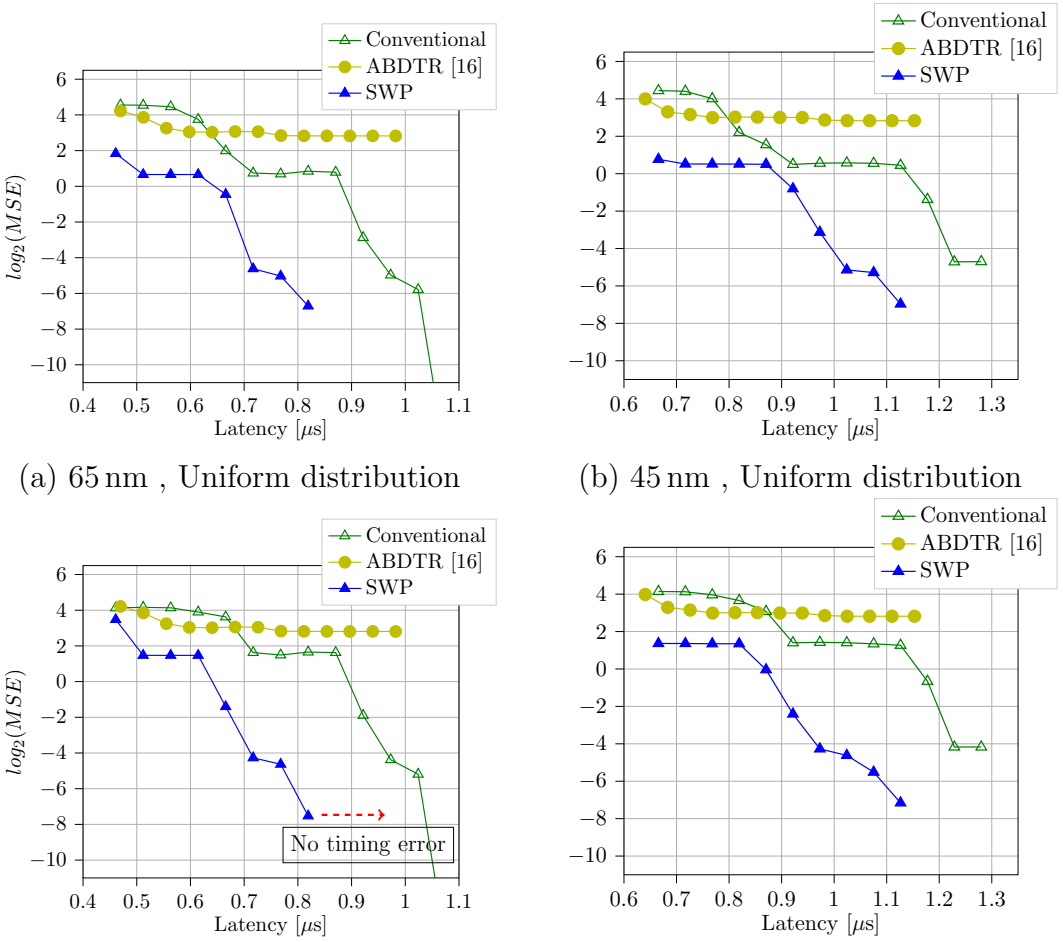
2. The error versus the overall delay of transmission of all transitions (clock period × number of transmissions to transmit) for Conventional data transmission, Stochastic Wave-Pipelining (SWP) and *ABDTR* compression techniques.

[1]More detailed regarding effective capacitance-based pattern classification are explained in Chapter 2

We use standard 3-wire effective capacitance-based pattern classification, iC for $i \in 0, 1, 2, 3, 4$, to split the transition patterns into different classes and group them into different colors. The experiments are carried out on the wire 1 of a 4-bit link. Fig. 6.10 shows the delay distribution of all transmitted patterns using 65 nm and 45 nm technologies. This figure helps observe the delay characteristic of different transition classes for different technology nodes. Also, it provides a rough estimation of performance improvement using different CAC codes. As it is shown in this figure, different transition classes are overlapped each other. Comparing Fig. 6.10(a) and Fig. 6.10(b), transition class $4C$ overlaps the total range of the class $3C$ by 35% and 52% for 65 nm and 45 nm technologies, respectively. It emphasizes using accurate models to develop coding techniques and the importance of approximate communication for future smaller technology nodes. As it is discussed earlier in Chapter 3, exact coding techniques, such as FPF and FTF Cross-talk Avoidance Codings (CACs), developed based on this classification approach cannot guarantee the error-free transmission of the signals and lead to serious reliability concerns. For example, it is assumed that 3C-free FPF-CAC ensures $4C$ and $3C$ patterns elimination and can reduce the worst-case delay from 2.06 ns to 1.14 ns. However, according to Fig. 6.10(a), $2C$ patterns produce errors 1.14 ns clock period. The alternative approach to classifying patterns and modeling delay is presented in Chapter 3.

Accepting erroneous output, approximate communication techniques can improve the interconnection delay and power performance drastically. Several different classes of approximate techniques are proposed in Chapter 4. We evaluate the effectiveness SWP technique (among all proposed techniques) in comparison to conventional data transmission and $ABDTR$ compression technique [16] for the individual link used in the NoC. To do so, synthetic data comprising of all transition patterns with uniform and correlated distributions are transmitted through the interconnect. The randomly generated correlated distribution is $\boldsymbol{\mu}=\left(\begin{smallmatrix}7.5\\7.5\end{smallmatrix}\right)$ and $\boldsymbol{\Sigma}=8\left(\begin{smallmatrix}1 & 0.4\\0.4 & 1\end{smallmatrix}\right)$. The resulting timing error is calculated for different delay values. The delay value is calculated as the multiplication of the clock period and the number of input data that is sent through the interconnect in one clock cycle. Using delay in the x-axis instead of the clock period allows us to make a fair comparison as $ABDTR$ compression technique reduces the amount of data to transmit. Thus, the amount of data to transfer for conventional and SWP are $2^B \times 2^B \times 2$, and for $ABDTR$ it is

reduces by $\frac{1}{1+\lambda}(2^B \times 2^B \times 2)$, where B is the number of wires and λ is interval value for *ABDTR* compression. We choose a λ of 5 that corresponds to 16.6% percent data reduction.



(a) 65 nm , Uniform distribution

(b) 45 nm , Uniform distribution

(c) 65 nm , Correlated distribution

(d) 45 nm , Correlated distribution

Fig. 6.11: $log_2(MSE)$ versus latency for conventional, ABDTR and SWP techniques. The results are obtained for 45 nm and 65 nm technology nodes

Fig. 6.11 shows $log_2(MSE)$ versus delay values for different techniques and using 45 nm and 65 nm technologies. To observe the effect of the input data distribution on the effectiveness of the techniques we compare the results for both correlated and uniform input distributions. According to the results, *SWP* outperforms the other techniques. For example, in Fig. 6.11(c), error-free data transmission can be achieved at delay value of 0.82 μs while it is 1.12 μs for conventional data transmission. Accepting an error, *SWP* can reduce the error drastically in comparison with *Conventional* and *ABDTR* methods. It

is important to highlight that *ABDTR* provides energy benefits due to data compression while it imposes quite large hardware area and energy overhead to approximate the missing values.

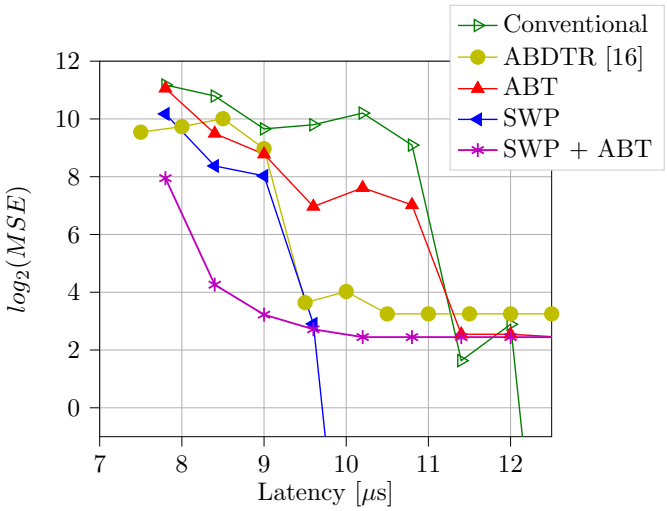
6.4.4 NoC Experimental Results

In this experimental results, we compare *Conventional* data transmission, *ABDTR* compression technique, Alternating Bit-Truncation(ABT), Stochastic Wave-Pipelining (SWP) and the combination of SWP and ABT using 65 nm and 45 nm commercial technologies. We transmit the data through two physical link as is discussed in subsection 6.4.2.

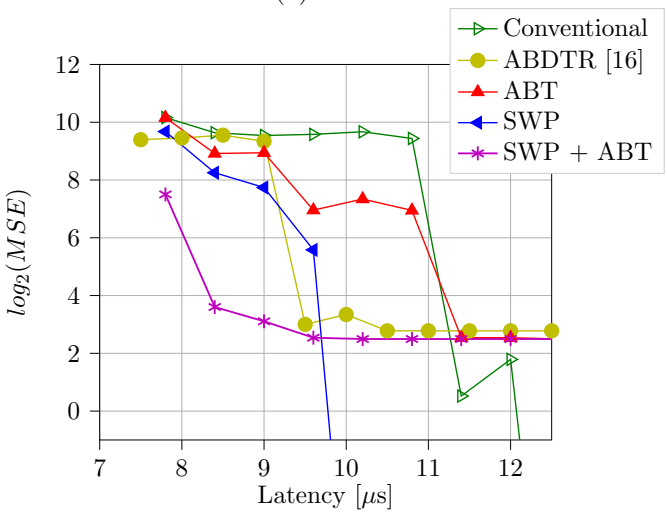
As in the previous subsection, the results of *ABDTR* compression technique are obtained for an interval value of 5 that results in a compression rate of 16.67 (more information regarding this compression technique can be found in subsection6.3.1). For ABT, 3 LSBs are truncated for each pixel, and thus, for the overall 32 bit, link 12 bits are truncated. For *SWP*, the optimal delta value is used for each clock period to obtain the optimal results.

Fig. 6.12 shows the $log_2(MSE)$ versus the latency for different approximate communication techniques using 65 nm technology. In principle, compression techniques reduce the number of data required to transmit and thus, minimize the delay. Therefore, to perform a fair comparison, the x-axis is the latency that is the number of packets required to be transmitted multiply by the clock period. According to the results, *SWP* technique can outperform other techniques for larger latencies. It can improve the data communication performance by about 20% in comparison to the Conventional technique by almost no timing error in results. On the other hand, in higher frequencies (smaller latency values), combination of *SWP* and *ABT* exhibits a solid result. For example, given an application that can tolerate $log_2(MSE)$ of less than 5, combination of *SWP* and *ABT* techniques outperforms Conventional and *ABDTR*, respectively, by 26% and 12% transmitting *Lena* image. Similar results are obtained transmitting *Cameraman* image.

The proposed techniques' significant performance improvements are achieved, while they impose negligible hardware energy and area overhead. Compared to compression techniques, the *SWP* and *ABT* are implemented using minimal hardware and, therefore, do not consume any extra energy or do not occupy the extra area.



(a) *Lena*

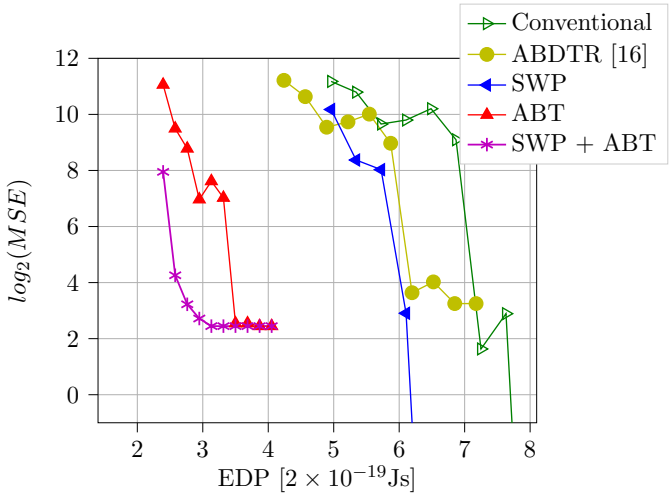


(b) *Cameraman*

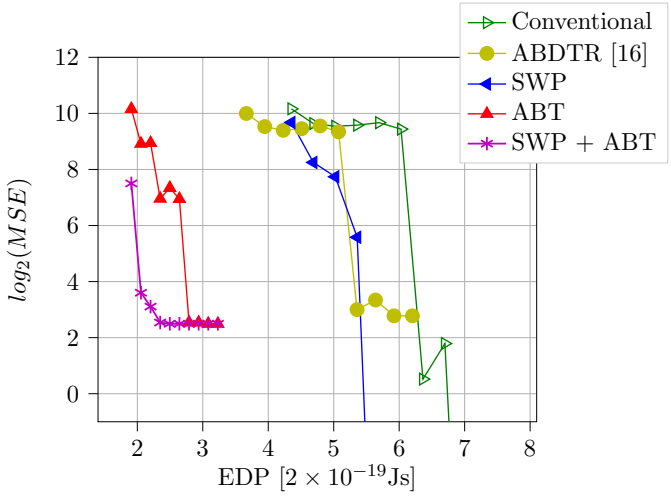
Fig. 6.12: $log_2(MSE)$ versus delay for a NoC architecture applying different approximate communication techniques. The results are obtained sending *Lena* and *Cameraman* images using 65 nm technology.

To explore the energy efficiency of the different techniques, we use the energy-delay product and compare the mean square of the error versus EDP for different techniques transmitting *Lena* and *Cameraman* images in Fig. 6.13. The results are obtained for 65 nm technology node. In this comparison, we do not consider any energy overhead of coder/decider or compressor/decompressor. As it is shown in this figure, combination of *SWP* and *ABT* can result in much smaller EDP for similar or even smaller integer error value. For example, in Fig. 6.13(b), for $log_2(MSE)$ of smaller than 4, using combination of *SWP*

and ABT results in EDP of 4.1×10^{-19} which is 60% smaller than using $ABDTR$ with EDP of 10.7×10^{-19}. The similar improvement can be observed in Fig. 6.13(a). For example, combination of SWP and ABT can improve EDP for more than 60% in comparison with conventional transmission of *Lena* image for $log_2(MSE)$ of smaller than 4. As we have discussed in previous chapter, the energy efficiency benefits of the ABT are due to two main characteristics of this technique: first, it removes the worst-case transitions that consumes the most energy; and second, it set number of LSBs to zero and in principle, the inactive lines do not consume energy.



(a) *Lena*



(b) *Cameraman*

Fig. 6.13: $log_2(MSE)$ versus EDP for a NoC architecture applying different approximate communication techniques. The results are obtained sending *Lena* and *Cameraman* images using 65 nm technology.

We use a two-step image processing algorithm in the processing element of the node (1,0). The received images are filtered with the average filtering using 3×3 square kernel. Afterward, the result of average filtering is fed to a Sobel filter to do the edge detection[2]. We compare the received *Lena* and *Cameraman* images and the corresponding filtered images using different approximate communication techniques in Fig. 6.14. The results are obtained for delay of 9 μs and that is 30% faster than nominal error-free delay of 13.2 μs using 65 nm technology. Considering combination of *SWP* and *ABT*, the substantial performance improvement comes along with more than 50% energy consumption reduction in comparison with conventional data transmission and quite insignificant area overhead. The proposed models improve the quality of received images in comparison of *Conventional* and *ABDTR*.

Table 6.2: Quality comparison of the received images and their corresponding filtered images for *Conventional*, *ABDTR*, *SWP*, and combination of *SWP* and *ABT* techniques. The results are compared using MSE, MAE , SSIM and PER metrics for transmission of *Lena* and *Cameraman* images.

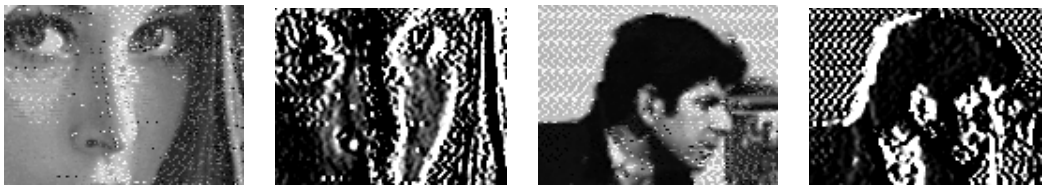
	Received Original Image				Received Sobel filtered Image				
	MSE	MAE	SSIM	PER	MSE	MAE	SSIM	PER	
Conventional	9.655	3.815	0.435	0.476	16.042	7.524	0.437	0.980	*Lena*
ABDTR [16]	8.969	3.133	0.592	0.416	15.056	6.882	0.631	0.987	
SWP	8.029	1.596	0.811	0.113	14.059	5.672	0.830	0.688	
ABT	8.779	2.945	0.609	0.900	14.914	6.626	0.626	0.992	
SWP + ABT	**3.387**	**1.187**	**0.951**	0.875	**9.431**	**4.249**	**0.971**	0.982	
Conventional	9.540	3.566	0.634	0.424	15.900	7.271	0.670	0.951	*Cameraman*
ABDTR [16]	9.343	3.412	0.661	0.400	15.272	6.973	0.701	0.967	
SWP	7.740	1.178	0.915	0.093	13.897	5.390	0.910	0.579	
ABT	8.943	2.966	0.686	0.916	15.278	6.632	0.745	0.989	
SWP + ABT	**3.108**	**1.134**	**0.954**	0.893	**9.251**	**4.142**	**0.977**	0.971	

Table 6.2 compares the quality of received *Lena* and *Cameraman* images and their corresponding filtered images. The comparison is made using different quality metrics. Structural Similarity Index (SSIM) is a perceptual metric that quantifies image quality degradation caused by processing such as data compression or data transmission losses. SSIM gives a value between 0 and 1

[2]More details of the image processing algorithm used in this section can be found in Chapter 4

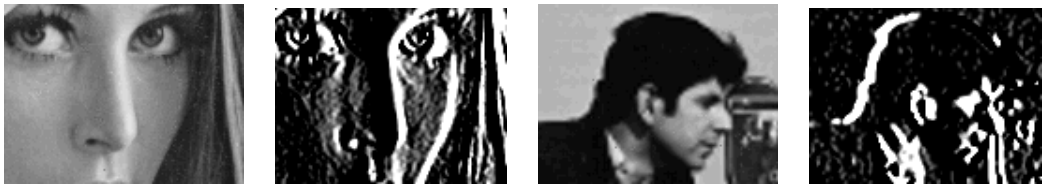
(a) *Original*

(b) *Conventional*



(c) *ABDTR [16]*

(d) *SWP*



(e) *SWP+ ABT*

Fig. 6.14: Comparison of the received *Lena* and *Cameraman* images and their corresponding filtered images at 1.2 V and latency of 9 μs. The image is transferred using *Conventional*, *ABDTR*, *SWP*, and combination of *SWP* and *ABT* techniques. The results are obtained using 65 nm technology

that the higher value indicates a better image quality. Unlike Mean Square of Error (MSE) and Mean Absolute Error (MAE), SSIM is based on visible structures in the image. Pixel Error Rate (PER) is the percentage of pixels that have errors relative to a total number of the received image pixels. We use MSE, MAE, SSIM, and PER to compare the images. According to the results, the *SWP + ABT* outperforms all other techniques for both *Lena* and *Cameraman* images. For example, *SWP + ABT* has a SSIM that is 38% and 35% higher than *ABDTR* technique for original and filtered images, respectively. Even though, *ABT* and *SWP + ABT* have higher PER, they result in a better received image quality. Results emphasize that the small frequent errors are negligible in most image processing applications where the small image quality degradation is tolerable.

For sake of completeness, we compare the results for 45 nm technology. The similar improvements are observed. Fig. 6.15 shows $log_2(MSE)$ versus EDP for different optimization techniques transmitting *Lena* from node (1,2) to (1,0). Like 65 nm technology, *SWP + ABT* shows a substantial EDP improvement in comparison with conventional technique. The inherent functionality of *SWP + ABT* to avoid crosstalk noise, highlights its effectiveness in smaller technology nodes.

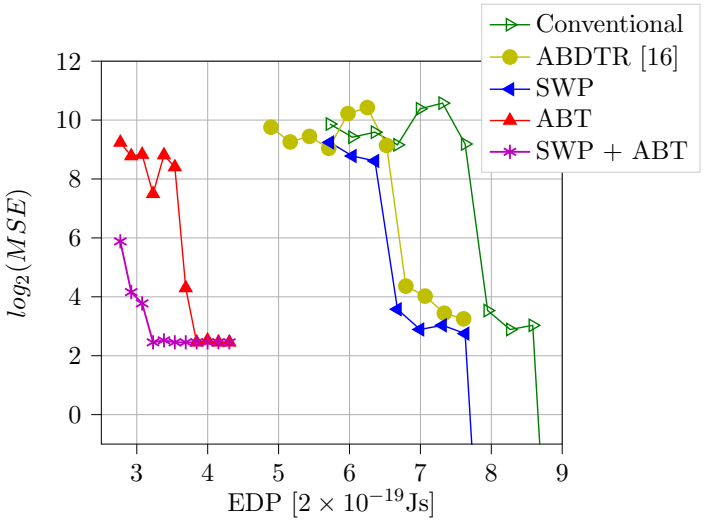
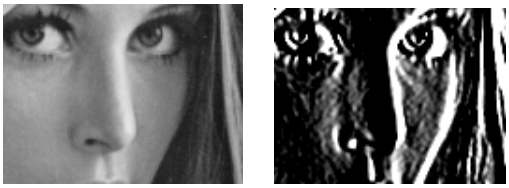
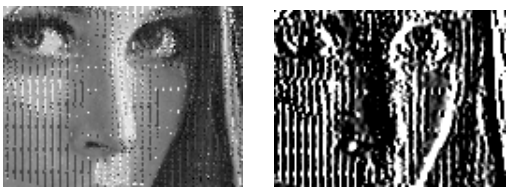


Fig. 6.15: $log_2(MSE)$ versus delay for a NoC architecture applying different approximate communication techniques. The results are obtained sending *Lena* image using 45 nm technology.

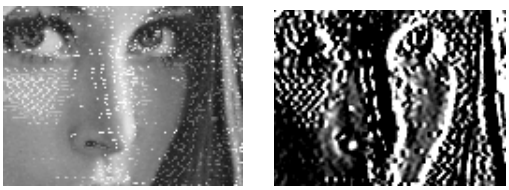
We also compare the received *Lena* images and the corresponding filtered



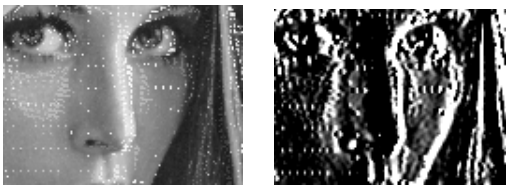
(a) Original



(b) Conventional



(c) ABDTR [16]



(d) SWP

(e) *SWP*+ ABT

Fig. 6.16: Comparison of the received *Lena* images at 1.1 V and time period of 12 μs.The image is transferred using *Conventional*, *ABDTR*, *SWP*, and combination of *SWP* and *ABT* techniques.The results are obtained using 45 nm technology

images for different techniques using 45 nm technology. The results are obtained for delay of 12 μs that is about 29% faster than the nominal error-free delay of 16.8 μs.

Table 6.3 compares the quality of received *Lena* images and their corresponding filtered images. Like 65 nm, the comparison is made using MSE, MAE, SSIM, and PER metrics. The proposed approximate techniques exhibit higher quality for metric. For example, *SWP + ABT* exhibits 39% and 38% quality enhancement in terms of SSIM in comparison with *ABDTR* technique for original and filtered images, respectively.

Table 6.3: Quality comparison of the received images and their corresponding filtered images for *Conventional*, *ABDTR*, *SWP*, and combination of *SWP* and *ABT* techniques. The results are compared using MSE, MAE , SSIM and PER metrics for transmission of *Lena* image using 45 nm technology node.

	Received Original Image				Received Sobel filtered Image			
	MSE	MAE	SSIM	PER	MSE	MAE	SSIM	PER
Conventional	9.589	3.746	0.426	0.463	15.982	7.491	0.440	0.979
ABDTR [16]	9.138	3.191	0.561	0.392	15.013	6.887	0.586	0.986
SWP	8.612	2.520	0.679	0.224	14.540	6.389	0.720	0.877
ABT	8.827	3.445	0.543	0.914	15.014	6.975	0.578	0.995
SWP + ABT	4.061	1.408	0.919	0.879	10.176	4.594	0.940	0.984

6.5 Conclusion

Networks-on-Chip (NoCs) have emerged as the most suitable method to connect an ever-increasing number of on-chip components in modern complex systems. The communication subsystem accounts for a significant fraction of the overall energy budget of multi-core systems. That having been said, fundamental improvement of the NoCs is essential.

Many applications can tolerate errors. Leveraging relaxed accuracy for high energy and performance efficiency has quickly become the accepted practice in the NoC communication fabric.

In this chapter, we described the fundamental characteristics of the NoC. We discussed approximate NoC schemes and the general frameworks that the approximate techniques, including compression and coding methods, can integrate into NoC architectures. Compressors are usually implemented before

the NI. The compressed data packetized and injected into the network using NI. However, using coding techniques like *ABT* or *SWP*, the data should be coded after the NI. To ensure the exact transmission of the head flit, in this case, we introduced two proposals: first, using two independent coding techniques dedicated to head and body flits; and, second, varying number of cycles for head and body flits.

We evaluated different techniques sending images through the NoC. Our evaluation results show that the proposed approximate communication techniques can improve EDP by up to 60% and 70% in comparison with the state of the art NoC compression mechanism, *ABDTR*, and *Conventional* data transmission. Also, the evaluation of received images after average and Sobel filtering using SSIM metric shows up to 39% quality improvement using *SWP + ABT* in comparison with ABDTR. We observe that regardless of a higher pixel error rate of the *SWP + ABT* technique, the quality of the received images are higher. It is mainly because the frequent smaller errors do not contribute significantly to the overall quality of the images.

CHAPTER 7

Conclusion

This dissertation contributes to the body of research by addressing two main problems identified in Chapter 2.

First, it has shown that the absence of precise models to estimate the power consumption and performance of on-chip communication leads to imprecise evaluation and development of only partially useful optimization techniques. Optimization techniques can only provide practical improvements if derived based on physically precise yet universally valid, high-level models. Signals' temporal alignment is an essential factor in modeling the interconnects that are conventionally overlooked.

Second, as discussed, the communication subsystems cannot keep pace with current advances in technology. The on-chip communication is facing budget constraints that make a fully reliable operation highly expensive. The untapped potential of full-system approximation can only be achieved by approximating all subsystems, including the communication.

To address the first problem, we analyzed neighboring wires' alignment behavior for worst-case and best-case transitions. Based on this analysis, Misalignment-Aware energy and delay models (MAA) have been developed. The superiority of MAA models has been proved through experimental results. For instance, the proposed energy model reduces the normalized mean squared error of the conventional energy estimation model by up to 13 times for

different bus characteristics. The proposed energy model provides a more precise evaluation for a fair comparison of different low-power techniques. For example, the conventional model mispredicts the energy reduction of Classical Bus Invert coding by up to more than 44% in comparison with simulation results in 5^{th} segment of the interconnect, while it is less than 2% for the proposed model.

To address the second problem, we proposed different approximate communication techniques. The **first** method is integer value coding. Restricted to simplest light-weighted coder and decoder, we introduced swap and inversion of selected input signals. Swap coding leverages the physical properties of the CMOS interconnects and changes the assignment of signals to wires. On the other hand, Inversion coding leverages the real-world data's spatial locality to exchange the probability of the worst-case transitions with the lowest probability transitions. The evaluation results show this coding method's effectiveness in reducing the magnitude of timing error by up to 200 times while it has negligible coder/decoder overhead. For applications with more relax area-constraint, a Crosstalk Avoidance based approximate coding, CA-IV, is suggested. Unlike conventional CACs that are very sensitive to timing error, we designed this technique for approximate communication. For example, CA-IV can improve MSE by up to 2^{10} in comparison with FPF CAC when sending a Cameraman image. The **second** approach is Stochastic Wave-Pipelining (SWP). SWP relaxes the stringent constraints of the classical wave-pipelining, which in some cases prevents the effective practical use of classical wave pipelining. Depending on the tolerable error of a given application, SWP can considerably improve classical wave-pipelining performance. For example, the transmission of Lena image and accepting MSE of smaller than 2^0, SWP enables 18% faster transmission of the signals. A key feature of SWP is that it enables additional improvements by combining with almost all existing stochastic and approximate communication approaches. The **third** proposed technique is Alternating Bit-Truncation (ABT), a simple but effective bit truncation technique in which a certain number of LSBs are set to zero. With a careful assignment of zeros to the wires, it is possible to prevent or reduce the crosstalk noise using virtual shielding. The inactive lines do not contribute to switching power, and hence ABT reduces the energy consumption drastically. According to the results, ABT can improve EDP by more than 50% in respect to conventional data

transmission for an application with a tolerable MSE of smaller than 2^3 and transmitting Cameraman image.

Finally, the proposed techniques are validated in the context of NoC. The results are obtained for 45 nm and 65 nm technology nodes and considering different simulation scenarios. The comparison is made with a state-of-the-art approximate compression technique. For example, for a given application with MSE smaller than 2^4, combination of SWP and ABT improves EDP by 60% using 65 nm technology and transmitting Cameraman image.

With contributions proposed in this thesis, we achieved our primary objective to alleviate interconnections problems in modern technology nodes. More specifically, we addressed the problem of current inaccurate high-level energy and delay models by introducing precise high-level misalignment-aware models. The current and future technology nodes cannot provide the required resources for the reliable operation of interconnections. Several optimization techniques for approximate communication are proposed in this thesis to address communication problems as the bottleneck of integrated circuits. The proposed techniques enable the full-system approximation with negligible hardware overhead.

Further works are required to explore the open research problems in the area of on-chip communication. We suggest the following future works:

1. Approximate communication techniques can be explored targeting alternative interconnection materials such as carbon nanotube interconnects. Applying the proposed techniques for new materials should be explored for the approximate communication's untapped potential.

2. One of the fundamental problems of current machine learning applications is communication. As future work, this dissertation's proposed techniques can be applied to critical applications such as machine learning. A high-performance improvement for a negligible accuracy loss is expected for such robust applications.

3. A hybrid approximate technique for quality configurable communication is valuable. It can be achieved by combining different light-weight techniques to meet the given application's requirements for a given technology.

4. Full-system approximation implementation and error analysis can provide a perfect opportunity for higher performance improvement. The error

resulting from various sub-systems can overlap each other. Thus, a careful system design considering multiple sub-systems may unlock an even higher potential of the approximation.

List of Figures

List of Tables

Bibliography

[1] Igor L. Markov. Limits on fundamental limits to computation. *Nature*, 512:147 EP –, Aug 2014. Review Article.

[2] J. A. Davis, R. Venkatesan, A. Kaloyeros, M. Beylansky, S. J. Souri, K. Banerjee, K. C. Saraswat, A. Rahman, R. Reif, and J. D. Meindl. Interconnect limits on gigascale integration (gsi) in the 21st century. *Proceedings of the IEEE*, 89(3):305–324, 2001.

[3] D. Blaauw et al. Driver modeling and alignment for worst-case delay noise. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 11(2):157–166, April 2003.

[4] R. Gandikota et al. Worst-case aggressor-victim alignment with current-source driver models. In *Design Automation Conference, 2009. DAC '09. 46th ACM/IEEE*, pages 13–18, July 2009.

[5] A. B. Kahng, S. Muddu, and D. Vidhani. Noise and delay uncertainty studies for coupled rc interconnects. In *Twelfth Annual IEEE International ASIC/SOC Conference (Cat. No.99TH8454)*, pages 3–8, 1999.

[6] F. Shi, X. Wu, and Z. Yan. Improved analytical delay models for rc-coupled interconnects. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 22(7):1639–1644, 2014.

[7] T. Uchino and J. Cong. An interconnect energy model considering coupling effects. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 21(7):763–776, Jul 2002.

[8] M. R. Stan and W. P. Burleson. Bus-invert coding for low-power i/o. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 3(1):49–58, 1995.

[9] L. Benini, G. De Micheli, E. Macii, D. Sciuto, and C. Silvano. Asymptotic zero-transition activity encoding for address busses in low-power microprocessor-based systems. In *Proceedings Great Lakes Symposium on VLSI*, pages 77–82, 1997.

[10] P. P. Pande, Haibo Zhu, A. Ganguly, and C. Grecu. Energy reduction through crosstalk avoidance coding in noc paradigm. In *9th EUROMICRO Conference on Digital System Design (DSD'06)*, pages 689–695, 2006.

[11] S. R. Sridhara, A. Ahmed, and N. R. Shanbhag. Area and energy-efficient crosstalk avoidance codes for on-chip buses. In *IEEE International Conference on Computer Design: VLSI in Computers and Processors, 2004. ICCD 2004. Proceedings.*, pages 12–17, 2004.

[12] Chunjie Duan, Brock J. LaMeres, and Sunil P. Khatri. *On and Off-Chip Crosstalk Avoidance in VLSI Design.* Springer Publishing Company, Incorporated, 1st edition, 2010.

[13] P. Stanley-Marbell and P. Hurley. Probabilistic Value-Deviation-Bounded Integer Codes for Approximate Communication. *ArXiv e-prints*, April 2018.

[14] Sparsh Mittal. A survey of techniques for approximate computing. *ACM Comput. Surv.*, 48(4):62:1–62:33, March 2016.

[15] V. K. Chippa, D. Mohapatra, K. Roy, S. T. Chakradhar, and A. Raghunathan. Scalable effort hardware design. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 22(9):2004–2016, 2014.

[16] L. Wang, X. Wang, and Y. Wang. ABDTR: Approximation-based dynamic traffic regulation for networks-on-chip systems. In *2017 IEEE*

International Conference on Computer Design (ICCD), pages 153–160, Nov 2017.

[17] A. B. Ahmed, D. Fujiki, H. Matsutani, M. Koibuchi, and H. Amano. AxNoC: Low-power approximate network-on-chips using critical-path isolation. In *2018 Twelfth IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*, pages 1–8, Oct 2018.

[18] D. J. Pagliari, E. Macii, and M. Poncino. Serial T0: Approximate bus encoding for energy-efficient transmission of sensor signals. In *2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6, June 2016.

[19] R. Boyapati, J. Huang, P. Majumder, K. H. Yum, and E. J. Kim. APPROX-NoC: A data approximation framework for network-on-chip architectures. In *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*, pages 666–677, June 2017.

[20] Phillip Stanley-Marbell and Martin Rinard. Value-deviation-bounded serial data encoding for energy-efficient approximate communication. *MIT technical report*, 2015.

[21] L. Wang, Y. Wang, and X. Wang. An approximate multiplane network-on-chip. In *2020 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 234–239, 2020.

[22] Filipe Betzel, Karen Khatamifard, Harini Suresh, David J. Lilja, John Sartori, and Ulya Karpuzcu. Approximate communication: Techniques for reducing communication bottlenecks in large-scale parallel systems. 51(1), January 2018.

[23] Andrew B. Kahng et al. Statistical crosstalk aggressor alignment aware interconnect delay calculation. In *Proceedings of the 2006 International Workshop on System-level Interconnect Prediction*, pages 91–97. ACM.

[24] G. Karakonstantis, G. Panagopoulos, and K. Roy. Herqules: System level cross-layer design exploration for efficient energy-quality trade-offs. In *2010 ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED)*, pages 117–122, 2010.

[25] A. Raha and V. Raghunathan. Approximating beyond the processor: Exploring full-system energy-accuracy tradeoffs in a smart camera system. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 26(12):2884–2897, Dec 2018.

[26] 2015 international technology roadmap for semiconductors (ITRS) - semiconductor industry association. (Accessed on 10/03/2019).

[27] Neil Weste and David Harris. *CMOS VLSI Design: A Circuits and Systems Perspective*. Addison-Wesley Publishing Company, USA, 4th edition, 2010.

[28] D. Bertozzi, L. Benini, and G. De Micheli. Error control schemes for on-chip communication links: the energy-reliability tradeoff. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24(6):818–831, June 2005.

[29] A. Ejlali, B. M. Al-Hashimi, P. Rosinger, S. G. Miremadi, and L. Benini. Performability/energy tradeoff in error-control schemes for on-chip networks. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 18(1):1–14, Jan 2010.

[30] A. Mineo, M. Palesi, G. Ascia, P. P. Pande, and V. Catania. On-chip communication energy reduction through reliability aware adaptive voltage swing scaling. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 35(11):1769–1782, Nov 2016.

[31] D. Fujiki, K. Ishii, I. Fujiwara, H. Matsutani, H. Amano, H. Casanova, and M. Koibuchi. High-bandwidth low-latency approximate interconnection networks. In *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 469–480, Feb 2017.

[32] Daniel Schinkel. *On-chip data communication*. PhD thesis, University of Twente, Netherlands, June 2011. 10.3990/1.9789036532020.

[33] Sudeep Pasricha and Nikil Dutt. *On-Chip Communication Architectures: System on Chip Interconnect*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2008.

[34] Sandeep Saini. *Low Power Interconnect Design*. Springer Publishing Company, Incorporated, 2015.

[35] T. Zhang and S. S. Sapatnekar. Simultaneous shield and buffer insertion for crosstalk noise reduction in global routing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 15(6):624–636, 2007.

[36] P. Saxena, N. Menezes, P. Cocchini, and D. A. Kirkpatrick. Repeater scaling and its impact on cad. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 23(4):451–463, 2004.

[37] V. Raghunathan, M. B. Srivastava, and R. K. Gupta. A survey of techniques for energy efficient on-chip communication. In *Proceedings 2003. Design Automation Conference (IEEE Cat. No.03CH37451)*, pages 900–905, 2003.

[38] Alberto Garcia-Ortiz et al. Low-power coding: Trends and new challenges. *Journal of Low Power Electronics*, 13(3), 2017.

[39] Ki-Wook Kim, Kwang-Hyun-Baek, N. Shanbhag, C. L. Liu, and Sung-Mo Kang. Coupling-driven signal encoding scheme for low-power interface design. In *IEEE/ACM International Conference on Computer Aided Design. ICCAD - 2000. IEEE/ACM Digest of Technical Papers (Cat. No.00CH37140)*, pages 318–321, 2000.

[40] M. Palesi, G. Ascia, F. Fazzino, and V. Catania. Data encoding schemes in networks on chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 30(5):774–786, 2011.

[41] N. Jafarzadeh, M. Palesi, A. Khademzadeh, and A. Afzali-Kusha. Data encoding techniques for reducing energy consumption in network-on-chip. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 22(3):675–685, 2014.

[42] A. Najafi, L. Bamberg, A. Najafi, and A. Garcia-Ortiz. Misalignment-aware delay modeling of narrow on-chip interconnects considering variability. In *2018 7th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, pages 1–4, 2018.

[43] Amir Najafi, Lennart Bamberg, and Alberto GarcÃa-Ortiz. Misalignment-aware energy modeling of narrow buses for data encoding schemes. *Integration*, 72:58 – 65, 2020.

[44] Ki-Wook Kim et al. Coupling-driven signal encoding scheme for low-power interface design. In *IEEE/ACM International Conference on Computer Aided Design. ICCAD - 2000.*, pages 318–321, Nov 2000.

[45] P. Stanley-Marbell, A. Alaghi, M. Carbin, E. Darulova, L. Dolecek, A. Gerstlauer, G. Gillani, D. Jevdjic, T. Moreau, M. Cacciotti, A. Daglis, N. Enright Jerger, B. Falsafi, S. Misailovic, A. Sampson, and D. Zufferey. Exploiting Errors for Efficiency: A Survey from Circuits to Algorithms. *ArXiv e-prints*, September 2018.

[46] John Sartori and Rakesh Kumar. Stochastic computing. *Foundations and Trends in Electronic Design Automation*, 5(3):153–210, 2011.

[47] C. E. Shannon. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(1):3–55, January 2001.

[48] S. Ramprasad, N. R. Shanbhag, and I. N. Hajj. A coding framework for low-power address and data busses. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 7(2):212–221, 1999.

[49] C. . Su, C. . Tsui, and A. M. Despain. Saving power in the control path of embedded processors. *IEEE Design Test of Computers*, 11(4):24–31, 1994.

[50] L. Benini, A. Macii, M. Poncino, and R. Scarsi. Architectures and synthesis algorithms for power-efficient bus interfaces. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(9):969–980, 2000.

[51] G. Ascia, V. Catania, M. Palesi, and A. Parlato. Switching activity reduction in embedded systems: a genetic bus encoding approach. *IEE Proceedings - Computers and Digital Techniques*, 152(6):756–764, 2005.

[52] Yan Zhang, J. Lach, K. Skadron, and M. R. Stan. Odd/even bus invert with two-phase transfer for buses with coupling. In *Proceedings of the*

International Symposium on Low Power Electronics and Design, pages 80–83, 2002.

[53] Chunjie Duan, Anup Tirumala, and S. P. Khatri. Analysis and avoidance of cross-talk in on-chip buses. In *HOT 9 Interconnects. Symposium on High Performance Interconnects*, pages 133–138, 2001.

[54] B. Victor and K. Keutzer. Bus encoding to prevent crosstalk delay. In *IEEE/ACM International Conference on Computer Aided Design. ICCAD 2001. IEEE/ACM Digest of Technical Papers (Cat. No.01CH37281)*, pages 57–63, 2001.

[55] A. Najafi, M. Weißbrich, G. Payá-Vayá, and A. Garcia-Ortiz. Coherent design of hybrid approximate adders: Unified design framework and metrics. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 8(4):736–745, Dec 2018.

[56] A. Najafi and A. Garcia-Ortiz. Stochastic Mixed-PR: A stochastically-tunable low-error adder. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 67(10):2144–2148, 2020.

[57] A. Ghofrani, A. Rahimi, M. A. Lastras-Montano, L. Benini, R. K. Gupta, and K. Cheng. Associative memristive memory for approximate computing in GPUs. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 6(2):222–234, June 2016.

[58] B. Zeinali, D. Karsinos, and F. Moradi. Progressive scaled stt-ram for approximate computing in multimedia applications. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 65(7):938–942, 2018.

[59] Jacob R. Stevens, Ashish Ranjan, and Anand Raghunathan. AxBA: An approximate bus architecture framework. In *Proceedings of the International Conference on Computer-Aided Design*, ICCAD '18, pages 43:1–43:8. ACM, 2018.

[60] A. Najafi, L. Bamberg, A. Najafi, and A. Garcia-Ortiz. Integer-value encoding for approximate on-chip communication. *IEEE Access*, 7:179220–179234, 2019.

[61] Purushottam Kulkarni, Fred Douglis, Jason LaVoie, and John M. Tracey. Redundancy elimination within large collections of files. In *Proceedings of the Annual Conference on USENIX Annual Technical Conference*, page 5, USA, 2004. USENIX Association.

[62] Purushottam Kulkarni, Fred Douglis, Jason LaVoie, and John M. Tracey. Redundancy elimination within large collections of files. In *Proceedings of the Annual Conference on USENIX Annual Technical Conference*, ATEC '04, page 5, USA, 2004. USENIX Association.

[63] D. J. Pagliari, E. Macii, and M. Poncino. Approximate differential encoding for energy-efficient serial communication. In *2016 International Great Lakes Symposium on VLSI (GLSVLSI)*, pages 421–426, 2016.

[64] A. Najafi, L. Bamberg, G. P. VayÃ¡, and A. Garcia-Ortiz. A coding approach to improve the energy efficiency of approximate nocs. In *2019 14th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC)*, pages 74–81, 2019.

[65] A. Najafi, L. Bamberg, A. Najafi, and A. Garcia-Ortiz. Misalignment-aware delay modeling of narrow on-chip interconnects considering variability. In *2018 7th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, pages 1–4, 2018.

[66] J. s. Seo, H. Kaul, R. Krishnamurthy, D. Sylvester, and D. Blaauw. A robust edge encoding technique for energy-efficient multi-cycle interconnect. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 19(2):264–273, Feb 2011.

[67] A. B. Kahng, S. Kang, R. Kumar, and J. Sartori. Recovery-driven design: A power minimization methodology for error-tolerant processor modules. In *Design Automation Conference (DAC), 2010 47th ACM/IEEE*, pages 825–830, June 2010.

[68] L. Wan and D. Chen. Dynatune: Circuit-level optimization for timing speculation considering dynamic path behavior. In *2009 IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers*, pages 172–179, Nov 2009.

[69] P. P. Sotiriadis and A. P. Chandrakasan. A bus energy model for deep submicron technology. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 10(3):341–350, June 2002.

[70] Lennart Bamberg and Alberto García-Ortiz. High-level energy estimation for submicrometric TSV arrays. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 25(10):2856–2866, 2017.

[71] C. N. Taylor, S. Dey, and Yi Zhao. Modeling and minimization of interconnect energy dissipation in nanometer technologies. In *Proceedings of the 38th Design Automation Conference (IEEE Cat. No.01CH37232)*, pages 754–757, June 2001.

[72] J. A. Davis and J. D. Meindl. Compact distributed rlc interconnect models-part ii: Coupled line transient expressions and peak crosstalk in multilevel networks. *IEEE Transactions on Electron Devices*, 47(11):2078–2087, 2000.

[73] Shang-Wei Tu, Jing-Yang Jou, and Yao-Wen Chang. Rlc coupling-aware simulation for on-chip buses and their encoding for delay reduction. In *2005 IEEE International Symposium on Circuits and Systems*, pages 4134–4137 Vol. 4, 2005.

[74] P. P. Sotiriadis and A. Chandrakasan. Reducing bus delay in submicron technology using coding. In *Proceedings of the ASP-DAC 2001. Asia and South Pacific Design Automation Conference 2001 (Cat. No.01EX455)*, pages 109–114, Feb 2001.

[75] Francesc Moll, Joan Figueras, and Antonio Rubio. Data dependence of delay distribution for a planar bus. pages 409–418, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[76] W. C. Elmore. The transient response of damped linear networks with particular regard to wideband amplifiers. *Journal of Applied Physics*, 19(1):55–63, 1948.

[77] R. Gupta, B. Tutuianu, and L. T. Pileggi. The elmore delay as a bound for rc trees with generalized input signals. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 16(1):95–104, 1997.

[78] E. Mensink. *High Speed Global On-Chip Interconnects and Transceivers*. PhD thesis, University of Twente, Netherlands, 6 2007.

[79] J. S. Miguel, J. Albericio, A. Moshovos, and N. E. Jerger. A cache for approximate computing. In *2015 48th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 50–61, 2015.

[80] A. Najafi, A. Najafi, and A. Garcia-Ortiz. Stochastic wave-pipelined on-chip interconnect. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 67(5):841–845, 2020.

[81] "the USC-SIPI image database: Version 6", allan weber, february 2018.

[82] Leonard W. Cotten. Circuit implementation of high-speed pipeline systems. In *Proceedings of the November 30–December 1, 1965, Fall Joint Computer Conference, Part I*, AFIPS '65 (Fall, part I), pages 489–504, New York, NY, USA, 1965. ACM.

[83] W. P. Burleson, M. Ciesielski, F. Klass, and W. Liu. Wave-pipelining: a tutorial and research survey. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 6(3):464–474, Sep. 1998.

[84] A. Datta, A. Mohanta, and B. Mishra. Approximation and bit truncation techniques in hardware for edge detection. In *2019 9th International Symposium on Embedded Computing and System Design (ISED)*, pages 1–5, 2019.

[85] W. R. Bennett. Spectra of quantized signals. *The Bell System Technical Journal*, 27(3):446–472, 1948.

[86] Kevin M. Lepak, Irwan Luwandi, and Lei He. Simultaneous shield insertion and net ordering under explicit rlc noise constraint. In *Proceedings of the 38th Annual Design Automation Conference*, pages 199–202, New York, NY, USA, 2001. Association for Computing Machinery.

[87] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014.

[88] Timothy J. O'Shea and Johnathan Corgan. Convolutional radio modulation recognition networks. *CoRR*, abs/1602.04105, 2016.

[89] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov 2004.

[90] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 963–968, 2011.

[91] A Geiger, P Lenz, C Stiller, and R Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[92] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. ICDAR 2003 robust reading competitions. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, pages 682–687, Aug 2003.

[93] R. Smith. An overview of the Tesseract OCR engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 629–633, Sep. 2007.

[94] Oreste Villa, Daniel R. Johnson, Mike O'Connor, Evgeny Bolotin, David Nellans, Justin Luitjens, Nikolai Sakharnykh, Peng Wang, Paulius Micikevicius, Anthony Scudiero, Stephen W. Keckler, and William J. Dally. Scaling the power wall: A path to exascale. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 830–841. IEEE Press, 2014.

[95] B. Bohnenstiehl, A. Stillmaker, J. J. Pimentel, T. Andreas, B. Liu, A. T. Tran, E. Adeagbo, and B. M. Baas. Kilocore: A 32-nm 1000-processor computational array. *IEEE Journal of Solid-State Circuits*, 52(4):891–902, 2017.

[96] Shekhar Borkar. Future of interconnect fabric: A contrarian view. In *Proceedings of the 12th ACM/IEEE International Workshop on System Level Interconnect Prediction*, pages 1–2, New York, NY, USA, 2010. Association for Computing Machinery.

[97] Hangsheng Wang, Li-Shiuan Peh, and S. Malik. Power-driven design of router microarchitectures in on-chip networks. In *Proceedings. 36th Annual IEEE/ACM International Symposium on Microarchitecture, 2003. MICRO-36.*, pages 105–116, 2003.

[98] Y. Chen and A. Louri. An approximate communication framework for network-on-chips. *IEEE Transactions on Parallel and Distributed Systems*, 31(6):1434–1446, 2020.

[99] S. R. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, V. Erraguntla, C. Roberts, Y. Hoskote, N. Borkar, and S. Borkar. An 80-tile sub-100-w teraflops processor in 65-nm cmos. *IEEE Journal of Solid-State Circuits*, 43(1):29–41, Jan 2008.

[100] E. Beigne, F. Clermidy, H. Lhermet, S. Miermont, Y. Thonnart, X. Tran, A. Valentian, D. Varreau, P. Vivet, X. Popon, and H. Lebreton. An asynchronous power aware and adaptive noc based circuit. *IEEE Journal of Solid-State Circuits*, 44(4):1167–1177, April 2009.

[101] H. Matsutani, M. Koibuchi, D. Ikebuchi, K. Usami, H. Nakamura, and H. Amano. Performance, area, and power evaluations of ultrafine-grained run-time power-gating routers for cmps. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 30(4):520–533, April 2011.

[102] G. Ascia, V. Catania, S. Monteleone, M. Palesi, D. Patti, and J. Jose. Improving energy consumption of noc based architectures through approximate communication. In *2018 7th Mediterranean Conference on Embedded Computing (MECO)*, pages 1–4, 2018.

[103] D. Bertozzi, L. Benini, and G. De Micheli. Low power error resilient encoding for on-chip data buses. In *Proceedings 2002 Design, Automation and Test in Europe Conference and Exhibition*, pages 102–109, 2002.

[104] M. Dehyadgari, M. Nickray, A. Afzali-kusha, and Z. Navabi. Evaluation of pseudo adaptive xy routing using an object oriented model for noc. In *2005 International Conference on Microelectronics*, pages 5 pp.–, 2005.