ENERGY EFFICIENT ENCODING METHODS FOR CHIP-TO-CHIP COMMUNICATION

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF SCIENCE AND ENGINEERING

2022

Eleni Maragkoudaki

Department of Computer Science

Contents

Word Count: 33456

List of Tables

List of Figures

List of Acronyms

3-D Three-Dimensional

2.5-D 2.5-Dimensional

I/O Input/Output

ICs Integrated Circuits

ICT Information and Communication Technologies

PCIe Peripheral Component Interconnect Express

VLSI Very Large Scale Integration

EM Electromagnetic

FPGA Field Programmable Gate Array

SNR Signal to Noise Ratio

EMI Electromagnetic Interference

TSVs Through Silicon Vias

HBM High Bandwidth Memory

DRAM Dynamic Random Access Memory

V-NAND Vertical-NAND

CAD Computer Aided Design

GPU Graphic Processing Unit

RDLs Redistribution Layers

BEOL Back-End-of-Line

EMIB Embedded Multidie Interconnect Bridge

ISI Intersymbol Interference

BI Bus Invert

DDR4 Double-Data-Rate fourth generation

PBI Partial Bus Invert

APBI Adaptive Partial Bus Invert

TTA Total Transition Accumulator

WC Window Counter

MUC Mask Update Counter

SINV Shift Invert

BS Bus Shifting

ABE Adaptive Bus Encoding

FV Frequent Value

CAM Content Addressable Memory

ADES Adaptive Dictionary Encoding Scheme

MSBs Most Significant Bits

MTF Move-To-Front

DESC Data Exchange using Synchronized Counters

ATE Adaptive Time-based Encoding

STFL Slow Transition Fast Level

BEAM Bus Encoding based on instruction-set-Aware Memories

POD Pseudo Open Drain

GDDR5 Graphics Double Data Rate 5

DBI Data Bus Inversion

MiL More is Less

BD Bitwise-Difference

CAFO Cost-Aware Flip Optimization

PCM Phase Change Memory

STT-RAM Spin-Transfer Torque Random Access Memory

I^{2}C Ifrijf

SPI Serial Peripheral Interface

SILENT Serialized Low-Energy Transmission

TIC Transition Inversion Coding

ADE Approximate Differential Encoding

LSBs Least Significant Bits

VDBS Value-Deviation-Bounded Serial

AXSERBUS Approximate Serial Bus

SerDes Serialiser/Deserialiser

USB Universal Serial Bus

CDR Clock Data Recovery

PRBS Pseudorandom Bit Sequence

LFSR Linear Feedback Shift Register

PCB Printed Circuit Board

TPV Through Package Via

ESD Electrostatic Discharge

ABF Ajinomoto Build-up Film

DLST-TX Dynamic Low Swing Tunable Transmitter

IVT-RX Inverter-based Tunable Receiver

LS Low Swing

FS Full Swing

AWR Adaptive Word Reordering

NN Nearest Neighbour

DMA Direct Memory Access

LPDDR3 Low-Power Double Data Rate 3

TSP Travelling Salesman Problem

TT Typical/Typical

STTE Serial Tuned Transition Encoding

CRC Cyclic Redundancy Check

MPSoC Multiprocessor System on Chip

SS Spiral Strip

PHY Physical Layer

HDL Hardware Description Language

SAIF Switching Activity Interchange Format

CPUs Central Processing Units

LLC Last Level Cache

Advanced eXtensible Interface AXI

Abstract

ENERGY EFFICIENT ENCODING METHODS FOR CHIP-TO-CHIP COMMUNICATION
Eleni Maragkoudaki
A thesis submitted to The University of Manchester
for the degree of Doctor of Philosophy, 2022

As traditional scaling slows down, the number of cores and the amount of memory per system increase to satisfy the performance demand. This drive for more parallelism increases data movement requirements rapidly. However, as technology scales the energy dissipated in data communication scales at a much slower pace compared to computation energy. Therefore, new methods to reduce the energy of data transmission are explored in this thesis.

Three different techniques that decrease the dynamic power of interconnects are discussed: low-swing signalling, 3-D and 2.5-D integration technologies that reduce the interconnect length, and signal encoding. An investigation of the energy benefits and limitations of low-swing signalling when applied to interposer technologies is provided, where different interposer materials are considered. Although the potential energy savings are high, low-swing signalling is susceptible to noise or induces high area penalty. Therefore, this thesis focuses on signal encoding, which reduces the bit transitions of the transmitted data to save power.

Two encoding schemes are proposed, named Adaptive Word Reordering (AWR) and Serial Tuned Transition Encoding (STTE), for parallel and serial interfaces, respectively. AWR achieves a high decrease in transitions and a novel custom circuit implementation is provided to constrain the overhead in power. The power savings of AWR reach up to 23% for a 1 mm interposer-based interconnect without affecting the communication bandwidth. Alternatively, STTE is designed for source asynchronous

serial interfaces, where the receiver recovers the clock from the incoming data. STTE regulates the number of transitions such that the clock can be reliably recovered while the communication energy is lowered. STTE provides at least 25% decrease in energy for a 1 mm interposer-based interconnect compared to scrambling, which is typically used in these interfaces. The ability to maintain clock recovery and, thus, link integrity is evaluated experimentally using both an electrical and an optical link that interconnect two FPGA devices.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Acknowledgements

First and foremost, I would like to thank my main supervisor Dr. Vasilis F. Pavlidis for his guidance throughout my PhD. He consistently provided advice and motivated me to develop and research my own ideas. His constructive criticism, his experience in research, and his knowledge on integrated circuits helped me become a better researcher.

A big thank you to Przemyslaw Mroszcz and Nguyen Dao for the technical support at different stages of my PhD. Their technical advice truly helped me overcome many challenges. I would also like to thank Will Toms for the fruitful discussions and collaboration.

Furthermore, I am thankful to Dr. Jim Garside, Dr. Dirk Koch, and my co-supervisor Dr. Christoforos Moutafis for their valuable advice and discussions about my research at the first and second year examinations as well as throughout my PhD. I would also like to thank all my colleagues from the APT group and especially Minmin Jiang for creating a pleasant and productive working environment. I am grateful to the European Commission and Pr. Mikel Luján for providing the funding for my research through the EuroEXA project.

Additionally, I would like to express my gratitude to my friends Danae, Rigina, Katerina, Eleni, Christos, Mary, Markos, Ioanna, and Thanos for their support and understanding and for making this journey enjoyable. I am also thankful to Joe for his support during the writing of this thesis.

Last but not least, I am deeply grateful to my family for supporting me and especially my mother, Popi, for believing in me and inspiring me to forge my own path. I dedicate this thesis to her.

Chapter 1

Introduction

1.1 Trends of Modern Integrated Circuits

Traditional transistor scaling has been slowing down due to manufacturing challenges including process variations, mask cost, and difficulty in shrinking the gate length [1]. Alternative routes are explored by the semiconductor industry to satisfy the emerging performance demand of data-intensive applications, such as machine learning. The clock frequency has plateaued and replaced by increasing the number of processing cores per system as well as memory to provide the required performance [2].

The integration of multiple dies in a single package has also facilitated the increase in core count and functional density as depicted in Figure 1.1. Advanced packaging technologies, such as wire bonding, three-dimensional (3-D) and 2.5-dimensional (2.5-D) integration, are utilised to interconnect multiple dies in close physical proximity, thus, reducing form factor [6]. This increase in processing cores and memory and the drive for more parallelism has led to an increasing demand of data movement and communication bandwidth.

Data bandwidth has been increasing rapidly in various systems from smartphones to supercomputers. This trend is illustrated in Figure 1.2, where the data rate per lane of a wide range of input/output (I/O) standards is depicted for the past 22 years. This exponential growth in data bandwidth poses significant challenges in terms of signal integrity and especially power consumption.

Constraining the energy demand of Integrated Circuits (ICs) is more important than ever. The Information and Communication Technologies (ICT) sector is responsible for a large part of the global electricity demand. Around 2% of the global carbon emissions is attributed to the ICT sector [3]. Due to the widespread use of ICs, from

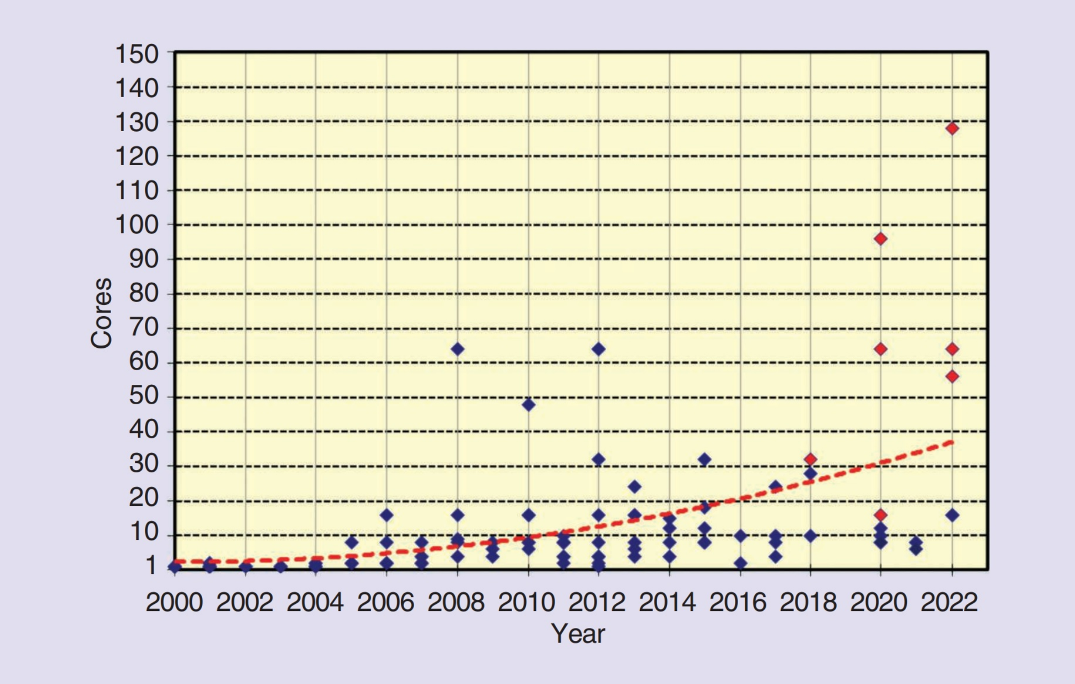


Figure 1.1: Number of cores over the years. The red diamonds denote multichip modules [2].

personal digital devices and sensors to data centers, low-power circuit design has to become even more sophisticated to support sustainability. Low-power circuit design is, also, essential for battery-operated devices, such as smartphones. These devices require significant performance and high communication bandwidth. Hence, increasing energy efficiency is required to extend battery longevity.

1.2 Power Consumption of Data Communication

A large part of the energy of ICs is dissipated in data communication. The energy dissipated in data transmission is estimated between 18% and 40% of the total system energy for scientific applications and could further increase in upcoming exascale systems [43]. Alternatively, in smartphone devices, the energy spent on moving data across the memory hierarchy is 34.6%, on average, and reaches 41% for realistic web browsing [45]. Consequently, data transmission becomes the bottleneck in satisfying power constraints of ICs.

An imbalance between computation and communication energy has emerged with

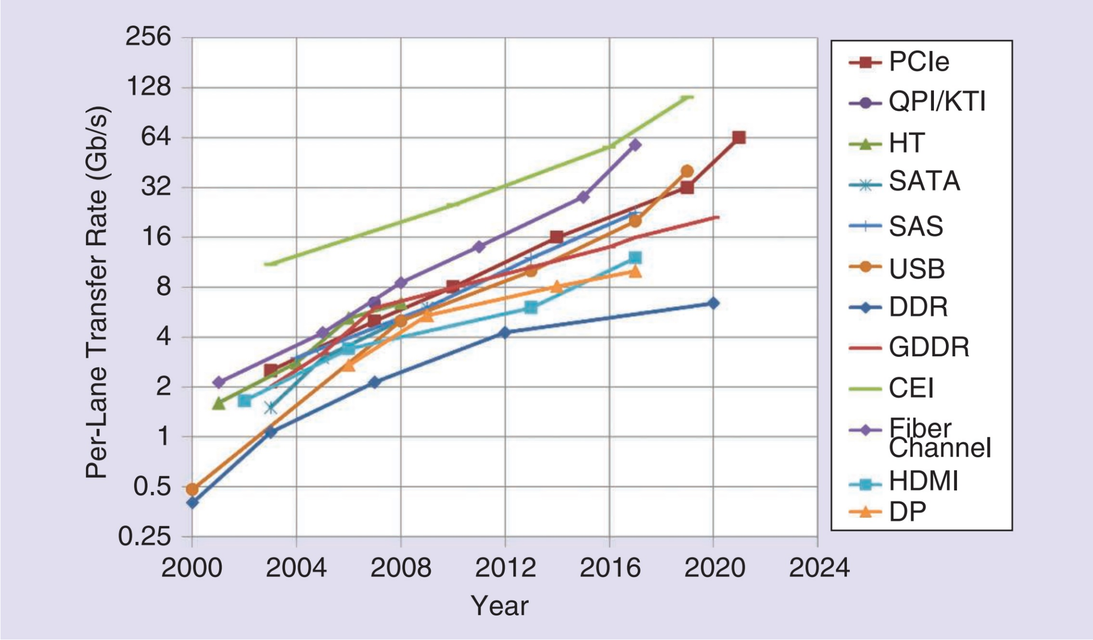


Figure 1.2: Data rate per lane of various I/O standards over the years [2].

technology scaling. The energy of data movement scales at a much slower pace compared to computation energy [44], [42]. For instance, the energy of a load instruction that requires transmitting data off-chip is estimated 115× higher than the energy of an add operation in smartphone devices [45]. The reason is that wires and I/O pins scale gradually, therefore, interconnect capacitance scales at a slower pace compared to gate capacitance [4]. Thus, a larger amount of power is dissipated for charging and discharging interconnect capacitance. Hence, due to the increasing demand of communication bandwidth as shown in Fig. 1.2, the power of inter-die communication becomes a key challenge of modern ICs.

The static power of ICs becomes more prominent with technology scaling as the gate capacitance decreases. However, the interconnect capacitance does not scale as much [4], therefore, the dynamic power dissipated in charging and discharging the interconnect capacitance is a major contributor of the total power consumption. Hence, in this thesis, the reduction of the dynamic power consumption of off-die communication is explored. Nevertheless, the developed low-power techniques are evaluated in terms of the total power as results include both static and dynamic power throughout the thesis.

Different techniques that decrease power are discussed: low voltage swing transmission, advanced packaging technologies, and signal encoding methods. Low-swing signalling reduces the voltage range of transmitted signals. Although the potential power improvement is significantly high (4× decrease in power over full swing communication [30]), low-swing signalling is susceptible to noise or induces high area overhead, such as double number of I/O pins and wires [25]. 3-D and 2.5-D are advanced integration technologies that support the integration of multiple dice and decrease form factor [6]. These technologies effectively reduce the interconnect length and, thus, capacitance, however, introduce manufacturing challenges or suffer from high cost.

Alternatively, signal encoding methods decrease the switching activity, *i.e.* bit transitions, of the transmitted signals [5]. Therefore, the frequency of charging and discharging the interconnect load is reduced and, thus, power is saved. An encoder and a decoder logic is added to the transmitter and the receiver, respectively, to reduce bit transitions and retrieve the original data. In this way, a low amount of on-chip computation power is exchanged for a decrease in I/O power. With technology scaling, the effectiveness of signal encoding is expected to increase, as I/O power scales disproportionately to computing power. Hence, the impact of the overhead due to encoding and decoding decreases for smaller process nodes.

The focus of this thesis is particularly on signal encoding. This work is of interest to digital and mixed-signal IC designers of data encodings for off-die communication. Two types of data transmission are targeted. The first type is wide parallel links implemented on advanced packaging technologies that support high wire density, such as interposers and Embedded Multidie Interconnect Bridge (EMIB) [96]–[97]. The second type of communication is high-speed, source asynchronous, Serialiser/Deserialiser (SerDes) interfaces. An encoding technique is proposed that can be integrated in the physical layer of a SerDes protocol, where voltage-mode drivers are utilised with one-to-one mapping between digital signals and transmitted symbols, such as Stub Series Terminated Logic (SSTL) [116].

Encoding schemes tailored to parallel buses reduce bit transitions between consecutive data words. A variety of encoding schemes have been proposed for parallel interconnects [46], [48], [51]-[81], [85]-[91]. The majority of the techniques exploit specific characteristics of the data stream to reduce transitions, such as data locality or the sequential nature of instruction addresses. These techniques are effective only for specific applications. On the contrary, general purpose encoding schemes either provide a moderate reduction in bit transitions or exhibit substantial encoding/decoding overhead. Thus, the latter allow savings only for long interconnects with large loads.

In serial communication, the target of encoding schemes is the decrease of transitions within data words. Existing low-power encoding techniques [104]-[110] are only suitable for source synchronous interconnects, despite the omnipresence of high-speed, source asynchronous serial interfaces, such as Ethernet and Peripheral Component Interconnect Express (PCIe) [111]. In source asynchronous interfaces, the receiver extracts the clock signal from the incoming data stream, instead of relying on an external clock signal. Consequently, the transmitted data must exhibit frequent bit transitions to facilitate clock recovery [101], [111]. Therefore, the minimisation of bit transitions in this type of interconnects obstructs clock recovery and degrades link integrity.

1.3 Contributions and Publications

A large part of the material in this thesis has been published. The contributions and publications are the following:

- E. Maragkoudaki, P. Mroszczyk, and V. F. Pavlidis, "Energy Efficiency of Low Swing Signaling for Emerging Interposer Technologies," *Proceedings of the IEEE/ACM Conference on Nanoscale Architectures*, pp. 124–130, July 2018.
  The effectiveness of low-swing signalling on 2.5-D integration scenarios is investigated. The energy consumed by a low swing scheme is, therefore, compared with a full swing solution. An electrical model of an interposer-based interconnect is developed to quantify the energy improvement as well as the limitations of low-swing signalling. Specifically, the critical length of the interconnect, above which the low swing solution starts to pay off is determined. Three different interposer materials are investigated: silicon, glass, and organic. The low swing solution provides higher energy savings and achieves the shortest critical length for silicon interposers, which have the highest wire capacitance. Furthermore, the effect of typical interconnect parameters such as width and spacing on the energy efficiency of low swing communication is, also, evaluated. In this way, useful rules of thumb are provided regarding the use of low swing communication mechanisms for 2.5-D integrated systems.

- E. Maragkoudaki, P. Mroszczyk, and V. F. Pavlidis, "Adaptive Word Reordering for Low-Power Inter-Chip Communication," *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 980–983, March 2019.

A general purpose encoding technique named Adaptive Word Reordering is proposed to reduce the power consumption of parallel interfaces by changing the order of data. This new scheme reduces bit transitions up to 48% for diverse data types without affecting the communication throughput. A novel encoder circuit is proposed, which exploits the time domain for complex computations of Hamming distances, to decrease the power overhead of encoding. AWR achieves significant power savings, up to $\sim$23% for 1 mm interposer-based interconnects. The proposed scheme outperforms state-of-the-art encoding techniques in terms of reduction in transitions.

- E. Maragkoudaki and V. F. Pavlidis, "Energy-Efficient Time-Based Adaptive Encoding for Off-Chip Communication," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 28, No. 12, pp. 2551–2562, Dec. 2020.
  In this publication, the effectiveness and limitations of the proposed encoding technique are further investigated. A comparison with state-of-the-art techniques in terms of savings in power as well as delay and latency overheads for diverse benchmark applications is provided. The power of the encoding and decoding circuits for each technique is also included, thus, an accurate estimation of the efficiency of each technique is provided. Furthermore, the power benefits of using the time domain for the complex computations are quantified. Additionally, the resilience of the proposed encoder and decoder circuits under process variations is explored. The trade-offs between accuracy, power, and timing are discussed.

- M. Jiang, E. Maragkoudaki, and V. F. Pavlidis, "Mitigating EM Side-Channel Attacks with Dynamic Delay Insertion and Data Bus Inversion," *Proceedings of the IEEE International Symposium on Circuits and Systems*, May 2022.
  In this work, a technique is proposed to mitigate security vulnerabilities in cryptographic circuits. These circuits are susceptible to electromagnetic (EM) side-channel attacks, which detect EM emissions to extract sensitive data. The proposed technique reduces the correlation between processed data and electromagnetic emissions by dynamically inserting a short delay (up to 110 ps) to signals based on the Hamming Distance of consecutive data. My contribution to this work is the implementation of the circuit of the proposed dynamic delay insertion scheme. This work is not presented in the thesis as the scope is beyond my PhD.

- E. Maragkoudaki, W. Toms, and V. F. Pavlidis, "Energy-Efficient Encoding for

High-Speed Serial Interfaces," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Submitted.
An encoding technique for high-speed, source asynchronous serial interfaces is proposed that reduces energy while maintains link integrity. Typically in these interfaces, techniques that insert bit transitions, such as scrambling, are employed to facilitate clock recovery. On the contrary, this scheme regulates the frequency of transitions such that the receiver can recover the clock from the transmitted data, while the number of transitions is reduced compared to scrambling and, hence, energy is saved. The proposed technique is investigated in terms of both energy-efficiency as well as maintaining link integrity to avoid data errors. The energy savings are estimated for a 1 mm interposer-based interconnect, whilst the link integrity is experimentally evaluated using both an electrical and an optical link that interconnect two Field Programmable Gate Array (FPGA) boards.

1.4 Thesis Organisation

The rest of this thesis is organised as follows. In Chapter 2, the techniques for decreasing the dynamic power consumption of off-die interconnects are presented. Particularly, three methods are discussed, low-swing signalling, 3-D and 2.5-D integration that reduce the length of off-die links, and signal encoding. The latter is the main focus of this thesis, therefore, a review of existing encoding methods for both parallel and serial links is provided. Techniques for parallel buses are classified to static and adaptive, while serial interfaces are divided into source synchronous and asynchronous and encoding schemes for both categories are reviewed. The strengths and limitations of all techniques are highlighted.

In Chapter 3, the energy improvement and limitations of low-swing signalling are investigated for interposer-based interconnects. The energy efficiency of a low-swing scheme is estimated for different interposer materials, interconnect lengths, and wire densities.

A general purpose encoding scheme for parallel interfaces is proposed in Chapter 4 that reduces bit transitions for diverse data types. The circuit implementations of the encoder and the decoder are presented. The effectiveness of the technique is investigated in terms of power for an interposed-based link where the power overhead of encoding and decoding is also considered. The limitations and timing overheads of this technique are discussed.

Existing low-power encoding techniques for serial interconnects are only applicable to source synchronous interfaces. Therefore, an encoding technique for source asynchronous serial links is presented in Chapter 5. The energy savings are estimated for an interposer-based interconnect, while the link integrity is experimentally evaluated for both an electrical and an optical interconnect. The trade-offs between energy, link integrity, and throughput are discussed.

A summary of the thesis is provided in Chapter 6 and conclusions are drawn. Directions for future research topics are also presented.

Chapter 2

Power of Die-to-Die Communication

As technology scales, data movement becomes increasingly the bottleneck in satisfying the power constraints of modern processors. The ever growing compute density, enabled by parallelism, in combination with the non-decreasing system size has led to an excessive demand for data movement through long interconnects [42]. The energy cost for transferring data across the memory hierarchy is estimated between 18% and 40% of the total system energy for scientific applications and is expected to further increase in upcoming exascale systems [43]. In addition, the energy of transferring data does not scale as fast as the energy for computation [44]. For instance, the energy consumed for off-chip communication is estimated 115× higher than that of an add operation in smart phone devices [45]. Thus, the power reduction of inter-die communication is a key challenge for modern integrated systems.

The static power of ICs becomes more significant compared to dynamic power as technology scales since the capacitive load of gates shrinks. However, this trend does not apply to communication power. The reason is that wires and, thus, the interconnect capacitance do not scale as fast [4]. Consequently, to increase the energy efficiency of off-die communication, the dynamic power consumption has to be restricted, which is given by:

$$P = aC_L V_{DD}^2 f, \tag{2.1}$$

where a is the average switching activity, C_L is the load capacitance, V_{DD} is the supply voltage, and f is the operating frequency. Hence, the decrease of any of these factors is required to limit the dynamic power consumption.

There is a plethora of techniques for reducing the power demand of off-die communication, such as data-centric architectures that reduce the number of memory accesses and data movement, reduction of the voltage level, packaging technologies that decrease the distance between dies, and signal encoding schemes that decrease the switching activity factor. In this thesis, the focus is on the reduction of the energy cost of data transfers, therefore, techniques that decrease the factors of Equation 2.1 are discussed in this chapter.

One way to reduce the dynamic power dissipation is to decrease frequency by employing frequency throttling. This technique is useful to restrict power to control temperature and prevent overheating. However, this method is not effective in decreasing the energy demand [42]. Therefore, this mechanism is not investigated in this work.

Techniques that target the rest of the parameters are discussed in the following sections. Particularly, low-swing signalling schemes that reduce the voltage range of the signals transmitted over the wires are described in Section 2.1. In Section 2.2, emerging packaging technologies are presented that contribute in increasing the physical proximity of dies and, thus, reducing the capacitive load of the wirelines. The benefits as well as the limitations of low-swing signalling and packaging technologies are highlighted.

Another category of power reduction techniques, called signal encoding, decrease the average switching activity of the transmitted data. Low-power signal encoding is the main focus of this thesis, which can be categorised to encoding for parallel and serial interfaces. The characteristics of the two types of interfaces are described and a thorough review of the state-of-the-art encoding techniques is provided in Sections 2.3 and 2.4 for parallel and serial communication, respectively.

2.1 Low-Swing Signalling

Power exhibits a quadratic relationship with voltage, thus, one of the most effective ways to reduce the dynamic power consumption of signal transmission is to decrease the voltage range. The implementation of low-swing signalling is illustrated in Fig. 2.1. The driver converts the signal from full-swing to low-swing. The signal is transferred over the interconnect in a lower voltage range, while the receiver at the other end of the link fully restores the voltage swing.

A number of low-swing schemes have been proposed in the literature. The majority of these schemes rely on additional power supply or reference voltage and multiple threshold voltage devices for the voltage level conversion [25], [26]. Although the dual power rail schemes exhibit high signal to noise ratio (SNR), they suffer from high area and delay penalties [26]. Furthermore, the complexity of the physical design

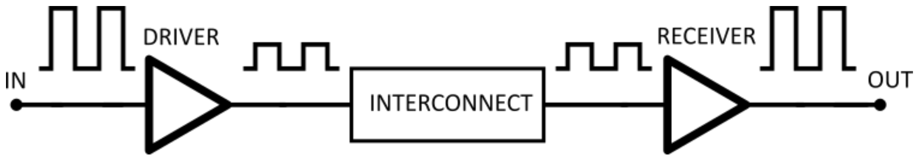


Figure 2.1: Low-swing signalling implementation.

is increased by the additional power supply and the technology node portability is confined due to the requirement of multi-threshold devices [28].

Alternatively, single supply, low-swing schemes address the drawbacks of the dual power rail. A set of single supply techniques exploit the threshold voltage of transistors to limit the voltage swing [27]–[29]. The energy efficiency of these techniques is limited since the energy savings depend on the threshold voltage of transistors, which is not that low compared to the supply voltage, especially for modern, scaled-down technologies. The last group of single supply schemes achieve the desired level of voltage by restraining the charging and discharging time of the load capacitance [30]–[33]. These circuits exhibit low voltage swing and low circuit complexity, therefore, high power savings. However, these schemes exhibit low SNR and are susceptible to crosstalk noise as well as process and environmental variability [25]. An additional restriction of these schemes is that the capacitive load must be known at design time, which is not always feasible, especially for off-die interconnects.

To enhance noise immunity of signal transmission, differential signalling has been widely employed. Two complementary signals are transmitted on a pair of wires to indicate the state of the transferred information. The receiver calculates the voltage difference of the pair of complementary signals instead of the difference between the data signal and the ground as in single-ended transmission. Differential signalling is highly resistive to crosstalk and electromagnetic interference (EMI) compared to single-ended transmission. The reason is that external noise similarly affects both wires, thus, is cancelled out [34]. Because of the noise-resilient nature of differential signalling, a further reduction in the signal swing can be achieved compared to single-ended signalling [25].

Although the potential power reduction is high by employing low-swing signalling reaching up to 4× [30], there are some important considerations. The major concern is reliability as single-ended, low-swing signalling is susceptible to noise. On the contrary, techniques that offer improved noise immunity have important limitations. Dual power rail methods depend on additional supply voltage for the voltage conversion, which induces a large area penalty and increases the physical design complexity. Differential signalling requires the double number of wires and I/O pins, which is an important limitation from both an area and a power perspective especially, as the number of I/Os does not scale as fast with technology evolution.

2.2 Increasing Physical Proximity

Technology scaling has increased computational density, but, at the same time, has increased data movement over longer interconnects. Accessing memory is one of the primary bottlenecks in terms of both power and performance [6]. The cost of data movement can be alleviated by placing memory modules closer to processing elements, thus, increasing the physical proximity. A primary way to improve physical proximity and, therefore, decrease the interconnect length is 3-D and 2.5-D integration, described in subsections 2.2.1 and 2.2.2, respectively.

2.2.1 Three-Dimensional Integration

In 3-D integration, the third dimension is exploited by stacking multiple dice, which are vertically interconnected. There are various ways of interconnecting the different tiers, such as wire bonding or inductive links, however, the most common approach is the utilisation of Through Silicon Vias (TSVs) [7]. An example of 3-D integration is depicted in Fig. 2.2, where memory dies are vertically stacked and interconnected using TSVs and microbumps (μbumps).

3-D integration offers significant benefits in terms of performance, size, and heterogeneity. The primary advantage of 3-D integration is the drastic decrease in interconnect length, enabled by vertically stacking dice [8]. Shorter wire length leads to a decrease in delay and power dissipation since the capacitive load is lower. Another important benefit of 3-D ICs is the ability to combine heterogeneous tiers with disparate fabrication processes, such as sensors, analog and digital processors, and memories, in a single low form factor package. In this way, the system footprint is reduced compared to planar approaches [9]. Health monitoring, military, security, and environmental monitoring are just a few areas that can benefit from this heterogeneous integration paradigm. Finally, TSVs provide high connectivity density, thus, increased data bandwidth and performance.

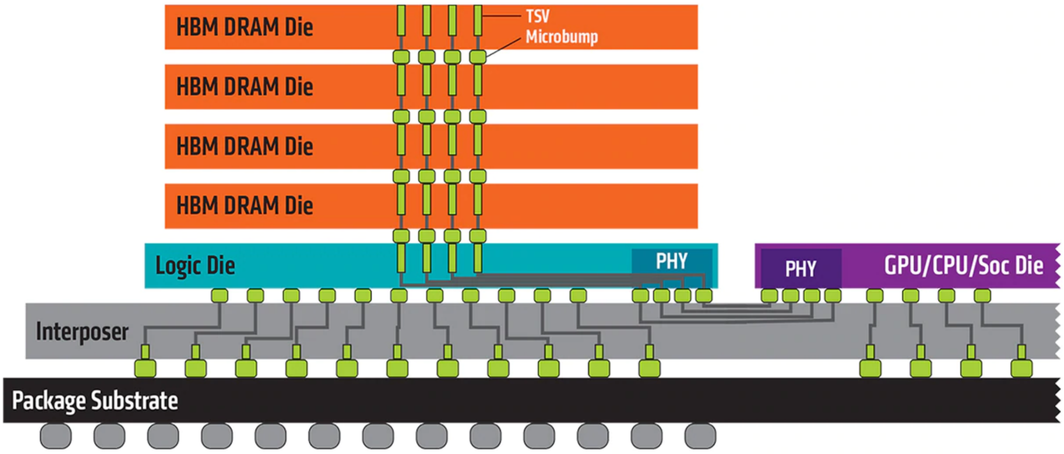


Figure 2.2: Schematic of AMD's High Bandwidth Memory [11].

The benefits and versatility of 3-D integration using TSVs has led to the commercialisation of this technology. Examples of commercial 3-D systems are High Bandwidth Memory (HBM) [10], [11], where dynamic random-access memory (DRAM) dies are vertically stacked and interconnected with TSVs and μbumps as depicted in Fig. 2.2, Hybrid Memory Cube [12], where four DRAM dice and one logic die are vertically stacked using TSVs in a single package, and Samsung's Solid State Drives using vertical-NAND (V-NAND) technology [13].

Despite the commercial success of 3-D systems, there are important challenges that restrict the applicability of this technology. Thermal issues are of major concern, as the high volumetric density of devices increases the power density and, thus, temperature [6]. Additionally, manufacturing complexities, testing for wafer level integration, design of global interconnects, and the development of efficient Computer Aided Design (CAD) tools for placement and routing are some of the challenges of 3-D integration [6]. These issues have limited the usage of 3-D integration mainly in memory systems.

2.2.2 2.5-Dimensional Integration

2.5-D integration is an alternative solution that resolves the manufacturing and thermal issues of 3-D ICs [14], whilst provides fine pitch interconnections. 2.5-D integration introduces a new substrate layer, named interposer, which interconnects the dice and the package substrate layer. An interposer is used in AMD's HBM to link the processing unit with the logic die of the memory stack and the package substrate as illustrated in Fig. 2.2. Other examples of commercial products employing interposer technologies are Xilinx Virtex-7 FPGA [15] and NVIDIA Tesla P100 Graphic Processing Unit (GPU) accelerator [16].

Interposers utilise μbumps and Redistribution Layers (RDLs) to interconnect the hosted dice. RDLs are horizontal metal layers similar to Back-End-of-Line (BEOL) interconnects, therefore, exhibit low pitch. Interposers provide high integration density and smaller form factor compared to traditional packaging technologies, as the size of interconnects, vias, and bumps is decreased. Therefore, higher bandwidth and lower power consumption are achieved [38].

The electrical characteristics of interposer-based interconnects are affected by the material of the interposer substrate. Silicon is the most widespread material used for interposers among the semiconductor fabrication processes, offering fine metal routing [17]. However, silicon substrate suffers from high cost and high insertion loss due to high conductivity [17]. Organic materials are a useful alternative that have been used for IC packaging. However, organic substrates require large pads, therefore, they are unsuitable for interconnecting high I/O density dice [18]. Finally, glass is a promising solution as exhibiting good electrical properties, such as low insertion loss and high resistivity, [19] and the cost per I/O is lower compared to silicon. Glass interposers also pose challenges, including cost of via formation and inferior thermal characteristics compared to silicon [19].

2.3 Signal Encoding for Parallel Transmission

Although 3-D and 2.5-D technologies increase the physical proximity of the interconnected dies, the high power of interconnects remains a major problem. Therefore, alternative low-power techniques are employed as well. An effective method to decrease the dynamic power is to reduce the switching activity, a of Equation 2.1. This aim can be effectively implemented through signal encoding. An encoder circuit is added to the transmitter to reduce bit transitions and a decoder is included at the receiver to restore the data as illustrated in Fig. 2.3 for a single bus lane. Signal encoding techniques have been developed for both parallel and serial data communication. The focus of this section is on parallel communication, while encoding for serial interfaces is discussed in Section 2.4.

Parallel transmission is the most straightforward way of transferring data between dies as each bit of data is transmitted simultaneously on a separate bus lane. In addition, parallel communication is source synchronous, *i.e.* a clock signal is transmitted along with data that is utilised by the receiver to sample the incoming data. Hence, parallel transmission provides better design simplicity over serial transmission, since no added circuitry is required for serialisation/deserialisation or clock recovery. Wide parallel links are especially preferred in advanced packaging technologies such as 2.5-D and 3-D integration and EMIB [96]–[97], where higher wire densities are supported compared to Printed Circuit Board (PCB) interconnects.

Encoding techniques tailored to parallel interfaces reduce the power dissipation of data transmission by decreasing the bit transitions between consecutively transmitted data words. This aim is accomplished without affecting noise resilience and without inducing any manufacturing challenges. In addition, encoding schemes become more beneficial with the scaling of technology. By employing signal encoding, a significant portion of the power dissipated in interconnects can be saved at the expense of a low increase in on-chip power for encoding and decoding. As technology scales, the power dissipated for computation decreases faster compared to the interconnect power [44]. Consequently, the impact of the added power for encoding and decoding on the overall savings in power diminishes.

Although some techniques aim at latency reduction [50], the vast majority of encoding methods target primarily to decrease power. The focus of this thesis is on the latter objective. Encoding schemes can be classified to either static or adaptive. Static schemes exploit the statistical properties of data streams, considered known at design time, and are discussed in subsection 2.3.1. Alternatively, adaptive schemes assume that there is no prior knowledge of these properties, observe data at runtime, and apply the appropriate encoding. These methods are described in subsection 2.3.2.

2.3.1 Static Encoding Schemes

The main requirement of static schemes is that the statistical properties of the transmitted data must be known at design time. An analysis of these properties is conducted before static schemes are developed. In this way, static schemes usually exhibit lower

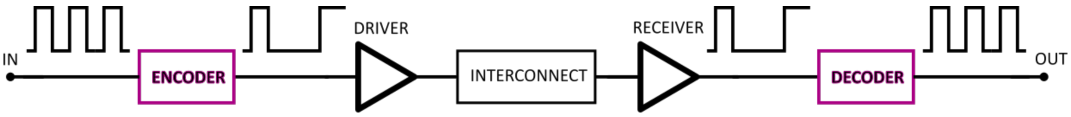


Figure 2.3: Basic structural diagram for signal encoding.

overheads compared to adaptive. Static techniques that decrease bit transitions of both address buses and data buses have been proposed.

The transmitted signals in address buses tend to be sequential as both data and instructions are usually stored in consecutive memory addresses. This attribute is exploited by some encoding techniques to reduce the number of transitions. Bit transitions in this type of buses can be reduced by encoding addresses with Gray code [51]. Gray code ensures a single bit transition between consecutive addresses. T0 code is another encoding technique that is based on the observation that addresses are usually incremental with a fixed stride, in which case the content of the bus is not altered and an extra bit signal, named INC, is set [52]. In this way, T0 ensures that no transition occurs for consecutive addresses. Both Gray code and T0 code decrease bit transitions only for consecutive addresses and are ineffective when the frequency of sequential addresses is low such as in case of frequent branch instructions.

The Beach Solution is an encoding technique that aims to address this problem [54]. This technique relies on the assumption that even non-sequential addresses are frequently correlated. A prior analysis of the address stream produced by one or more executions of a program is required to group the bus lines in clusters. For each cluster an appropriate encoding function is generated. However, this technique is application oriented since a program has to be executed at least once to generate the address stream to analyse. Therefore, this scheme is not suitable for general purpose processors where different applications can run concurrently.

In the case of data buses or multiplexed address and data buses, the transmitted data words are not sequential, therefore, the previous techniques are rather not effective. A technique that is appropriate for these types of buses is probability-based mapping [55]. The data words with high probability of occurrence are mapped to code words with low Hamming weight, *i.e.* low number of 1's, to decrease the overall transitions. An analysis of the data stream at design time is required to obtain these statistical properties. This idea is implemented using reversible circuits in [56], where only part of the statistics of the data stream is given. Another technique that assigns limited-weight codes, *i.e.* codes with low Hamming weight, is proposed in [57]. The assignment of codes is determined based not only on the frequency of occurrence of each memory trace but, also, on the frequency with which each trace is followed by another trace. By exploiting the information regarding the sequence of traces, a further reduction in bit transitions is achieved. A prior analysis of the memory traces of the system is required for the design of the code assignment.

In [53], an encoding technique is proposed for multiplexed buses, which combines T0 coding [52] and Bus Invert (BI) [46]. BI is an adaptive encoding scheme suitable for data buses and is further discussed in the next subsection. The method applies the T0 code in case of consecutive addresses, otherwise BI is employed. The power savings of this technique are limited if the addresses are not consecutive and, additionally, the circuit overhead of the technique is higher as two different techniques are combined. An algorithm that generates low activity codes based on a detailed statistical characterisation of the data stream is proposed in [58]. Particularly, the probability distribution of all pairs of successive values of data words is required to assign a code to each pair. The optimal code can be generated for buses with limited width as the complexity of the encoding and decoding increases rapidly with the number of bus lines. Hence, this technique is more effective for narrow buses and requires a detailed analysis of the data stream statics.

2.3.2 Adaptive Encoding Schemes

Adaptive techniques do not require a *a priori* knowledge of the statistical properties of data, therefore, address the limitations of static methods. Specifically, static schemes rely on the statistics of the transmitted data stream for the design of the encoding mechanism. Therefore, these statistics must be known at design time, which is not always feasible. In addition, the power savings diminish if the statistical properties vary temporally. Alternatively, adaptive techniques observe data during runtime to apply the appropriate encoding. Hence, these techniques are more versatile and can be applied to a wide range of interconnects.

A plethora of adaptive techniques for parallel interfaces have been proposed. These encoding schemes are categorised, based on their characteristics, in Table 2.1. General purpose techniques can be applied to a wide range of parallel buses and are effective for all types of data. On the contrary, there are encoding techniques that exploit the correlation of the transmitted data to achieve a higher reduction in bit transitions. However, these techniques are ineffective for uncorrelated data. In time-based techniques, data are represented as delays in the time domain to reduce transitions.

Techniques aiming to balance performance and power use the energy delay product as an optimisation metric. Techniques based on value prediction add two identical predictors to the encoder and the decoder and whenever the prediction is successful, no data are transmitted over the bus, thus, power is saved. Coupling-based techniques are designed for interconnects with high coupling capacitance between adjacent wires.

Table 2.1: Classification of adaptive encoding techniques for parallel interconnects based on their characteristics.

Categories	Techniques
General purpose	[46], [60], [61], [62], [72], [48]
Exploiting data correlation	[65], [67], [66], [63], [64]
Time-based	[68], [69]
Balancing performance and energy	[70], [71], [73]
Value prediction	[74], [75], [76]
Coupling-based	[77]–[81]
Tailored to interfaces with asymmetric termination	[71], [85]–[91]

Therefore, these techniques focus on reducing the power dissipated in charging and discharging the coupling capacitance. Finally, techniques tailored to interconnects with asymmetric termination reduce power by decreasing the direct current flowing due to the termination resistor of these links.

General Purpose Techniques

BI is one of the early encoding schemes for decreasing bit transitions [46]. BI works as follows. In each clock cycle, BI calculates the Hamming distance, *i.e.* the number of bits that two data words differ, of the current data word and the word transmitted in the previous clock cycle. If the Hamming distance is higher than $M/2$, where M is the bus width, then the data word is inverted before transmission, otherwise it remains unaltered. One redundant bus line is used to inform the decoder whether the data word is inverted or not. Hence, the decoder conditionally inverts the input data depending on the value of the redundant bus line.

A possible circuit implementation of the technique is depicted in Fig. 2.4 for a 4-bit bus. The bits of the two words are compared using 4 XOR gates, while the majority voter calculates the Hamming distance and decides whether to invert the data word or not. If the data word is inverted then *inv_out* is set. The remaining 4 XOR gates are used to invert the data word. The implementation of the decoder is simple, as M XOR gates are required for the conditional inversion of the data word.

This technique remains popular and is adopted in commercial bus interfaces such

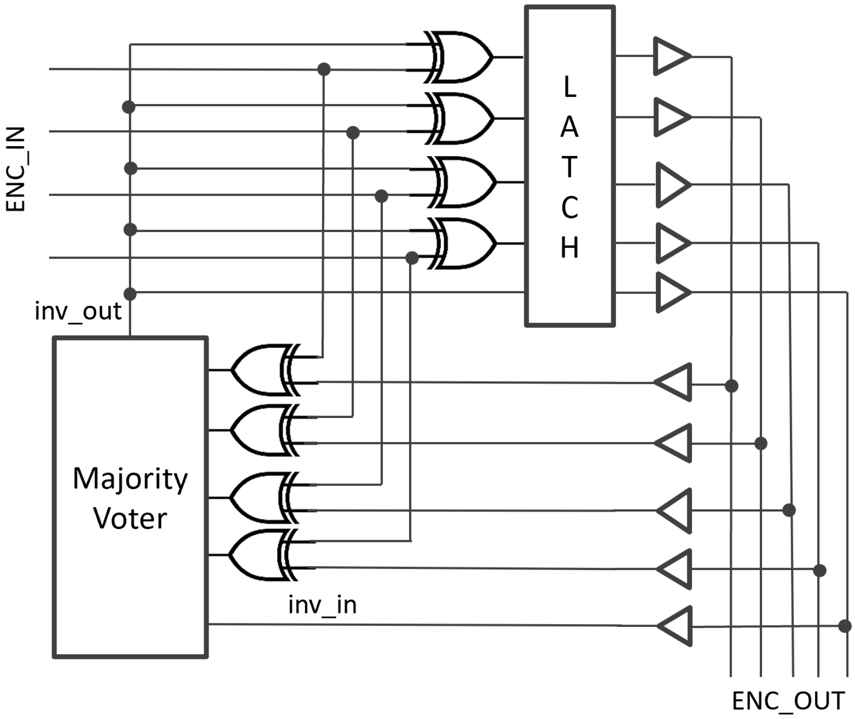


Figure 2.4: Circuit implementation of the BI encoder [46].

as in HBM [10] because of its simple circuit implementation, and low overhead (∼1 mW in a 65 nm technology, operating at 400 MHz). The benefits of using BI are investigated in [47]. When BI is applied to double-data-rate fourth generation (DDR4) memory, the I/O power as well as the power-supply noise decrease. Despite its commercial success, BI provides limited savings in transitions compared to more elaborate techniques, especially in cases where data are highly correlated, such as image data, where BI exhibits 4.42% reduction in transitions for a 32-bit bus [60]. This behaviour is, also, verified in Chapter 4 of the thesis, where BI exhibits a mere 6.04% reduction in transitions for the transmission of image data over a 64-bit bus.

Some techniques are based on BI and aim at enhancing the reduction in bit transitions by inverting only a part of the data word. In [59], the Partial Bus Invert (PBI) method is proposed, where a subgroup of bus lines is composed. Only the bits of the bus lines in the subgroup are considered for inversion according to BI. The subgroup is formed based on the transition probabilities and the transition correlations among the bus lines. PBI has both static and adaptive features as these statistics must be known at design time, however, the decision about inverting the subgroup of bus lines is taken at runtime. Nevertheless, PBI is not effective when the statistics of the data stream are not known in advance.

Adaptive Partial Bus Invert (APBI) is an extension of PBI that is fully adaptive and, thus, addresses the restriction of PBI [60]. The data stream is observed during runtime and the subgroup of encoded bus lines is changed periodically. Specifically, APBI observes the data stream for a window of a fixed number of words and forms a subgroup with the bus lines with the highest number of transitions, called mask. BI is then applied only to the bus lines in the mask and one extra bit (INV) is used to inform the decoder about inversion.

APBI encoding is defined as follows:

$$X^t_{Bus} = \{X^t_d, INV\} = \begin{cases} \{Q^t, 0\}, & \text{if } W(X^t_{Diff}) \leq \frac{W(mask(t))+1)}{2} \\ \{Q^t \oplus mask(t), 1\}, & otherwise, \end{cases} \tag{2.2}$$

where X^t_{Bus} is the encoded data word including the redundant bit, INV, that is transmitted over the bus at time t. Q^t is the original data word unencoded and $W(x)$ is the Hamming weight, *i.e.* the number of 1's of a binary vector. The mask at time t is equal to $mask(t) = \{m^t_0, m^t_1, ..., m^t_{n-1}\}$, where $m^t_i \in \{0,1\}$ and n is the bus width. If $m^t_i = 1$, then, the i^{th} bus line is included in the mask at time t, thus, in the subgroup of bus lines where BI is applied. X^t_{Diff} is defined as:

$$X^t_{Diff} = (X^{t-1}_{Bus} \oplus \{Q^t, 0\}) \cdot mask(t), \tag{2.3}$$

where X^{t-1}_{Bus} is the encoded data word transmitted over the bus at time $t-1$. The decoder conditionally inverts the bus lines of the mask depending on the value of the redundant bit INV. Hence, the decoder restores the original word as follows:

$$Q^t = \begin{cases} \{X^t_d\}, & \text{if } INV = 0 \\ \{X^t_d \oplus mask(t)\}, & \text{if } INV = 1. \end{cases} \tag{2.4}$$

To determine which bus lines are in the mask, the number of bit transitions of each bus line, as well as the total number of transitions across the bus have to be calculated over a specified window of N data words. The mask is, therefore, calculated as follows:

$$m^t_i = \begin{cases} 1, & \text{if } \alpha_i \geq \alpha_{tot}/n \\ 0, & \text{if } \alpha_i < \alpha_{tot}/n. \end{cases} \tag{2.5}$$

where α_i is the number of transitions of the i^{th} bus line and α_{tot} is the total number of transitions of the bus over a specified window of N words. Hence, if the number of transitions of a bus line is higher than the average, then, the bus line is included in the mask.

The mask information is not transmitted to the decoder, instead the calculation of the number of transitions per bus line is performed by the decoder over the same window of words to define the mask. The circuit diagram of APBI is illustrated in Fig. 2.5. The Bus Invert Coder can be implemented according to Fig. 2.4 and the Decoder with a number of XOR gates as in BI. The implementation of the Mask Computation block is depicted in Fig. 2.6. The current data word $(D_0, ..., D_{n-1})$ is XOR-ed with the previous word to detect if a transition occurs in each bus line. The previous word is stored in registers denoted as R in the diagram. The outputs of the XOR gates are fed to the enable inputs of n counters (TC$_0$,...,TC$_{n-1}$), which count the number of transitions of each bus line. In addition, the total number of transitions of the bus has to be computed. Therefore, the Weight block calculates the number of 1's of the XOR outputs in each cycle and the output of Weight is fed to Total Transition Accumulator (TTA), where the total number of transitions over N cycles is calculated.

The mask is computed in the F$_0$,...,F$_{n-1}$ blocks according to Equation 2.5. To simplify the division operation of Equation 2.5, the bus width is restricted only to values that are a power of 2. Thus, the division can be replaced by a shift operation. The output of the F blocks is the resulting mask and is stored in registers, R. The computation of the mask in F is only performed at the end of each window, therefore, a Window Counter (WC) is used that produces an update signal that is fed to the F blocks and the final registers.

The power dissipated in the encoder and decoder circuits of APBI can be reduced by disabling the mask computation for some windows. Hence, instead of calculating a

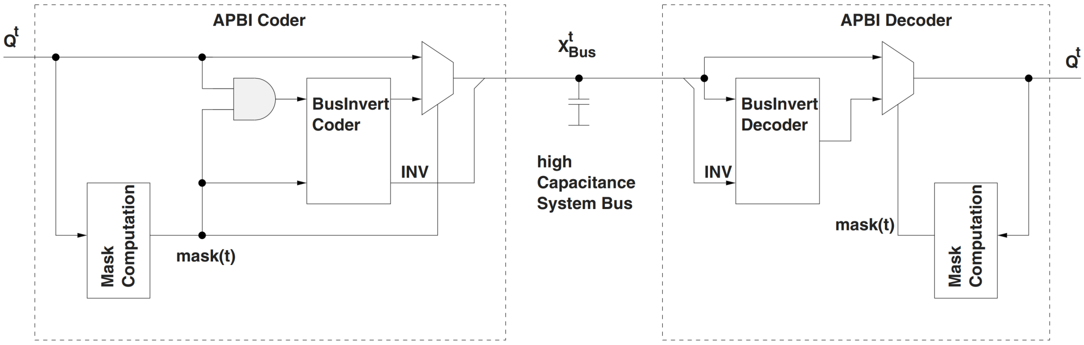


Figure 2.5: Circuit diagram of the APBI technique [60].

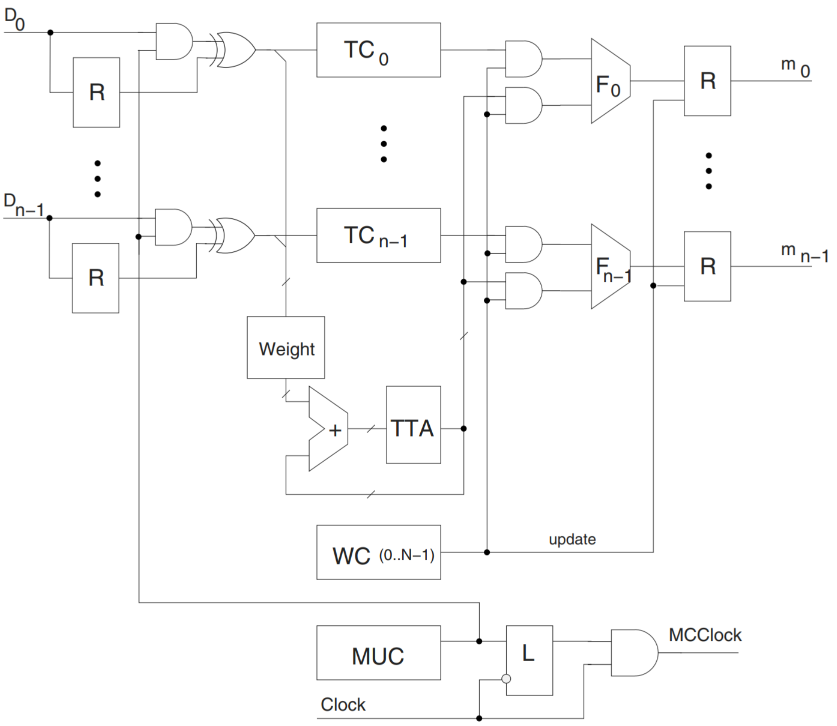


Figure 2.6: Circuit diagram of the Mask Computation block [60].

new mask in every window, the mask can be updated every k^{th} window. Consequently, the mask computation is disabled for windows 1 to $k-1$. This function is implemented in the Mask Update Counter (MUC) and the clock gate circuit, composed of a latch (L) and an AND gate, that generates the *MCClock* clock signal. The output of the MUC block is only high during the k^{th} window, thus, the mask is computed only during this window.

APBI is particularly effective and outperforms BI in terms of reduction in transitions for highly correlated data streams, such as image files [60]. However, this technique suffers from high circuit overhead as the calculation of the mask must be performed twice, both in the encoder and the decoder, especially when the mask computation takes place in every window. Alternatively, when the mask update interval is increased, the reduction in transitions decreases, thus, restricting the power savings. Furthermore, in cases where the bus lines are not correlated, the simple BI technique is more efficient, as shown in Chapter 4.

The effectiveness of bus inversion diminishes for large bus widths [62], therefore, alternative methods are explored. Shift Invert (SINV) coding selects between shifting and inverting data words to further decrease the number of transitions compared to BI

[61]. Specifically, SINV, first, evaluates the number of transitions of transmitting the data word in the original form, inverted, left-shifted, and right-shifted. Then, the data word is transmitted in the form that leads to the lowest number of transitions. Two additional bus lines are required to indicate the form of the transmitted data such that the decoder can retrieve the original data.

Although this technique further reduces transitions compared to BI, the required hardware complexity is higher as more computations of Hamming distances are required. The circuit overhead of the technique is not estimated, therefore, the effectiveness of the technique in decreasing power cannot be properly evaluated. Finally, the effectiveness of SINV in reducing transitions is only evaluated for random data, however, data are correlated in many applications.

Bus shifting (BS) rotates the data word to be transmitted by a number of bits to minimise transitions and uses spatial redundancy to send this number to the decoder [62]. To determine the optimal number of rotations, the Hamming distance for all possible rotations of each data word is calculated. The number of rotations with the lowest Hamming distance is then selected. This scheme is only tested for uniformly distributed random data, where the probability of transition of each bus line is 0.5.

BS is most effective for random data when the bus width is 32 or 64 bits. However, in this case 32 or 64, respectively, calculations of Hamming distances per clock cycle are required, which increase the hardware complexity. Alternatively, for narrow buses the reduction in transitions is limited due to the redundant bus lines. For example, 2 redundant bus lines are required when the bus width is 4 bits, which diminish significantly the transition savings [62]. Furthermore, a circuit implementation of this scheme is not provided, therefore, the power overhead of the encoding mechanism can not be assessed. Hence, the power savings of the technique can not be fairly estimated. In addition, the performance of BS in reducing bit transitions is only evaluated for random data and not for real types of data, which often are correlated.

Adaptive Bus Encoding (ABE) is another technique that encodes only the highly correlated bus lines [72]. This technique observes the data characteristics over a window of a fixed number of words to select a bus line as a basis line and form a cluster with the bus lines that exhibit high correlation with the basis line. The lines in the cluster are finally XOR-ed with the basis to achieve a high reduction in bit transitions.

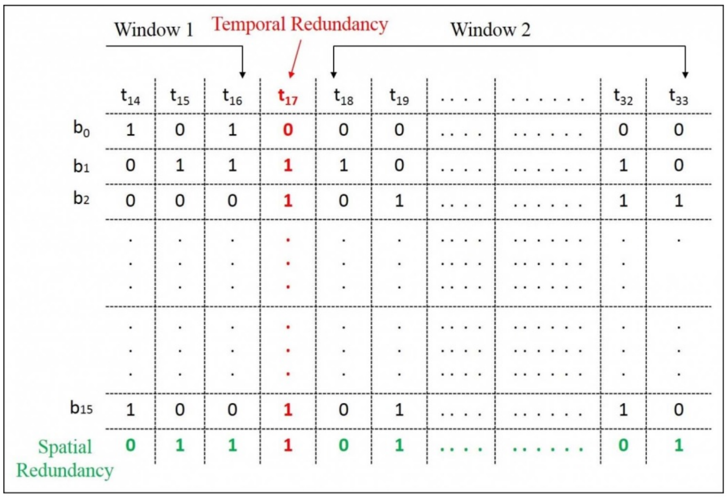
The pseudocode of the ABE technique is shown in Algorithm 1, where N is the number of bus lines. To determine the optimal basis line and cluster, each bus line is considered as basis and the number of transitions of every other line XOR-ed with the basis is calculated. For every basis line, a cluster is formed with the bus lines that exhibit less transitions when XOR-ed with the basis line. Subsequently, the clustered lines are XOR-ed with the basis line and the savings in transitions are calculated. The bus line that leads to the maximum savings is selected as basis. This process is repeated for each observation window. The basis and cluster information are transmitted using a redundant bus line and an additional clock cycle at the beginning of the window, respectively, as depicted in Fig. 2.7. The observation window comprises 16 words, the bus lines are denoted as $b_0 - b_{15}$ and clock cycles as $t_{14} - t_{23}$.

A high level circuit diagram of the encoder is depicted in Fig. 2.8. The encoder comprises a decision block, which determines the basis line and the cluster, a delay element, a block of XOR gates used to XOR the clustered lines with the basis line and a multiplexer is used to insert the temporal redundancy, *i.e.* the cluster information. The decision block is structured as a $N \times N$ array, as illustrated in Fig. 2.9. In each row (i) the corresponding bus line (b_i) is considered as basis. The number of transitions of the b_i are calculated in the self transition computation block, while in the bit wise savings computation block the number of savings in transitions contributed by every bus line b_j ($j \neq i$) when XOR-ed with the basis line b_i are calculated and the presence of b_j in the cluster is decided.

The overall savings computation unit calculates the total number of savings in transitions for each b_i. In the basis selection unit, the basis and the cluster that exhibit the highest savings are selected. The role of the eliminator block is to restrict the power dissipated in the encoder. Therefore, this block removes the bus lines with a low number of transitions from consideration as basis line, since the probability of being selected as basis is low. The circuit implementations of the individual blocks of Fig. 2.9

Algorithm 1 Pseudocode of the ABE technique.

```
Find total number of transitions in all the bit lines $s_{ini}$
for $i \leftarrow 0$ to $N-1$ do
    Choose $i^{th}$ line as basis
    for $j \leftarrow 0$ to $N-1$, $j \neq i$ do
        Put $j^{th}$ line in cluster if $j^{th}$ line has more transitions than $j^{th}$ line XOR-ed with basis line
    XOR all the clustered lines with the basis line
    Find number ($s_i$) of transitions in this modified set
    Savings $p_i = s_{ini} - s_i$
If $b_k = argmax_i(p_i)$, $0 \leq i \leq N-1$, then the $k^{th}$ bus line is the basis line for that observation window
```


Window 1
Temporal Redundancy
Window 2

	t_{14}	t_{15}	t_{16}	t_{17}	t_{18}	t_{19}	....	......	t_{32}	t_{33}
b_0	1	0	1	0	0	0	....	......	0	0
b_1	0	1	1	1	1	0	....	......	1	0
b_2	0	0	0	1	0	1	....	......	1	1
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	·
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
b_{15}	1	0	0	1	0	1	....	......	1	0
Spatial Redundancy	0	1	1	1	0	1	....	......	0	1

Figure 2.7: Redundancy added by the ABE technique [72].

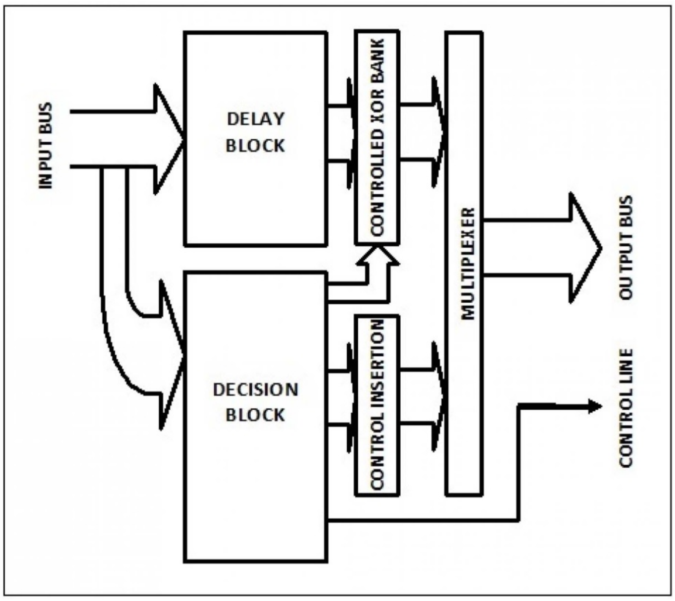


Figure 2.8: Block diagram of the ABE encoder [72].

are provided in [72].

The circuit diagram of the decoder is illustrated in Fig. 2.10. *N* registers are used to store the cluster information, which are updated in the beginning of each observation window. A control signal is transmitted in the beginning of each window and the cluster is extracted by XOR-ing the control signal and the encoded data of the previous observation window. The bus lines in the cluster are XOR-ed with the basis line to restore the original data.

The ABE technique requires a large number of computations even with the use

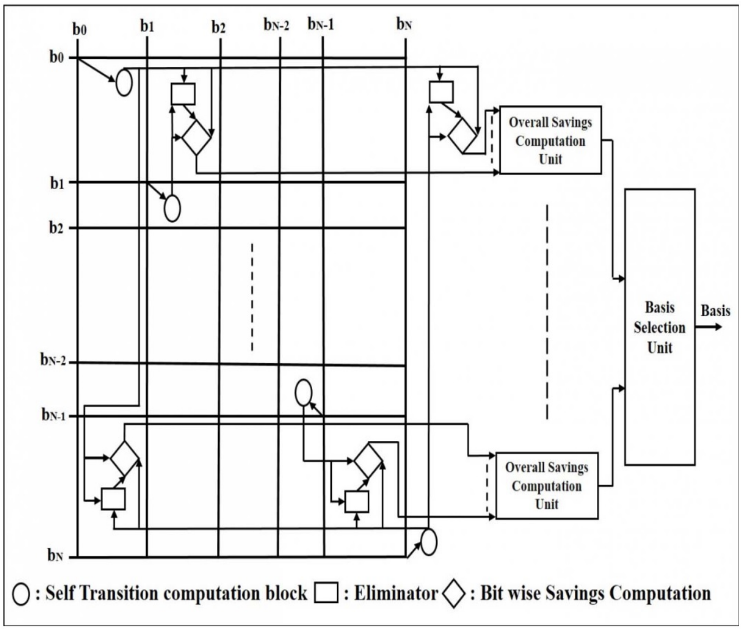

b0
b1
b2
bN-2
bN-1
bN
Overall Savings Computation Unit
Basis Selection Unit
Basis
Overall Savings Computation Unit
○ : Self Transition computation block □ : Eliminator ◇ : Bit wise Savings Computation

Figure 2.9: Block diagram of the decision block of the ABE encoder [72].

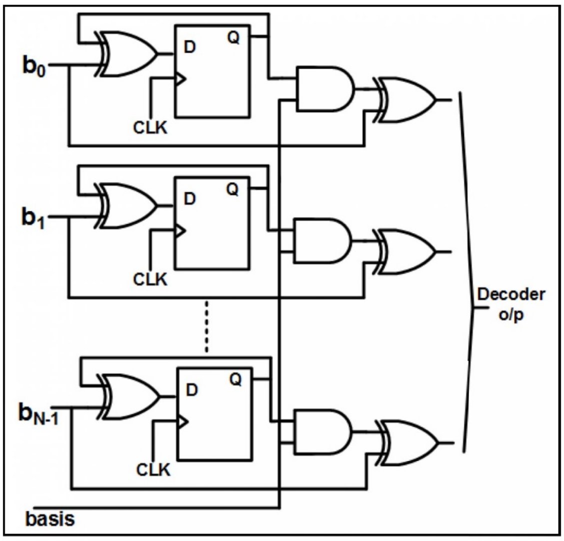


Figure 2.10: Circuit diagram of the the ABE decoder [72].

of the eliminator block, which results in high hardware complexity and overhead in power. This behaviour is demonstrated in Chapter 4, where the power overhead of the technique is estimated to be ∼17 mW for a 64-bit bus when implemented in a 65 nm technology and operating at 400 MHz. Hence, the overhead of ABE can undermine the power savings, especially, for short interposer-based interconnects, where the length can be just 1 mm (Chapter 4).

Sequence-switch coding [48] is based on the reordering of words to reduce transition activity. The algorithm for the reordering mechanism considers only the two most recent words in the reordering operation and, therefore, exhibits a moderate decrease in bit transitions, which is comparable to BI. This scheme, however, requires a more complex implementation than BI. Consequently, the overall performance of this technique is inferior to BI. Since only two data words are considered for reordering, the potential power savings of the idea of word reordering are not fully explored. This idea is further explored in this dissertation in Chapter 4, where by selecting a higher number of data words for reordering, a higher decrease in bit transitions is achieved. Furthermore, the circuit architecture of this technique is not provided, therefore, the added circuit overhead can not be determined and the efficiency of this technique can not be fully and properly evaluated.

Exploiting Correlation of data words

Several encoding techniques exploit the correlation between data words to decrease bit transitions. Frequent Value (FV) is a technique based on storing frequent values to encode the frequently transmitted words [65]. A table with the frequent values is used both at the encoder and at the decoder. The content of the tables is updated during runtime to adapt to changes. If a data word is found in the table, then, one-hot encoding is used to transmit the word, where the ‘1’ indicates the index of the table instead of the actual value of the data word. However, in case a data word is not found in the table the bit transitions are not reduced since the word is transmitted in its original form. Hence, FV is effective only when identical data values are transmitted frequently. Another drawback of this technique is that the two tables are implemented using Content Addressable Memories (CAM), which significantly increase the power and hardware requirements.

Adaptive Dictionary Encoding Scheme (ADES) is also based on storing recurring data values to reduce bit transitions and is inspired by dictionary compression [67]. A dictionary is used in the encoder and the decoder to store data values. Each data word is split in three parts. The encoder uses the middle part to look-up the dictionary. In case the first part with the Most Significant Bits (MSBs) of the data word matches the value of the dictionary, then, only the middle and last part of the word are transmitted, while the MSBs in the bus remain unchanged to reduce bit transitions. In case of a miss, the data word is transmitted unchanged and the dictionaries are updated. Consequently, if frequent misses occur, the bit transitions are not decreased. Hence, ADES is only effective for correlated data.

In techniques that use tables to store frequent values, such as FV and ADES, by increasing the size of the table in which frequent values are stored, the hit rate increases as well as the bit transitions when a hit occurs. This trade-off is balanced by implementing a hierarchical cache design [66]. Although this technique successfully reduces bit transitions, the energy, delay, and area penalties are high, which are estimated to be 50.2 pJ, 6.2 ns, and 0.039 mm^2, respectively, for a 0.180 μm technology. In addition, a large number of control signals is required to be transmitted such that the data can be recovered by the decoder.

Data locality can be exploited to decrease bit transitions on data address buses. To achieve that, two encoding schemes assign codes with low Hamming distance to the most frequently accessed addresses by using self-organising lists [63]. Two heuristics are used to assign codes to the accessed addresses in the lists, Move-To-Front (MTF) and Transpose. In both methods, all possible symbols are stored in a linked-list and each input symbol is encoded as the index of the symbol instead of the value of the symbol. Since all symbols are stored in the list, the length of the index is equal to the length of the symbol. In MTF, once a symbol is encoded, the symbol is moved to the top of the list. Alternatively, the Transpose scheme swaps the positions of the encoded symbol and the preceding symbol.

In both techniques, the frequently incoming symbols are moved to the top of the lists, where indices exhibit low Hamming distance. Thus, when frequently accessed addresses are transmitted consecutively, the bit transitions are reduced. In addition, both the encoder and the decoder initialise and update the lists in the same way, thus, the decoder can retrieve the original symbol without redundancy. To reduce the power and delay overhead of storing and accessing all possible addresses in the lists, the address bus is divided into smaller parts and MTF or Transpose is applied to each part separately. The limitation of these techniques is that their effectiveness is restricted in data address buses, where the likelihood of transferring two different address sequences is high, such as reading from one address space and writing to another space.

Working zone is an adaptive technique that is, also, tailored to address buses and is based on the assumption that applications use specific working zones of the address space [64]. Therefore, if an address belongs to one of these working zones, only an identifier of the zone and an offset are required to be sent. To reduce the number of transitions, the offset is one-hot encoded and transition signalling is used. According to transition signalling, the word is XOR-ed with the previous word before transmission.

Thus, when an one-hot encoded word is transmitted, a single transition occurs. In case of a miss, *i.e.* the address does not belong in the working zones, the address is transmitted unaltered. Consequently, the effectiveness of this technique is restricted for address buses under the condition that applications use only a few working zones.

Time-base Representation of Data

Time-based representation of data values has been explored to reduce the number of bit transitions. In Data Exchange using Synchronized Counters (DESC), data values are represented by delays and, specifically, by the number of clock cycles elapsed between successive pulses [68]. In this way, the number of wires is reduced and the number of bit transitions is lowered since a fixed number of transitions per data word is exhibited. The limitation of this scheme is that the bandwidth is decreased by 3.15× on average [69], which is prohibitive for many applications.

Adaptive Time-based Encoding (ATE) [69] is a technique that improves the energy-delay product of DESC for last level cache interconnects by applying either DESC or binary encoding depending on the workload. However, this technique suffers from higher design complexity and circuit overhead compared to DESC since both time-based and binary encoding are combined. The peak power consumption of the technique is 0.519 W for a 22 nm technology [69], which can diminish the power savings. Additionally, ATE is specifically designed to monitor characteristics of applications at runtime and, particularly, the execution phases and cache access bursts to apply the appropriate encoding for cache links. Therefore, ATE is not applicable to general purpose interconnects.

Balancing Performance and Energy

Other techniques focus on balancing performance and energy consumption. Slow Transition Fast Level (STFL) signalling encodes data to optimise the energy delay product of low power wires in last level caches [70]. This technique is based on the observation that reducing the supply voltage of wire repeaters leads to a reduction in power and, also, an increase in the delay of signal transmission. Reducing the supply voltage from 0.7 V to 0.48 V leads to a 100% increase in delay for a 22 nm technology. Therefore, STFL applies the appropriate encoding to mitigate this delay and, thus, balance bandwidth and power [70]. In [71], STFL is applied on off-chip interconnects and, specifically, different DRAM interfaces. However, STFL is designed for low swing communication, which has considerable limitations, such as noise susceptibility or large area penalties as discussed in Section 2.1.

Energy Control and Metadata Consolidation are toggle-aware compression techniques that combine encoding with compression [73]. These methods are based on the observation that transferring compressed data increases the bit transitions and, thus, the power dissipation compared to transferring the original data uncompressed. Therefore, these two techniques decrease a part of the bit transitions introduced by compression at the expense of a low decrease in throughput, thereby balancing performance and power.

Energy Control decides whether to transmit a data word compressed or uncompressed based on the number of transitions of the original word and the compressed word, the compression ratio, and the bandwidth utilisation. In this way, bandwidth and energy can be traded-off. Alternatively, Metadata consolidation aligns the compressed data words to reduce bit transitions. The alignment of data words is usually disturbed by the metadata generated by compression. Therefore, this technique consolidates these metadata into a single block to prevent misalignment of similar bit patterns. Energy Control and Metadata Consolidation effectively balance performance and energy, however, these techniques are only applicable to compressed data transfers.

Techniques based on Value Prediction

Adaptive encoding techniques based on value prediction have, also, been explored. A bit prediction technique utilises a table with 2^k entries for each bus line both in the transmitter and the receiver [74]. The last k values of the bus line are used to index the table. Transition signalling is used. Consequently, if the prediction is correct, the value of the bus line is not altered. In case of a wrong prediction, a transition takes place and the corresponding value in the tables is complemented. The advantage of this technique is that no redundancy is required, however, the size of the table increases exponentially with k, therefore, the value of k must be low. Finally, this technique is only effective for address buses where values tend to be more predictable and not for data buses or multiplexed data and address buses. BI [46] outperforms this technique for data buses.

Bus Encoding based on instruction-set-Aware Memories (BEAM) is a value prediction technique for instruction and data address buses [75]. BEAM requires memories to attain some knowledge about the instruction set architecture to make predictions. Hence, the memory collects information about the executed instructions to predict addresses. The processor does not need to transmit the correctly predicted addresses, therefore, energy is saved. The limitation of this technique is that the memory requires knowledge of the instructions that generate the addresses and the instruction-set architecture. These requirements are challenging since the processor and the memory might operate at different frequencies and, also, limit portability. Additionally, this technique cannot be applied to data buses.

In another technique utilising value prediction, a value predictor is included both in the encoder and the decoder, which provide identical predictions [76]. The encoder checks if the the prediction of the value is correct. If the prediction is successful, then the receiver utilises the predicted value and no data is transmitted over the bus, thus, no transition takes place. In case of incorrect prediction, the original data value is transmitted and an additional control bit switches.

To improve the transition savings, a predictor confidence mechanism is added, according to which multiple predicted values are sorted in a table based on confidence and a code-word is assigned to each value. The value with the highest confidence is mapped to an all-zero code-word and by using transition signalling, no transition is induced on the bus. Code-words with a single bit set are matched to the next M values in the table and so forth, where M is the bus width. In case of a hit, *i.e.* the data value is found in the table, the assigned code-word is transmitted, which reduces the transitions, otherwise the data value is transmitted either in its original form or inverted.

The savings of this technique heavily depend on the prediction accuracy and table size. More accurate predictors require higher hardware complexity and exhibit higher overhead. Value predictors are effective only when data values are repeated, thus, not for random values. The implemented predictor in [76] saves energy for long interconnects ($>$ 10 mm for memory interconnects), where the capacitive load is higher.

Coupling-based Techniques

Several techniques focus on decreasing the power dissipated on charging and discharging the coupling capacitance of adjacent wires. In Fig. 2.11, a simplified bus model is depicted, where C_L is the ground capacitance of each line, C_I is the coupling capacitance between adjacent lines, and V_1, V_2, ... V_n are the voltages of the bus lines. To decrease the power dissipated on C_I, the transitions between adjacent wires, which are named relative transitions [77], must be decreased.

Various encoding schemes [77]–[81] target the decrease in relative transitions between adjacent bus lines, rather than the bit transitions of individual lines. The applicability of these methods is restricted to those interconnects, where the coupling

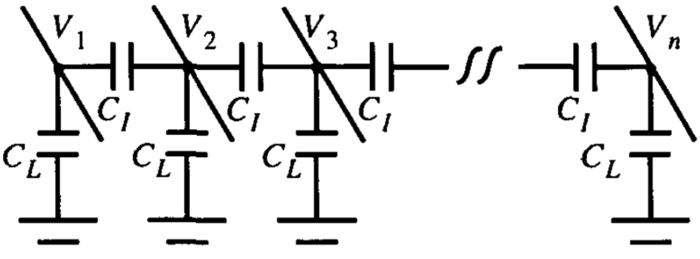


Figure 2.11: Simplified bus model [77].

capacitance is dominant and the ground capacitance is less significant. Hence, these techniques are more effective for on-chip interconnects and not suitable for inter-die interconnects, considering the large capacitive load added from I/O cells.

Techniques for Interfaces with Asymmetric Termination

The last set of techniques for parallel interconnects is tailored to interfaces with asymmetric termination. An example is the Pseudo Open Drain (POD) interface illustrated in Fig. 2.12, adopted in many DRAM interfaces, DDR4 [83] and Graphics Double Data Rate 5 (GDDR5) [84]. POD adds a termination resistor that connects the line with V_{DDQ}. A static current flows when the line is driven to 0 V, which dissipates a high amount of energy [88]. Consequently, the energy of transmitting a logic 0 value is orders of magnitude higher compared to the energy of transmitting a logic 1.

The objective of encoding schemes for asymmetric interfaces is the reduction of the number of 0's or 1's depending on the type of termination (V_{DDQ} or V_{SSQ} termination) rather than the bit transitions. Data Bus Inversion can be adapted for these interfaces [82], where the criterion for inversion is the number of 0's or 1's rather than the number of bit transitions. DC/AC Data Bus Inversion (DBI) [85] and DBI-AC/DC [86] are both extensions of DBI, which consider both the energy due to the termination resistor and the bit transitions.

More is Less (MiL) uses sparse codes to reduce the number of 0's at the expense of longer data words [87]. Base+XOR Transfer [88] and Bitwise-Difference Encoding (BD-Encoding) [89] both exploit the similarity of data. Base+XOR divides each memory transaction into four parts. The first part is transmitted unaltered, whilst the rest are XOR-ed with the preceding data part. In this way, the number of 1's is reduced, since data within a memory transaction tend to be correlated. BD-Encoding, uses data tables to store the most recently transmitted data words both in the DRAM and the memory controller of DRAM. The data to be transmitted is XOR-ed with the most similar data

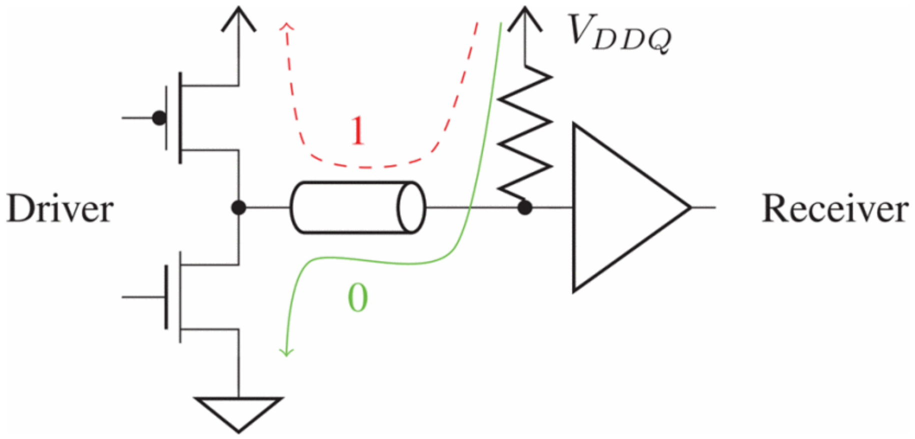


Figure 2.12: Pseudo Open Drain interface [85].

word in the table to minimise the number of 1's.

Online Data Clustering [90] groups data into different clusters based on their resemblance, evaluated according to Hamming distance. A center is assigned to each cluster and each data word is XOR-ed with the nearest cluster center to reduce the number of 1s. Alternatively, Cost-Aware Flip Optimisation (CAFO) focuses on reducing the total cost of write operations in Phase Change Memory (PCM) and spin-transfer torque random access memory (STT-RAM) in terms of endurance and error rate, respectively [91]. All of these techniques are only effective for parallel interfaces with asymmetric termination.

Summary of adaptive techniques

The characteristics of adaptive encoding techniques for parallel interfaces are summarised in Table 2.2. BI [46], APBI [60], ABE [72], sequence-switch [48], SINV [61], and BS [62] are general purpose encoding techniques and can be applied to both address and data buses. In addition, these schemes reduce bit transitions of both correlated and uncorrelated data. However, BI [46], APBI [60], and ABE [72] provide moderate reduction in bit transitions or exhibit significant overhead in encoding/decoding power, while a circuit implementation is not provided for sequence-switch [48], SINV [61], and BS [62], thus, the overhead cannot be evaluated.

As shown in [49], the effectiveness of encoding in reducing power consumption is often restricted by the high implementation overhead, thereby diminishing and often eliminating the benefits, offered theoretically, by data encoding techniques. Hence,

Table 2.2: Characteristics of adaptive encoding techniques for parallel interconnects.

Techniques	Address buses	Data buses	Correlated data	Uncorrelated data	Redundancy required
BI [46]	✓	✓	✓	✓	✓
APBI [60]	✓	✓	✓	✓	✓
SINV [61]	✓	✓	✓	✓	✓
BS [62]	✓	✓	✓	✓	✓
ABE [72]	✓	✓	✓	✓	✓
Sequence-switch [48]	✓	✓	✓	✓	✓
FV [65]	✓	✓	✓	✗	✓
ADES [67]	✓	✓	✓	✗	✓
[66]	✓	✓	✓	✗	✓
MTF and Transpose[63]	✓	✗	✓	✗	✗
Working zone [64]	✓	✗	✓	✗	✓
DESC [68]	✓	✓	✓	✓	✗
ATE [69]	✓	✓	✓	✓	✗
STFL [70]	✓	✓	✓	✓	✓
Energy Control[73]	✓	✓	✓	✓	✗
Metadata Consolidation [73]	✓	✓	✓	✓	✗
[74]	✓	✗	✓	✗	✗
BEAM [75]	✓	✗	✓	✗	✗
[76]	✓	✓	✓	✗	✓

techniques with high power overhead allow savings only for extremely large capacitive loads, thus, are not applicable for advanced packaging technologies, such as interposers with short interconnect length (<20 mm).

Several techniques ([63]–[66], [74]–[76]) rely on data correlation or specific characteristics of data, such as in address buses, to reduce bit transitions. Therefore, these techniques become ineffective when applied to uncorrelated data. Other techniques, such as STFL [70], Energy Control, Metadata Consolidation [73], and ATE [69], focus on optimising the energy delay product, thus, balance energy and performance. Time-based techniques, DESC [68] and ATE [69], suffer from high penalties either in bandwidth or circuit complexity and, thus, power, which are prohibitive for many applications.

The coupling-based techniques and the techniques designed for interfaces with asymmetric termination are excluded from Table 2.2 as encoding schemes of these categories are applicable to specific kind of links. The former group of schemes is based on the assumption that the coupling capacitance of interconnects is higher than the ground capacitance, hence, these schemes are not effective for off-die links. The latter set of techniques decreases the number of 0's or 1's depending on the type of termination to reduce power. Therefore, these techniques are only effective for links with asymmetric termination.

Finally, encoding techniques designed for parallel interfaces are not suitable to serial communication. To lower the power dissipated in serial communication, the bit transitions within each data word has to be decreased, instead of the transitions between successive words. Encoding techniques applicable to both source synchronous and asynchronous serial interfaces are discussed in the next section.

2.4 Signal Encoding for Serial Transmission

In serial communication, the transmitter sends data sequentially over an interconnect. One of the main advantages of serial communication is the small number of I/O pins and wires as serial transmission can operate on just a single wire. Therefore, serial interfaces are preferred for long interconnects and I/O-limited systems. Serial interfaces can be divided to source synchronous and source asynchronous discussed in subsections 2.4.1 and 2.4.2, respectively.

2.4.1 Source Synchronous Serial Interfaces

In source synchronous communication, a clock signal is transmitted to the receiver along with data, which is used to capture the data signals. The clock signal can either be sent by the transmitter or a reference clock can be used by both the transmitter and the receiver [101]. On the contrary, in asynchronous communication the receiver extracts the clock information from the incoming data. Examples of serial, source synchronous interfaces are Inter-Integrated Circuit (I^2C) [102] and Serial Peripheral Interface (SPI) [103].

The objective of encoding techniques for serial links is the decrease of bit transitions within each data word and not between consecutive words. Therefore, encoding techniques for parallel interconnects are not suitable to serial links and encoding techniques for the latter have been designed. Serialized Low-Energy Transmission (SILENT) is a simple encoding scheme that XOR's successive data words and transmits the result [104]. For similar data words, the result of the XOR operation is a stream of mostly 0's with few bit transitions. This technique reduces transitions only when the consecutive data words are highly correlated and can even increase transitions, if data exhibit poor correlation.

Encoding techniques that do not rely on data correlation to decrease transitions have been proposed. The encoding scheme of [105] prevents consecutive transitions regardless of the type of the transmitted data. Specifically, this technique checks every 3-bit string of a byte and swaps the two last bits if two consecutive transitions are exhibited. However, two redundant bits for every byte are required to keep track of the modifications such that the original data can be retrieved by the decoder. This large amount of redundant bits results in 20% decrease in throughput. Transition Inversion Coding (TIC) inverts the even bits of a word in case the number of transitions within the word is higher than the word length [106]. TIC uses one redundant bit per word to indicate whether the even bits are inverted or not. TIC is general purpose and decreases transitions regardless of similarities in data values. However, in case of highly correlated data, such as in image files, the decrease in transitions is lower compared to SILENT.

Some techniques exploit the error-tolerant nature of applications that process sensor data to decrease the transition activity. Approximate Differential Encoding (ADE) rounds some of the Least Significant Bits (LSBs) of data words before transmitting the bitwise difference of successive words [107]. In this way, the number of transitions in the LSBs is minimised, however, some information loss is inevitable.

Value-Deviation-Bounded Serial encoder (VDBS) inverts bits of the words to generate long sequences of 0's or 1's and, hence, reduce the transitions within a word [108]. A maximum deviation between the original and the encoded word is defined. Serial-T0 transmits a zero transition pattern when consecutive words are correlated and is especially effective for camera captured data [109]. Approximate Serial Bus

(AXSERBUS) transmits a zero transition pattern when the difference between consecutive words is low, while the difference is encoded as a single transition pattern when it is considered intermediate [110]. All these techniques are suitable only to error-tolerant applications.

The objective of all of these encoding schemes is to achieve the highest decrease in the number of transitions. Consequently, these techniques are only compatible with source synchronous serial interfaces. Asynchronous interfaces have prerequisites that these techniques do not satisfy as discussed in the next subsection.

2.4.2 Source Asynchronous Serial Interfaces

High speed, SerDes devices, such as PCIe, Universal Serial Bus (USB), Ethernet, and Serial Advanced Technology Attachment (SATA) are ubiquitous. These interfaces are source asynchronous. Thus, the clock is recovered from the incoming data at the receiver instead of relying on an external clock signal as in source synchronous communication [101].

The architecture of a generic SerDes device is illustrated in Fig. 2.13. The transmitter is composed by encoding logic, a serialiser (P2S) that converts data from parallel to serial format, a Phase Locked Loop (PLL) circuit, and a differential driver. The PLL generates a high-speed clock for the serialiser. Differential signalling is commonly used in SerDes devices, such as PCIe, to increase noise resiliency [111]. The receiver's physical interface comprises a differential line receiver, a Clock Data Recovery (CDR) block, a deserialiser that converts the serial data to parallel, and decoding logic that restores the original data. The differential line receiver detects the potential difference between the two incoming signals and determines whether a logic '0' or '1' is represented. The CDR block is responsible for recovering the clock from the received data.

The transmitted data are required to exhibit frequent bit transitions to ensure that the CDR logic is able to recover the clock from the incoming data and prevent data reception errors [101], [111]. The CDR block generates a reference clock by phase-aligning the clock to the transitions of the incoming data signal using a PLL [101], [111]. The receiver samples the data stream using the generated reference clock. Therefore, large run lengths must be avoided, *i.e.*, the number of consecutive 0's or 1's. If very few transitions are detected by the CDR logic, the timing information cannot be extracted and the PLL's clock can drift away from the transmitter's frequency and phase [101], [111]. Thus, data signals are not sampled at the correct time instance

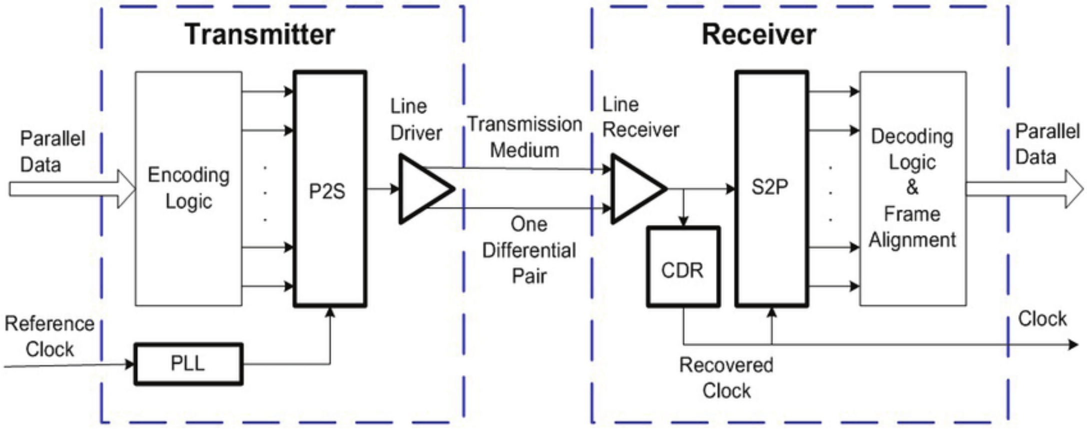


Figure 2.13: A generic architecture of SerDes devices [112].

leading to receiving erroneous data. Hence, to perform and maintain clock locking, the data stream is required to exhibit sufficiently frequent bit transitions [101], [111]. The amount of bit transitions required for CDR depends on how robust the link and the CDR logic are. In Chapter 5, the effect of run length on error rate is investigated for an electrical and an optical link. Results show that the optical interconnect requires a higher number of bit transitions compared to the coaxial link to avoid data reception errors.

In addition, the transmitted data should exhibit an approximately equal number of 0's and 1's to maintain signal integrity and avoid data errors [111]. This attribute is called DC balance. The prevalence of one voltage level causes the drifting of the communication medium towards that voltage level. Consequently, the signal may not transition within the required time. Therefore, the absence of DC balance can lead to data loss [111].

Encoding and decoding logic is added to SerDes devices, as depicted in Fig 2.13, to avoid data loss. This logic ensures that the data stream contains both a sufficient number of edges for CDR and, additionally, an approximately equal number of 0's and 1's. 8B/10B encoding [113] is a popular technique that maps 8-bit characters to 10-bit symbols and guarantees that the run length does not exceed 5 consecutive bits. In addition, data disparity control is performed to achieve DC balance. Disparity of a block of data is defined as *the difference between the number of 1's and 0's* and can be positive (excess number of 1's), negative (excess number of 0's) or zero (number of 1's and 0's is equal). By interleaving blocks of data with positive and negative disparity, DC balance is achieved over time.

A crucial drawback of 8B/10B encoding is the 20% overhead in throughput due to the two additional bits for every byte of data, which significantly limits the performance of the link. Alternatively, 64B/66B and 128B/130B encoding methods exhibit 3.03% and 1.53% overhead in throughput as only 2 bits are added to every 64 and 128 bits, respectively [101]. However, these schemes guarantee only one transition every 66 bits and 130 bits, respectively, which may not be sufficient for clock recovery depending on the transition requirements of the CDR circuit. This behaviour is verified experimentally in Chapter 5, where the application of 64B/66B encoding without scrambling leads to a high data error rate ($\sim 10^{-1}$ errors per packet) and degrades link's performance. In addition, these techniques do not have a mechanism for DC balance preservation, consequently, data errors can occur due to DC bias.

In the early generations of PCIe, 8B/10B encoding is used, whilst in the latest versions, scrambling is utilised to provide sufficient transitions and preserve DC balance by randomising the data stream [111]. The scrambler generates a Pseudorandom Bit Sequence (PRBS) using a Linear Feedback Shift Register (LFSR). The output of the LFSR is XOR-ed with the data stream. In this way, although a maximum run length is not guaranteed, the probability of occurrence of a long bit sequence without a transition is drastically reduced [101]. Furthermore, scrambling maintains DC balance over long periods of time, however, this balance is not ensured for short periods since long sequences of 0's or 1's are still possible [111]. The data are descrambled by the receiver using the inverse function. Therefore, no redundant bits are required and, thus, no overhead in throughput is induced.

8B/10B encoding and scrambling preserve signal integrity, however, at a cost of a drastic increase in transitions and, thus, energy consumption. These techniques do not consider the energy dissipated due to the excess amount of transitions. On the contrary, the encoding techniques [104]-[110] discussed in subsection 2.4.1 that reduce bit transitions are not compatible with high speed, asynchronous, serial interfaces such as PCIe [73]. The reason is that the objective of all of the existing schemes is to achieve the highest decrease in the number of bit transitions. Consequently, CDR is obstructed and, also, DC balance is not considered. Hence, these techniques can lead to data errors.

The reduction in energy consumption of asynchronous, serial interfaces by decreasing bit transitions while maintaining link integrity is investigated in Chapter 5. An encoding scheme is proposed that regulates the number of transitions such that CDR and DC balance are maintained and, at the same time, energy consumption is restricted. This technique can adjust the number of bit transitions to provide either higher energy efficiency or a more reliable CDR.

2.5 Summary

A major bottleneck in meeting the power constraints of modern processors is the power dissipated in interconnects and, especially, in inter-die data transfers. Therefore, in this chapter, methods that decrease the dynamic power consumption of off-die interconnects are reviewed, particularly, low-swing signalling, increasing physical proximity of dies, and signal encoding.

Low-swing signalling decreases the voltage range of the transmitted signals to effectively reduce power, however, this type of communication is either susceptible to noise or exhibits high area penalty and design complexity, such as double number of I/O pins and wires. Emerging system integration solutions, such as 3-D and 2.5-D integration increase physical proximity and, thus, decrease the length of interconnects. However, the capacitive load of interconnects still remains high, therefore, alternative techniques have to be employed to reduce the energy cost of data transfers.

Signal encoding techniques reduce the bit transitions of the transmitted data through data encoding. In this way, the dynamic power consumption is lowered. These techniques address the limitations of the other methods and become more beneficial as technology scales. A review of the existing encoding schemes is provided, which target either parallel or serial interfaces. The former decreases the bit transitions between successive words, whilst the latter reduce transitions within each word.

The existing general purpose encoding techniques for parallel interconnects either provide a moderate reduction in bit transitions or exhibit substantial overhead in power. Alternatively, the low-power encoding techniques for serial communication are only applicable to source synchronous interfaces since asynchronous communication employs clock recovery to extract timing information from the transmitted data. The basic architecture of high speed, source asynchronous, serial interfaces along with the prerequisites for reliable clock recovery are, therefore, discussed.

In the next chapter, the effectiveness of low-swing signalling in reducing the energy dissipation of emerging interposer-based interconnects is explored. Useful rules of thumb are provided regarding the implementation of low-swing signalling in 2.5-D integration.

Chapter 3

Low-Swing Signalling for 2.5-D Technologies

A drastic way to accomplish remarkable energy drop of data transmission is the use of low swing signalling [25] due to the quadratic relation with voltage. Several works have proposed different low swing techniques, where the supported communication includes chip-to-chip interconnects at the package or Printed Circuit Board (PCB) level ([26] -[29]). Consequently, the potential improvement in the energy efficiency of inter-die communication for interposers has yet to be determined. In this chapter, the benefits of utilising low swing for inter-die communication in 2.5-D integration are quantified.

The power benefits due to the shorter physical distance between components in 2.5-D systems utilising full swing interconnects have been demonstrated in [35], [36]. The circuits used to drive the interconnect in the prior art are not well-suited for low swing communication. Consequently, to investigate the effectiveness of the low swing paradigm in different interposer technologies, an appropriate single-ended transceiver proposed in [30] is employed. This circuit supports off-die communication in interposers and offers ultra low level of voltage swing. This transceiver exhibits a 4× decrease in power over full swing communication as shown in [30] for a specific, however, link length. Therefore, this circuit is used as the baseline transceiver to investigate the potential of low swing communication in several physical media developed for 2.5-D integrated systems.

The energy efficiency of the low swing transceiver is accurately determined and the primary energy consuming factors are highlighted for three different substrate materials, silicon, glass, and organic. Additionally, the critical length above which the application of the low swing scheme starts to offer energy savings in comparison with the full swing is determined for each substrate material. Finally, the effect of wire density on the energy efficiency of the low swing circuit is ascertained. In this way, useful rules of thumb are provided regarding the use of low swing communication mechanisms for 2.5-D integrated systems.

The chapter is structured as follows. In Section 3.1, the characteristics of different technologies along with electrical models are presented. The utilised low swing transceiver along with the test circuit is discussed in Section 3.2. The energy efficiency for different technologies, interconnect lengths and wire densities is investigated in Section 3.3, and conclusions are drawn in Section 3.4.

3.1 Interposer Technologies and Modeling

Interposer technologies, or 2.5-D integration, have emerged as a promising solution to decrease the interconnect length and achieve higher integration density [6]. This integration paradigm allows multiple dies to be integrated on both sides of the interposer and be connected through RDLs, allowing an interconnect pitch comparable to on-chip wires and several times smaller than PCB traces. As silicon-based 2.5-D ICs have been commercialised [11], different materials for interposer substrates are developed, moving from standard silicon [21] to other alternatives, such as glass [22] and organic substrate [23], [24]. Although these materials exhibit lower conductivity and thus provide better insulation, the problem of high interconnect power remains crucial, due to additional load capacitance from Through Package Via (TPV), bump bonding and Electrostatic Discharge (ESD) protection.

In this section, the models for different types of interconnects and the electrical characteristics of each type of interconnect are described. The investigated interconnect technologies and the related RDL parameters of each technology are listed in Table 3.1. Three types of interposers are considered. For the silicon interposer, the top metal layers are utilised, thus the physical parameters of global wires are used according to [37]. However, since the RDL on top of interposers is usually larger, a second case with more realistic wire models is also considered [21], where SiO_2 is used as dielectric with relative dielectric permittivity $\varepsilon_r = 3.9$. The parameters of the interconnects on a glass and organic interposer are derived from [22] and [23], respectively. An Ajinomoto Build-up Film (ABF) is used for the insulation of metal layers on the glass interposer, with $\varepsilon_r = 3.35$. In the remaining cases where the dielectric material is not specified, polyimide is assumed for the sake of simplicity, with $\varepsilon_r = 3.5$.

Table 3.1: Interconnect parameters including the minimum width (W), space (S), and thickness (T_W) of the wires, the interlayer dielectric thickness (T_D) and dielectric constant of the passivation layer.

Interposer Technology	W [μm]	S [μm]	T_W [μm]	T_D [μm]	ε_r
Silicon [37]	0.45	0.45	1.2	0.2	3.5
Silicon [21]	2	2	3	1.5	3.9
Glass [22]	3	3	4	7	3.35
Organic [23]	3	3	2.5	2.5	3.5

Table 3.2: *RLC* elements of microbumps and package.

R_{pkg} [Ω]	$R_{\mu bump}$ [Ω]	$L_{\mu bump}$ [nH]	$C_{\mu bump}$ [fF]
0.014	0.095	0.053	5.4

The equivalent circuit of an interposer-based interconnect consisting of three adjacent lines is illustrated in Fig. 3.1. In this model, in addition to the wire electrical parameters, ESD protection, as well as the microbump impedance characteristics are included according to [38]. Furthermore, the *RLC* elements of microbumps, reported in Table 3.2, are assumed equal for all of the interposer materials.

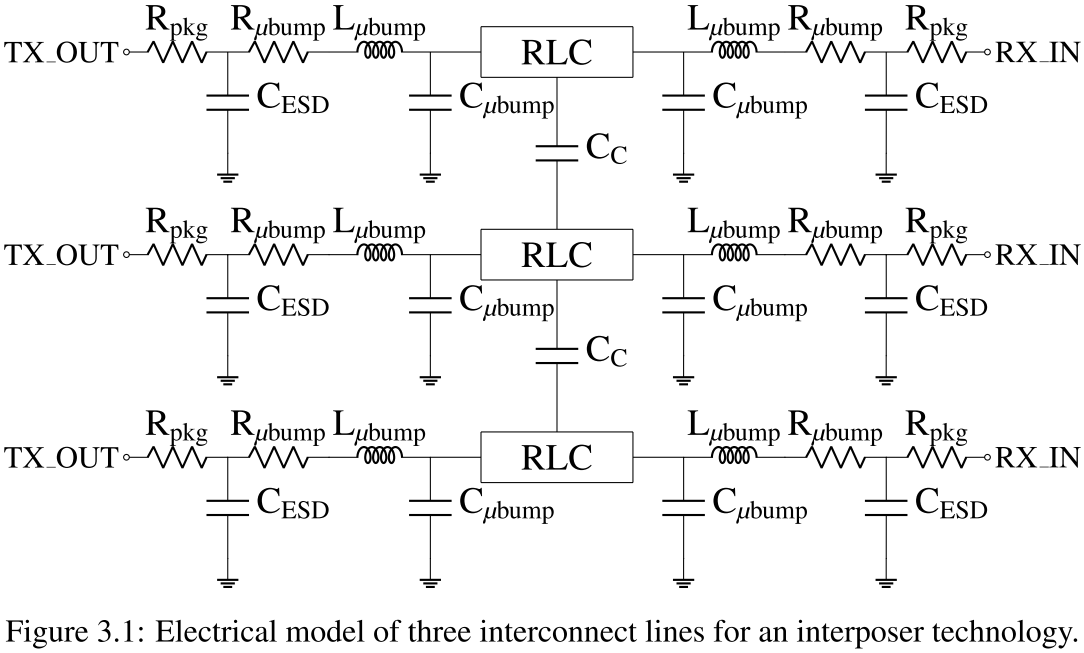


Figure 3.1: Electrical model of three interconnect lines for an interposer technology.

The wires are modeled as distributed interconnects with multiple π-type segments as depicted in Fig. 3.2. The resistance and wire inductance are estimated according to [37] and the mutual inductance is not considered as its effect is negligible [35], while expressions from [39] are utilised for the wire capacitance. Two primary structures are utilised for modeling the capacitance of the interconnects. In the first geometry (S1), the wires are adjacent to a ground plane (Fig. 3.3a), while the second structure (S2) consists of wires sandwiched in-between two ground planes (Fig. 3.3b).

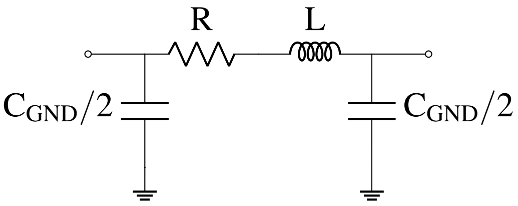
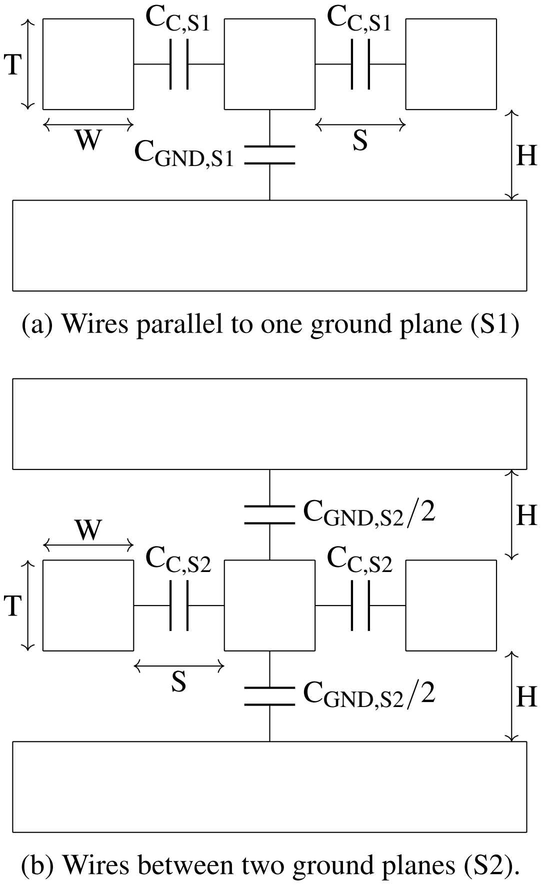


Figure 3.2: A π-type segment.



(a) Wires parallel to one ground plane (S1)

(b) Wires between two ground planes (S2).

Figure 3.3: Cross-sectional view of typical interconnect structures, where (a) one ground plane is present and (b) interconnects are flanked by two ground planes.

Table 3.3: Electrical characteristics of wires for minimum pitch.

Interposer	Structure	R [Ω/mm]	L [nH/mm]	C_{GND} [fF/mm]	C_C [fF/mm]	C_{total} [fF/mm]
Silicon [37]	2	40.74	1.52	222.55	52.45	327.45
Silicon [21]	1	3.67	1.29	85.49	73.49	232.47
Glass [22]	1	1.83	1.23	24.09	69.16	162.41
Organic [23]	1	2.93	1.28	68.79	46.09	160.97

To calculate the wire capacitances, each interconnect geometry is matched to one of the two structures. Specifically, in case of the 2 + 0 + 2 glass interposer stack-up of [22], the glass core is 100 μm thick and, therefore, is safely assumed to behave as an insulator, as the glass resistivity is very high, from 1×10^{12} to 1×10^{16} Ω·cm [40]. Hence, the capacitances of the bottom wires follow the closed-form expressions for S1, where only one ground plane is considered. The organic interposer of [23], is modeled in the exact same way as glass, due to the low permittivity ($\varepsilon_r = 4.4$ [41]) of the organic materials. Alternatively, the silicon substrate, with $\varepsilon_r = 11.2$ [41], is electrically modeled by a ground plane. Thus, the capacitances of the silicon interposer with wire parameters described in [37] are evaluated by utilising the expressions for the second structure (S2). The estimated electrical characteristics of the wires per unit length for each technology are reported in Table 3.3. The total capacitance is calculated as $C_{total} = C_{GND} + 2C_C$ for each of the two structures S1 and S2.

3.2 Simulation Setup

In this section, the setup used for simulations is described. First, a brief description of the low swing transceiver of [30] is provided in subsection 3.2.1. In subsection 3.2.2, the test circuit is presented along with details of the implementation.

3.2.1 Low-Swing Transceiver

The transceiver circuit consists of a dynamic low swing tunable transmitter (DLST-TX) and an inverter-based tunable receiver (INVT-RX). The schematic diagram of the transmitter is depicted in Fig. 3.4. The conversion of signal from full to low swing is achieved through a short propagation delay, introduced by a delay line (three inverters

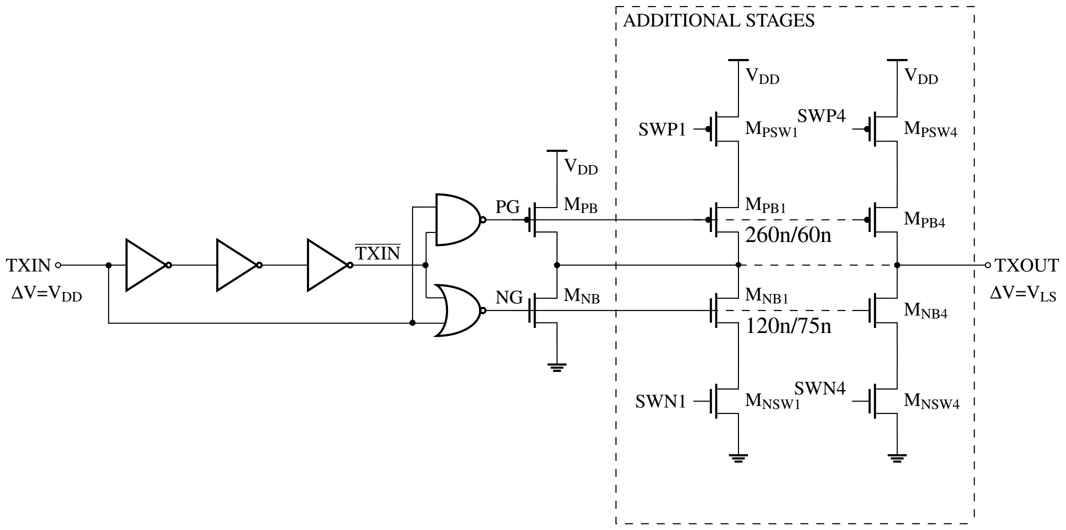


Figure 3.4: Transmitter schematic diagram.

connected in series), during which the output of one of the transistors $\mathrm{M_{NB}}$ or $\mathrm{M_{PB}}$ is turned on, while in steady mode (i.e. the input TXIN does not switch) both are turned off. This short delay time is not sufficient for the output buffer to fully charge or discharge the load capacitance, hence, the voltage range decreases. Hence, the voltage varies by $\pm\Delta$V around a constant V_{DC}, which is selected equal to $V_{DD}/2$. To restore voltage swing, the receiver incorporates a CMOS inverter used as a front end amplifier.

In [30], the additional stages illustrated in Fig. 3.4 are used for trimming the driving strength of the transceiver in order to compensate process variability. However, in this work, the transistors added in parallel to $\mathrm{M_{NB}}$ and $\mathrm{M_{PB}}$ are rather used to adjust the level of the low voltage swing V_{LS}, since V_{LS} depends on the load capacitance, which, in turn, also depends on the interposer technology. V_{LS} is regulated by the number of the activated transistors $\mathrm{M_{PB1}}$ to $\mathrm{M_{PB4}}$ and $\mathrm{M_{NB1}}$ to $\mathrm{M_{NB4}}$ through the respective switches $\mathrm{M_{PSW1}}$ to $\mathrm{M_{PSW4}}$ and $\mathrm{M_{NSW1}}$ to $\mathrm{M_{NSW4}}$. Note that the driving strength of each stage is double compared to the previous stage.

3.2.2 Test Circuit

The transceiver test circuit, designed in a 65 nm technology with 1.2 V supply voltage (V_{DD}), is illustrated in Fig. 3.5. The test circuit consists of three parallel wires; the middle wire is connected to a signal generator that operates at 1 Gb/s speed with 20 ps transition times, while the two neighbouring wires are connected to ground. The energy per bit of both full and low swing transceivers is evaluated for a pseudo-random

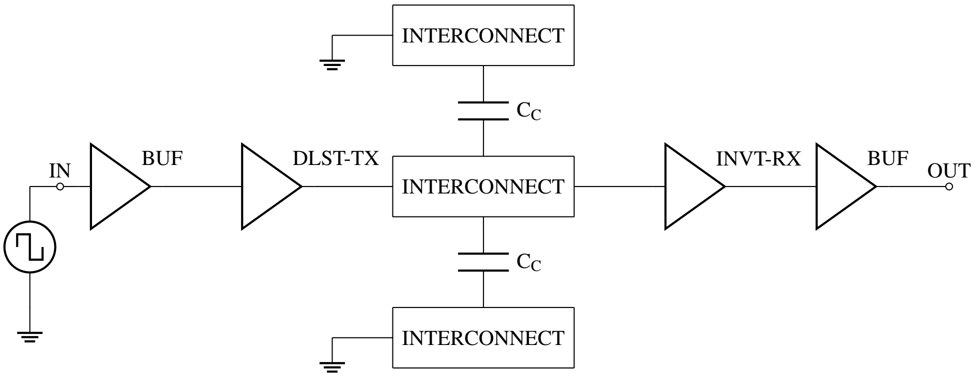


Figure 3.5: Transceiver test circuit.

200 bit long sequence in nominal conditions (typical device corners, 27°C). The two buffers (noted as BUF) preceding the transmitter and following the receiver are used to, respectively, model the driving strength and load of the core logic circuits connected by the transceiver. Finally, the electrical models of the three interconnect lines are depicted in Fig. 3.1.

Note that the energy results do not critically depend on the choice of the low swing (LS) circuit, but primarily on the level of voltage swing and the characteristics of the communication medium. A different implementation of the LS transceiver would only affect the overhead in power, introduced by the transmitter and receiver circuits, required to reduce and restore the voltage level. Therefore, the energy of the low swing scheme would be shifted based on the efficiency of the implementation, however, the behaviour of the interconnect energy would remain the same.

3.3 Energy Improvement and Limitations

The results relating to the energy efficiency of the low swing transceiver for the investigated communication channels are presented in this section. In subsection 3.3.1, for each interposer technology the energy per bit of the LS interconnect is compared with that of the full swing (FS) with and without ESD capacitance. A minimum voltage swing of $V_{LS} = 100$ mV and minimum wire pitch are considered in this scenario. The technology that can benefit more from the implementation of low swing signalling is determined. Furthermore, the length at which the two schemes have equal energy consumption is evaluated. In subsection 3.3.2, the energy efficiency of the low swing versus the full swing transceiver is explored for varying wire densities by adjusting the width and space of the wires.

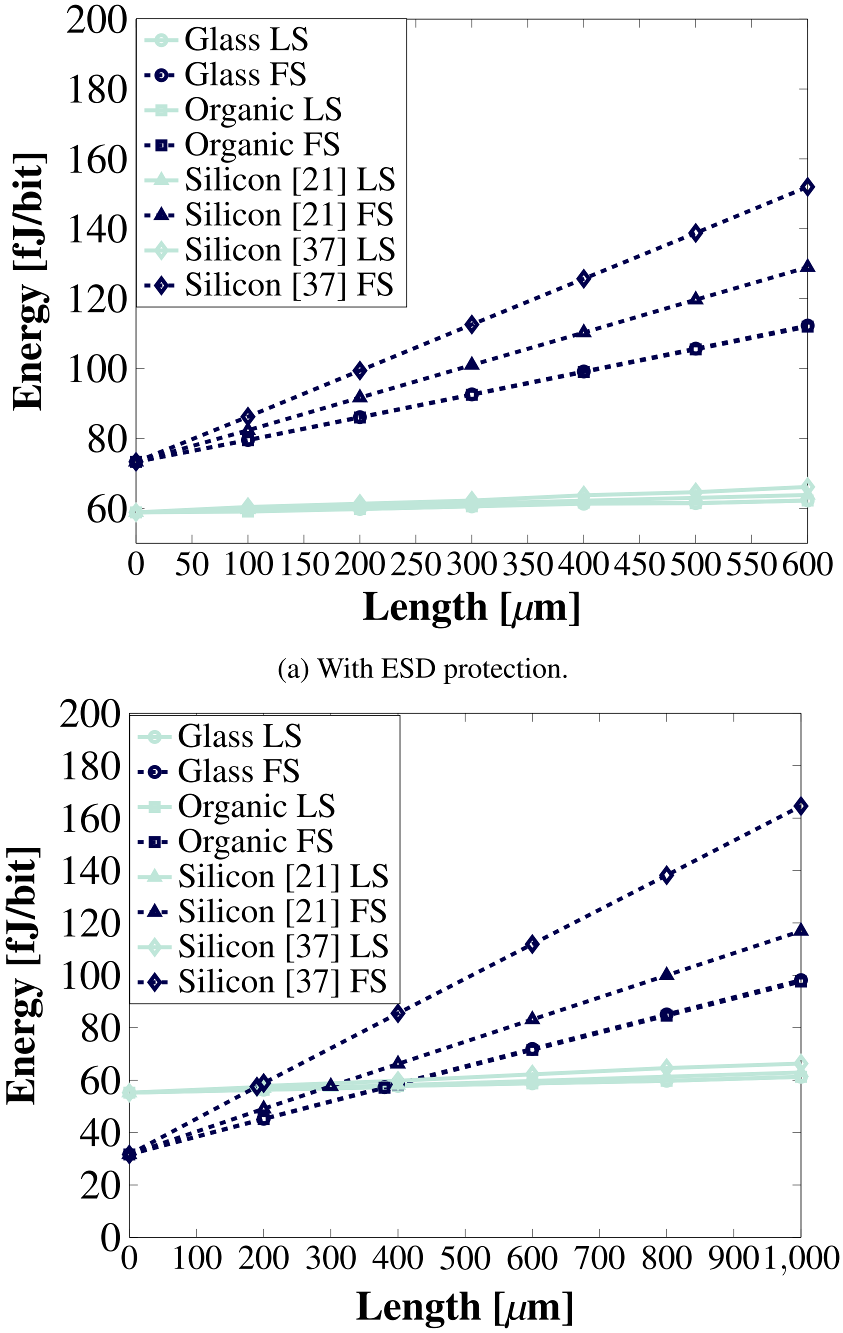
The employed low-swing transceiver does not affect the communication bandwidth. The transmission rate is 1 Gb/s per bus line in both the full swing and the low swing transceivers. Other low swing schemes decrease bandwidth. In this case, the energy per bit increases as the required transmission time per bit increases.

3.3.1 Critical Length for Minimum Wire Pitch

The simulated energy efficiency versus the interconnect length for each interposer technology is plotted in Fig. 3.6. The low swing voltage is kept constant at 100 mV and the interconnect length ranges from 10 μm up to 1 mm. The low boundary of the explored range corresponds to the case where the link is so short that the wire parasitics are negligible. As shown, the energy per bit increases linearly with the increase in length for all different technologies. This behaviour is expected since the power dissipation is proportional to capacitance and longer length corresponds to higher load capacitance. In addition, the slope of the energy depends primarily on the level of voltage swing and secondarily, on the wire capacitance of each technology. Therefore, the energy consumed by the low swing scheme remains relatively steady for each technology. On the contrary, the energy of the full swing scheme increases at a higher pace and the interconnect technology with the steepest slope (i.e. the greatest increase in energy) is the silicon interposer with wire parameters described in [37], which has the highest total wire capacitance.

In Fig. 3.6a, where the ESD capacitance is equal to 50 fF based on [36], [38], the low swing transceiver for all of the interposer technologies features always higher energy efficiency than the full swing regardless the link length. This behaviour highlights the limited benefit of interposer technologies to reduce the power dissipation of interconnects and the capability of low swing signalling to effectively enhance energy efficiency. The energy improvement is even higher for longer interconnects. At 500 μm length the ratio of the energy of the full swing to the low swing scheme is 2.15, while at 1 mm length is estimated 2.9.

Removing the ESD capacitance leads to a considerable reduction of energy for the full swing transceiver, as illustrated in Fig. 3.6b, whereas the shift of the low swing is hardly noticeable. For short interconnect lengths, the energy per bit of the low swing solution exceeds that of the full swing. This behaviour indicates that the power overhead of the low swing circuit, which is dominantly generated by the transmitter to decrease the voltage level, is higher than the power dissipation of full swing within this length range. Hence, this case indicates that the low swing scheme is not suitable for



(a) With ESD protection.

(b) Without ESD protection.

Figure 3.6: Energy *vs* interconnect length for different interposer technologies, where (a) the ESD capacitance is $C_{ESD} = 50$ fF and (b) the C_{ESD} is not considered.

technologies that can support high physical proximity of components, such as in [36].

The critical length along with the total wire capacitance per mm for each interconnect technology are listed in Table 3.4. As shown, the critical length and the C_{total}

Table 3.4: Critical length for different interposer technologies.

Interposer technology	Silicon [37]	Silicon [21]	Glass [22]	Organic [23]
Critical length [μm]	190	300	380	380
C_{total} [fF/mm]	327.45	232.47	162.41	160.97

are strongly correlated. The higher the C_{total} is, the shorter the critical length becomes. Results show that the critical length is the shortest for the silicon interposer technology with the lowest pitch described in [37] and the longest for the glass and organic interposers, demonstrating that for silicon substrates the application of low swing signalling is more beneficial.

3.3.2 Energy Efficiency for Different Wire Densities

To provide the boundaries of energy savings of low swing signalling in the case of relaxed area constraints, an exploration of the energy efficiency with respect to wire parameters, such as width and space, is conducted. The energy efficiency is estimated as the ratio of the energy per bit of the full swing to that of the low swing solution (denoted as E_{FS}/E_{LS}). The V_{LS} is kept constant at 100 mV as well as the length at 300 μm, while ESD protection is excluded. The upper limits of width and space are considered 10× and 5×, respectively, larger than the minimum corresponding value supported by each technology for the glass, organic and the silicon interposer described in [21], while for the silicon interposer in [37] are 5× and 2.5×, respectively.

Results demonstrate that the highest energy savings in absolute power terms are exhibited by the interconnects of silicon interposers, as shown in Figures 3.7 and 3.8. The highest ratio E_{FS}/E_{LS} is demonstrated for maximum width and minimum space, which is 2.1 and 1.8 for the interconnect in [37] and [21], respectively. In this case, the power dissipation of the full swing scheme is high since both C_{GND} and C_C are significant, resulting in a high capacitive load. On the contrary, the application of low swing signalling is less beneficial for low width and large space, where the energy consumption of the full swing decreases. Especially, the efficiency of the interconnect in [21] drops below 1 in this area, which means that the energy dissipation of the low swing is higher than the full swing circuit. Furthermore, the energy for both types of interconnects shows a strong dependence on the width of the wires, while the effect of space is rather weak, diminishing fast farther from the minimum value.

Interconnects on glass and organic interposers feature lower energy efficiency by

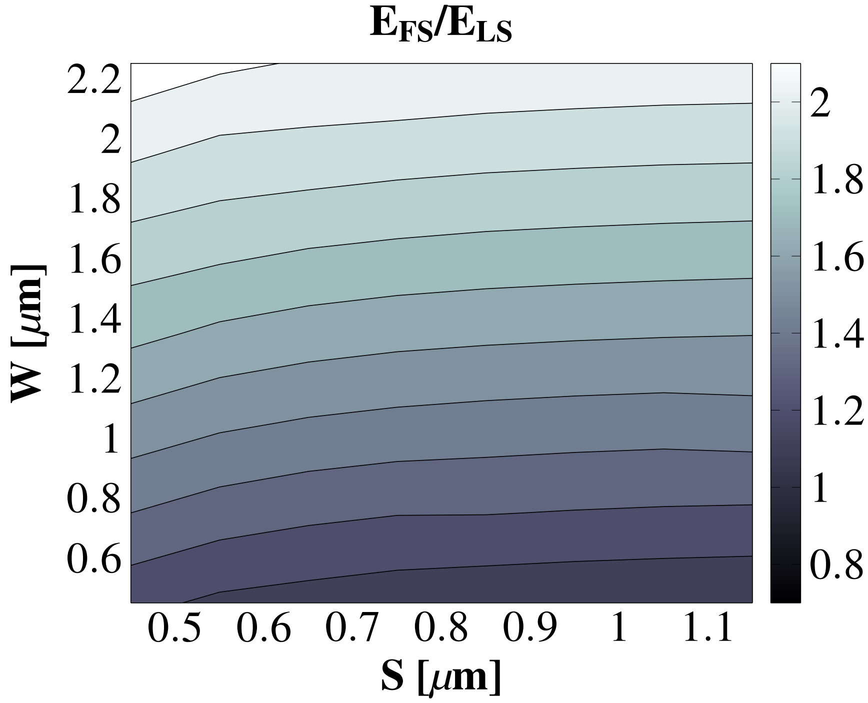


Figure 3.7: Energy efficiency for the silicon interposer with wire parameters described in [37].

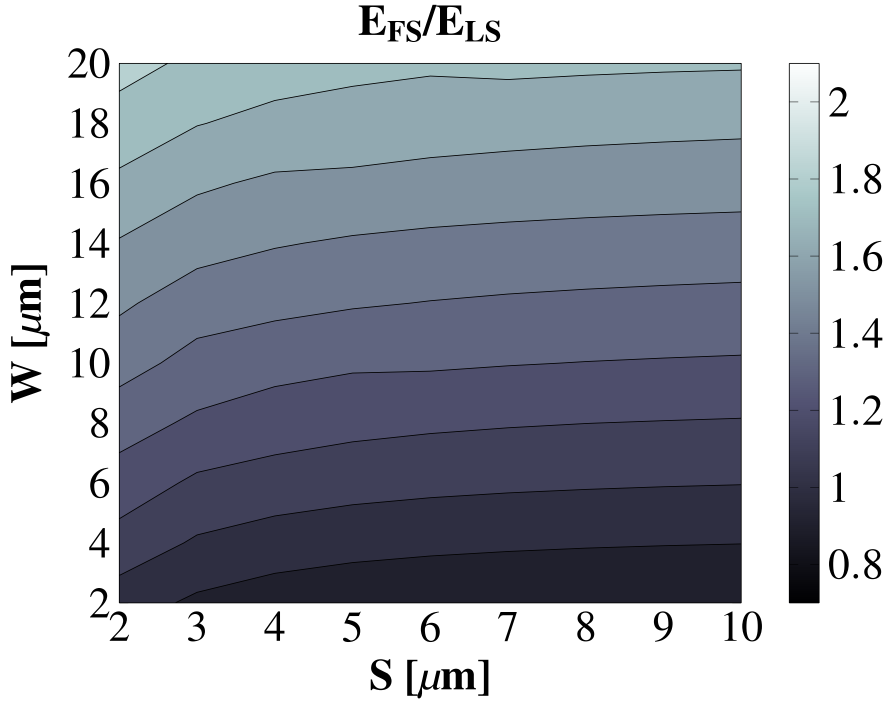


Figure 3.8: Energy efficiency for the silicon interposer described in [21].

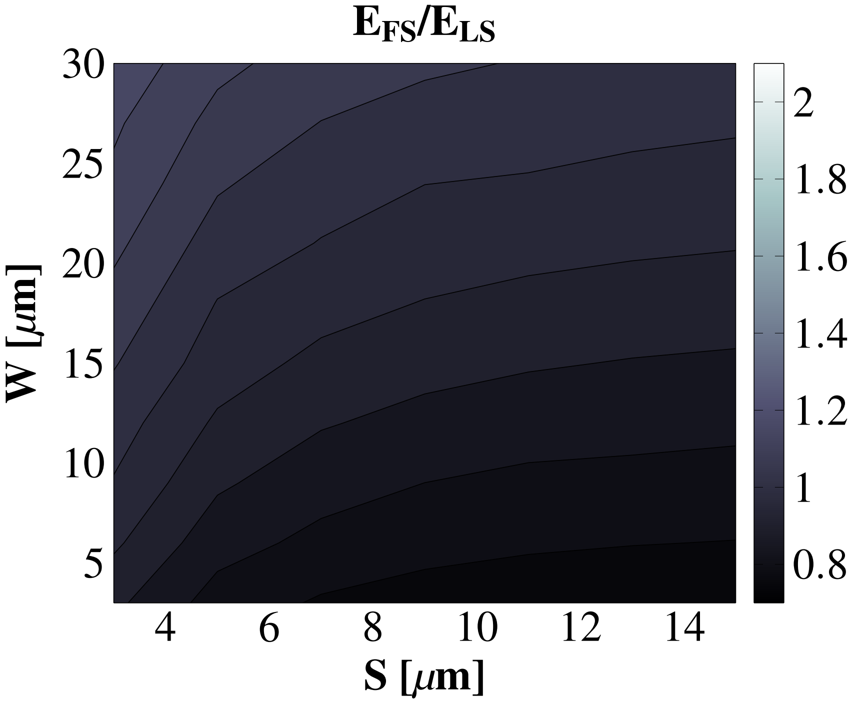


Figure 3.9: Energy efficiency for the glass interposer.

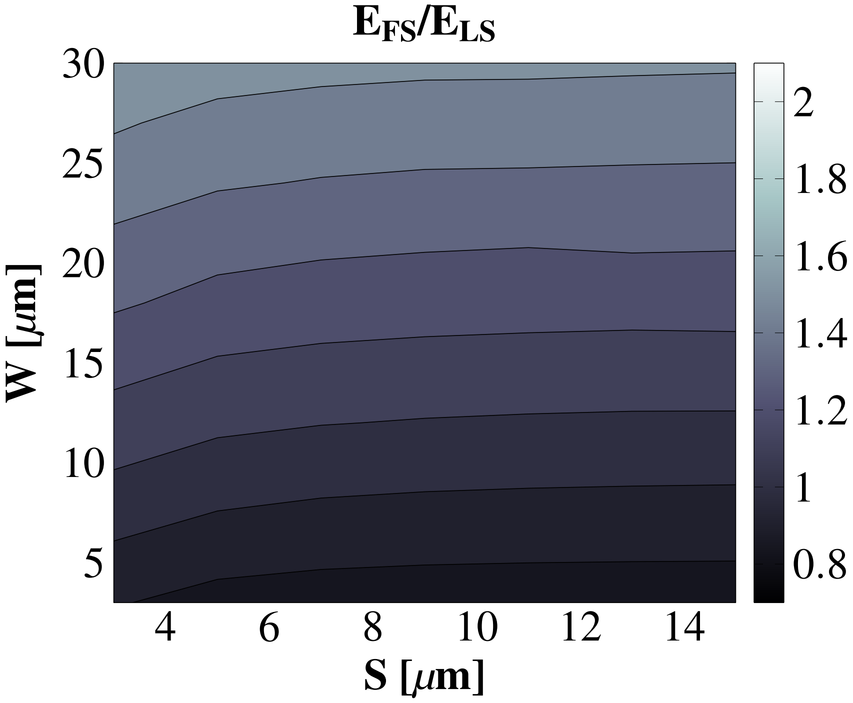


Figure 3.10: Energy efficiency for the organic interposer.

the implementation of the low swing scheme compared to silicon, as illustrated in Figures 3.9 and 3.10. E_{FS}/E_{LS} follows the same trend for the organic interposer as for the silicon interposers, which means that the width of wires has the highest impact on power and thus, on the energy efficiency. E_{FS}/E_{LS} reaches the highest value of 1.5 for maximum width and decreases at about 0.9 for minimum width. Alternatively, the power dissipation for the glass interposer is highly affected by the spacing of wires. Due to the large thickness of the passivation layer of the glass interposer, the largest component of the C_{total} is due to the coupling of the wires. Consequently, larger spacing results in the drop of the power dissipated in the link and diminishes the efficiency of low swing. The glass interposer benefits less by the low swing scheme. E_{FS}/E_{LS} ranges from 0.8 for maximum space and low width to 1.15 for minimum space and maximum width.

3.4 Summary

Interconnects often constitute a major bottleneck in the design process of low power integrated circuits. Although 2.5-D integration technologies support physical proximity, the dissipated power in the communication links remains high. In this chapter, the power savings enabled by low swing signalling is investigated for interposer-based interconnects. The energy consumed by a low swing scheme is, therefore, compared with a full swing solution. The critical length of the interconnect, above which the low swing solution starts to pay off, is determined for diverse interposer technologies. The energy consumption is compared for three different substrate materials, silicon, glass, and organic.

Results imply that energy consumption depends highly on the total capacitance of the driven load and not just the interconnect length. This behaviour shows that only the physical proximity enabled by the interposers may not be sufficient to provide the energy efficiency required by future interconnects. Low swing techniques can decrease the power dissipation of inter-die communication for 2.5-D integrated systems. The low swing solution provides higher power savings for silicon interposers which have the highest wire capacitance.

Furthermore, when ESD protection is required, the low swing signalling is always superior to the full swing due to the high capacitive load of the ESD circuit, regardless the substrate material and the link length. Alternatively, without ESD protection and for short interconnect lengths, the power overhead of the low swing circuit is higher than the power dissipation of the full swing. In this case, the critical length at which the two circuits demonstrate equal energy consumption is the shortest for silicon interposers and reaches up to 380 μm for glass and organic interposers for the explored interconnect parameters.

Finally, to further explore the limits of power reduction from low swing signalling for 2.5-D ICs, the effect of typical interconnect parameters, such as width and space on the energy efficiency of low swing communication is evaluated. The energy efficiency of the low swing signalling increases in the case where large wire width in combination with low spacing are required.

Chapter 4

Adaptive Time-based Encoding for Parallel Interfaces

In the previous chapter, the energy benefits by applying low-swing signalling are investigated. The potential decrease in energy is shown to be high and the critical interconnect length above which low-swing signalling starts to save energy is just 190 μm for silicon interposer-based links. Nevertheless reducing the voltage swing can lead to a decrease in the noise margin [114]–[115].

An alternative way to reduce the energy of communication is signal encoding. A new technique is presented in this chapter, which encodes data to reduce bit transitions and, thus, the frequency of charging and discharging the capacitive load associated with the interconnect. This load comprises the ground capacitance and the coupling capacitance. Therefore, in addition to the power benefits, encoding can also reduce crosstalk by reducing the frequency of charging and discharging the coupling capacitance. A novel encoding scheme is proposed that is capable of attaining significant power savings.

Encoding schemes can significantly reduce power particularly when the switching behaviour of the data stream is known in advance. However, this may not be feasible or the characteristics of the data stream may vary over time. Therefore, the proposed encoding scheme, namely Adaptive Word Reordering (AWR), is adaptive, hence, the statistics of the data stream are not required to be known *a priori*. AWR decreases the switching activity for wide off-die buses without affecting the communication throughput. The core idea of this technique is the reordering of the words of the data stream such that signal transitions are minimum. Optimal reordering requires a formidable amount of computations, therefore, a heuristic algorithm, namely Nearest Neighbour

(NN) is preferred. The proposed scheme is envisioned for block data transfers, such as Direct Memory Access (DMA) where the data is transmitted only after a block is fully available and, thus, the latency introduced by encoding does not affect the performance of the communication. In addition, AWR is compared with state-of-the-art techniques in terms of total power savings including the power overhead of the encoder and decoder logic. Furthermore, the resilience of the proposed circuit implementation to process variations is explored and the related trade-offs between accuracy, power, and timing are discussed.

Although reordering data has been attempted by the Sequence-switch scheme [48], the related algorithm does not fully exploit the potential of this approach in reducing switching activity as only the two most recent words are considered in the reordering operation. Consequently, the provided reduction in switching activity is comparable to BI which is shown to typically be inferior to AWR in Section 4.5. Furthermore, Sequence-switch does not provide a circuit implementation, therefore, the circuit overhead in power is not considered. However, as shown in [49], the effectiveness of encoding in reducing power consumption is often restricted by the high implementation overhead, thereby diminishing and often eliminating the benefits offered theoretically by data encoding techniques. Hence, the efficiency of the Sequence-switch technique can not be evaluated properly.

On the contrary, in this work the NN search is considered for the reordering, which effectively decreases bit transitions. To the best of the author's knowledge, NN has not been used for data encoding before. Additionally, a novel circuit that implements NN is proposed, which exploits the time domain for complex computations of Hamming distances, to decrease the power overhead of encoding. AWR decreases the power consumption of transferring data while the circuit power overhead is low compared to the saved power. Therefore, AWR provides significant power savings ($\sim$23%) as compared to state-of-the-art techniques discussed in Chapter 2.

The remainder of this chapter is structured as follows. The proposed technique is described in Section 4.1, while, in Section 4.2, is compared with competitive algorithms in terms of decrease in toggle rate. The circuit implementing this method is described in Section 4.3 and the simulation setup utilised for power measurements is presented in Section 4.4. The power savings and limitations of the proposed technique are quantified in Section 4.5 and are also compared with existing encoding schemes. Finally, a summary is provided in Section 4.6.

4.1 Encoding Algorithm

The proposed encoding scheme (AWR) is based on the observation of the data stream over a fixed window of N words and the dynamic reordering of these words in order to decrease the total number of transitions on the encoded bus. The proposed scheme can be applied to memory interfaces that do not use asymmetric termination such as Low-Power Double Data Rate 3 (LPDDR3) where power can be saved by reducing the number of bit transitions. Furthermore, access to LPDDR3 is burst oriented. Therefore, a block of read or write data is fully available at the beginning of the transmission, required by the encoder of AWR in order to not introduce additional latency.

The problem of the optimal word reordering can be described by considering each word as a vertex of a complete, undirected, weighted graph $G(V,E)$. The weight of each edge, w, is the Hamming distance of each pair of words. The problem can be reformulated as the identification of the minimum weight route that visits all the vertices of the graph exactly once. This is very similar to the Travelling Salesman Problem (TSP) classified as NP-hard.

The difference between the two problems is that the route in TSP is cyclic and has to end to the starting vertex. However, this difference does not change the computational complexity of the problem. An example of a graph is illustrated in Fig. 4.1a, where the number of reordered words is $N = 4$ and the word length is $M = 8$ bits. The minimum weight route is highlighted.

The most straightforward solution to the TSP is an exhaustive search, for which the calculation of the cost of all possible rearrangements ($N!$) is required. Exact algorithms that determine the optimal solution and alleviate the computational cost have been developed [92]. However, the computational cost remains high and the hardware complexity of these algorithms is prohibitive if any savings in power are to be harvested.

There is a plethora of heuristic algorithms that determine near-optimal solutions to the TSP problem. In this work, the Nearest Neighbour search is implemented as a less complex approach. The NN algorithm is properly adapted for the specific problem. In each clock cycle, the search starts with the previously transmitted word and the weights with all the unvisited vertices (words that have not been transmitted) are calculated. To keep track of the order, a unique code of $K = \log_2 N$ bits is assigned to each word before transmission. In this way, data can be placed back in the initial order at the receiver side. These K bits are included in the calculation of the Hamming distance to ensure that these bits do not notably increase the switching activity. In the example of

Algorithm 2 Pseudocode of the AWR algorithm.

```
while Queue with words not empty do
    Update the $N$ words to be reordered
    Assign a unique code of $K$ bits to all $N$ words
    for $i \leftarrow 0$ to $N-1$ do
        for $j \leftarrow 0$ to $N-1$ do
            if word[$j$] not transmitted then
                Calculate the Hamming distance between word[$j$] and previously transmitted word including the unique code of $K$ bits
        Transmit the word with the minimum Hamming distance
```

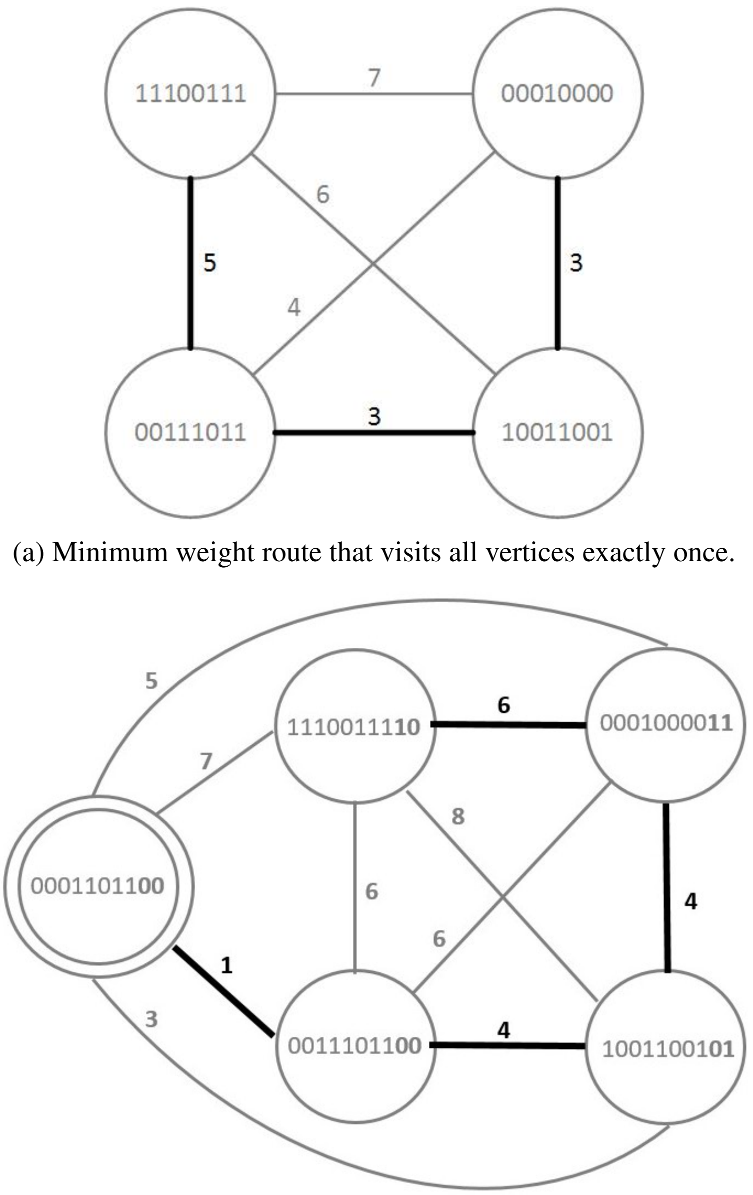
Fig. 4.1b, the previous word is assumed to be 00011011, thus, the search starts from this vertex. The last two bits of each word are the order bits (shown with boldface). The route followed by the NN algorithm to reduce transitions is highlighted.

Using $K = \log_2 N$ redundant bits is the simplest way to represent the order and allows the decoder to retrieve the order of each word at the cycle the word is received. However, this method requires the highest overhead of $\log_2 N$ redundant bits per word and, thus, the highest number of I/O pins and wires. For example, the overhead is 5 bits per word when $N = 32$. Ways to reduce this overhead exist. Since the number of all possible rearrangements is $N!$, $\log_2(N!)$ bits in total can be used to represent the order. Hence, $\log_2(N!)/N$ redundant bits per word are sufficient. For example, 4 bits per word are required when $N = 32$.

The pseudocode of the algorithm is shown in Algorithm 2. N transmissions are required to transfer the N words over the interconnect. Therefore, index i is used to count the number of transmissions. Index j is used to iterate between all the words to find the one with the minimum Hamming distance for each transmission. If a word has been transmitted, then the Hamming distance is not calculated and the word is not considered for retransmission.

4.2 Algorithmic Performance Evaluation

The theoretical performance of the proposed reordering technique is determined by the switching activity, which is compared with three other encoding schemes. These schemes are described in detail in the related work in Chapter 2 and are also briefly discussed in the following subsection 4.2.1, while the related results are presented in subsection 4.2.2.



(a) Minimum weight route that visits all vertices exactly once.

(b) Route followed by the Nearest Neighbour algorithm. The search is assumed to start at the vertex 00011011.

Figure 4.1: Examples of routes that reduce bit transitions.

4.2.1 State-of-the-Art Techniques

Three general purpose encoding techniques are selected that do not require *a priori* knowledge of data statistics and a circuit implementation is provided. BI [46] is a low power adaptive scheme that calculates the Hamming distance of consecutive data words and inverts the transmitted data word if the Hamming distance is higher than half of the word length. To indicate whether a word is inverted or not, one extra bus line is used. APBI [60] observes data for a window of a fixed number of N words and forms a mask with the bus lines with the higher probability of switching. BI is applied to these bus lines and one extra bit is used to inform the decoder about inversion. The mask is computed by both the encoder and the decoder for the same N words.

ABE [72] selectively encodes a cluster of highly correlated bus lines. First, observes the data characteristics over a window of N words. Based on these characteristics a cluster is formed and the line with the maximum correlated transitions with the lines of the cluster is selected as the basis line. The lines in the cluster are finally XOR-ed with the basis. The basis and cluster information are transmitted using a redundant bus line and an additional clock cycle at the beginning of the window, respectively.

The parameters of all the encoding schemes as well as the related (spatial and temporal) redundancies are listed in Table 4.1. The parameters of the compared techniques APBI and ABE are specifically selected such that the techniques yield the highest reduction in switching activity according to [60] and [72]. In this way, a fair comparison is conducted that provides the highest savings for each technique. For all of the simulations, the bus width is considered to be $M = 64$ bits whilst the observation window is $N = 32$ words for APBI and $N = 16$ words for ABE. The mask computation of the APBI technique can be executed in each window of N words (APBI$_1$), but intervals of 16 windows (APBI$_{16}$) are also explored to further reduce the power consumption of encoding and decoding and provide a fairer comparison. Therefore, both of these scenarios are included. Furthermore, two cases for ABE are considered. In the first case, ABE is applied to the entire bus (ABE$_1$) while in the second case, the bus is split in 4 groups of $M/4$ bits and ABE is applied individually to each group (ABE$_4$) to best exploit this technique.

The decrease in switching activity of AWR is reported using both $N = 32$ (AWR$_{32}$) and $N = 64$ (AWR$_{64}$) reordered words. The savings increase with a higher number of reordered words, however, the power overhead of the encoding and decoding circuits increases faster. For lower number of reordered words the savings in bit transitions and, thus, the power savings are lower. Therefore, these two values ($N = 32$ and $N = 64$) are selected as high savings in switching activity are provided while the power overhead is restrained, therefore, leading to maximum power savings. This behaviour is demonstrated in subsection 4.5.2.

AWR with window size $N = 32$ is applicable to 32 byte cache line transactions in Central Processing Units (CPUs). In cases where the data burst size is higher, such

Table 4.1: Parameters of the investigated encoding techniques. The parameters are selected such that each technique yields the highest decrease in switching activity.

	AWR_{32}	AWR_{64}	BI	APBI_1	APBI_{16}	ABE_1	ABE_4
Observation window N	32	64	1	32	32	16	16
Bus width M	64	64	64	64	64	64	16
Update of observation window in clock cycles	32	64	1	32	512	17	17
Spatial redundancy in bits	5	6	1	1	1	1	4
Temporal redundancy in clock cycles	0	0	0	0	0	1	1

as in Last Level Cache (LLC) and DMA transfers, the burst can be split to blocks of 32 or 64 words and AWR can be applied individually to each block. However, to prevent inducing a large latency penalty, AWR should be applied to each block in parallel, hence, increasing the area overhead. In case of Advanced eXtensible Interface (AXI) transactions, the size of a data burst ranges from 1 to 256 transfers in AXI4. AWR requires a fixed size of data bursts and the power overhead of AWR increases rapidly with size. Hence, AWR is not directly applicable to AXI interfaces. AWR can be improved to handle multiple size bursts. Specifically, in case the burst size is lower than the window size of AWR, the extra delay lines in the encoder circuit can be disabled by properly initialising register REG1 (see Fig. 4.2).

4.2.2 Savings in Switching Activity

The switching activities are determined for diverse data types, both synthetic and real, reported in Table 4.2. The LFRic, InfOli, LITE LOOP, LU SOLVER, and NASTY EXPS benchmarks comprise a mix of memory addresses and data and are generated from different applications. LFRic [93] is a weather forecasting and climate research atmospheric model, InfOli is a simulator of the brain Inferior Olive, and LITE LOOP, LU SOLVER, and NASTY EXPS are microkernels from the Integrated Forecast System [94]. Furthermore, an image file is used as well as uniformly distributed random data generated with Matlab.

The results of the decrease in switching compared to the unencoded data streams are listed in Table 4.3. The results for the proposed AWR technique and BI are reported in Table 4.4a and for APBI and ABE in Table 4.4b. In these results, the spatial and temporal redundancies of each technique are also considered in the calculation of the switching activity. Moreover, in addition to the self transitions, the relative transitions are also listed as coupling is not negligible even for off-die interconnects.

The AWR scheme provides greater self-switching savings for most types of data streams. For the multiplexed address and data streams, AWR_{64} yields the highest savings in switching activity with 46–48% reduction in bit transitions. Savings for the image file are, also, high ($\sim$33%) while the lowest are observed for random data. In the latter case, ABE_1 provides the highest decrease in switching, yielding $\sim$17%.

The results of Table 4.3 highlight that the transitions contributed from the redundant bus lines of AWR add a low overhead as demonstrated by the high savings in bit switching for $N = 32$ and $N = 64$. For lower numbers of reordered words (N) the number of redundant bit lines decreases. However, this situation leads to lower savings in switching activity and, hence, in power. This trade-off is also demonstrated in the results of subsection 4.5.2.

Relative transitions are calculated based on the transitions of adjacent bus lines according to Table 2 in [80] and can cause the charging or discharging of the coupling capacitance. The ground capacitance of off-chip wires is, typically, much higher compared to the coupling capacitance, therefore, the reduction of self transitions is crucial. However, the coupling capacitance is not negligible, hence, the impact of relative switching on power is notable. Although all of the techniques in Table 4.3 target minimising self transitions, relative transitions are also affected mostly in a positive way (*i.e.* transitions are reduced). The reduction in relative transitions of AWR_{64}

Table 4.2: Characteristics of data streams.

Data	Type	Size	Total Transitions	
Streams			Self	Relat.
LFRic	Address-data	2.44 MB	2,987,633	3,650,307
InfOli	Address-data	2.44 MB	2,961,115	3,537,631
LITE_LOOP	Address-data	2.44 MB	2,961,773	3,539,832
LU_SOLVER	Address-data	2.44 MB	2,962,321	3,540,071
NASTY_EXPS	Address-data	2.44 MB	2,961,212	3,539,298
Image	Data	17.14 MB	29,189,114	61,056,347
Random	Data	2.44 MB	5,120,736	10,079,981

Table 4.3: Decrease in both self and relative switching activity for the different encoding schemes for diverse data streams.

(a) Proposed AWR and BI.

Data	AWR$_{32}$		AWR$_{64}$		BI [46]	
Streams	Self	Relat.	Self	Relat.	Self	Relat.
LFRic	36.88%	10.89%	**46.55%**	**25.20%**	14.37%	0.57%
InfOli	35.99%	10.55%	**47.99%**	**26.02%**	14.92%	0.54%
LITE_LOOP	36.02%	10.59%	**48.00%**	**26.06%**	14.91%	0.52%
LU_SOLVER	35.91%	10.61%	**48.00%**	**26.05%**	14.95%	0.53%
NASTY_EXPS	36.01%	10.58%	**47.99%**	**26.03%**	14.94%	0.52%
Image	29.87%	31.25%	**32.79%**	**34.10%**	6.04%	5.91%
Random	13.08%	13.03%	15.64%	15.53%	8.54%	8.50%

(b) APBI and ABE.

Data	APBI$_1$ [60][1]		APBI$_{16}$ [60][1]		ABE$_1$ [72][1]		ABE$_4$ [72][1]	
Streams	Self	Relat.	Self	Relat.	Self	Relat.	Self	Relat.
LFRic	13.48%	-13.18%	12.55%	-13.99%	17.91%	-0.78%	18.90%	-6.04%
InfOli	14.29%	-13.31%	13.45%	-14.73%	18.74%	-0.34%	19.80%	-5.68%
LITE_LOOP	14.30%	-13.35%	13.46%	-14.78%	18.74%	-0.32%	19.80%	-5.67%
LU_SOLVER	14.29%	-13.35%	13.46%	-14.73%	18.75%	-0.32%	19.81%	-5.66%
NASTY_EXPS	14.30%	-13.35%	13.47%	-14.74%	18.74%	-0.32%	19.80%	-5.67%
Image	9.89%	10.50%	7.76%	8.26%	17.27%	18.68%	15.69%	17.04%
Random	6.45%	6.48%	6.41%	6.46%	**16.66%**	**16.67%**	15.82%	15.32%

[1] Minus sign implies an increase in the number of transitions.

range from ∼15% to ∼34%. The other techniques also provide high reduction for the image file and the random data. In case of the multiplexed address-data benchmarks, APBI and ABE cause a slight increase in the relative transitions. The likely cause of this behaviour is that whilst aiming to minimise self transitions, transitions of adjacent bus lines in opposite directions are produced. This type of transition dissipates more power compared to other switching events (e.g., relative transitions in the same direction) [79], [80]. Consequently, increasing the number of transitions in opposite directions can increase the contribution of the coupling capacitance in the overall power figures.

4.3 Circuit Architecture

The circuit architecture of the proposed encoding scheme is provided in this section. The encoder and decoder circuits are described in subsections 4.3.1 and 4.3.2, respectively.

4.3.1 Encoder

The encoding of data is implemented as follows. In each clock cycle, the Hamming distances between the previous word and the words that have not yet been transmitted are evaluated. The word with the lowest Hamming distance is then transmitted through the bus. This method requires N registers at the transmitter and the receiver to store the reordered words.

Conventionally, the computation of the Hamming distance is implemented using adder trees. This approach is highly inefficient in terms of power, especially for wide buses, counteracting any energy savings produced from encoding. Therefore, a different approach is followed, where the Hamming distance is determined as a delay in the time domain, drastically reducing the overhead in power due to encoding.

The proposed encoder circuit comprises three stages, as depicted in Fig. 4.2. The first stage is the race stage, where a variable delay line is assigned to each word. In each delay line, a clock pulse is propagated and delayed according to the number of bits that switch. The delay is shorter for a lower number of transitions and, thus, the fastest signal corresponds to the word with the lowest Hamming distance.

The DEL signal that arrives first at the finish line of the race stage prevents the others from propagating. This condition is implemented in the finish stage. Before any signal arrives, a "0" is stored in all of the latches (LA) and the PER signal is set to 1 using a weak pull-up resistor. The signal that arrives first, sets the respective latch and resets PER. After all the DEL signals are reset, PER switches slowly back to 1 due to the weak pull-up resistor.

The winner stage is composed by the selection block and two registers. The selection block is a digital circuit that decides which word wins the race according to the received SEL[0..N] signals. In case two or more signals arrive at the same time (i.e. these words yield the same number of switching bits), such that more than one of the SEL signals is equal to 1, the word with the lowest index is chosen. Transmitting the word with the lowest index can potentially decrease the delay of decoding. For example, if WORD[0] and WORD[7] arrive at the same time, there is no benefit in

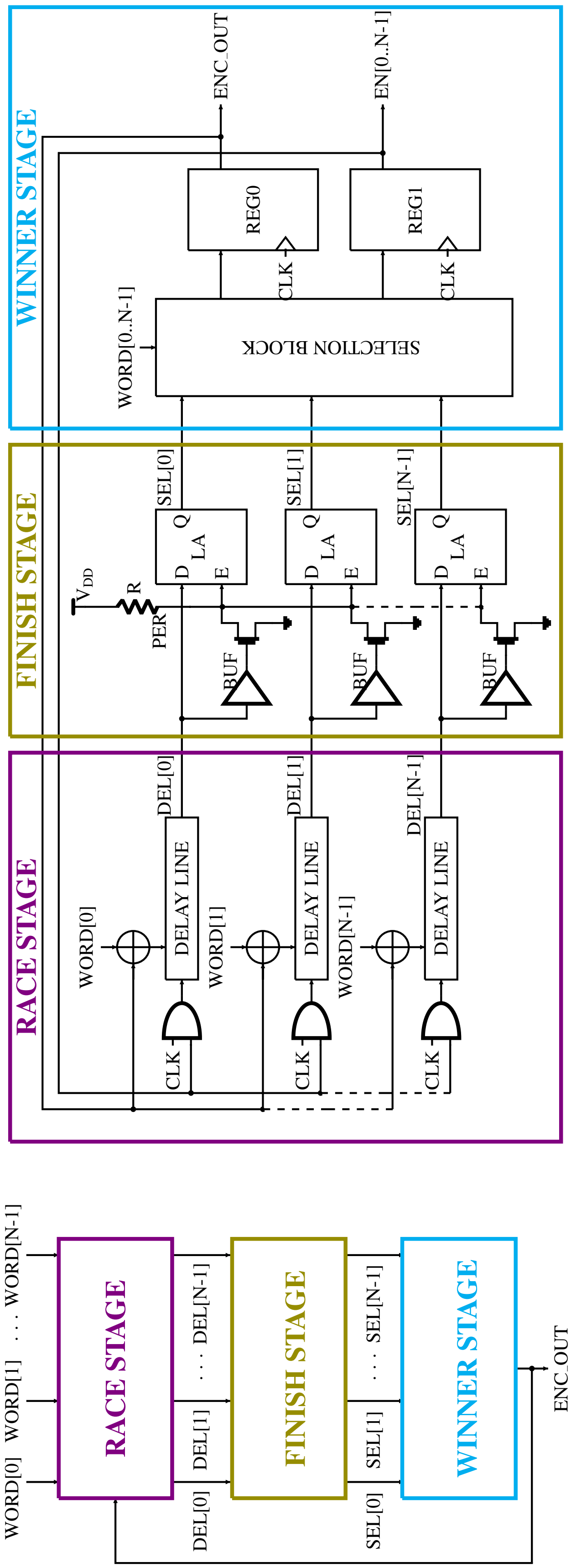


Figure 4.2: Encoder circuit.

transmitting WORD[7] since the receiver can not utilise WORD[7] if all the previous words have not been read. The winning word is stored in the register REG0. To keep track of the transmitted words, a second register (REG1) is used, where the enable signals EN[0..N-1] are stored. Once the word to be transmitted is selected, the respective EN signal is switched to 0 and remains low until all N words are transmitted. All EN signals switch to high whenever a new block of N words is to be transmitted. EN signals enable or disable the respective delay lines allowing the propagation of the clock only for the words that have not been transmitted. To enable and disable the delay lines, an AND gate is added in the beginning of each delay line. This mechanism not only prevents from transmitting a word twice but also saves dynamic power since the inverters in the delay lines of transmitted words do not switch.

An example of signal propagation with $N = 2$ is illustrated in Fig. 4.3. It is assumed that in the first clock cycle the Hamming distance of Word[1] is the lowest, therefore, DEL[1] is set to 1 faster than DEL[0]. DEL[1] causes both SEL[1] and PER signals to transition. Thus, SEL[1] is set to 1 and the latches are disabled before DEL[0] transitions to 1. PER switches slowly back to 1 after both DEL[1] and DEL[0] are reset. In the second clock cycle, the state of EN[1] changes to 0 as Word[1] was selected, while the clock pulse does not propagate through the delay line of Word[1], therefore, DEL[1] remains 0. Word[0] is selected in this cycle, since DEL[0] is the only signal that transitions to 1, causing SEL[0] and PER to flip. In the third clock cycle, both EN[0] and EN[1] are equal to 1 in order to enable the reordering of the next two words.

The delay line consists of a modified inverter chain as illustrated in Fig. 4.4, where W is the minimum width and the length is the minimum for all devices as determined by the utilised technology library. Each inverter is connected either to the ground or the supply voltage through a pair of devices. A detailed description of this delay line can be found in [95]. Briefly, when the i^{th} bit switches, T(i) switches to 1 and the i^{th} inverter is connected to ground (i is odd) or V_{DD} (i is even) through only one device. In case no transition takes place, the inverter is connected to either V_{DD} or ground through two devices connected in parallel. Hence, in the former case the delay is larger than in the latter case. The delay of only the first edge of each inverter (falling for even and rising for odd) is affected by the Hamming distance since the pair of devices is only added to either the pull-down (even inverters) or pull-up (odd inverters). The delay of the second edge of each inverter is constant. This implementation ensures that the delay of the falling edge of the DEL signals is constant. Therefore, the PER signal

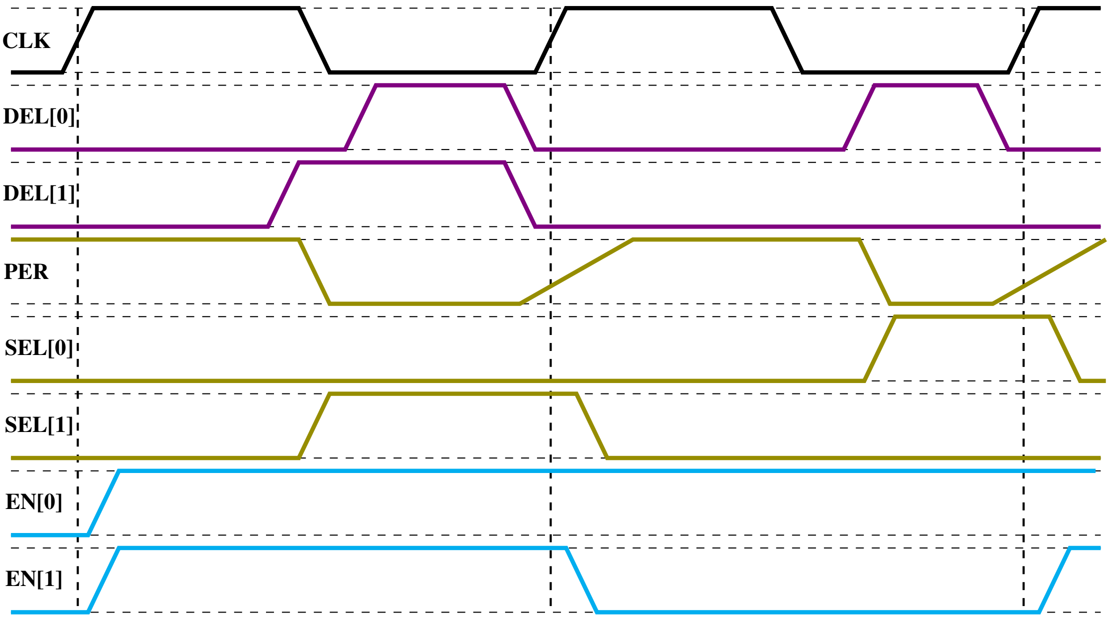


Figure 4.3: Timing diagram of the encoder circuit.

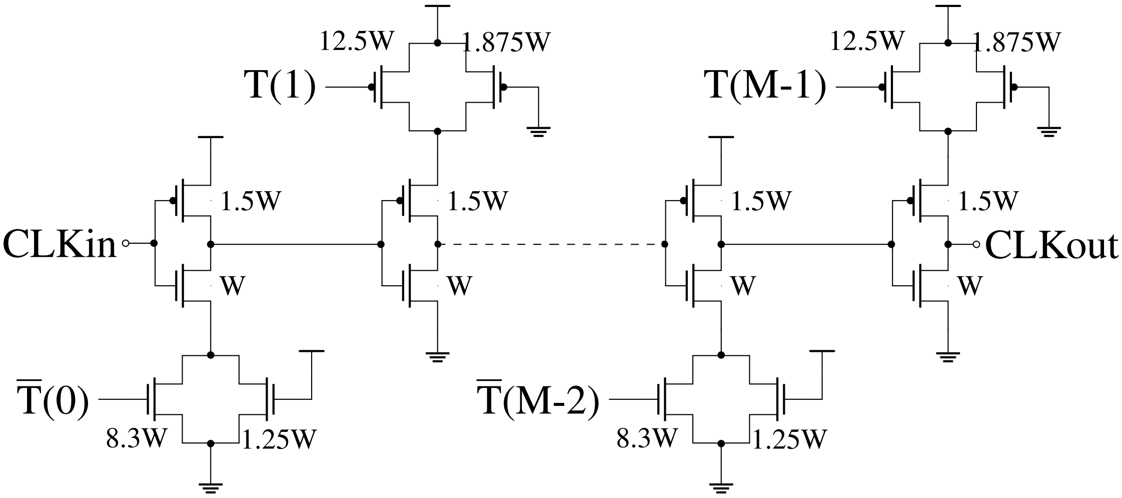


Figure 4.4: Delay line circuit.

switches to 1 at a specific time in every clock cycle as this happens exactly when the DEL signals switch to 0. Hence, the risk of a DEL signal switching to 1 while PER is still low is eliminated.

4.3.2 Decoder

The role of the decoder at the receiver side is to place the words back in the initial order. To achieve that, the decoder uses *N* registers to store the words in the right order

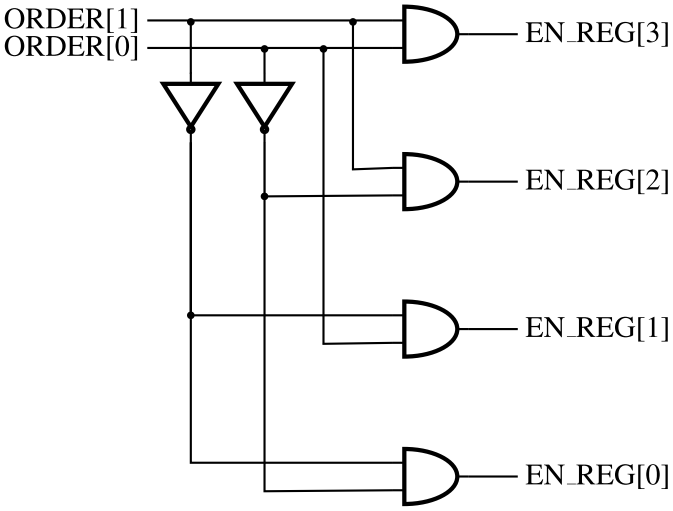


Figure 4.5: Decoder circuit.

as they arrive. The order of each word is transmitted by using spatial redundancy. K more bits are added to each word that indicate the order. Thus, for N words, $K = \log_2 N$ additional bus lines are required. The decoder reads the K bits and enables the corresponding register to store the word while the rest of the registers remain disabled. The circuit that implements this function is a basic K-to-N decoder with each output connected to the enable input of the appropriate register. A 2-to-4 decoder circuit is depicted in Fig. 4.5. The K bits are also considered in the encoding process, thus, they are included in the calculation of the Hamming distance. Consequently, it is ensured that they do not considerably increase the power overhead.

4.4 Simulation Setup

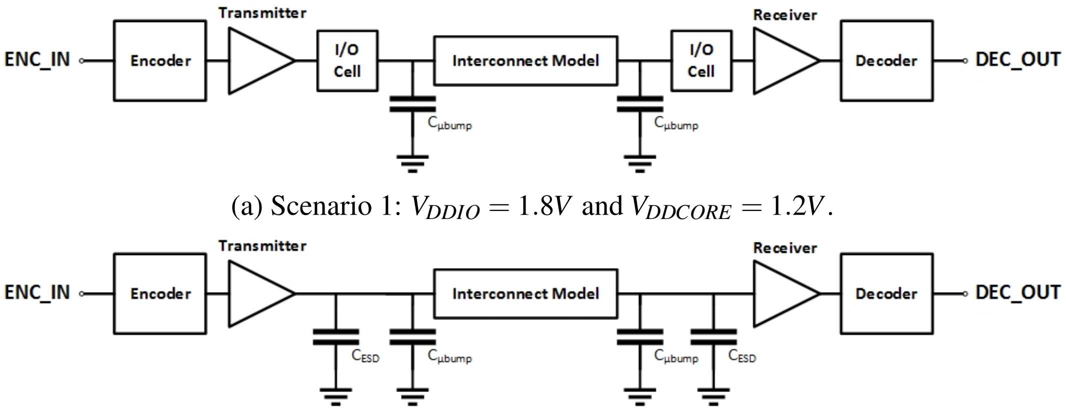
The simulation setup used to evaluate the efficiency of the proposed AWR scheme is described in this section. The effectiveness of the proposed AWR method is explored for an inter-die link for 2.5-D integration as interposers support a high wire density. However, note that the method is equally applicable to other state-of-the-art or emerging packaging approaches, such as EMIB [96]–[97] and more broadly to applications that require wide and slow links. The link is assumed to connect two dies that are bump bonded on top of a silicon-based interposer.

To fully explore the savings as well as the limitations of the technique, two different scenarios are investigated. In the first scenario, separate voltage supplies $V_{DDCORE} = 1.2V$ and $V_{DDIO} = 1.8V$ are, respectively, considered for the core logic and the I/O.

Alternatively, in the second scenario, the I/O voltage is considered equal to the core voltage ($V_{DDIO} = V_{DDCORE} = 1.2V$). The two circuits used for simulations that contain the electrical model of the interconnects are illustrated in Fig. 4.6.

Both circuits consist of the encoder and decoder logic described in Section 4.3, the distributed wire model, the parasitic capacitance of μbumps, $C_{\mu bump} = 30$ fF [30], and the transmitter and receiver circuits. The latter comprise buffers used to, respectively, model the driving strength and load of the transmitter and receiver circuits. For the distributed wire model, the global wire dimensions for a 65 nm technology are used according to [37]. The electrical characteristics of the wires for minimum pitch are listed in Table 4.5. For the sake of simplicity, the mutual inductances are not considered. In the circuit of the first scenario (Fig. 4.6a), I/O cells are included following the transmitter and preceding the receiver to convert voltage from $1.2V$ to $1.8V$ and vice versa. In the second simulation circuit (Fig. 4.6b), the voltage remains stable, therefore, the I/O cells are replaced by an ultra low ESD capacitance ($C_{ESD} = 115$ fF [30]), that models the ESD protection circuit of the I/O cells.

The interconnect length ranges from 1 mm to 5 mm. The length of interposer-based interconnects can range from ∼100 μm [98] up to 20 mm [38]. For example, the length of EMIB interconnects ranges from ∼1 mm to 20 mm [97]. Hence, the effectiveness of the proposed technique is demonstrated for relatively short interconnects (1 mm). The capacitive load of the interconnect increases with wire length, therefore, the power saved in the interconnect by encoding increases, while the power overhead of encoding



(a) Scenario 1: $V_{DDIO} = 1.8V$ and $V_{DDCORE} = 1.2V$.

(b) Scenario 2: $V_{DDIO} = V_{DDCORE} = 1.2V$.

Figure 4.6: Electrical model for an interposer-based interconnect.

and decoding remains fixed. Thus, encoding becomes more beneficial. This behaviour is demonstrated in subsection 4.5.2.

4.5 Results

The efficiency of the proposed AWR scheme is estimated in terms of power savings. However, note that the energy savings are equal to the power savings. The reason is that AWR does not affect the communication throughput as one word is transmitted per clock cycle both when data are transmitted unencoded and when data are encoded with AWR. This attribute is further discussed in subsection 4.5.1, where the timing and power overheads of AWR and the state-of-the-art encoding techniques are explained. Furthermore, the reduction in power of AWR is quantified in subsection 4.5.2 and is compared with the state-of-the-art techniques in subsection 4.5.3. Finally, the robustness of AWR under process variations is investigated in subsection 4.5.4.

4.5.1 Circuit Implementation and Overheads

The encoder and decoder circuits of AWR as well as the other three techniques are implemented with a 65 nm technology [99]. The digital part of the encoder circuit and the decoder circuit are described in Verilog and synthesised using Design Compiler, Synopsys to produce the gate level netlist. The schematic of the custom part of the encoder is designed using Virtuoso, Cadence®. The low leakage and regular threshold voltage library is used to reduce the power overhead of the encoder and decoder circuits. In addition, minimum gate length devices are used for the custom circuit, while the widths of the devices are depicted in Fig. 4.4, where W is the minimum width determined by the 65 nm library. For these values the power is minimised while timing constraints are met, *i.e.* 2.5 ns period. Furthermore, the used library does not support vertically stacked channels, therefore, 1-channel devices are used. Simulations are conducted at schematic level both at nominal conditions (typical device corners, 27°C) and under process variations. The latter results are presented in subsection 4.5.4.

Table 4.5: Electrical characteristics of wires.

R [Ω/mm]	L [nH/mm]	C_{GND} [fF/mm]	C_C [fF/mm]
40.74	1.52	222.55	52.45

Table 4.6: Power consumption of encoder and decoder circuits in mW.

(a) Proposed AWR and BI.

Data Streams	Type	AWR_{32}		AWR_{64}		BI [46]	
		Enc.	Dec.	Enc.	Dec.	Enc.	Dec.
LFRic	Address-data	3.76	0.01	6.25	0.02	0.79	0.39
InfOli	Address-data	3.29	0.01	5.62	0.02	0.67	0.35
LITE_LOOP	Address-data	3.29	0.01	5.62	0.02	0.67	0.35
LU_SOLVER	Address-data	3.29	0.01	5.62	0.02	0.67	0.35
NASTY_EXPS	Address-data	3.29	0.01	5.62	0.02	0.67	0.35
Image	Data	3.37	0.01	5.87	0.02	0.88	0.46
Random	Data	5.78	0.01	10.86	0.02	1.09	0.51

(b) APBI and ABE.

Data Streams	APBI_1 [60]		APBI_{16} [60]		ABE_1 [72]		ABE_4 [72]	
	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.
LFRic	3.26	2.61	1.45	0.88	49.27	2.29	14.30	1.95
InfOli	3.01	2.43	1.34	0.81	48.17	2.29	14.04	1.94
LITE_LOOP	3.01	2.44	1.34	0.81	48.17	2.29	14.04	1.94
LU_SOLVER	3.01	2.44	1.34	0.81	48.17	2.29	14.04	1.94
NASTY_EXPS	3.01	2.44	1.34	0.81	48.17	2.29	14.04	1.94
Image	3.67	2.93	1.68	1.01	57.18	2.32	16.56	2.01
Random	3.97	3.09	1.88	1.08	60.40	2.47	17.69	2.10

The gate level netlist is imported to Virtuoso and the complete encoder and decoder circuits are simulated at transistor level using Spectre, Cadence®. In this way, the total power dissipated in the circuits is calculated with sufficient accuracy. Simulations at layout level with clock tree synthesis would provide more accuracy. However, the simulation time and the memory required for a full layout simulation of the entire circuit are prohibitive. The methodology used to combine the digital and custom part of the circuit to simulate the entire mixed signal design at schematic level is described in Appendix A. The total power consumption, *i.e.* static and dynamic, of the encoding schemes is listed in Table 4.6, while delay is reported in Table 4.7. The bus is assumed to be 64 bits wide and the operating frequency is 400 MHz.

Table 4.7: Timing overheads of encoding techniques. The critical path delay is listed in ns and the latency in clock cycles. The bus width is $M = 64$ bits unless stated otherwise and N is the window size.

(a) Proposed AWR and BI.

	AWR_{32}, M=32		AWR_{32}		AWR_{64}		BI [46]	
	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.
Critical path delay	1.60	0.25	2.21	0.25	2.40	0.33	1.63	0.32
Latency in cycles	1 – N+1		1 – N+1		1 – N+1		1	
Throughput decrease	0		0		0		0	

(b) APBI and ABE.

	APBI_1 [60]		APBI_{16} [60]		ABE_1 [72]		ABE_4 [72]	
	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.	Enc.	Dec.
Critical path delay	1.86	1.92	1.65	1.88	2.16	1.07	2.11	1.20
Latency in cycles	1		1		N+1		N+1	
Throughput decrease	0		0		1/(N+1)		1/(N+1)	

Note that to apply AWR to LPDDR3, the frequency has to be adjusted to 800 MHz. The frequency of AWR can be increased by implementing AWR in a more advanced technology as well as by decreasing the bus width. By narrowing the bus, the number of inverters in the delay line (Fig. 4.4) is considerably reduced since one inverter is required for each bit. The reduction of the critical path delay by reducing the bus width and, hence, the number of inverters is evident in the results of Table 4.7.

The power of the encoding and decoding circuits has a significant effect on the total power savings. A high overhead in power can diminish the efficiency of encoding and even increase power consumption [49]. BI exhibits the lowest overhead in power since the least amount of computations is required with this technique. Alternatively, ABE exhibits the highest power consumption, especially when applied to the entire bus (ABE_1) as the complexity of calculations increases exponentially with the number of bus lines. AWR and APBI have moderate power overheads.

To highlight the benefits of the time-based circuit of AWR, a digital implementation of the encoder of AWR with adder trees is described in Verilog and synthesized using Design Compiler, Synopsys. The power consumed by the encoder of this digital implementation for $N = 32$ and $M = 64$ is 20.13 mW on average. Thus, the conventional implementation consumes $\sim 5\times$ more power than the time-based encoder.

In terms of timing, AWR and ABE exhibit higher delays. However, the delay of AWR decreases significantly for lower bus widths since the delay lines are shorter. For example, for $M = 32$ bus lines and $N = 32$ reordered words the critical path delay is 1.6 ns. Furthermore, the latency of AWR ranges from 1 to N+1 cycles in total, where N is the number of words. In the best case scenario where the words are transmitted in the initial order, the decoding latency is just 1 clock cycle, which is the time required to store the word in the respective buffer. In the worst case scenario, where the first word is transmitted last, the received words can only be utilised by the receiver after $N+1$ cycles after the first word is received and decoded, assuming that the order at the receiver matters. Thus, in case the number of reordered words is $N = 32$, latency can be up to 33 cycles.

Usually in memory interfaces, a mechanism called Critical Word First is employed [122]. The critical word requested from the CPU is transmitted first to reduce the penalty of a cache miss. In case the critical word is transmitted last by AWR, the CPU has to wait until the end of the transmission to continue execution, which degrades both performance and power and, therefore, counteracts the power saved on the bus by encoding. To alleviate the large latency penalty caused by AWR in these bus interfaces, the scheme can be amended to always transmit the critical word first and reorder only the remaining words. The effect of this modification on the power efficiency of AWR will be rather negligible as the remaining $N-1$ words will be reordered.

ABE always introduces a latency of N+1 cycles as the data are decoded after the information about the cluster and the basis line is received. The basis line information is only available after N cycles. The decoding requires 1 clock cycle, consequently, the total latency is N+1 cycles. BI and APBI increase latency by only one cycle. Finally, AWR, BI and APBI do not affect the communication throughput since 1 word is transmitted per clock cycle which is the same rate with the unencoded transmission. However, ABE requires N+1 cycles to transmit N words due to the temporal redundancy required to transmit the cluster information (see Table 4.1). Therefore, ABE decreases throughput by 1 word per N+1 cycles.

Overall, increasing the window size, N, of AWR has a negative impact on performance since the maximum latency increases with N as reported in Table 4.7a. In addition, the power overhead of encoding and decoding increases with the window size as shown in Table 4.6a. However, a larger window size leads to a higher decrease in bit transitions and, thus, in the power dissipated in the interconnect as demonstrated in Table 4.4a. Therefore, the total power is minimised for a moderate window size ($N = 32$ for our simulation setup) as discussed in subsection 4.5.2. AWR does not affect the resilience of the off-die communication to process variations as demonstrated in subsection 4.5.4. Hence, the robustness of the link is not affected by the window size.

Often the energy-delay product is utilised to evaluate the efficiency of circuits. However, optimising the energy delay product of the encoder of AWR is not straightforward. It is possible to decrease the delay but this decrease adversely affects the accuracy of the Hamming distance calculations as well as the power overhead. Techniques that reduce the delay in inverter chains such as tapering can not be applied in the delay lines since the delays of all the inverters should nominally be equal. Sizing up the devices uniformly in the delay lines can speed up the encoder, however, the accuracy of the delay lines as well as the power consumption are still affected. Particularly, increasing the width of the devices in the pull up and pull down that are always on (gate is connected to ground or VDD, respectively) results in a significant decrease of the total delay. However, this increase causes the difference in the delay of an inverter when a transition does and does not occur to shrink. This can lead to many DEL signals arriving at about the same time in the winner stage (see Fig. 4.2). As a result, the optimal word that exhibits the lowest Hamming distance is not always selected, thus, reducing the accuracy of the technique. On the contrary, sizing up the devices connected to $T(i)$ (see Fig. 4.4) mitigates this problem at the cost of higher power consumption. Finally, increasing the size of the inverters up to a certain value speeds up the whole inverter chain, nonetheless, power and accuracy are negatively impacted. By performing a parametric analysis with the widths of the devices, the specific combination of sizes illustrated in Fig. 4.4 is found to provide a balance between delay and power while accuracy is not compromised for the typical device corner.

4.5.2 Power Savings of AWR

The efficiency of AWR in saving power is evaluated for different interconnect lengths, bus widths (M), and number of reordered words (N). For this exploration, the LFRic benchmark is used, for which AWR$_{32}$ yields about 37% and 11% decrease in self and relative switching, respectively. A sequence of 1,600 words are transmitted over the interconnect, both encoded and unencoded. The total power is measured, which includes static and dynamic power of the encoder and decoder circuits and interconnect.

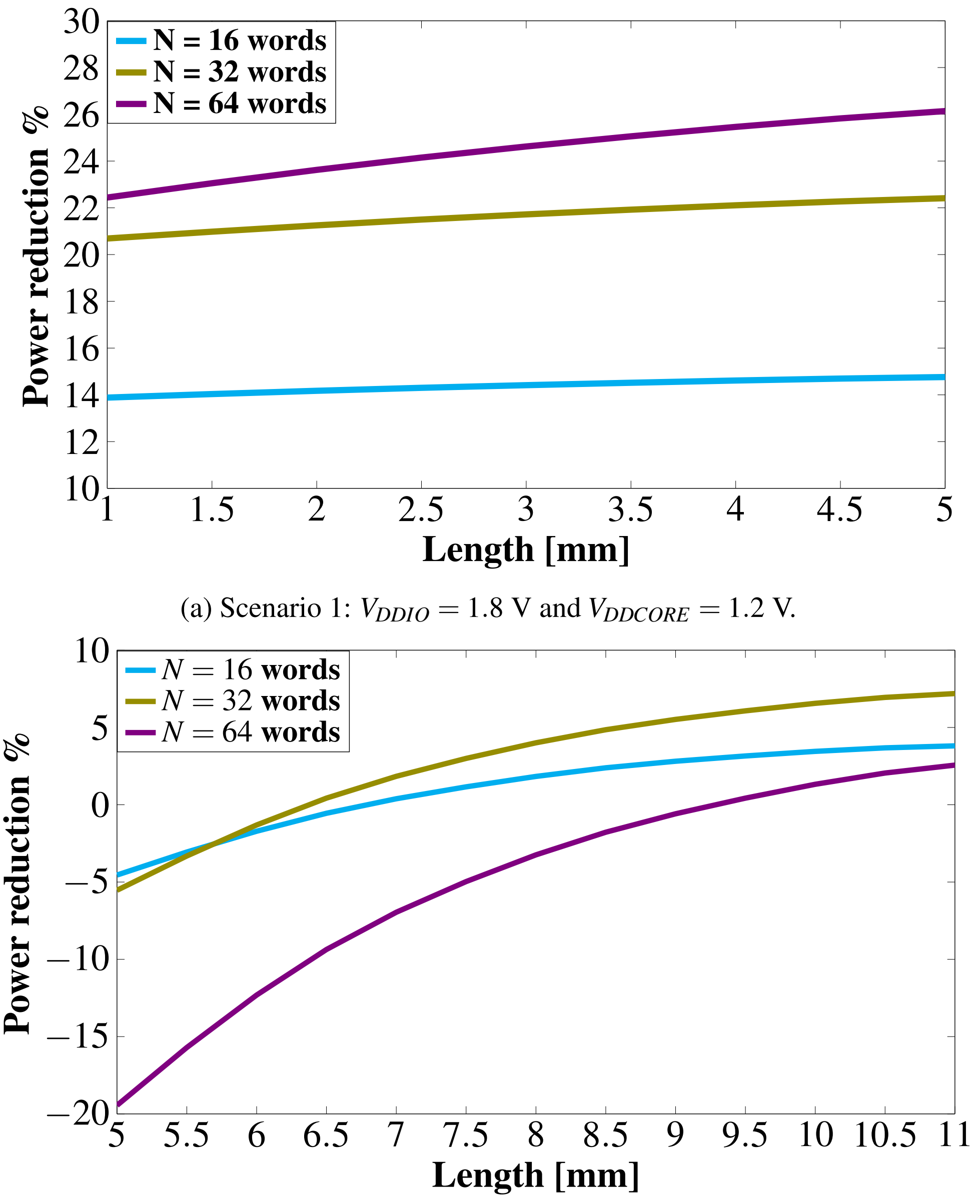
In Fig. 4.7, the percentage of power reduction is illustrated for a fixed bus width,

$M = 64$ bits, and different number of reordered words and interconnect length. In Fig. 4.7a, the results for the first scenario (Fig. 4.6a) are displayed, where the typical I/O voltage is considered for the 65 nm technology, $V_{DDIO} = 1.8$ V. In this case, the power reduction is higher for larger N even though the circuit power overhead and the required spatial redundancy are higher. This result demonstrates that the power overhead of encoding and decoding is low for AWR compared to the high switching savings, yielding, in total, up to 23% power savings at just 1 mm wire length. Furthermore, the power savings increase for longer interconnects, which is expected since the capacitive load of wires increases, while the circuit overhead in power remains unaltered.

In Fig. 4.7b, the limitations of the proposed technique are shown using the second setup (Fig. 4.6b), where the I/O voltage is equal to the core voltage ($V_{DDIO} = V_{DDCORE} = 1.2$ V). As expected, for short interconnects < 6 mm, AWR is not effective as the power consumed in the interconnect decreases, while the power overhead is constant. In this case, $N = 32$ words is more efficient because of the lower power overhead compared to $N = 64$.

AWR is also effective for different bus widths. This efficiency is demonstrated in Fig. 4.8, where the savings in power with respect to the bus width are illustrated. The number of reordered words is equal to 32 while the I/O voltage is 1.8 V according to the first scenario. The power savings range from 18% to 21% at 1 mm and from 20% to 23% at 5 mm. Thus, the difference in savings is maximum 3% for different bus widths. Note that for $M = 128$ bits the frequency is reduced to 200 MHz due to the longer delay lines in the encoder circuit. However, the communication throughput is not affected considering that compared to the case of $M = 64$ bits, the bus width is doubled.

Note that for more advanced technologies, AWR yields higher power savings as encoding techniques become more beneficial. The power consumed in logic is lower due to the smaller feature size of the transistors in these technologies. The power consumed in interconnects scales disproportionately compared to the power consumed in computations [44]. Therefore, the power overhead of encoding schemes decreases faster whilst the power consumed by interconnects decreases at a much slower pace as technology scales. Hence, the total power savings are expected to be even higher for advanced technologies.



(a) Scenario 1: $V_{DDIO} = 1.8$ V and $V_{DDCORE} = 1.2$ V.

(b) Scenario 2: $V_{DDIO} = V_{DDCORE} = 1.2$ V.

Figure 4.7: Decrease in power for $M = 64$ bits and different number of reordered words, N.

4.5.3 Comparison with State-of-the-art

The proposed technique is compared with the discussed state-of-the-art schemes in terms of total power savings in Fig. 4.9 for the first scenario of Fig. 4.6a and all the benchmark data of Table 4.2. In these results the power overhead, including both static and dynamic power, of each technique is considered. The bus width is $M = 64$ bits

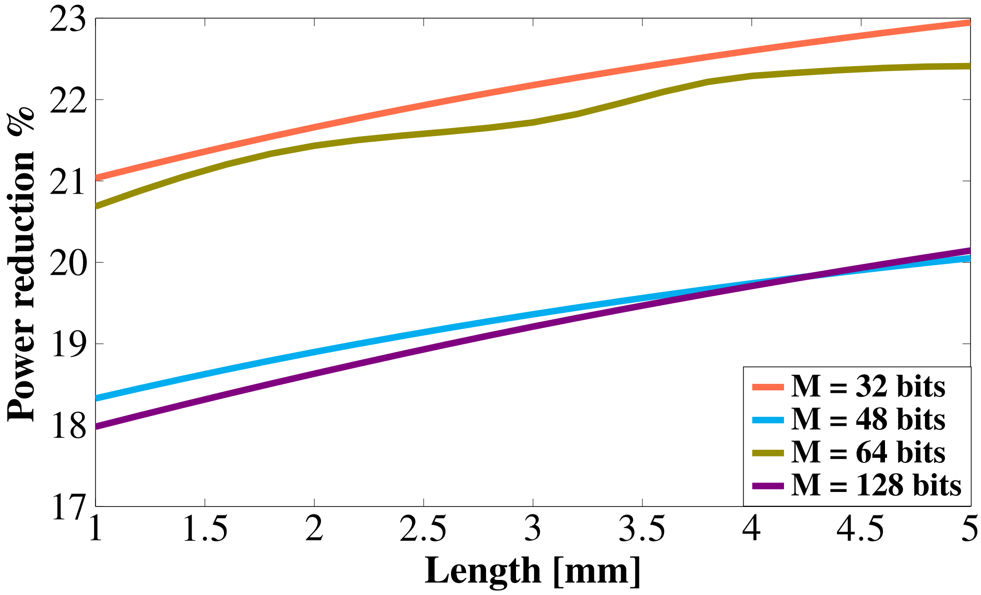


Figure 4.8: Decrease in power for fixed number of reordered words $N = 32$ and different bus widths M. $V_{DDIO} = 1.8$ V, and $V_{DDCORE} = 1.2$ V.

and the wire length is 1 mm. As expected the savings in power are lower than the theoretical savings of Table 4.3 for all the encoding schemes due to the added power overhead of the encoding and decoding circuits.

AWR outperforms the other techniques for all the real data streams as the high reduction in bit transitions greatly counteracts the low overhead in circuit power even at just 1 mm interconnect length. The low overhead is enabled by properly mapping the Hamming distances as delays in the time domain. The power savings of AWR for the multiplexed address-data benchmarks are ∼23% while for the image file, AWR improves power by ∼21%.

In contrast, the other techniques are less effective. BI yields low power savings compared to AWR ranging from 4.5% to 9% for the address-data benchmarks and ∼5% for the image data. For the address-data benchmarks the savings of APBI$_{16}$ range from -3% to 4%, while for the image file APBI$_{16}$ yields just ∼2% decrease in power. The efficiency of BI and APBI is limited as the encoding is less effective in decreasing bit transitions compared to AWR and ABE. In other words, these methods emphasize mostly circuit simplicity, which is only one aspect of data encoding for power reduction.

Although ABE effectively reduces bit transitions, the high power overhead undermines the power savings. For the image benchmark, ABE$_4$ increases the power

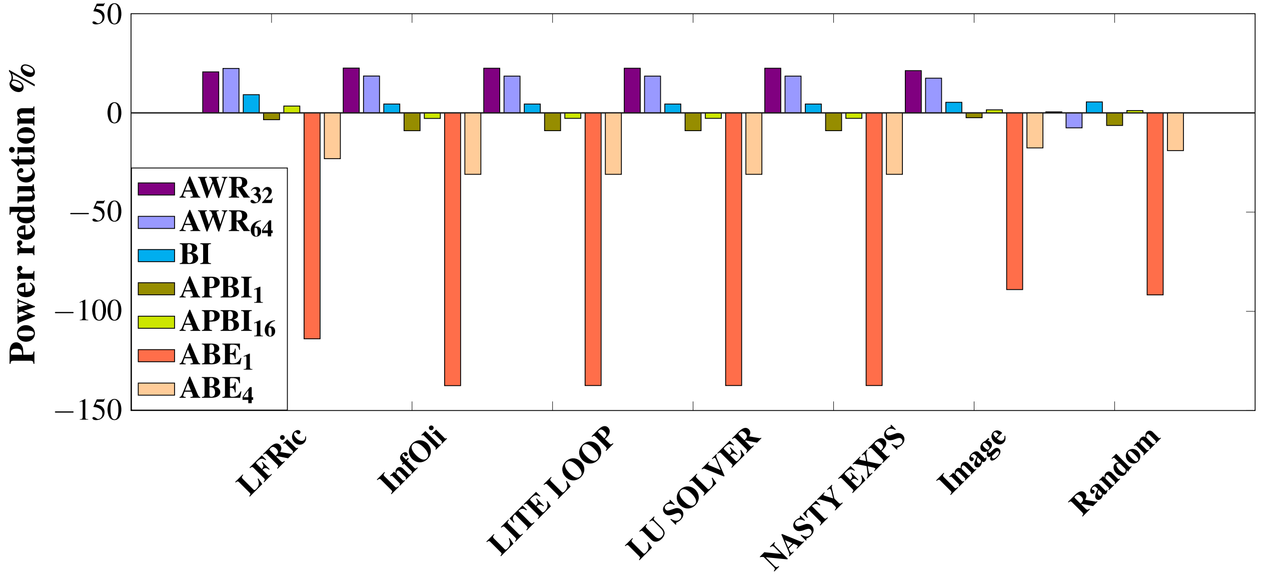


Figure 4.9: Comparison of AWR with existing encoding techniques in terms of total power savings for a 1 mm long, 64-bit interconnect, $V_{DDIO} = 1.8$ V, and $V_{DDCORE} = 1.2$ V.

consumption by at least 17% due to the encoding and decoding overhead. When ABE is applied to the entire bus (ABE_1), the power penalty is even greater due to the increased power of the encoder. Consequently, compared to BI and APBI, ABE places care mostly on the strength of the encoding, overlooking, however, the implications of this practice on the power of the encoding and decoding circuits. Alternatively, AWR carefully balances these two rather conflicting objectives, enabling higher savings in the total power from all of these techniques.

The benefits of data encoding diminish for random data. BI provides just 5% decrease in power while AWR_{32} and APBI_{16} only "break even". Thus, the power overhead of the encoder and the decoder is equal to the power saved in the interconnect. The rest of the techniques increase power. However, note that for longer interconnects the power reduction is higher as the power saved on the bus increases while the power overheads are the same.

From the results of Fig. 4.9, the conclusion that AWR provides higher savings for applications with more correlated data can be drawn. The efficiency of AWR is high for multiplexed address-data and image files where the data words have more similarities. In contrast, the bit transitions can not be significantly reduced by reordering the data words if the Hamming distance between all the most recent data words is not sufficiently high.

In [89], the frequency of occurrence of 90% data similarity among the 64 most recent data words is measured for SPEC 2006 workloads [100]. Specifically, how often 58 bits or more out of a 64-bit word match exactly with the bits of one of the 64 most recent words is calculated. The frequency of 90% data similarity ranges between $\sim$22% and $\sim$100%. AWR would provide higher savings for applications with high frequency of occurrence of 90% similarity such as libquantum, a library for the simulation of quantum computers, omnetpp, a large ethernet simulator, soplex, a Simplex Linear Program (LP) solver, and zeusmp, a computational fluid dynamics code. Alternatively, the savings of AWR would be lower for bzip, a compression code, gromacs, a package for molecular dynamics, and milc, a simulator of four dimensional SU lattice gauge theory on MIMD parallel machine.

Finally, a detailed analysis of the power consumption is presented in Table 4.8 using the LFRic and the Image benchmarks. The overheads of encoding and decoding as well as the power consumed in the interconnect are reported for all the techniques along with the power consumed in transmitting data unencoded. The lowest power

Table 4.8: Detailed analysis of power consumption using the LFRic and the Image benchmarks.

Encoding	Overhead [mW]				Interconnect power [mW]			
schemes	LFRic		Image		LFRic		Image	
	Enc.	Dec.	Enc.	Dec.	Enc.	Not enc.	Enc.	Not enc.
AWR_{32}	3.76	0.01	3.13	0.01	27.41	39.41	37.20	51.26
AWR_{64}	6.25	0.02	5.45	0.02	24.29	39.41	36.82	51.26
BI [46]	0.79	0.39	0.96	0.45	34.61	39.41	47.11	51.26
APBI_1 [60]	3.26	3.26	3.37	2.69	34.87	39.41	46.44	51.26
APBI_{16} [60]	1.45	0.88	1.54	0.93	35.70	39.41	47.99	51.26
ABE_1 [72]	49.27	2.29	52.63	2.13	32.70	39.41	42.14	51.26
ABE_4 [72]	14.30	1.95	15.24	1.85	32.25	39.41	43.21	51.26

consumption in the interconnect due to encoding is provided by AWR for both benchmarks and the circuit overhead of AWR is low (3.5 mW for $N = 32$ and 5.8 mW for $N = 64$ on average), hence, the overall power improvement is significant. BI and APBI are less effective in decreasing the interconnect power, however, the total power savings of BI are considerable because of the significantly low circuit overhead (1.3 mW on average) of this technique. Alternatively, ABE achieves a substantial decrease in the interconnect power for both benchmarks, however, the circuit overhead is high. Especially, in the case that ABE is applied to the entire bus (ABE_1), the overhead is higher than the power consumed in the interconnect for both address-data and data benchmarks, which increases the total power consumption. This behaviour is due to the fact that the circuit complexity of ABE increases quadratically with the number of bus lines.

4.5.4 Resilience of AWR to process variations

The robustness of the proposed encoder and decoder circuits under process variations and the effect of these variations on saving power are investigated in this subsection. The results of corner analysis are reported in Table 4.9, where $N = 32$ words and $M = 64$ bits. The savings in power are reported both with and without including the overhead in power of the encoding and decoding circuits. AWR saves power for all the explored process corners, however, the efficiency of encoding is lower in case of

Table 4.9: Process corner analysis results using the LFRic benchmark.

Corner	Overhead [mW]		Power savings	
	Encoder	Decoder	without circuit power	with circuit power
TT	3.76	0.01	30.26%	20.68%
FF	4.09	0.01	28.15%	17.70%
SS	5.54	0.01	26.35%	12.25%
SF	4.07	0.01	26.13%	15.77%
FS	4.11	0.01	26.13%	15.67%

process variations since the savings on the bus are lower and the circuit power overhead is higher compared to the typical/typical (TT) corner.

The low drop of the savings on the interconnect (without including the circuit power overhead) is due to the delay lines (see Fig. 4.4). Particularly, for the SF and FS corners, the delay of the odd inverters (connected to ground through a pair of devices connected in parallel) deviates from the delay of the even inverters (connected to the supply voltage through a pair of devices connected in parallel). Therefore, the difference in delay resulting from the optimal word (i.e. the word with the lowest Hamming distance) and near-optimal words shrinks. Consequently, often a near-optimal word is selected which has a similar delay and, therefore, Hamming distance. This behaviour leads to a low increase in the switching activity of the transmitted data and, thus, a low drop in the power savings in the interconnect ($\sim$4%). However, the robustness of the link does not degrade as the encoder always selects a word to transmit even if two or more words have identical delays. Hence, one word is transmitted per clock cycle and, consequently, the throughput is not affected.

For the FF corner, the delay of the inverters in the delay line is shorter compared to TT and, hence, the delay difference when a bit transition does and does not occur can not be sufficiently high to ensure that the “best” word (lowest Hamming distance) is always selected. Consequently, a mere $\sim$2% increase in switching activity compared to TT is observed. Finally, to meet the timing constraints for the SS corner, the devices in the delay line are sized up leading to shorter delays and higher power overhead. The shorter delays affect the accuracy of the calculation of the Hamming distances as in the case of the FF corner, leading to slightly higher switching activity.

Although AWR is less efficient in case of process variations, the savings in power ($>$12%) remain significant and the reliability of the communication is not affected. A way to mitigate the impact of process variations and maintain the savings on the interconnect is by regulating the delays of the delay lines such that the optimal word, i.e. the word with the lowest Hamming distance, is always the fastest. A way of regulating the delays is the sizing up of the devices of the delay lines. However, the deterioration of the efficiency of AWR in reducing the power consumed in the interconnect is just $\sim 4\%$ in the FS and SF corners, where the difference in the delays of consecutive inverters is maximum. Consequently, the additional power overhead due to sizing up can surpass this low drop of 4%.

An alternative solution would be a self-calibration scheme to adjust the delay of each inverter and minimise the deviations. However, a self-calibration scheme would require additional circuits and routing resources. Considering that each stage of the delay line would require at least two devices to compensate for mismatch between the pull up and down network, the increase in the power of the delay line would be greater than the mere loss of 4% due to non-optimal word transmission. If the power of the additional circuitry required for the control of these calibrating devices is also incorporated in the power of the encoder, the savings in power will rather vanish. Hence, a more prudent approach is to design for the nominal case instead of the worst case, trading off a small reduction in the efficiency of the method with a significant increase in the circuit power overhead due to the sizing up of the devices or the addition of a self-calibrating circuit.

4.6 Summary

In this chapter, the reduction of the power consumption of interposer-based interconnects by employing encoding techniques is investigated. These techniques target the decrease of the switching activity of the transmitted data stream and, thus, the dynamic power consumption. An adaptive encoding scheme (AWR) is proposed for wide, parallel inter-die interconnects. AWR decreases the interconnect power by changing the order of the transmitted words to effectively reduce the switching activity. The Nearest Neighbour algorithm is utilised to determine the order of the words. AWR is adaptive, thus, *a priori* knowledge of the statistical characteristics of data is not required.

The power consumption of the circuitry implementing the encoding and decoding plays a significant role in the total power savings. Often this power overhead is high and counteracts the power saved in the interconnect. Notably, the circuit power overhead of AWR is restrained by proposing a novel encoder circuit, which exploits the time domain for the computation of Hamming distances by replacing power-hungry adder trees with delay lines. Hence, the calculation of the next word to be transmitted is converted to a race. The data word with the lowest Hamming distance propagates faster through the delay lines, therefore, arrives first to the finish stage and subsequently is transmitted over the interconnect.

The effectiveness of AWR is investigated for an interposer-based interconnect and both the savings in power as well as the limitations in terms of timing are presented. AWR outperforms state-of-the-art techniques in terms of decrease in switching activity and overall power savings for real data streams for several applications relating to high performance computing where energy reduction is a primary objective. AWR yields up to 23% savings in power for multiplexed address-data benchmarks, respectively, at just 1 mm interconnect length. Finally, the robustness of AWR is investigated under process variations. Although AWR is less efficient compared to the typical corner, the provided savings in power remain substantial where process variations are considered.

AWR is specifically designed for parallel interfaces and, thus, is not applicable to serial communication. Therefore, in the next chapter, an encoding technique is presented that is tailored to high-speed, serial interfaces.

Chapter 5

Adaptive Time-based Encoding for Serial Interfaces

In the previous chapter, AWR is proposed, an encoding technique that is suited to wide, parallel interconnects. However, techniques designed for parallel interfaces are not directly applicable to serial interfaces. High-speed SerDes interfaces are widely used offering important advantages over parallel buses especially for long interconnects. Highly optimised SerDes devices such as PCIe, USB, Ethernet, and SATA have gained popularity. Hence, in this chapter, the decrease in energy consumption of SerDes interfaces by reducing bit transitions is investigated. A novel encoding technique, namely Serial Tuned Transition Encoding (STTE), is proposed that is tailored to high-speed, serial interconnects and is inspired by AWR.

The proposed encoding is designed for SerDes interfaces with voltage-mode drivers, where there is a one-to-one mapping between the digital signal and the transmitted analog symbol. An example of these drivers is SSTL [116]. STTE is not applicable when digital modulation schemes are used, where each symbol represents two or more digital bits.

Encoding techniques for serial interfaces that target the minimisation of bit transitions have been proposed in [104–110]. However, these methods are only suitable for source synchronous communication and *not for high-speed SerDes interfaces that are source asynchronous* [73]. The reason is that there are two requirements to maintain link integrity and prevent data loss in source asynchronous interfaces that these techniques do not consider. The transmitted data stream is required to exhibit frequent bit transitions since the clock is recovered from the incoming data at the receiver by employing Clock Data Recovery (CDR) [111]. In addition, DC balance has to be preserved, i.e. the number of 0's and 1's has to be approximately equal [111]. These prerequisites are discussed in detail in Chapter 2.

Encoding techniques, such as 8B/10B encoding [113] and scrambling [111] are applied to SerDes devices. These techniques increase bit transitions and maintain DC balance to facilitate CDR and avoid data loss. However, these methods increase drastically the transitions of the transmitted data and, thus, the energy consumption.

In this chapter, the reduction in energy consumption of source asynchronous, serial interfaces is investigated, by decreasing bit transitions while maintaining link integrity. An encoding technique is proposed that is inspired by AWR (Chapter 4) and is tailored to source asynchronous interfaces. STTE regulates the number of bit transitions such that the clock can be recovered and DC balance is maintained, while the communication energy is lowered. The number of transitions can be tuned to provide either higher energy efficiency or a more reliable CDR. The energy savings are estimated for a short interposer-based interconnect (1 mm length), which exhibits a low capacitive load. For longer interconnects, the energy savings of the technique are greater as the interconnect load increases with length. The link integrity is experimentally evaluated using both an electrical and an optical link that interconnect two FPGA devices.

The rest of this chapter is organised as follows. The proposed encoding technique is presented in Section 5.1. The emerging trade-off between energy reduction and reliability is discussed in Section 5.2. The circuit implementation of the encoding and decoding algorithms is presented in Section 5.3. The energy-efficiency of STTE is presented in Section 5.4, while the ability of STTE to preserve link integrity and curtail data errors is experimentally evaluated in Section 5.5. Finally, a summary is provided in Section 5.6.

5.1 Encoding Algorithm

The proposed encoding scheme is based on the idea of reordering a finite number of words prior to transmission as explored in the AWR method described in Chapter 4. As AWR is developed for wide parallel buses, applying a technique that shares the same principles as AWR to serial communication is not straightforward. The objective of encoding in parallel buses is to reduce bit transitions between consecutive words, whilst the purpose of encoding in serial buses is the reduction of bit transitions within each word. In addition, source synchronous communication is assumed in AWR, while this is not the case for all serial communication, such as PCIe interface, where the clock is recovered from the transmitted data [111]. The proposed STTE scheme is tailored to the latter type of communication, consequently, the reduction in transitions has to be restricted to facilitate CDR.

The reordering of the words within each packet is performed using the Nearest Neighbour algorithm as in AWR (Chapter 4), since NN exhibits low hardware complexity. The NN algorithm is adapted to reduce the transitions within each word and, thus, to support serial data transmission. Furthermore, a partial inversion mechanism is added that regulates the frequency of transitions to facilitate CDR.

STTE is well suited for SerDes communication. In SerDes devices, data words are organised in packets and reordering is applied to the words within each packet. In this way, stalling cannot be caused by STTE, as all the data words of a packet must be fully available before transmission commences.

The pseudocode of the new algorithm is shown in Algorithm 3. Once a block of N words is available, first, a unique code of $K = \log_2 N$ bits is assigned to each word. Hence, assuming that each word is M bits long, the new word length is $M + K$. This unique code is used by the receiver to place data words back in the original order. Subsequently, the reordering commences.

The reordering mechanism works as follows. Each word in the block ($word[j]$) that has not yet been transmitted is XOR-ed with the previously transmitted word ($word\_pre$). The index j is used to iterate between all the words in the block. The number of transitions of $word\_xored[j]$, which is the result of the XOR, is calculated. The K bits are included in this calculation to ensure that these bits do not

Algorithm 3 Pseudocode of the STTE algorithm.

```
Initialise $word\_pre$
while Queue with words not empty do
    Update the block of $N$ words to be reordered
    Assign a unique code of $K$ bits to all $N$ words
    for $i \leftarrow 0$ to $N-1$ do
        for $j \leftarrow 0$ to $N-1$ do
            if $word[j]$ not transmitted then
                $word\_xored[j] = word[j] \oplus word\_pre$
                Calculate the number of transitions of $word\_xored[j]$
        Select the $word\_xored[j_{min}]$ with the minimum number of transitions
        Update $word\_pre = word[j_{min}]$
```

notably increase the transitions. The result of the XOR with the least number of transitions (*word_xored*$[j_{min}]$) is then selected for transmission. The index j_{min} denotes the *word_xored* with the least number of transitions. In this way, the number of transitions is reduced. To transfer the N words over the interconnect, N transmissions are required. Therefore, index i is used to count the number of transmissions in Algorithm 3. Once all the N words are transmitted, the block is updated and the process is repeated.

In SerDes interfaces, the transmitted data are required to exhibit frequent bit transitions, hence, a mechanism that prevents long run lengths is required. In Algorithm 3, if the *word_xored*$[j_{min}]$ is transmitted unaltered (hence, leading to maximum savings in power), the risk that the number of transitions is insufficient for CDR is high. For example, if a word is the complement of the *word_pre*, then the number of transitions of *word_xored* is equal to 0, thus, the run length is $M+K$ bits. In this case, the clock of the receiver can get out of phase, which results to errors in sampling the input data. To prevent these errors, an appropriate number of bit transitions are added by inverting parts of *word_xored*$[j_{min}]$. Specifically, a step size S is selected, where $1 \leq S \leq \frac{M+K+1}{2}$. The bits *word_xored*$[j_{min}][0..S-1]$ are transmitted unaltered, while the next S bits (*word_xored*$[j_{min}][S..2*S-1]$) are inverted and so forth. Hence, regions of unaltered and inverted bits are interleaved. In this way, in the previous example where the word to be transmitted is the complement of the *word_pre*, a transition is added every S bits. The pseudocode that describes this function is shown in Algorithm 4, where L is the number of segments, thus, is an integer equal to $L = \lceil \frac{M+K}{S} \rceil$.

5.2 Energy vs Reliability Trade-off

There are two prerequisites to decrease the energy consumption of serial links without increasing the bit error rate; the bit transitions after encoding have to be sufficient for CDR and, at the same time, fewer than the transitions generated by the typical encodings, such as 8B/10B and scrambling. The more bit transitions the data stream

Algorithm 4 Partial inversion of the transmitted word.

```
Select the $word\_xored[j_{min}]$ with the minimum number of transitions
for $i \leftarrow 0$ to $M+K-1$ do
    if $((i \geq S)$ and $(i < 2*S))$ or $((i \geq 3*S)$ and $(i < 4*S))$ or $((i \geq 5*S)$ and $(i < 6*S))$ ... or $((i \geq L*S)$ and $(i < (L+1)*S))$ then
        Invert $word\_xored[j_{min}][i]$
Transmit $word\_xored[j_{min}]$
```

exhibits, the fewer errors occur in sampling the input data and the energy consumption is higher. Hence, the selection of the step size, S, is crucial as this parameter determines how frequently transitions are added.

The effect of S on energy consumption and on CDR are demonstrated in Figs. 5.1 and 5.2, respectively. In Fig. 5.1, the decrease in bit transitions of the encoded and serialized data stream compared to the scrambled and serialized stream is illustrated as a function of the step size, S. In Fig. 5.2, the mean run length (*i.e.* number of consecutive bits without a transition) and the standard deviation are depicted with respect to S. For these simulations, the number of reordered words is $N = 16$, each word is $M = 64$ bits long, and $K = 4$ bits. The input traces used are generated using gem5 simulator [117] for the application InFoli of Table 4.2.

The reduction in bit transitions is higher for larger values of S as demonstrated in Fig. 5.1. This behaviour is expected since fewer transitions are injected. The energy consumption is directly proportional to the number of transitions, thus, a significant decrease in transitions is equivalent to a significant decrease in energy. The mean run length as well as the standard deviation, also, increase with S, as shown in Fig. 5.2. However, long sequences of bits without transitions adversely affect CDR and can induce errors in data sampling at the receiver. Hence, *energy efficiency increases for high values of S, while link integrity is enhanced for low values of S*. This trade-off is crucial for the application of the technique and, therefore, useful insight is offered in

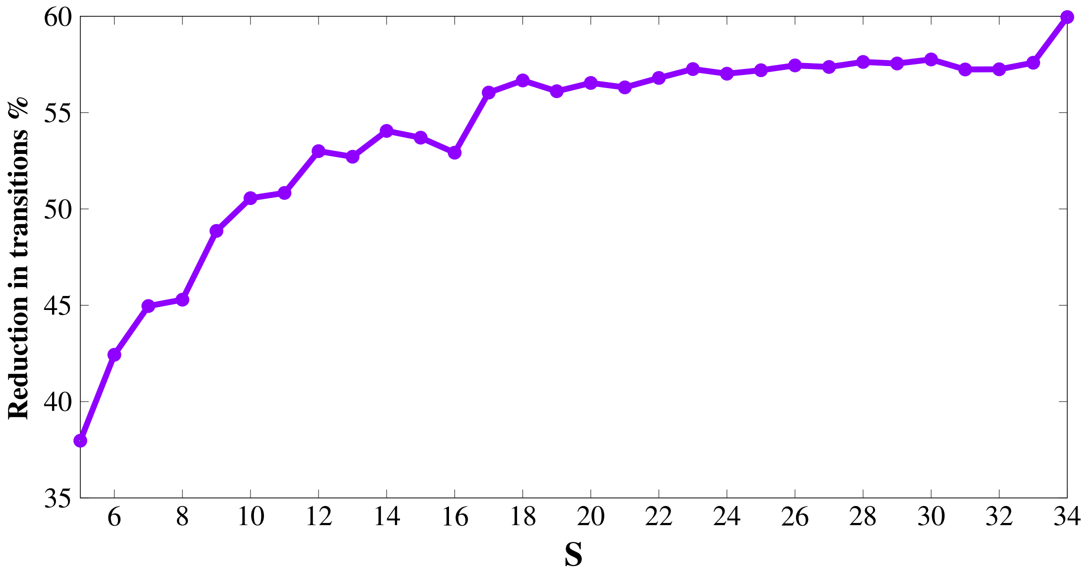


Figure 5.1: Reduction in the number of transitions compared to a scrambled data stream as a function of the step size S.

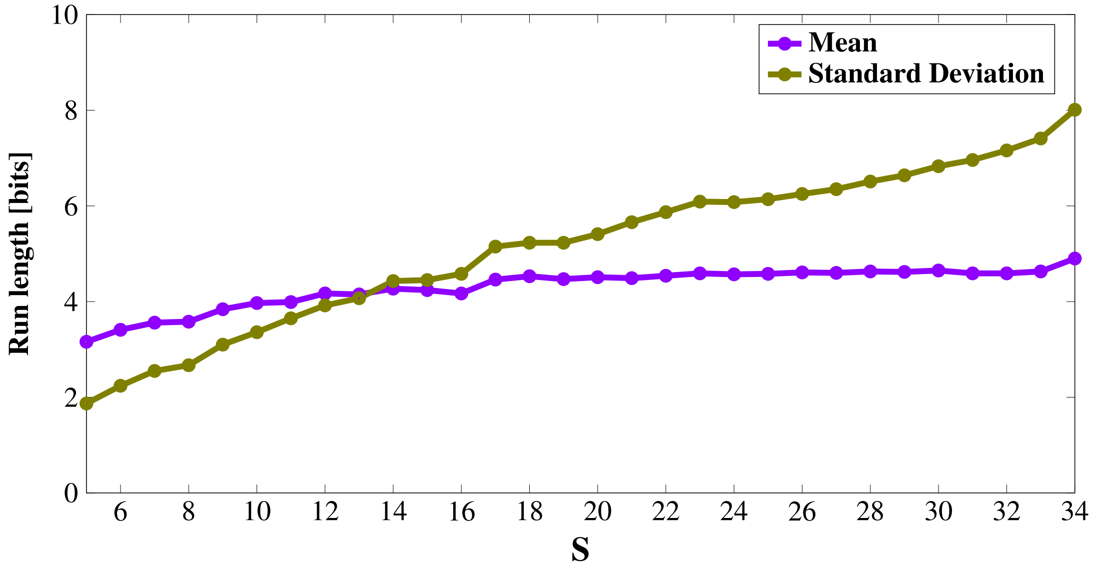


Figure 5.2: The mean run length (*i.e.* number of consecutive bits without a transition) and the standard deviation in relation to the step size S.

the following sections.

The scrambler randomises the data stream and, hence, induces a higher number of transitions compared to the proposed encoder. The mean run length of the scrambled stream is equal to 1.84 bits and the standard deviation is 1.13 bits. However, this high amount of transitions is not necessary for CDR, as shown in the following sections (Section 5.5.2). STTE exploits this fact to reduce the energy demand of serial links.

In contrast, without the application of any encoding, the mean run length is 4.48 bits and the standard deviation is 27.86 bits. Hence, the original traces contain long sequences of 0's or 1's and, therefore, link integrity is attenuated as verified in subsection 5.5.2. This behaviour highlights the importance of encoding the traces. STTE provides a balance in the number of transitions so that energy is decreased while link integrity is not compromised.

5.3 Circuit Architecture

The circuit architecture of the proposed encoding technique, STTE, is described in this section. The implementation of the encoder and decoder circuits are presented in subsections 5.3.1 and 5.3.2, respectively.

5.3.1 STTE Encoder

The encoding algorithm is implemented as follows. In each clock cycle, the words that have not been transmitted and the previous word are XOR-ed. The result of the XOR with the least number of transitions is selected for transmission. Segments of this result are inverted before transmission according to the partial inversion mechanism (Algorithm 4). The circuit diagram of the encoder is illustrated in Fig. 5.3.

Instead of calculating the number of transitions for each word and compare them to identify the minimum, the time-domain is exploited and the number of transitions are represented as delays, as in AWR (Chapter 4). This approach is more energy-efficient since the overhead in power of the encoder is restricted.

The encoder circuit of STTE consists of three stages, the race stage, the finish stage, and the winner stage. The race stage comprises the XOR block and the delay lines. Within the XOR block, each word is XOR-ed with the word that was previously transmitted (WORD_PRE) to generate the signals WORD_XORED, as shown in Fig. 5.4. The objective is to identify the WORD_XORED with the minimum number of transitions. Thus, consecutive bits within each WORD_XORED[i] are pairwise XOR-ed. The last bit (WORD_XORED[i][$M+K-1$], where $0 \leq i \leq N-1$) is XOR-ed with the last bit transmitted over the bus (ENC_OUT[0]). The output of this block is denoted as DIFF[i][$M+K-1:0$]. The number of 1's in DIFF corresponds to the number of transitions within WORD_XORED.

The DIFF signals are fed to the delay lines, where a clock pulse is propagated and delayed according to the number of 1's in DIFF[i][$M+K-1:0$]. Therefore, the clock pulse that corresponds to the WORD_XORED with the lowest number of transitions propagates faster and arrives first at the finish stage. The delayed clock pulses are denoted as DEL in Fig. 5.3. The circuit of the delay line is illustrated in Fig. 4.4 and is explained in depth in Chapter 4.

The finish stage is composed by latches, where the index of the pulse that arrives first is stored. In the beginning of a clock cycle, the DEL signals are low, therefore, the PER signal is kept high through a weak pull-up resistor and the latches are enabled. Once a DEL signal switches to 1, the value 1 is stored in the corresponding latch and, subsequently, the PER signal switches to 0 and disables the latches. In this way, the latch that corresponds to the fastest pulse (*i.e.* the WORD_XORED with the minimum number of transitions) is set while the rest are reset. The SEL signals that are the outputs of the latches are fed to the final stage, which is the winner stage. The basic functionality of this stage is described in detail in Chapter 4, however, circuitry is

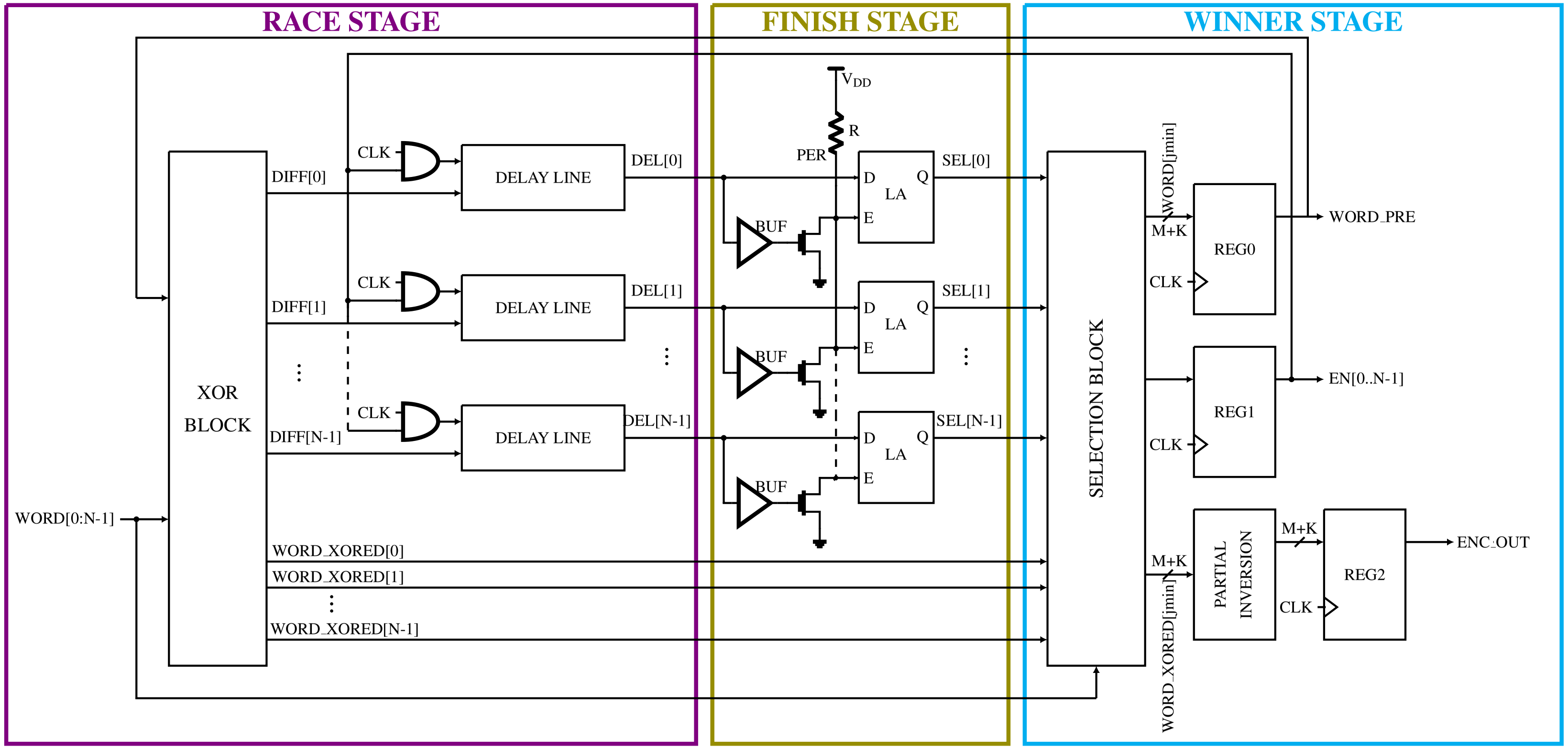


Figure 5.3: Encoder circuit of STTE.

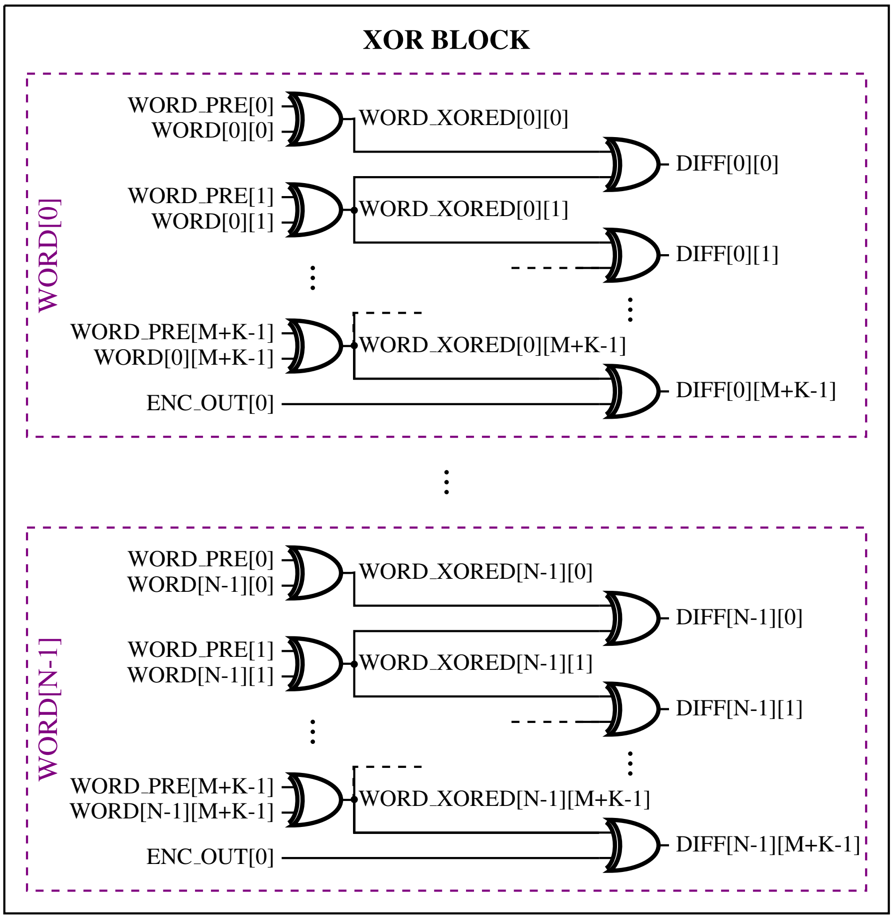


Figure 5.4: Circuit of the XOR block of the STTE encoder.

added to this circuit to suit the requirements of this application.

The first part of the winner stage is the selection block. This block selects the word that won the race (WORD[jmin]) according to the SEL signals and stores it in REG0 in order to be used in the next cycle. In addition, this stage keeps track of the words that have already been transmitted. This information is stored in REG1. The outputs of this register, EN[0..N-1], are initialised with 1 and in each cycle EN[jmin], which corresponds to WORD[jmin], is switched to 0. All of the EN signals switch back to high once all *N* words are transmitted. In case of a tie, *i.e.* two or more DEL signals switch to 1 at the same time at the finish stage, all the corresponding SEL signals are equal to 1. In this case, the selection block is designed to select the word with the lowest index as in AWR (Chapter 4). However, the selection of WORD[jmin] is not sufficient for STTE. In STTE, the selection block selects the corresponding WORD_XORED[jmin] as well since the data word is transmitted over the bus in this form. To balance the number of transitions and facilitate CDR, parts of WORD_XORED[jmin] are inverted,

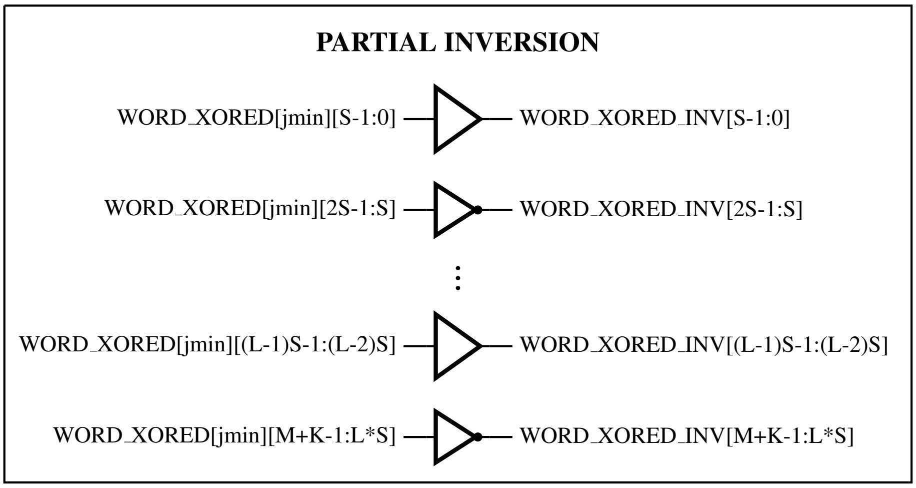

PARTIAL INVERSION
WORD_XORED[jmin][S-1:0]
WORD_XORED_INV[S-1:0]
WORD_XORED[jmin][2S-1:S]
WORD_XORED_INV[2S-1:S]
WORD_XORED[jmin][(L-1)S-1:(L-2)S]
WORD_XORED_INV[(L-1)S-1:(L-2)S]
WORD_XORED[jmin][M+K-1:L*S]
WORD_XORED_INV[M+K-1:L*S]

Figure 5.5: Circuit of the Partial Inversion block of the STTE technique.

as discussed in Section 5.2. This is implemented in the Partial Inversion block, where inverted and not inverted segments of *S* bits are interleaved as illustrated in Fig. 5.5. Note that the value of *S* is predetermined during design time. The output of this block is stored in REG2 and is transmitted in the next clock cycle.

5.3.2 STTE Decoder

The decoder provides the originally unencoded and in the right order data. The circuit diagram of the decoder is displayed in Fig. 5.6. The input of the decoder is first processed by the Partial Inversion block. This part is identical to the Partial Inversion block in the encoder (Fig. 5.5) since the same bits must be inverted to restore the original word. The output of this block is XOR-ed with the previous decoded word DEC_PRE. A register is used to store DEC_PRE, which is initialised with 0 as the register REG0 of the encoder circuit (see Fig. 5.3).

To restore the initial order of the data words, *N* buffers are required where the words are stored in the original order. The decoder utilises the *K* last bits, which indicate the word order, to enable the appropriate buffer to store the DEC_WORD. Therefore, a simple *K*-to-*N* decoder is used, the output of which enables the corresponding buffer.

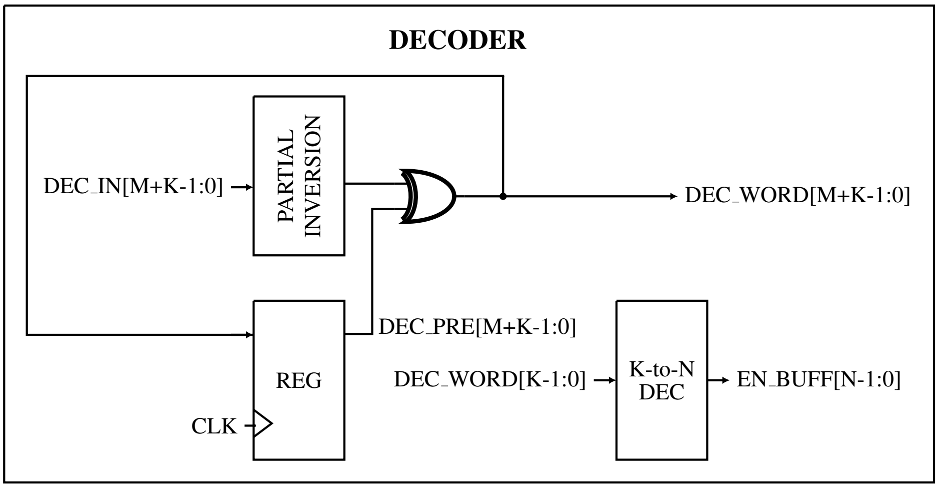


Figure 5.6: Circuit of the STTE decoder.

5.4 Energy-efficiency Results

The results in energy reduction provided by STTE are presented in this section. The methodology used to investigate the energy-efficiency of STTE is described in subsection 5.4.1. The overheads of the STTE encoder and decoder circuits are discussed in subsection 5.4.2, while the decrease in energy consumption provided by STTE is presented in subsection 5.4.3.

5.4.1 Evaluation Methodology

The energy savings of the proposed scheme are estimated for a 2.5-D integration scenario. An interposer-based interconnect is assumed to link two dies that are bump bonded on a silicon interposer. The circuit diagram that models this link is depicted in Fig. 5.7. The link model is composed of two bus lanes since differential signalling is used in high speed SerDes interfaces, such as PCI express [111]. Each bus lane comprises two buffers that model the driver and receiver logic, two I/O cells that convert the core voltage ($V_{DDCORE} = 1.2$ V) to I/O voltage ($V_{DDIO} = 1.8$ V) and *vice versa* and, also, model the ESD protection circuit, the parasitic capacitance of μbumps ($C_{\mu bump} = 30$ fF), and the interconnect, which is a distributed wire model. The *RC* characteristics of the latter model are listed in Table 4.5 and are calculated for the global wire dimensions reported in [37] for a 65 nm technology with minimum pitch.

To estimate the energy savings of the proposed technique, the link model (Fig. 5.7) is simulated with both scrambled traces and encoded with STTE traces using

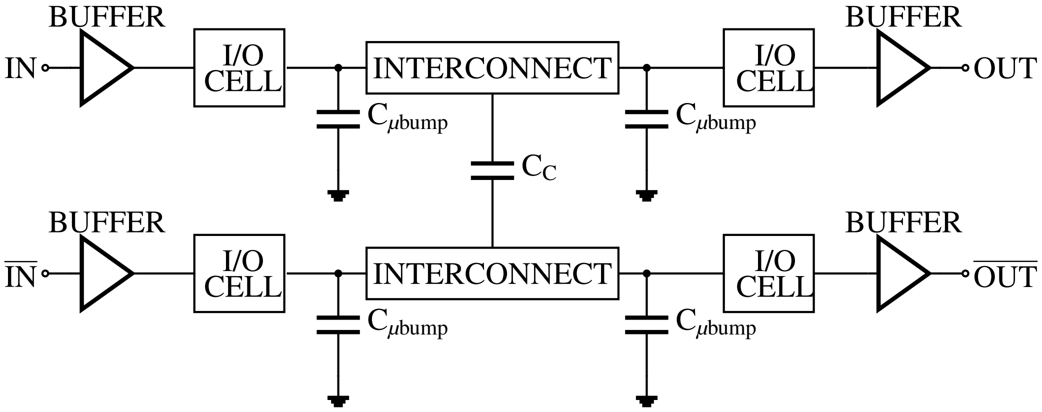


Figure 5.7: Test circuit of an interposer-based interconnect.

Spectre, Cadence. The benchmarks LFRic, InfOli, LITE_LOOP, LU_SOLVER, and NASTY_EXPS, reported in Table 4.2, are used to determine energy savings. These traces are generated using the gem5 simulator [117] and comprise a mix of memory addresses and data. Therefore, these traces provide realistic traffic for high-speed SerDes interfaces. Further information is provided in subsection 4.2.2.

To provide a fair evaluation of the energy savings of the proposed technique, the overhead in power of encoding and decoding must be considered. Therefore, the STTE circuits as well as the scrambler and the descrambler are implemented with a 65 nm technology [99]. The STTE circuit is described in Verilog and synthesized with Synopsys Design Compiler, apart from the finish stage and the delay lines of the race stage (see Fig. 5.3), which are implemented in schematic using Virtuoso, Cadence. The STTE encoder and decoder circuits and the bus model are simulated using Spectre, Cadence at nominal conditions, *i.e.*, typical device corners and 27°C. The operating frequency of the STTE encoding and decoding and for the scrambler and descrambler circuits is 250 MHz. The power consumed by the scrambler and descrambler circuits is estimated using Synopsys Design Compiler.

5.4.2 Power and Timing Overheads

The overheads of the STTE circuit are provided in this subsection. The overhead in power of the proposed STTE technique is reported in Table 5.1. Power is measured and averaged for the first five benchmarks of Table 4.2, which comprise a mix of addresses and data. The power consumption is estimated for different numbers of reordered words, N, and different step sizes, S. The number of reordered words, N, has a substantial role in the power overhead of the encoder circuit. The power overhead of the

Table 5.1: Power overhead in mW of the STTE encoder and decoder circuits operating at 250 MHz.

Step	$N = 16$		$N = 32$	
	Enc.	Dec.	Enc.	Dec.
$S = 5$	2.35	0.21	3.45	0.21
$S = 18$	2.35	0.21	3.49	0.21
$S = 34/35$ [1]	2.36	0.21	3.48	0.21

[1] $S = 34$ for $N = 16$ and $S = 35$ for $N = 32$.

Table 5.2: Timing overheads of the STTE encoder and decoder circuits operating at 250 MHz.

Overheads	$N = 16$		$N = 32$	
	Enc.	Dec.	Enc.	Dec.
Critical Path Delay [ns]	2.79	0.42	2.89	0.40
Throughput decrease	5.88%		7.24%	

encoder with $N = 32$ is about 1.5× higher compared to the overhead with $N = 16$. This behaviour is expected since the circuitry increases with N, specifically, the size of the XOR and selection blocks and the number of delay lines and latches (Fig. 5.3). On the contrary, the effect of S on power is not significant as the circuitry does not change much.

The timing overheads of STTE are reported in Table 5.2. The critical path delay of the encoder is slightly higher for $N = 32$, whilst the step value does not affect the delay. Due to the K redundant bits added to each 64-bit word, the throughput of the communication decreases. Higher number of reordered words leads to a higher decrease in throughput as more bits are required for decoding ($K = \log_2 N$). Hence, the throughput decrease is 5.88% for $N = 16$ and 7.24% for $N = 32$. This overhead is higher compared to the overhead of 64B/66B (∼3.03%), however, significantly lower compared to the overhead of 8B/10B encoding, which is 20%.

The decoding of STTE does not induce any latency as long as words are reordered within a packet. The reason is that in SerDes devices, the transmitter adds fields to the beginning and end of frames (packets), which must be received and processed before data can be utilised by the receiver. Typically, a header is added to the front of each packet to provide information about the packet followed by a sequence number and a Cyclic Redundancy Check (CRC) field is attached to the end of the packet for error correction [111]. The whole packet has to arrive and be checked for CRC errors, before the data can be utilised. The CRC function is further discussed in subsection 5.5.1. The decoder of STTE places the data words back in the initial order as they arrive. Hence, by the time the CRC bits arrive, the order of data is retrieved. Therefore, the decoding process of STTE does not induce a latency overhead.

Overall, the number of reordered words has a negative effect on the power, throughput, and delay overheads. However, as shown for AWR in subection 4.5.2, a higher N leads to a larger decrease in transitions and, hence, more energy is saved on the link. This behaviour is demonstrated in subection 5.4.3, where the energy savings of STTE are presented and the trade-off between timing overheads and energy consumption is further discussed.

5.4.3 Energy Savings

The STTE technique achieves higher decrease in energy dissipation compared to the scrambler that is typically used to maintain an adequate number of transitions and provide DC balance. The energy reduction provided by STTE compared to scrambling is illustrated in Fig. 5.8 and Fig. 5.9 for $N = 16$ and $N = 32$ reordered words, respectively. In addition, three different step sizes, S, are considered. The energy savings range from 24.92% to 58.12%. Both the overhead in power of the STTE encoder and decoder circuits and the overhead of the scrambler and descrambler circuits are included in determining the energy savings.

The energy savings are greatly affected by the value of S. Higher S leads to a significant reduction in energy since the number of transitions decrease, as shown in Fig. 5.1. For $S = 18$ the energy savings improve by $\sim$20% compared to $S = 5$ and a further $\sim$4% improvement is achieved for $S = 34$ or $S = 35$. This difference in energy is expected since more transitions are injected for a lower value of S, which leads to a higher energy dissipation. S does not affect timing overheads, hence, high energy savings are achieved without any impact on throughput or latency. However, a high value of S leads to lower number of transitions, which can have a negative impact on link integrity as described in Section 5.2. Effectively, S must be chosen considering the characteristics of the channel, as demonstrated experimentally in the following section.

Furthermore, the number of reordered words, N, affects energy savings. A larger

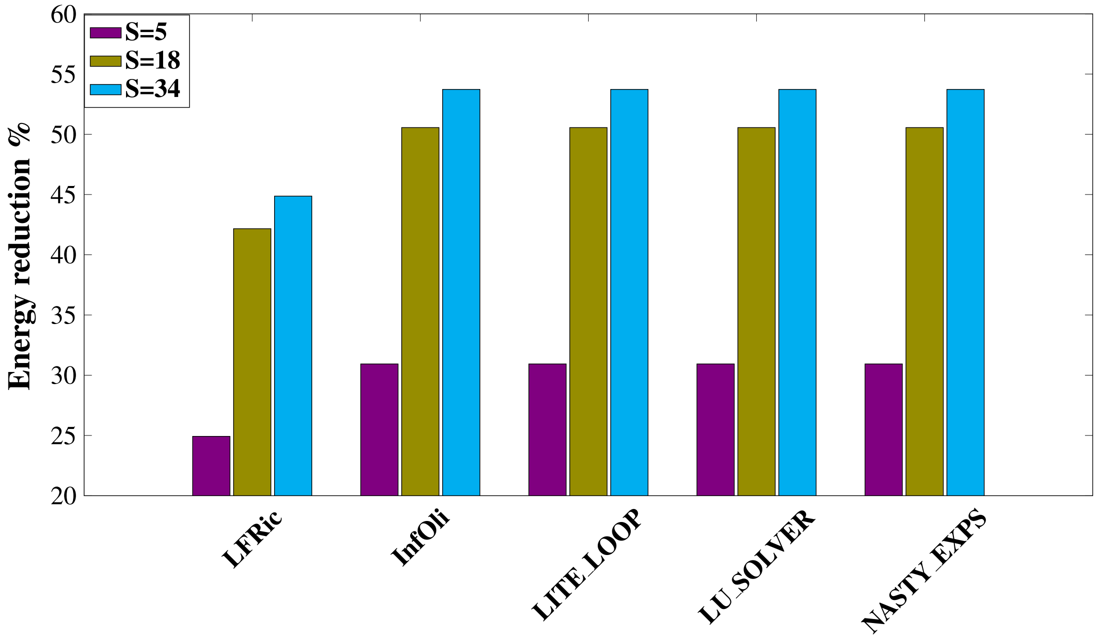


Figure 5.8: Energy savings for $N = 16$ reordered data words.

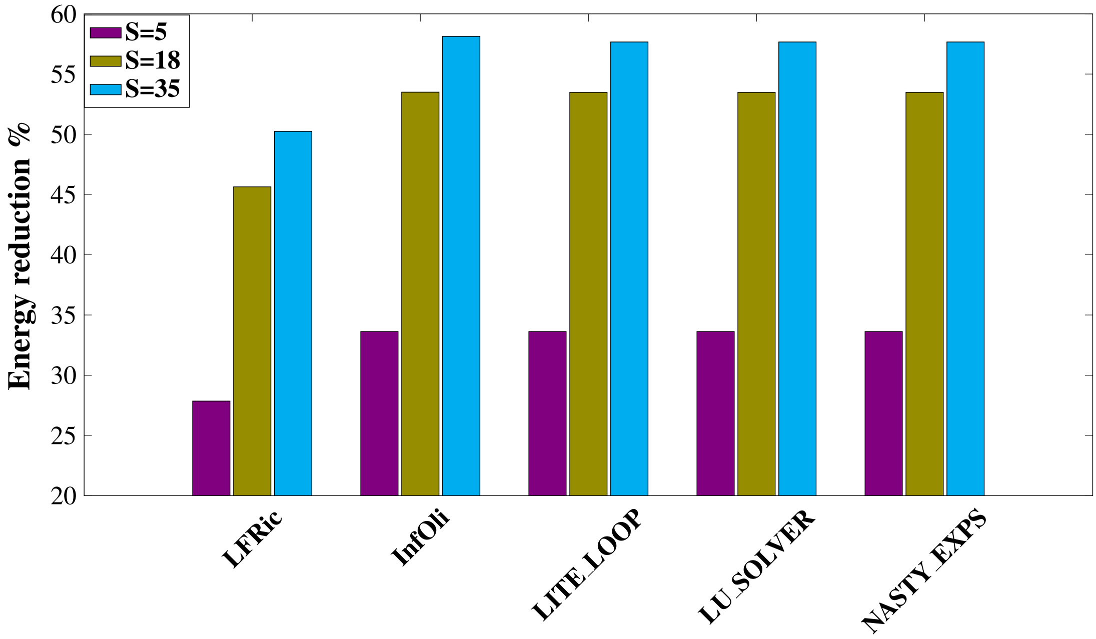


Figure 5.9: Energy savings for $N = 32$ reordered data words.

value of N results in a higher decrease in energy dissipation of about 3-5%. This behaviour is expected, since as the number of reordered words increases, the correlation among the N words increases and, therefore, fewer transitions occur where encoding is applied. However, this increase in energy savings comes at a cost in throughput and delay as shown in Table 5.2. Specifically, throughput is adversely affected. In this way, performance can be traded-off for energy reduction. Note, however, that this decrease in throughput is moderate as compared to other prior art techniques, such as 8B/10B encoding, where throughput loss can reach 20%.

5.5 Link Integrity Results

In SerDes devices, data encoding is crucial for maintaining DC balance, performing CDR, and, thus, preserving the link's correct operation. Therefore, the link integrity of data transmission using STTE is investigated experimentally using two FPGA boards, where a SerDes protocol is implemented. The experimental setup used to evaluate link integrity is described in subsection 5.5.1. The reliability of the communication is measured in terms of error rate and results are presented in subsection 5.5.2.

5.5.1 Experimental setup

The effectiveness of STTE in preserving clock recovery and DC balance is investigated for a physical link that connects two Field Programmable Gate Array (FPGA) boards as shown in Fig. 5.10. A custom, high-performance SerDes protocol, namely MAXILINK [118], is implemented on two Xilinx Zynq UltraScale+ Multiprocessor System on Chip (MPSoC) ZCU102 boards and a Xilinx Ultrascale GTH transceiver [119] is used. Two different cables are used as transmission media; a 30 cm long Spiral Strip (SS) coaxial cable [120] operating up to 18 Gb/s and a 50 cm long optical cable rated at 10 Gb/s. The Avago AFBR-703SDZ SFP+ transceiver [121] is utilised to convert the electrical signal to optical and *vice versa*.

MAXILINK is a configurable, high-speed, multi-lane SerDes protocol with a data link layer and a physical layer. The data link layer is responsible for frame assembling and disassembling and provides resiliency features such as CRC, CRC-based re-transmission, and lane failure management. The transmitter adds a CRC code to the end of each packet. This code is calculated as the remainder of a polynomial division of the content of the packet. The receiver repeats the calculation to determine whether any CRC error occured or not and forwards the contents of only the error-free packets [101]. The receiver informs the transmitter about the result of the check and, in case any CRC errors are detected, the whole packet is re-transmitted.

The physical layer of MAXILINK provides an interface to Xilinx GTH physical layer (PHY). Specifically, MAXILINK implements 64B/66B encoding and scrambling to provide sufficient bit transitions for CDR and preserve DC balance. In addition, this layer also provides symbol and block alignment, lane synchronisation and deskew, and a handshake process. During the latter process, dynamic line rate switching is performed. The data rate is selected based on the error rate and ranges between 7.5 to 16.25 Gb/s. CDR and serialisation/deserialisation are performed by the GTH PHY.

64B/66B encoding adds a 2-bit tag to each 64-bit data block. There are only two possible values that the 2-bit tag can take. The value 01 indicates that the block contains data, while 10 is used to specify a control block [101]. In this way, at least one transition every 66 bits is ensured. The 2-bit tag is used for error detection as well. A sync error is issued in case an illegal value is detected by the receiver, which causes the link to re-synchronise.

Each packet contains a 64-bit header, 18 data words of 64 bits, and a 16-bit CRC code. The 2-bit tag of 64B/66B encoding is added to both scrambled and encoded with STTE data words as the two additional bits are used for error detection. The control and idle packets are always scrambled, however, the header of each packet is

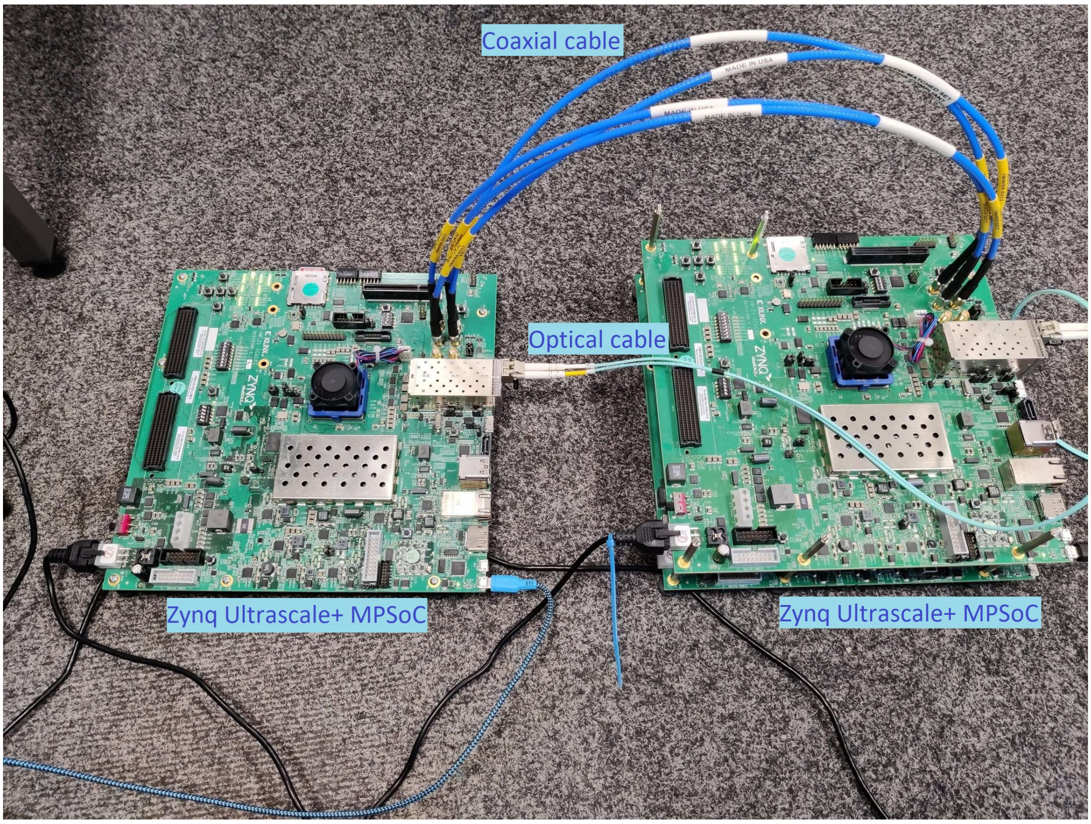


Figure 5.10: Experimental setup.

not scrambled when data is encoded with STTE. Finally, a single bus lane is used for conducting experiments for the sake of simplicity.

Note that this experimental setup can be used only for evaluating the integrity of the link and not the power savings of the method. The reason is that the analogue-like delay lines depicted in Fig. 5.3, designed at transistor level, cannot be implemented on the FPGA boards. Instead, the implementation of the encoder with the available resources on the FPGAs leads to high overhead in power, thereby making the available setup suitable for evaluating only the link integrity.

5.5.2 Link Integrity and Performance

Both frequent bit transitions and a roughly equal number of 0's and 1's are required to maintain DC balance and clock synchronisation between the transmitter and the receiver. In this way, the correct sampling of incoming data is ensured and data errors are avoided. Therefore, in this subsection, the error rate is used to characterise the resulting link integrity of the transmitted data for various scenarios. The SerDes protocol, MAXILINK, detects three types of errors; CRC, frame, and sync errors. A CRC error is issued when the transmitted CRC code does not match the CRC code calculated by the receiver. A frame error occurs in case the receiver detects a frame that is not expected or the frame length is different from the expected length. Finally, a sync error is issued when the 2-bit tag added by the 64B/66B encoding takes an illegal value, *i.e.* 00 or 11.

First, an investigation of the link's resilience in run length and DC bias is conducted. Synthetic bitstreams are generated, which comprise a repeated stream of consecutive 0's and 1's. The number of successive 0's and 1's are defined by the duty cycle and off-period. The duty cycle and the off-period range from 10% to 50% and from 144 to 2,032 bits, respectively, for the coaxial cable. For the optical link, the duty cycle ranges between 25% and 50%, while the off-period spans from 48 to 288 bits. In this way, the effects of DC bias and frequency of transitions on link integrity are investigated. Each experiment is repeated five times and the total number of errors as well as the data rate are measured.

The mean error rate per packet is illustrated in Figs. 5.11 and 5.12 for the coaxial and the optical link, respectively. No errors are detected with the coaxial cable when the run length is lower than $\sim$1,300 bits and the number of 0's are between 50% and 60%. In this case, the disparity in the number of 0's and 1's is low and the frequency of transitions is sufficient to maintain clock synchronisation. Alternatively, the optical

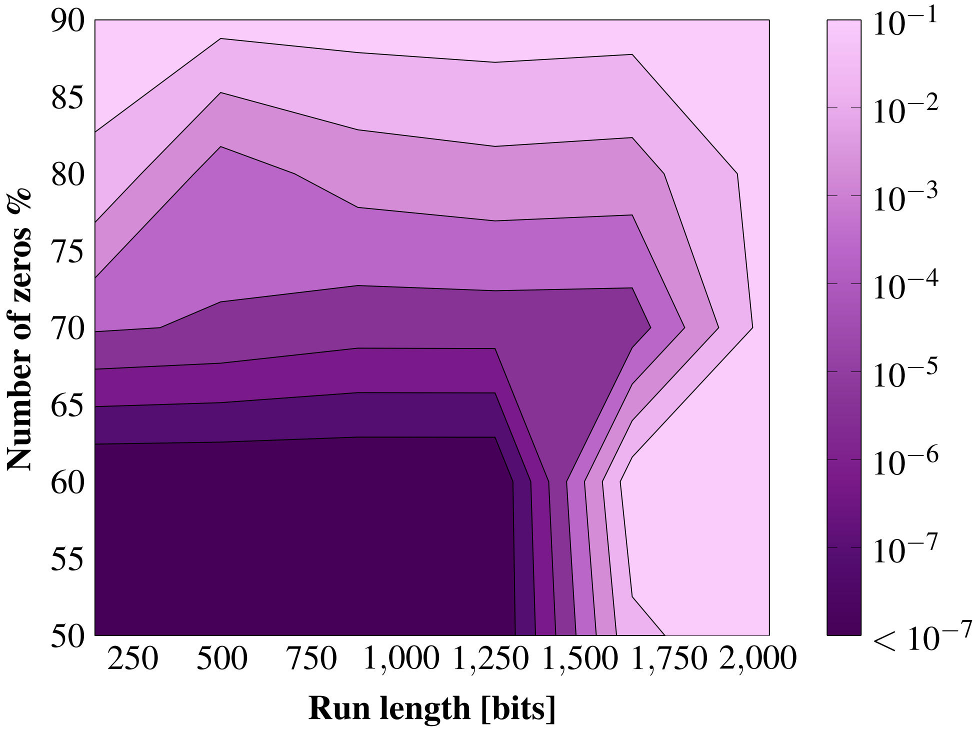


Figure 5.11: Errors per packet using a coaxial cable.

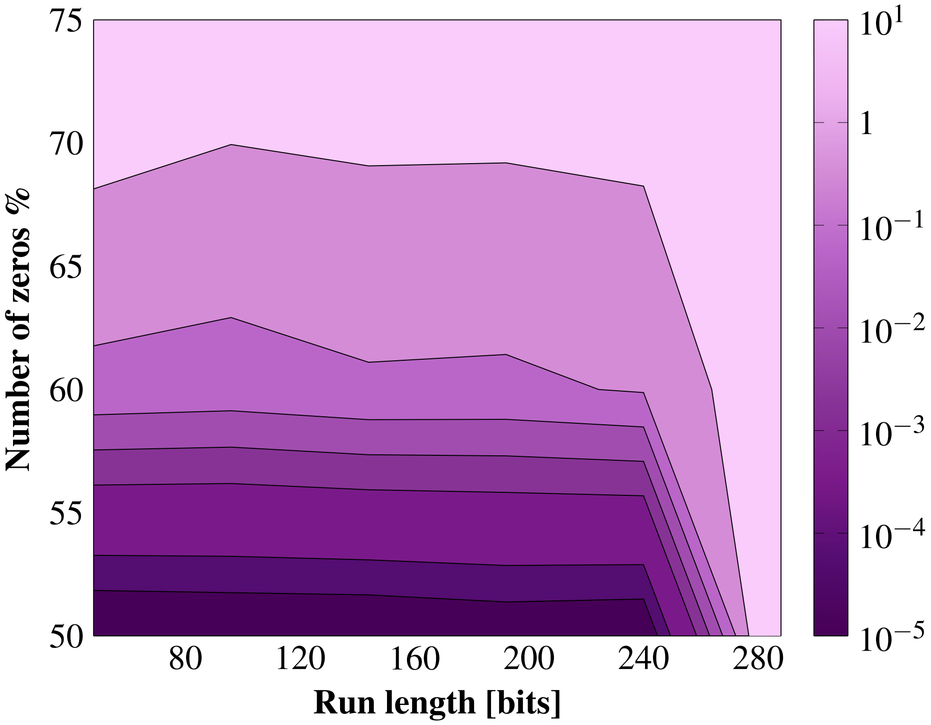


Figure 5.12: Errors per packet using an optical cable.

link is more prone to errors. The lowest error rate ($\sim 10^{-5}$) is exhibited when the run length is lower than $\sim$250 bits and the number of 0's is between 50% and 52%. As the run length and the number of 0's increase, the errors increase for both links and more rapidly for the optical link. Note that the error rate has the same behaviour when the number of 0's are less than 50%, since DC balance is lost as the number of 0's diverges strongly from 50%.

MAXILINK decreases the link's data rate when sync errors occur. Hence, the error rate affects the performance of the link. The data rate of the synthetic data transmission is depicted in Figs. 5.13 and 5.14 for the coaxial and the optical link, respectively. The data rate ranges between 7.5 Gb/s and 16.25 Gb/s for the coaxial cable and between 7.5 Gb/s and 13.75 Gb/s for the optical cable. Although the optical cable is rated at 10 Gb/s, the link reaches up to 13.75 Gb/s when data are transmitted scrambled without affecting the error rate. The link rate follows the same trend with the error rate for both links. The maximum speed is achieved for the dark areas where the error rate is the lowest. Furthermore, the data rate degrades rapidly with the number of sync errors, thus, the minimisation of sync errors is crucial for the performance of the link.

The effectiveness of the STTE scheme in maintaining link integrity is investigated for real data streams, particularly, the first five benchmarks of Table 4.2. The results of the STTE encoding are compared with the results of transmitting the same data scrambled and unencoded. DC balance and run length are the two parameters that determine the error and data rate as shown in Figs. 5.11-5.14. Therefore, the percentage of 0's compared to the total number of bits as well as the run length exhibited by STTE over the transmission of 2.38 GB of data are reported in Tables 5.3 and 5.4, respectively. Both $N = 16$ and $N = 32$ reordered words are considered and different step sizes are selected; $S = 5$, $S = 18$, and $S = 34$ for $N = 16$ and $S = 35$ for $N = 32$.

As with scrambling, STTE does not guarantee DC balance for short periods of time, nonetheless, maintains DC balance over long data streams, as shown in Table 5.3. Note that the ideal number of 0's is 50% as in this case the number of 0's and 1's are equal. The proposed STTE scheme provides DC balance, as the number of 0's is close to 50%. In fact, STTE provides better DC balance compared to the scrambler since the number of 0's ranges between 48.52% and 50.66% for the STTE encoded stream and between 47.80% and 47.90% for the scrambled stream.

The maximum run length of the STTE encoded data ranges from 25 to 70 bits depending on S, as reported in Table 5.4. Although STTE's maximum run length is considerably higher compared to the maximum run length of scrambled data, it

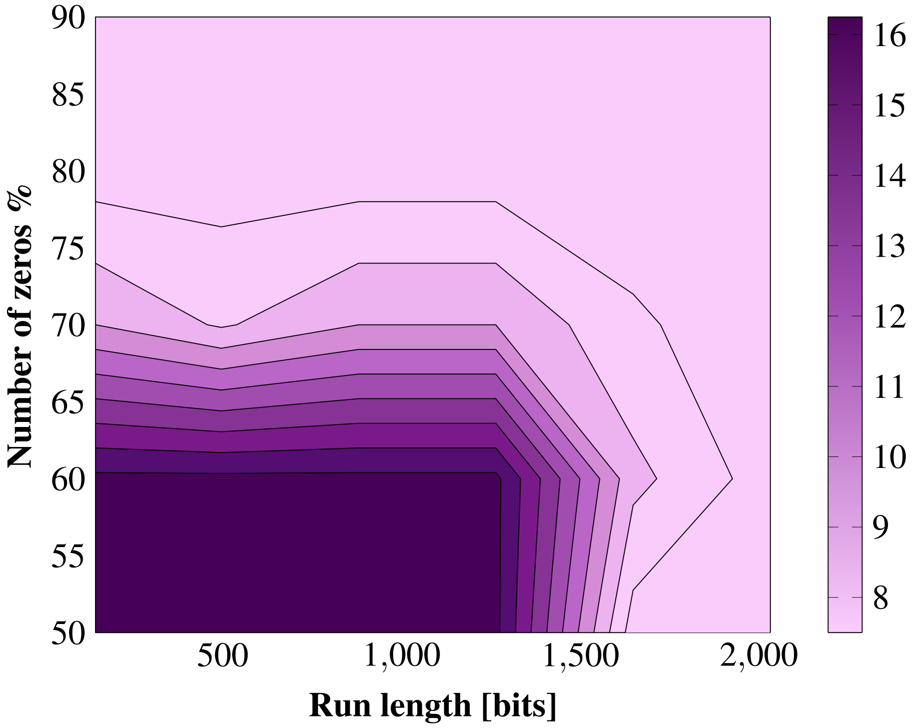


Figure 5.13: Data rate [Gb/s] using a coaxial cable.

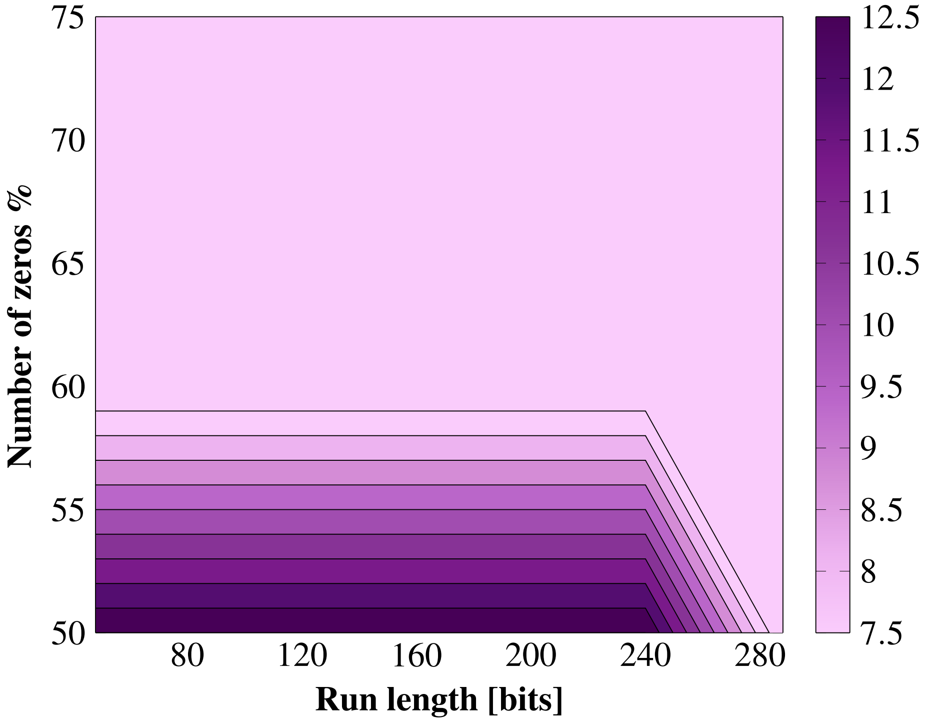


Figure 5.14: Data rate [Gb/s] using an optical cable.

Table 5.3: Percentage of 0's. DC balance is achieved for ∼50%.

Encoding	LFRic	InfOli	LITE LOOP	LU SOLVER	NASTY EXPS
Unencoded	69.92%	70.39%	70.38%	70.38%	70.39%
Scrambled	47.90%	47.80%	47.80%	47.80%	47.80%
STTE_{16}, S=5	48.69%	48.68%	48.68%	48.68%	48.68%
STTE_{16}, S=18	50.58%	50.66%	50.66%	50.66%	50.66%
STTE_{16}, S=34	50.32%	50.61%	50.60%	50.60%	50.61%
STTE_{32}, S=5	48.54%	48.52%	48.52%	48.52%	48.52%
STTE_{32}, S=18	49.94%	50.01%	50.02%	50.02%	50.02%
STTE_{32}, S=35	49.29%	49.53%	49.52%	49.52%	49.52%

Table 5.4: Maximum run length in bits.

Encoding	LFRic	InfOli	LITE LOOP	LU SOLVER	NASTY EXPS
Unencoded	5126	4127	4127	4127	4127
Scrambled	6	6	6	6	6
STTE_{16}, S=5	25	26	26	26	26
STTE_{16}, S=18	39	41	41	41	41
STTE_{16}, S=34	69	68	69	69	69
STTE_{32}, S=5	26	26	26	26	26
STTE_{32}, S=18	35	41	41	41	41
STTE_{32}, S=35	70	70	70	70	70

is significantly lower than the run length of unencoded data and below the threshold where errors occur for the coaxial cable according to Fig. 5.11. For the optical cable, STTE's maximum run length lies within the dark area of Fig. 5.12, where a low error rate ($\sim 10^{-5}$ errors per packet) is exhibited.

To measure the link's error rate and the data rate, the data streams are transmitted

Table 5.5: Errors per packet using a coaxial cable.

Encoding	LFRic	InfOli	LITE LOOP	LU SOLVER	NASTY EXPS
Unencoded	2.68e-4	5.30e-4	9.64e-4	9.19e-4	8.16e-4
Scrambled	0	0	0	0	0
STTE_{16}, S=5	0	0	0	0	0
STTE_{16}, S=18	0	0	0	0	0
STTE_{16}, S=34	0	0	0	0	0

over the link encoded by the STTE scheme, scrambled, and unencoded. All the experiments are repeated five times and the mean error rate and data rate are calculated. The number of reordered words is $N = 16$ for STTE, since the run length and DC balance exhibited for $N = 32$ are similar, hence, the values of the error and data rate are approximately equal.

The error rate and data rate for the coaxial link are reported in Tables 5.5 and 5.6, respectively. STTE maintains link integrity as the error rate is 0 and the link operates at the maximum speed even for $S = 34$ in case of the coaxial link. According to Table 5.4, the maximum run length is 70 bits, which is much lower compared to the threshold (∼1,300 bits) where errors start to occur as shown in Fig. 5.11. Hence, STTE provides sufficient bit transitions for CDR. In addition, DC balance is preserved as the number of 0's is close to 50%, as reported in Table 5.3. Therefore, STTE can replace the scrambler without inducing data errors or affecting speed. At the same time, the number of transitions of STTE is significantly lower compared to scrambling, thus, a large portion of the data transmission energy is saved as demonstrated in subsection 5.4.3. Hence, STTE successfully decreases energy up to 58% (see Figs. 5.8 and 5.9) without compromising link integrity for the coaxial cable.

The mean values of the error and data rate for the optical link are listed in Tables 5.7 and 5.8, respectively. STTE maintains link integrity for the optical cable. Specifically, when $S = 5$, STTE does not induce any errors and the data rate is maximum (13.75 Gb/s) as when scrambling is used. Therefore, a ∼25%-30% of the transmission energy is saved without affecting the link integrity. For higher values of S, the maximum run length is longer as reported in Table 5.4. Therefore, the error rate increases and the data rate decreases. However, the error rate for $S = 18$ and $S = 34$ ($\sim 10^{-5}$) is

Table 5.6: Link data rate in Gb/s using a coaxial cable.

Encoding	LFRic	InfOli	LITE LOOP	LU SOLVER	NASTY EXPS
Unencoded	7.5	7.5	7.5	7.5	7.5
Scrambled	16.25	16.25	16.25	16.25	16.25
STTE_{16}, S=5	16.25	16.25	16.25	16.25	16.25
STTE_{16}, S=18	16.25	16.25	16.25	16.25	16.25
STTE_{16}, S=34	16.25	16.25	16.25	16.25	16.25

Table 5.7: Errors per packet using an optical cable.

Encoding	LFRic	InfOli	LITE LOOP	LU SOLVER	NASTY EXPS
Unencoded	2.42e-1	2.65e-1	2.65e-1	2.58e-1	2.60e-1
Scrambled	0	0	0	0	0
STTE_{16}, S=5	0	0	0	0	0
STTE_{16}, S=18	1.88e-5	4.66e-5	9.47e-6	5.07e-6	7.66e-6
STTE_{16}, S=34	1.50e-5	1.40e-5	1.56e-5	2.20e-5	1.53e-5

low compared to unencoded transmission ($\sim 10^{-1}$) since the maximum run length (70 bits) is considerably lower than the threshold (∼240 bits), where the error rate starts to increase rapidly. In addition, the decrease in data rate is low and the rate remains higher than the optical transceiver's nominal operating rate (10 Gb/s). As demonstrated in subsection 5.4.3, more energy is saved for higher values of S. Consequently, STTE utilises parameter S to enable an important trade-off among energy, error rate, and data rate for those links, which are more susceptible to transmission errors, such as the investigated optical link.

5.6 Summary

The reduction of the energy demand of high speed, source asynchronous, serial interconnects is investigated in this chapter. In these SerDes interfaces, the data are required

Table 5.8: Link data rate in Gb/s using an optical cable.

Encoding	LFRic	InfOli	LITE LOOP	LU SOLVER	NASTY EXPS
Unencoded	7.5	7.5	7.5	7.5	7.5
Scrambled	13.75	13.75	13.75	13.75	13.75
STTE$_{16}$, S=5	13.75	13.75	13.75	13.75	13.75
STTE$_{16}$, S=18	13.125	13.125	13.125	13.125	13.125
STTE$_{16}$, S=34	13.75	11.875	11.875	12.25	11.875

to exhibit regular bit transitions such that the clock can be recovered by the receiver. Another prerequisite for the prevention of data errors is DC balance, which means that the number of 0's and 1's of the data stream must be approximately equal. Therefore, techniques such as 8B/10B encoding and scrambling are employed to satisfy these conditions. However, these techniques increase the transition rate significantly and, hence, consume a high amount of energy. Alternatively, encoding schemes that reduce the energy demand of serial communication are only applicable to synchronous interfaces as their objective is to achieve the maximum reduction in transitions. Therefore, CDR is obstructed, which can lead to the incorrect sampling of data signals. In addition, these schemes do not consider DC balance, which can lead to data loss.

An encoding technique is proposed, namely STTE, tailored to source asynchronous, serial interfaces. The proposed scheme is inspired by AWR, as the number of transitions are reduced by reordering the data words using the NN algorithm. In addition, a partial inversion mechanism is employed to inject an appropriate amount of transitions to facilitate CDR. In this way, STTE regulates the frequency of transitions such that the clock can be recovered, while the number of transitions is reduced compared to scrambling and, hence, energy is saved. The partial inversion mechanism contributes to the balancing of the number of 0's and 1's, hence, signal distortion is prevented.

The STTE technique is investigated in terms of both energy-efficiency as well as maintaining link integrity to avoid data errors. An encoder and a decoder circuit are designed and implemented in a 65 nm technology and an interposer-based interconnect model is developed to estimate the energy dissipated in encoding and decoding and in data transmission. STTE provides high energy savings compared to scrambling, which is typically used in high speed SerDes interfaces. The savings range from $\sim$25% to

$\sim$58% depending on the number of reordered words and the amount of transitions injected by the partial inversion mechanism.

Furthermore, to investigate link integrity, a custom SerDes protocol is implemented on two FPGA boards that are interconnected using an electrical and an optical cable. The error and data rates are measured to determine link integrity and performance. The STTE encoding maintains link integrity as no error occurs and the data rate of the link does not decrease in case of the electrical cable. Hence, STTE saves up to 58% of the energy dissipated for communication, without inducing any data errors and, thus, preserving link integrity. Alternatively, the optical link is more prone to errors, therefore, STTE saves 25%-30% of the transmission energy without affecting link integrity. Finally, a higher reduction in energy can be achieved at the cost of a low increase in error rate.

Chapter 6

Conclusions and Future Work

With the advent of parallel computing and the increasing number of cores per system, the rapid scaling of I/O bandwidth is essential. Constraining power consumption is a key challenge to continue scaling bandwidth in a sustainable manner. Power dissipated in communication is disproportionate to the power required for computations and the gap between them expands as technology scales. Therefore, increasing the energy-efficiency of communication is of high importance. Novel techniques for reducing the power demand of die-to-die interconnects are discussed in this thesis. A summary of the outcomes of the thesis is provided in Section 6.1. Directions to extend this work are presented in Section 6.2.

6.1 Summary

Three different ways of decreasing the dynamic power of off-die communication are discussed in this thesis, low-swing signalling, 2.5-D integration, and signal encoding. Low-swing signalling decreases the voltage range of transmitted signals. 2.5-D technologies increase integration density and reduce the distance between dice. Therefore, the capacitive load is decreased. Signal encoding schemes reduce the bit transitions of the transmitted data and, thus, the frequency of charging and discharging the interconnect load. A review of existing encoding techniques is provided in Chapter 2 for both parallel and serial links. The strengths and limitations of these encoding schemes are discussed.

Low-Swing Signalling for 2.5-D Technologies

Low-swing signalling is an effective way to decrease the energy demand of die-to-die communication. The energy benefits of this method have been evaluated for chip-to-chip interconnects only at the package and PCB level. The potential energy improvement for interposer-based interconnects is estimated in this work by comparing the energy consumed by a low-swing scheme with a full swing solution. Three different interposer materials are investigated: silicon, glass, and organic. The limitations of the low-swing scheme are, also, quantified. The critical length, *i.e.* the link length above which the low-swing scheme pays off, is determined.

The low-swing solution provides higher energy savings for silicon interposers, which exhibit higher wire capacitance compared to the other materials. Hence, the critical length is the shortest for silicon interposers and reaches 380 μm for glass and organic interposers for minimum wire pitch. Furthermore, the effect of wire parameters on the energy efficiency of low-swing communication is evaluated. Results demonstrate that the energy efficiency of low swing signalling is the highest when large wire width in combination with low spacing are required.

Adaptive Time-based Encoding for Parallel Buses

Despite the high energy savings of low-swing signalling, noise sensitivity and large area overhead are important limitations of this method. Therefore, this thesis emphasises signal encoding methods. The majority of existing encoding techniques rely on specific characteristics of the data stream to reduce transitions. Alternatively, general purpose schemes either provide a moderate reduction in bit transitions or exhibit a high overhead in power, which counteracts the power saved on the interconnect.

An encoding technique, named AWR, is proposed in Chapter 4 that addresses the issues of existing techniques. AWR provides a significant reduction in bit transitions (up to 48%) for diverse data types by changing the order of the transmitted data words. A novel encoder circuit is designed, where the time-domain is exploited for complex computations. Instead of using power-hungry adder trees for the computations of Hamming distances, delay lines are used, which comprise inverter chains. In this way, the power consumed for encoding and decoding is restrained.

The power savings of the technique are estimated for a link interconnecting two dies on a silicon interposer. AWR yields up to 23% savings in power for multiplexed address-data benchmarks at just 1 mm interconnect length and outperforms state-of-the-art techniques. Furthermore, the limitations of the technique in terms of delay, latency, and saving power are demonstrated. Additionally, the robustness of AWR is investigated under process variations. Although AWR is less efficient under process variations compared to the typical corner, the provided savings in power remain substantial.

AWR adds a number of redundant bits to each transmitted word to allow the decoder to retrieve the order. Different ways to transfer the order information exist. In this work, the simplest method is used, where a unique code is added to each word that denotes the order of the word. This implementation leads to the highest overhead in terms of the number wires and I/O pins. A way to reduce this overhead is discussed.

Another main limitation of AWR is the latency penalty. In the worst case scenario, where the first word is transmitted last, the received words can only be utilised by the receiver after the transmission of the entire burst is finished. Therefore, if the critical word is transmitted last by AWR, the CPU has to wait until the end of the transmission to continue execution, which degrades both performance and power and, therefore, counteracts the power saved in the bus by encoding. AWR can be amended to always transmit the critical word first and reorder only the remaining words to reduce latency.

AWR provides the highest power savings when the window size, N, is between 32 and 64 words. Therefore, AWR is directly applicable to 32 byte cache line transactions. However, other interfaces use larger burst sizes or the size of bursts is not constant, such as in DMA and AXI transactions. AWR can be improved to handle larger data bursts and bursts with variable size.

Adaptive Time-based Encoding for Serial Interfaces

Encoding techniques for parallel links are not applicable to serial interfaces. The objective in the latter is the reduction of bit transitions within each data word rather than between successive words as in parallel links. Low-power encoding techniques for serial links have been proposed, however, these schemes are only applicable to source synchronous interfaces. In source asynchronous links, such as PCIe, the receiver recovers the clock from the incoming data, therefore, the transmitted data are required to exhibit regular bit transitions. Another prerequisite for the prevention of data errors is DC balance, *i.e.* the number of 0's and 1's of the data stream must be approximately equal. Encoding techniques, such as 8B/10B and scrambling are typically employed to satisfy these conditions. However, these techniques induce a high increase in power dissipated on the interconnect.

In Chapter 5, an encoding technique is presented that reduces the power demand of source asynchronous interfaces. The new scheme, namely STTE, regulates the frequency of transitions such that the clock can be recovered, while the number of transitions is lowered compared to scrambling, thereby, saving energy. The STTE scheme reduces bit transitions by reordering the data words and, then, a partial inversion mechanism is applied, which injects an adjustable amount of transitions to facilitate clock recovery and preserve DC balance. In this way, STTE allows energy-efficiency and link integrity to be traded-off. A circuit implementation of the technique is provided, where the circuit calculating Hamming distances in the time domain is reused to restrict the power overhead of the encoder.

The effectiveness of this technique is investigated both in terms of saving energy and maintaining link integrity. STTE provides between 25% and 58% savings in energy compared to scrambling for an interposer-based link. These savings depend on the number of reordered words and mainly on the amount of transitions injected by the partial inversion mechanism. Furthermore, the provided link integrity is estimated experimentally for an electrical and an optical cable that interconnect two FPGA boards. The error and data rates are measured to determine link integrity and performance. In case of the coaxial cable, no error occurs and the data rate does not decrease, hence, STTE saves up to 58% energy without degrading link integrity. Alternatively, the optical cable is more prone to errors, thus, a higher amount of injected transitions is required to maintain link integrity. STTE saves 25%–30% without affecting the error and data rate. These trade-offs between energy savings, error rate, and data rate are discussed in detail.

STTE saves energy only in interfaces with voltage-mode drivers, where there is a one-to-one mapping between the digital signal and the transmitted symbol, such as in SSTL drivers. STTE is not applicable when digital modulation schemes are used, where each symbol represents two or more digital bits. Additionally, STTE adds redundancy to transmit the order information to the receiver, thereby introducing an overhead. As in AWR, the simplest way to transmit the order is used, where a binary code denoting the order of each word is added. However, there are techniques to reduce this overhead.

6.2 Future Work

The two proposed encoding methods achieve substantial savings in power for parallel and high-speed serial links. However, certain challenges can restrict the applicability of these techniques in different technologies. Directions to address these challenges are provided in this section. In addition, future directions are provided on how the findings of this thesis can be used in different applications to enhance energy efficiency.

Increasing Portability of Proposed Encoding Techniques

FPGA devices become more and more popular and are used as accelerators for computationally intensive applications, such as machine learning. Therefore, applications on these devices require high data bandwidth and communication power is often the bottleneck in satisfying power constraints. Hence, reducing power dissipated in these links is essential. Applying the proposed AWR and STTE techniques on FPGA devices is not straightforward due to the custom designed circuitry.

In both proposed encoding techniques, AWR and STTE, time-based circuitry is utilised to enhance the power efficiency of encoding. Therefore, custom circuitry is developed in a specific technology to implement delay-based calculations. Particularly, this circuitry consists of the delay lines, depicted in Fig. 4.4 and the finish stage, illustrated in Figs. 4.2 and 5.3. Despite increasing energy efficiency, these parts hinder the portability of the techniques. Consequently, altering the design such that it can be described in an Hardware Description Language (HDL), while maintaining the low-power properties of the delay lines is an important task. That would also increase portability of the design in different technologies including FPGA devices, facilitate verification, and decrease simulation time.

Integration of STTE in Ser/Des Protocols

High-speed, source asynchronous serial interfaces require complex protocols to reliably transmit data. These protocols consist of two or three different layers, such as data link layer and physical layer, which perform different operations, such as frame assembling and disassembling, error correction, sending acknowledgements, symbol and block alignment, serialisation and deserialisation, clock recovery, and handshaking. Data encoding is part of the physical layer [111], [118] and the basic architecture of this layer is described in Chapter 2. The integration of STTE in the physical layer is not straightforward and should be investigated.

STTE has to be adapted in different aspects to facilitate integration. STTE reorders the data words in a single frame to prevent adding latency. However, the number of data words within a packet is not constant in most Ser/Des protocols, therefore, STTE must be adapted to regulate transitions when the number of data words is different from the desirable. Furthermore, STTE is designed to adjust bit transitions and maintain DC balance for a single bus lane. However, most Ser/Des protocols including PCIe [111] can support multiple bus lanes. Therefore, STTE has to be modified to ensure that sufficient transitions are exhibited and DC balance is preserved in each bus lane.

AWR in Interfaces with Asymmetric Termination

Modern DRAM interfaces, such as DDR4 [83] and GDDR5 [84], use asymmetric termination to achieve higher bandwidth rates. Restricting the energy demand of these popular interfaces is important. As discussed in Chapter 2, the number of 0's or 1's (depending on the type of termination) has to be reduced to decrease the energy consumption of these interfaces. AWR is designed to decrease bit transitions, therefore, AWR must be adapted to effectively restrict the power of these interfaces. Moreover, memory traces exhibit a large number of 0's. Transmitting a long stream of 0's unencoded or inverted can achieve a high reduction in energy. Hence, the application of BI might be more effective than reordering in these cases. Consequently, combining AWR with BI for DRAM interfaces should be explored.

Combination of Compression with Proposed Encoding Techniques

Compression techniques effectively increase data throughput in both parallel and serial I/O interfaces. However, compression increases bit transitions by removing redundancy in data and, therefore, power consumption increases [73]. Energy Control and Metadata Consolidation are two techniques that combine encoding with compression to decrease the energy delay product of interconnects [73]. Combining different encoding and compression techniques should be investigated more to further decrease the energy of transmitting compressed data. The presented AWR and STTE techniques are effective in reducing power for diverse data types, therefore, these schemes would be good candidates to combine with compression. Challenges that arise from the variable word length of compressed data should be addressed.

Bibliography

[1] Gareth Kenyon, "Heterogeneous Integration and the Evolution of IC Packaging," *EE Times Europe*, April 2021.

[2] S. Mirabbasi, L. C. Fujino, and K. C. Smith, "Through the Looking Glass–The 2022 Edition: Trends in Solid-State Circuits from ISSCC," *IEEE Solid-State Circuits Magazine*, Vol. 14, No. 1, pp. 54–72, January 2022.

[3] N. Jones, "How to Stop Data Centres from Gobbling up the World's Electricity," Nature 561, pp. 163–166, 2018.

[4] R. Ho, K. W. Mai and M. A. Horowitz, "The Future of Wires," *Proceedings of the IEEE*, Vol. 89, No. 4, pp. 490–504, April 2001.

[5] A. García-Ortiz, D. Gregorek, and C. Osewold, "Optimization of Interconnect Architectures through Coding: A Review," *Saudi International Electronics, Communications and Photonics Conference*, pp. 1–6, April 2011.

[6] V. F. Pavlidis, I. Savidis, and E. G. Friedman, *Three-Dimensional Integrated Circuit Design* 2^{nd} *Edition*, Morgan Kaufmann Publishers, 2017.

[7] M. Koyanagi, T. Fukushima, and T. Tanaka, "High-Density Through Silicon Vias for 3-D LSIs," Proceedings of the IEEE, Vol. 97, No. 1, pp. 49–59, January 2009.

[8] J. W. Joyner, *et al.*, "A Three-Dimensional Stochastic Wire-Length Distribution for Variable Separation of Strata," *Proceedings of the IEEE International Interconnect Technology Conference*, pp. 126–128, June 2000.

[9] V. K. Jain, *et al.*, "A Highly Reconfigurable Computing Array: DSP Plane of a 3-D Heterogeneous SoC," Proceedings of the IEEE International SOC Conference, pp. 243–246, September 2005.

[10] "High Bandwidth Memory (HBM) DRAM (JESD235D)," *JEDEC*, March 2021.

[11] High Bandwidth Memory | AMD website, `https://www.amd.com/en/technologies/hbm`, 21 January 2022.

[12] "Hybrid Memory Cube – HMC Gen2," Micron Technology, 2018.

[13] Samsung 2bit V-NAND Technology | Samsung Semiconductor, `https://www.samsung.com/semiconductor/newsroom/tech-leadership/samsung-2bit-v-nand-technology`, 21 January 2022.

[14] M. Detalle *et al.*, "Interposer technology for high band width interconnect applications," Proceedings of the IEEE Electronic Components and Technology Conference, pp. 323-328, May 2013.

[15] K. Saban, "Xilinx Stacked Silicon Interconnect Technology Delivers Breakthrough FPGA Capacity, Bandwidth, and Power Efficiency," Xilinx, December 2012.

[16] Tesla P100 Data Center Accelerator | NVIDIA `https://www.nvidia.com/en-gb/data-center/tesla-p100/` 21 January 2022.

[17] Y. Kim, *et al.*, "Signal and power integrity analysis in 2.5D integrated circuits (ICs) with glass, silicon and organic interposer," *Proceedings of the IEEE Electronic Components and Technology Conference (ECTC)*, pp. 738–743, May 2015.

[18] R. R. Tummala, *et al.*, "Fundamental limits of organic packages and boards and the need for novel ceramic boards for next generation electronic packaging," *Journal of Electroceramics*, vol. 13, pp. 417-422, July 2004.

[19] V. Sukumaran *et al.*, "Through-package-via formation and metallization of glass interposers," *Proceedings of the IEEE Electronic Components and Technology Conference (ECTC)*, pp. 557–563, June 2010.

[20] L. Frenzel, *Handbook of Serial Communications Interfaces,* Elsevier, 2016.

[21] M. Sunohara *et al.*, "Studies on Electrical Performance and Thermal Stress of a Silicon Interposer with TSVs," *Proceedings of the IEEE Electronic Components and Technology Conference*, pp. 1088–1093, June 2010.

[22] B. M. D. Sawyer *et al.*, “Design and Demonstration of a 2.5-D Glass Interposer BGA Package for High Bandwidth and Low Cost,” *IEEE Transactions on Components, Packaging and Manufacturing Technology*, Vol. 7, No. 4, pp. 552–562, April 2017.

[23] Z. Wu *et al.*, “Modeling, Design and Fabrication of Ultra-thin and Low CTE Organic Interposers at 40μm I/O Pitch,” *Proceedings of the IEEE Electronic Components and Technology Conference*, pp. 301–307, May 2015.

[24] K. Oi *et al.*, “Development of New 2.5D Package with Novel Integrated Organic Interposer Substrate with Ultra-fine Wiring and High Density Bumps,” *Proceedings of the IEEE Electronic Components and Technology Conference*, pp. 348–353, May 2014.

[25] H. Zhang, V. George, and J. M. Rabaey, “Low-Swing On-Chip Signaling Techniques: Effectiveness and Robustness,” *IEEE Transactions on Very Large Scale of Integration (VLSI) Systems*, Vol. 8, No. 3, pp. 264–272, June 2000.

[26] S. H. Kulkarni and D. Sylvester, “High Performance Level Conversion for Dual $\mathrm{V_{DD}}$ Design,” *IEEE Transactions on Very Large Scale of Integration (VLSI) Systems*, Vol. 12, No. 9, pp. 926–936, September 2004.

[27] J. C. Garcia Montesdeoca, J. A. Montiel-Nelson, and S. Nooshabadi, “CMOS Driver-Receiver Pair for Low-Swing Signaling for Low Energy On-Chip Interconnects,” *IEEE Transactions on Very Large Scale of Integration (VLSI) Systems*, Vol. 17, No. 2, pp. 311–316, February 2009.

[28] M. Ferretti and P. A. Beerel, “Low Swing Signaling Using a Dynamic Diode-Connected Driver,” *Proceedings of the IEEE European Solid-State Circuits Conference*, pp. 369–372, September 2001.

[29] J. C. Garcia, J. A. Montiel-Nelson, and S. Nooshabadi, “Adaptive Low/High Voltage Swing CMOS Driver for On-Chip Interconnects,” *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 881–884, May 2007.

[30] P. Mroszczyk and V. F. Pavlidis, “Ultra-Low Swing CMOS Transceiver for 2.5-D Integrated Systems,” *Proceedings of the IEEE International Symposium on Quality Electronic Design*, pp. 262–267, March 2018.

[31] C. K. Kwon, K. M. Rho, and K. Lee, "High Speed and Low Swing Interface Circuits Using Dynamic Over-Driving and Adaptive Sensing Scheme," *Proceedings of the International Conference on VLSI and CAD*, pp. 388–391, October 1999.

[32] S. Fang and E. Salman, "Low Swing TSV Signaling Using Novel Level Shifters with Single Supply Voltage," Proceedings of the IEEE International Symposium on Circuits and Systems, pp. 1965–1968, May 2015.

[33] F. H. A. Asgari and M. Sachdev, "A Low-Power Reduced Swing Global Clocking Methodology," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 12, No. 5, pp. 538–545, May 2004.

[34] L. E. Frenzel, Jr, "Handbook of Serial Communications Interfaces: A Comprehensive Compendium of Serial Digital Input/Output (I/O) Standards," Elsevier, 2016.

[35] H. Kalargaris and V. F. Pavlidis, "Interconnect Design Tradeoffs for Silicon and Glass Interposers," *Proceedings of the IEEE New Circuits and Systems Conference*, pp. 77–80, June 2014.

[36] S. Jangam *et al.*, "Latency, Bandwidth and Power Benefits of the SuperCHIPS Integration Scheme," *Proceedings of the IEEE Electronic Components and Technology Conference*, pp. 86–94, May 2017.

[37] Predictive Technology Model (PTM) website, `http://ptm.asu.edu/`, 25 June 2021.

[38] M. A. Karim, P. D. Franzon, and A. Kumar, "Power Comparison of 2D, 3D and 2.5D Interconnect Solutions and Power Optimization of Interposer Interconnects," *Proceedings of the IEEE Electronic Components and Technology Conference*, pp. 860–866, May 2013.

[39] S. Wong, G. Lee, and D. Ma, "Modeling of Interconnect Capacitance, Delay, and Crosstalk in VLSI," *IEEE Transactions on Semiconductor Manufacturing*, Vol. 13, No. 1, pp. 108–111, February 2000.

[40] M. Lee *et al.*, "Noise Coupling of Through-Via in Silicon and Glass Interposer," *Proceedings of the IEEE Electronic Components and Technology Conference*, pp. 1806–1810, May 2013.

[41] Y. Kim *et al.*, “Signal and Power Integrity Analysis in 2.5D Integrated Circuits (ICs) with Glass, Silicon and Organic Interposer,” *Proceedings of the IEEE Electronic Components and Technology Conference*, pp. 738–743, May 2015.

[42] S. Borkar, “Role of Interconnects in the Future of Computing,” *Journal of Lightwave Technology*, Vol. 31, No. 24, pp. 3927–3933, December 2013.

[43] G. Kestor, R. Gioiosa, D. J. Kerbyson, and A. Hoisie, “Quantifying the Energy Cost of Data Movement in Scientific Applications,” *Proceedings of the IEEE International Symposium on Workload Characterization*, pp. 56–65, September 2013.

[44] P. Kogge and J. Shalf, “Exascale Computing Trends: Adjusting to the “New Normal” for Computer Architecture,” *Computing in Science and Engineering*, Vol. 15, No. 6, pp. 16–26, November 2013.

[45] D. Pandiyan and C. J. Wu, “Quantifying the Energy Cost of Data Movement for Emerging Smart Phone Workloads on Mobile Platforms,” *Proceedings of the IEEE International Symposium on Workload Characterization*, pp. 171–180, October 2014.

[46] M. R. Stan and W. P. Burleson, “Bus-Invert Coding for Low-Power I/O,” *IEEE Transactions on Very Large Scale of Integration (VLSI) Systems*, Vol. 3, No. 1, pp. 49–58, March 1995.

[47] H. Y To, “An Analysis of Data Bus Inversion: Examining Its Impact on Supply Voltage and Single-Ended Signals,” *IEEE Solid-State Circuits Magazine*, Vol 11, No 2, pp. 31–41, June 2019.

[48] M. Yoon, “Sequence-Switch Coding for Low-Power Data Transmission,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 12, No. 12, pp. 1381–1385, 2004.

[49] C. Kretzschmar, A. K. Nieuwland, and D. Muller, “Why Transition Coding for Power Minimization of on-Chip Buses does not work,” *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 512–517, February 2004.

[50] P. P. Sotiriadis and A. Chandrakasan, “Reducing Bus Delay in Submicron Technology Using Coding,” *Proceedings of the 2001 Asia and South Pacific Design Automation Conference*, pp. 109–114, 2001.

[51] C. L. Su, C. Y. Tsui, and A. M. Despain, “Saving Power in the Control Path of Embedded Processors,” *IEEE Design and Test of Computers*, Vol. 11, No. 4, pp. 24–30, October 1994.

[52] L. Benini *et al.*, “Asymptotic Zero-Transition Activity Encoding for Address Busses in Low-Power Microprocessor-Based Systems,” *Proceedings of Great Lakes Symposium on VLSI*, pp. 77–82, March 1997.

[53] L. Benini *et al.*, “Address Bus Encoding Techniques for System-Level Power Optimization,” *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 861–866, February 1998.

[54] L. Benini *et al.*, “System-Level Power Optimization of Special Purpose Applications: The Beach Solution,” *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 24–29, August 1997.

[55] 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 of Integration (VLSI) Systems*, Vol. 7, No. 2, pp. 212–221, June 1999.

[56] R. Wille *et al.*, “Synthesis of Approximate Coders for On-chip Interconnects Using Reversible Logic,” *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 1140–1143, March 2016.

[57] M. Choudhury *et al.*, “Single-ended Coding Techniques for Off-chip Interconnects to Commodity Memory,” *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 1–6, April 2007.

[58] L. Benini *et al.*, “Architectures and Synthesis Algorithms for Power-Efficient Bus Interfaces,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 19, No. 9, pp. 969–980, September 2000.

[59] Y. Shin, S. Chae, and K. Choi, “Partial Bus-Invert Coding for Power Optimization of System Level Bus,” *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 127–129, August 1998.

[60] C. Kretzschmar, R. Siegmund, and D. Mueller, “Adaptive Bus Encoding Technique for Switching Activity Reduced Data Transfer over Wide System Buses,” *Proceedings of the International Workshop on Integrated Circuit Design, Power*

and Timing Modeling, Optimization and Simulation, pp. 66–75, September 2000.

[61] J. Natesan and D. Radhakrishnan, "Shift invert coding (SINV) for low power VLSI," *Euromicro Symposium on Digital System Design*, pp. 190-194, September 2004.

[62] M. Alamgir, I. I. Basith, T. Supon, and R. Rashidzadeh, "Improved Bus-Shift Coding for Low-Power I/O," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 2940–2943, May 2015.

[63] M. Mamidipaka, D. Hirschberg, and N. Dutt, "Low power address encoding using self-organizing lists," *Proceedings of the IEEE International Symposium on Low Power Electronics and Design*, pp. 188-193, August 2001.

[64] E. Musoll, T. Lang, and J. Cortadella, "Working-Zone Encoding for Reducing the Energy in Microprocessor Address Buses," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 6, No. 4, pp. 568–572, December 1998.

[65] J. Yang and R. Gupta, "FV Encoding for Low-Power Data I/O," *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 84–87, August 2001.

[66] C. Lin, C. Yang, and K. King, "Hierarchical Value Cache Encoding for Off-Chip Data Bus," *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 143–146, October 2006.

[67] T. Lv, J. Henkel, H. Lekatsas, and W. Wolf, "An Adaptive Dictionary Encoding Scheme for SOC Data Buses," *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 1059–1064, March 2002.

[68] M. N. Bojnordi and E. Ipek, "DESC: Energy-Efficient Data Exchange using Synchronized Counters," *Proceedings of the IEEE/ACM International Symposium on Microarchitecture*, pp. 234–246, December 2013.

[69] P. Behnam, N. Sedaghati, and M. N. Bojnordi, "Adaptive Time-based Encoding for Energy-Efficient Large Cache Architectures," *Proceedings of the ACM International Workshop on Energy Efficient Supercomputing*, November 2017.

[70] P. Behnam and M. N. Bojnordi, “STFL: Energy-Efficient Data Movement with Slow Transition Fast Level Signaling,” *Proceedings of the IEEE/ACM Design Automation Conference*, pp. 1–6, June, 2019.

[71] P. Behnam and M. N. Bojnordi, “STFL-DDR: Improving the Energy-Efficiency of Memory Interface,” *IEEE Transactions on Computers*, March 2020.

[72] S. Sarkar, A. Biswas, A. S. Dhar, and R. M. Rao, “Adaptive Bus Encoding for Transition Reduction on Off-Chip Buses With Dynamically Varying Switching Characteristics,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 25, No. 11, pp. 3057–3066, November 2017.

[73] G. Pekhimenko *et al.*, “A Case for Toggle-Aware Compression for GPU Systems,” *Proceedings of the IEEE International Symposium on High Performance Computer Architecture*, pp. 188–200, March 2016.

[74] P. Ramos and A. Oliveira, “Low Overhead Encodings for Reduced Activity in Data and Address Buses,” *Proceedings of the IEEE International Symposium on Signals, Circuits and Systems*, pp. 21–24, 1999.

[75] Y. Aghaghiri, F. Fallah, and M. Pedram, “BEAM: Bus Encoding based on Instruction-Set-Aware Memories,” *Proceedings of the Asia and South Pacific Design Automation Conference*, pp. 3–8, January 2003.

[76] V. Wen, M. Whitney, Y. Patel, and J. D. Kubiatowicz, “Exploiting prediction to reduce power on buses,” *Proceedings of the International Symposium on High Performance Computer Architecture*, pp. 2-13, February 2004.

[77] P. P. Sotiriadis and A. Chandrakasan, “Low power bus coding techniques considering inter-wire capacitances,” *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 507–510, May 2000.

[78] K. Kim *et al.*, “Coupling-Driven Signal Encoding Scheme for Low-Power Interface Design,” *Proceedings of the IEEE/ACM International Conference on Computer Aided Design*, pp. 318–321, November 2000.

[79] M. Ghoneima and Y. Ismail, “Low Power Coupling-Based Encoding for On-Chip Buses,” *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 325–328, May 2004.

[80] Y. Zhang, J. Lach, K. Skadron, and M. R. Stan, “Odd/Even Bus Invert with Two-Phase Transfer for Buses with Coupling,” *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 80–83, August 2002.

[81] P. Subrahmanya, R. Manimegalai, V. Kamakoti, and M. Mutyam, “A Bus Encoding Technique for Power and Cross-talk Minimization,” *Proceedings of the IEEE International Conference on VLSI Design*, pp. 443–448, January 2004.

[82] K. Sohn *et al.*, “A 1.2 V 30 nm 3.2 Gb/s/pin 4 Gb DDR4 SDRAM With Dual-Error Detection and PVT-Tolerant Data-Fetch Scheme,” *IEEE Journal of Solid-State Circuits*, vol. 48, No. 1, pp. 168–177, January 2013.

[83] “DDR4 SDRAM Standard (JESD79-4D),” JEDEC, July 2021.

[84] “Graphics Double Data Rate (GDDR5) SGRAM Standard (JESD212C),” JEDEC, February 2016.

[85] J. Lucas, S. Lal, and B. Juurlink, “Optimal DC/AC Data Bus Inversion Coding,” *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 1063–1068, March 2018.

[86] T. M. Hollis, “Data Bus Inversion in High-Speed Memory Applications,” *IEEE Transactions on Circuits and Systems–II Express Briefs*, Vol. 56, No. 4, pp. 300–304, April 2009.

[87] Y. Song and E. Ipek, “More is Less: Improving the Energy Efficiency of Data Movement via Opportunistic Use of Sparse Codes,” *Proceedings of the IEEE/ACM International Symposium on Microarchitecture*, pp. 242–254, 2015.

[88] D. Lee, M. O’Connor, and N. Chatterjee, “Reducing Data Transfer Energy by Exploiting Similarity within a Data Transaction,” *Proceedings of the IEEE International Symposium on High Performance Computer Architecture*, February 2018.

[89] H. Seol *et al.*, “Energy Efficient Data Encoding in DRAM channels exploiting Data Value Similarity,” *Proceedings of the IEEE/ACM International Symposium on Computer Architecture*, June 2016.

[90] S. Wang and E. Ipek, “Reducing Data Movement Energy via Online Data Clustering and Encoding,” *Proceedings of the IEEE/ACM International Symposium on Microarchitecture*, October 2016.

[91] R. Maddah, S. M. Seyedzadeh, and R. Melhem, "CAFO: Cost Aware Flip Optimization for Asymmetric Memories", *Proceedings of the IEEE International Symposium on High Performance Computer Architecture*, pp. 320–330, February 2015.

[92] G. Laporte, "The Traveling Salesman Problem: An Overview of Exact and Approximate Algorithms," *European Journal of Operational Research*, vol. 59, No. 2, pp. 231–248, June 1992.

[93] LFRic, `https://www.metoffice.gov.uk/research/modelling-systems/lfric`, 25 August 2020.

[94] Modelling and Prediction, `https://www.ecmwf.int/en/research/modelling-and-prediction`, 25 August 2020.

[95] M. Fujino and V. G. Moshnyaga, "An Efficient Hamming Distance Comparator for Low-Power Applications," *Proceedings of the IEEE International Conference on Electronics, Circuits and Systems*, pp. 641–644, September 2002.

[96] R. Mahajan *et al.*, "Embedded Multidie Interconnect Bridge - A Localized, High-Density Multichip Packaging Interconnect," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 9, No. 10, pp. 1952–1962, October 2019.

[97] R. Mahajan *et al.*, "Embedded Multi-die Interconnect Bridge (EMIB) – A High Density, High Bandwidth Packaging Interconnect," *Proceedings of the IEEE Electronic Components and Technology Conference*, pp. 557–565, June 2016.

[98] J. Kim *et al.*, "Silicon vs. Organic Interposer: PPA and Reliability Tradeoffs in Heterogeneous 2.5D Chiplet Integration," *Proceedings of the IEEE International Conference on Computer Design*, pp. 80–87, October 2020.

[99] United Microelectronics Corporation (UMC), `https://www.umc.com/en/home/Index`, 1 April 2022.

[100] Standard Performance Evaluation Corporation, SPEC CPU 2006, `http://www.spec.org/cpu2006`, 1 April 2022.

[101] D. R. Stauffer *et al.*, "High Speed Serdes Devices and Applications," Springer, 2008.

[102] "The I2C-Bus Specification Version 2.1," Philips Semiconductors, January 2000.

[103] "Serial Peripheral Interface (RD1075)," Lattice Semiconductor Corporation, December 2010.

[104] K. Lee, S.–J. Lee, and H.-J. Yoo, "SILENT Serialized Low Energy Transmission Coding for On-Chip Interconnection Networks," *Proceedings of the IEEE/ACM International Conference on Computer Aided Design*, pp. 448–451, November 2004.

[105] S. Ghosh *et al.*, "Data Correlation Aware Serial Encoding for Low Switching Power On-Chip Communication," *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, pp. 124–129, July 2014.

[106] J. Zeng, J.–Y. Zhou, and R.–B. Lin, "Transition Inversion Coding with Parity Check for Off-Chip Serial Transmission," *Proceedings of the IEEE International Conference on Electronics, Circuits and Systems*, pp. 634–637, December 2014.

[107] D. J. Pagliari, E. Macii and M. Poncino, "Approximate Differential Encoding for Energy-Efficient Serial Communication," *Proceedings of the International Great Lakes Symposium on VLSI*, pp. 421–426, May 2016.

[108] P. Stanley–Marbell and M. Rinard, "Reducing Serial I/O Power in Error-Tolerant Applications by Efficient Lossy Encoding," *Proceedings of the ACM/EDAC/IEEE Design Automation Conference*, pp. 1–6, June 2016.

[109] D. J. Pagliari, E. Macii and M. Poncino, "Zero-Transition Serial Encoding for Image Sensors," *IEEE Sensors Journal*, vol. 17, no. 8, pp. 2563–2571, April 2017.

[110] Y. Kim *et al.*, "AXSERBUS: A Quality-Configurable Approximate Serial Bus for Energy-Efficient Sensing," *Proceedings of the IEEE/ACM International Symposium on Low Power Electronics and Design*, pp. 1-6, July 2017.

[111] M. Jackson, R. Budruk, *PCI Express technology : comprehensive guide to generations 1.x, 2.x, 3.0*, MindShare Press, September 2012.

[112] Y. Fan and Z. Zilic, “Accelerating Jitter Tolerance Qualification for High Speed Serial Interfaces,” *Proceedings of the IEEE International Symposium on Quality Electronic Design*, pp. 360–365, March 2009.

[113] A. X. Widmer and P. A. Franaszek,“A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code,” *IBM Journal of Research and Development*, vol. 27, no. 5, pp. 440-451, September 1983.

[114] A. Majumder *et al.*, “A variation tolerant current mode low swing signaling approach for gigascale on-chip interface circuit,” *AEU - International Journal of Electronics and Communications*, vol. 93, pp. 140–149, September 2018.

[115] H. Rezaei, S. A. Moghaddam, and A. Rahmati, “High-speed low-power on-chip global interconnects using low-swing self-timed regenerators,” *Microelectronics Journal*, vol. 58, pp. 76–82, December 2016.

[116] “Stub Series Terminated Logic for 1.8 V (SSTL_18),” JEDEC, September 2003.

[117] N. Binkert *et al.*, “The gem5 Simulator,” *ACM SIGARCH Computer Architecture News*, Vol. 39, No. 2, pp. 1–7, May 2011.

[118] Technical report, “D4.6: Design Summary for die-2- die low-swing interface,” EuroEXA, 2021.

[119] “UltraScale Architecture GTH Transceivers,” UG576 (v1.7.1), Xilinx, August 2021.

[120] High Performance Coax Cable: RF Microwave, High Frequency, Low Loss, Coaxial — Harbour Industries—SS, `https://www.harbourind.com/coaxial-cable/ss-spiral-strip-coaxial-cable-spiral-strip-flexible`, 6 January 2022.

[121] “AFBR-703SDZ 10Gb Ethernet, 850 nm, 10GBASE-SR, SFP+ Transceiver,” *Avago Technologies*, January 2013.

[122] N. Chatterjee *et al.*, “Leveraging Heterogeneity in DRAM Main Memories to Accelerate Critical Word Access,” *Proceedings of the IEEE/ACM International Symposium on Microarchitecture*, pp. 13–24, December 2012.

Appendix A

Simulation of AWR and STTE Encoder Circuits

In this appendix, the methodology followed for the simulation of the designed encoder circuits of the AWR and STTE techniques (Figs. 4.2 and 5.3) is presented. These two circuits include a custom part (delay lines and finish stage) designed in schematic and a fully digital part described in Verilog. Thus, the steps followed to estimate the power of the entire circuit are described.

Two different methods can be used to simulate these encoder circuits, a fully analog simulation and a mixed-signal simulation. The first method provides a more accurate estimation of power, however, the simulation time is higher. The accuracy and the simulation time depend on the size of the digital circuit. The simulation time can be prohibitive in case the digital part is large. Additionally, a fully analog simulation can only be conducted when the schematic views of the standard cells of the used technology are provided. This method is described in Section A.1. On the contrary, the schematic views of standard cells are not required for a mixed-signal simulation. This methodology is explained in Section A.2.

A.1 Analog Simulation

The digital part of the circuit has to be converted to a gate-level description to execute a fully analog simulation. Therefore, the RTL code is synthesised using Synopsys design compiler. The Tcl script used to produce the gate-level netlist, which includes design constraints, is shown in Listing A.1. The produced gate-level netlist is written in <gate_level_netlist_name>.v verilog file.

Listing A.1: Tcl script.

```
set target_library {<library_name >.db}
set link_library { "*" <library_name >.db}

analyze -format verilog <code1.v>
analyze -format verilog <code2.v>
analyze -format verilog <code3.v>

elaborate <top_module_name >

check_design

create_clock -period 4 -name clk \\
[get_ports <clock_name >]
set_clock_uncertainty 0.1 [all_clocks]
set_clock_latency 0.1 -source [get_ports <clock_name >]
set_clock_transition 0.05 [get_clocks clk]

set_input_delay 0.5 -max -network_latency_included \\
-clock clk [all_inputs]
set_input_delay 0.01 -min -network_latency_included \\
-clock clk [all_inputs]

set_output_delay 0.5 -max -network_latency_included \\
-clock clk [all_outputs]
set_output_delay 0.01 -min -network_latency_included \\
-clock clk [all_outputs]

set_load 0.1 -max [all_outputs]
set_load 0.001 -min [all_outputs]

set_driving_cell -min -library <library_name> \\
-lib_cell <buffer_name> -pin Z [all_inputs]
set_driving_cell -max -library <library_name> \\
-lib_cell <buffer_name> -pin Z [all_inputs]

report_clock
```

```
check_timing

set_host_options -max_cores <cores_number>
compile_ultra

check_design

report_timing
report_power
report_area

ungroup -all
change_names -hier -verb -rules verilog

write_file -format verilog -hier -o \\
<gate_level_netlist_name>.v
```

Subsequently, the gate-level netlist must be imported to Cadence Virtuoso, where the custom part of the circuit is designed. Design compiler only provides a connection for the input and output pins of the gates and not for the supply voltage, body, and ground pins. Therefore, these connections must be added manually to the netlist before importing to Virtuoso. The code in Listing A.2 must be added to the input and output declaration of <gate_level_netlist_name>.v. In addition, the code in Listing A.3 has to be added to the port list of each gate instantiation. VDD is the name of voltage supply pin, VSS is the name of the ground pin, and VBP and VBN are the body pins. These names may vary depending on the technology.

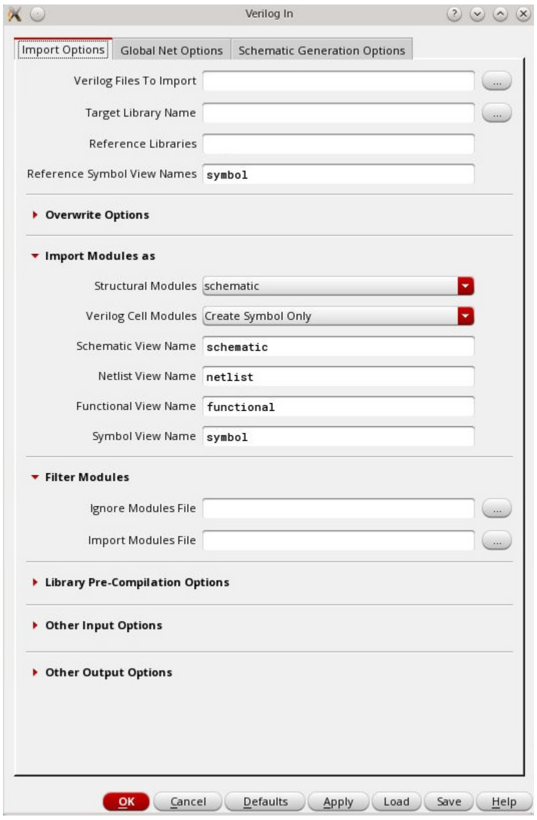
Listing A.2: Code to be added to the port list of the gate level netlist.

```
module <top_module_name> (<list_of_inputs_outputs>, \\
vdd, vss);
input vdd, vss;
```

Listing A.3: Code to be added to the port list of the cell instantiations in the gate level netlist.

```
.VDD(vdd), .VBP(vdd), .VSS(vss), .VBN(vss)
```

The next step is importing the modified gate-level netlist to Virtuoso. From the file menu of Virtuoso the **Verilog In** form, shown in Fig. A.1, is opened by selecting **File**



Verilog Files To Import
Target Library Name
Reference Libraries
Reference Symbol View Names: symbol
Structural Modules: schematic
Verilog Cell Modules: Create Symbol Only
Schematic View Name: schematic
Netlist View Name: netlist
Functional View Name: functional
Symbol View Name: symbol
Ignore Modules File
Import Modules File

Figure A.1: **Verilog In** form.

→ **Import** → **Verilog**. The <gate_level_netlist_name>.v file is selected in the **Verilog Files To Import** field. The target library is selected and the name of the technology library that contains the schematic views of the standard cells is used in the **Reference Libraries** field. The default settings are used in the rest of the fields. In this way, a new cell is created with a schematic and a symbol view.

The custom designed circuit and the imported gate-level circuit must be combined to simulate the whole circuit. Thus, a new cell is created using schematic view. This cell is used as a test circuit to simulate the entire design. An instance of the custom circuit and an instance of the imported circuit are added and connected. Furthermore, a DC voltage source, capacitors connected to the outputs, and power and ground nets are added.

Two verilogA scripts are used to read inputs and write outputs. A symbol is created for each script and the cells are added to the test circuit. The verilogA code used to read 32 words of 64 bits (2,048 bits in total) every 32 cycles is shown in Listing A.4. The verilogA code for writing the outputs of the encoder to a file is shown in Listing A.5.

Listing A.4: VerilogA code for reading inputs from a file.

```
`include "constants.vams"
`include "disciplines.vams"

`define N 2048

module read_enc_inputs(out_word, out_clk);

parameter real logic_low = 0 from [0:inf);
parameter real logic_high = 1.2 from [0:inf);
parameter real period = 2.5n from [0:inf);
parameter real rtime = 20p from (0:inf);
parameter real ftime = 20p from (0:inf);
parameter string file_path = "<input_file >";

output [`N-1:0] out_word;
output out_clk;

electrical [`N-1:0] out_word;
electrical out_clk;

real signal_clk;
real next_clk_rising_edge_time;
real next_clk_falling_edge_time;
real next_reading;
integer  word [`N-1:0];
integer j,file_id;

analog
begin
@(initial_step)
begin
        file_id = $fopen(file_path, "r");
        signal_clk = logic_low;
        next_clk_rising_edge_time = period/2;
        next_clk_falling_edge_time = period;
        next_reading = 2.5*period + 10p;
```

```
end

@(timer(next_clk_rising_edge_time))
begin
        signal_clk = logic_high;
        next_clk_falling_edge_time = \\
        next_clk_rising_edge_time + period/2;
end

@(timer(next_clk_falling_edge_time))
begin
        signal_clk = logic_low;
        next_clk_rising_edge_time = \\
        next_clk_falling_edge_time + period/2;
end

@(timer(next_reading))
begin
        for (j=0; j<2048; j=j+1)
        begin
                $fscanf(file_id, "%d", word[j]);
        end
        next_reading = next_reading + 32*period;
end

V(out_clk) <+ transition(signal_clk, 0, rtime, ftime);
generate i (2047,0) begin
        V(out_word[i]) <+ 1.2*transition(word[i], 0, \\
        rtime, ftime);
end
end
endmodule
```

Listing A.5: VerilogA code for writing outputs to a file.

```
`include "constants.vams"
`include "disciplines.vams"
```

```
‘define N 69

module write_enc_outputs(word,clk,rst);

parameter real logic_high = 1.2;
parameter real logic_low = 0;
parameter real logic_threshold = 0.6;
parameter string file_path =
"<enc_output_file>";

input [‘N-1:0] word;
input clk;
input rst;

electrical [‘N-1:0] word;
electrical clk;
electrical rst;

integer file_id, j;
integer word_bit[‘N-1:0];

analog
begin
generate i ((‘N-1),0)
begin
        word_bit[i] = ( (V(word[i]) > logic_threshold) \\
        ? 1 : 0 );
end

@(initial_step)
begin
        file_id = $fopen(file_path, "w");
end

@(cross( V(clk) - logic_threshold, +1 ))
begin
        if (V(rst) < logic_threshold)
        begin
```

```
                for (j=68; j>=0; j=j-1)
                begin
                        $fstrobe(file_id, "%d ", \\
                        word_bit[j]);
                end
                $fstrobe(file_id, "\n");
        end
end
end

endmodule
```

Once all of the components are added and connected, the simulation environment is launched by selecting **Launch** → **ADE L**. Spectre is used for simulation, therefore, in the simulation environment **Setup** → **Simulator/Directory/Host...** → **simulator** → **sprectre** is selected. Transient analysis is chosen and the duration of the simulation is filled out. The signals to be saved and plotted are selected. To estimate power, the expression **average(abs(IT("/V0/PLUS"))*VT("/vdd!"))** has to be created in the calculator tool, where V0 is the name of the DC voltage source and vdd! is the global power net. The simulation is started by clicking **Simulation** → **Netlist and Run**. The average power is shown in the **Outputs** section when the simulation is completed. In this way, an accurate estimation of the power of the entire circuit is attained.

A.2 Mixed-Signal Simulation

The power of the custom circuit and the digital circuit are calculated separately in this method. This process is recommended when either the digital part is large, thus, a fully analog simulation is slow or the schematic views of standard cells are not provided. In the latter case, a fully analog simulation is not possible. Therefore, the power of the digital circuit is calculated using design compiler and the power of the custom circuit is computed using Spectre.

To estimate the power of the digital circuit, the RTL code is synthesised as described in Section A.1 using the Tcl script, shown in Listing A.1. The command **report_power** provides an estimation of the power consumption. To obtain a more accurate estimation of the dynamic power for a specific input data stream, a Switching Activity Interchange Format (SAIF) file is used. This file provides information about the switching activity of signals and is generated using Synopsys functional verification tool, VCS.

The commands in Listing A.6 are added to the testbench used for functional verification at the point where the user wants to start monitoring the switching activity. For example, the code is added after the command **initial begin** to monitor the switching activity from the beginning of the simulation. The commands shown in Listing A.7 are used to stop monitoring and write a report with the switching activity in file <output_file>.saif. The time unit used by design compiler also has to be provided. For example, $1.0e-9$ is used if the time unit is 1 ns.

Listing A.6: Start monitoring switching activity.

```
$set_toggle_region \\
("<testbench_module_name>.<instance_name>");
$toggle_start;
```

Listing A.7: Writing report with switching activity.

```
$toggle_stop;
$toggle_report("<output_file>.saif", \\
<synthesis_time_unit>, \\
"<testbench_module_name>.<instance_name>");
```

The SAIF file is then used in design compiler to calculate power. The command in Listing A.8 is added to the Tcl script shown in Listing A.1 after setting the link library. This command accurately associates the switching activities in the SAIF file with the corresponding design objects of the produced gate-level netlist. The command in Listing A.9 is added to the TCL script shown in Listing A.1 before compile_ultra to read the SAIF file. The time unit is set to 1 ns.

Listing A.8: Tcl command for mapping SAIF names with gate-level counterparts.

```
saif_map -start
```

Listing A.9: Tcl command for reading SAIF file.

```
read_saif -input <output_file>.saif -instance \\
<testbench_module_name>/<instance_name> \\
-auto_map_names -verbose -unit_base ns -scale 1
```

Mixed-signal simulations are conducted to perform functional verification of the entire circuit as well as calculate the power of the custom circuit. Thus, the custom and the digital circuits must be connected in Virtuoso to simulate the entire design. The custom circuit is simulated using Spectre, while NC-Sim is used for the digital circuit. The digital part described in RTL is imported to Virtuoso. Note that for a fully analog simulation, the gate-level netlist instead of the RTL code of the digital part is imported as discussed in Section A.1.

The RTL code is imported by selecting **File** → **Import** → **Verilog** and the **Verilog In** form appears, depicted in Fig. A.1. The verilog file and the target library fields are filled in. In case other modules are instantiated in the imported RTL code, the verilog files that contain these modules must be imported first. In addition, the libraries, in which these verilog files are imported, must be included in the **Reference Libraries** field. Hence, the top level module must be imported last. The functional option is selected in the **Structural Modules** field and the default settings are used in the rest of the fields. For each imported RTL code, a new cell is created with a functional and a symbol view.

After importing the RTL files, a new cell is created with schematic type which constitutes the test circuit for the entire design as in the analog simulation setup (Section A.1). An instance of the custom circuit and the imported digital circuit are added along with a DC voltage source, power and ground nets, and the two verilogA scripts for reading input data and writing outputs, shown in Listings A.4 and A.5, respectively. An example of a test circuit is shown in Fig. A.2 Verilog scripts instead of verilogA can be used for a mixed-signal simulation if preferred. All the components are connected before the simulation commences.

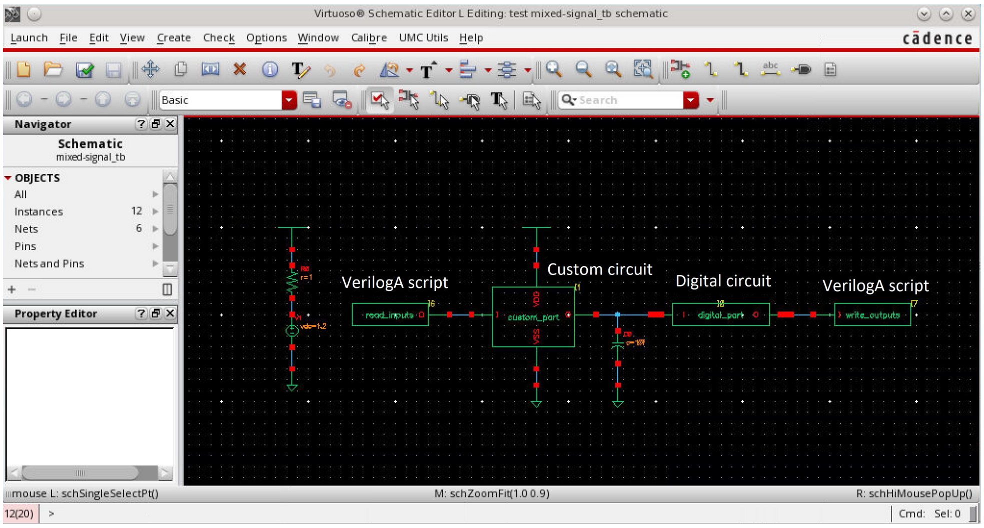
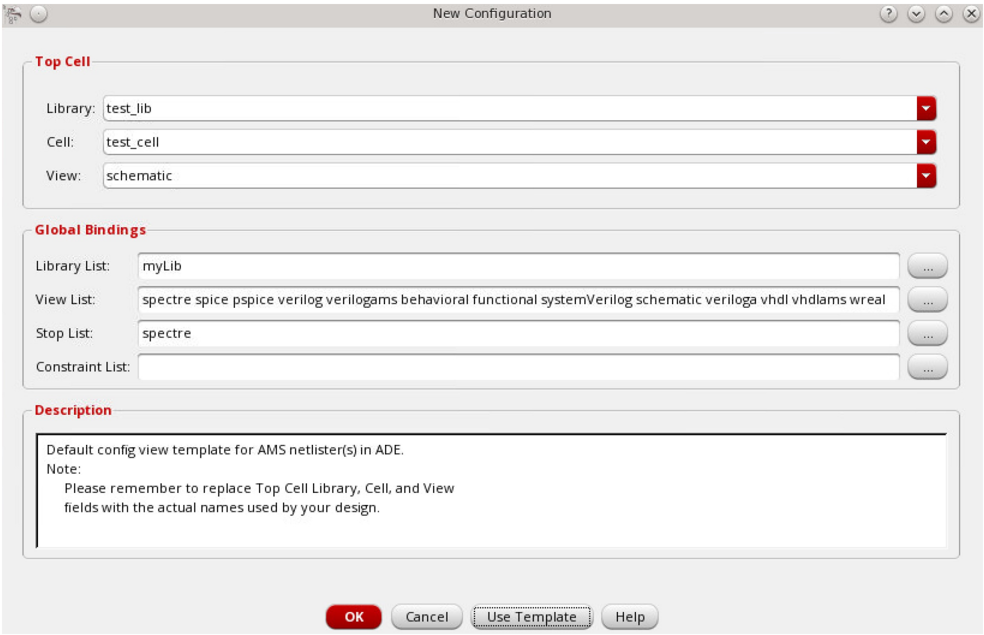


Figure A.2: Schematic view of the mixed-signal test circuit.

To run a mixed-signal simulation a config view of the test circuit has to be created. In this file, the type of simulation and the simulator for each component is configured. Thus, in the **Library Manager** window, a new cell with config type is created with the same name and library as the test circuit. A form named **New Configuration** appears, where schematic view is selected under the **Top Cell** section. Next a template is chosen by clicking on **Use Template** and **AMS** is chosen as **Name** in the **Use Template** form. Then, the **New Configuration** form should look like the form in Fig. A.3. After clicking **OK** on the form another window appears with the name **Virtuoso® Hierarchy Editor: New Configuration**. In this configuration window all the cells of the design along with their library and found view are listed under the **Table View** section, as depicted in Fig. A.4. In case a cell has multiple views, the desired view has to be specified under the **View To Use** column. After ensuring that the correct view is used for each cell, e.g. functional for the imported RTL cells and schematic for custom cells, the window is saved.

The **ADE L** simulation environment is launched from the config file. In the **ADE L** window **Setup** → **Simulator/Directory/Host...** → **simulator** → **AMS** is selected. The next step is to specify connect rules. The INCISIVE tool must be installed and the command in Listing A.10 must be in the cds.lib file, where $AMSHOME is an environment variable pointing to the path that INCISIVE is installed.



Top Cell

Library: test_lib

Cell: test_cell

View: schematic

Global Bindings

Library List: myLib

View List: spectre spice pspice verilog verilogams behavioral functional systemVerilog schematic veriloga vhdl vhdlams wreal

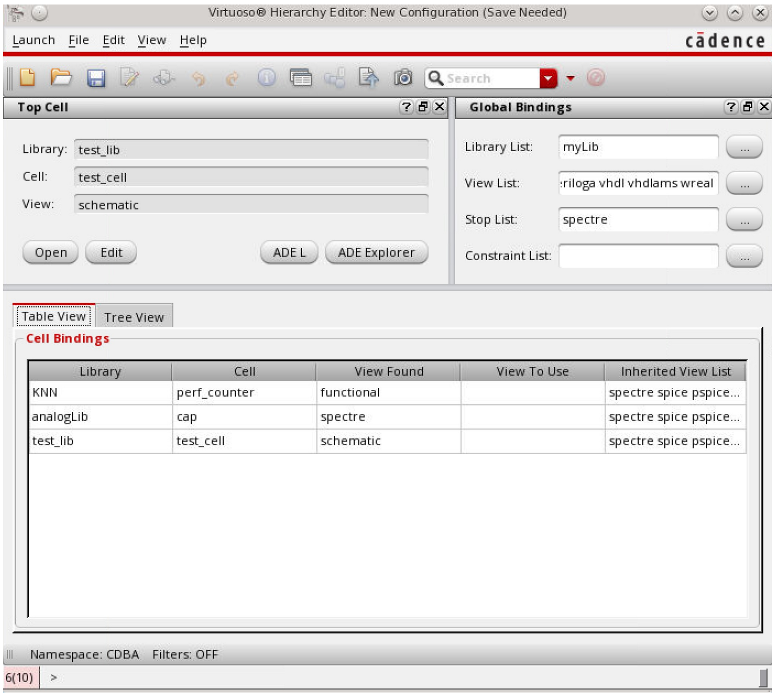
Stop List: spectre

Constraint List:

Description

Default config view template for AMS netlister(s) in ADE.
Note:
Please remember to replace Top Cell Library, Cell, and View
fields with the actual names used by your design.

Figure A.3: **New Configuration** form.



Virtuoso® Hierarchy Editor: New Configuration (Save Needed)
Launch File Edit View Help
cādence
Search
Top Cell
Global Bindings
Open
Edit
ADE L
ADE Explorer
...
Table View
Tree View
Namespace: CDBA Filters: OFF
6(10) >

Library: test_lib
Cell: test_cell
View: schematic

Library List: myLib
View List: :riloga vhdl vhdlams wreal
Stop List: spectre
Constraint List:

Cell Bindings

Library	Cell	View Found	View To Use	Inherited View List
KNN	perf_counter	functional		spectre spice pspice...
analogLib	cap	spectre		spectre spice pspice...
test_lib	test_cell	schematic		spectre spice pspice...

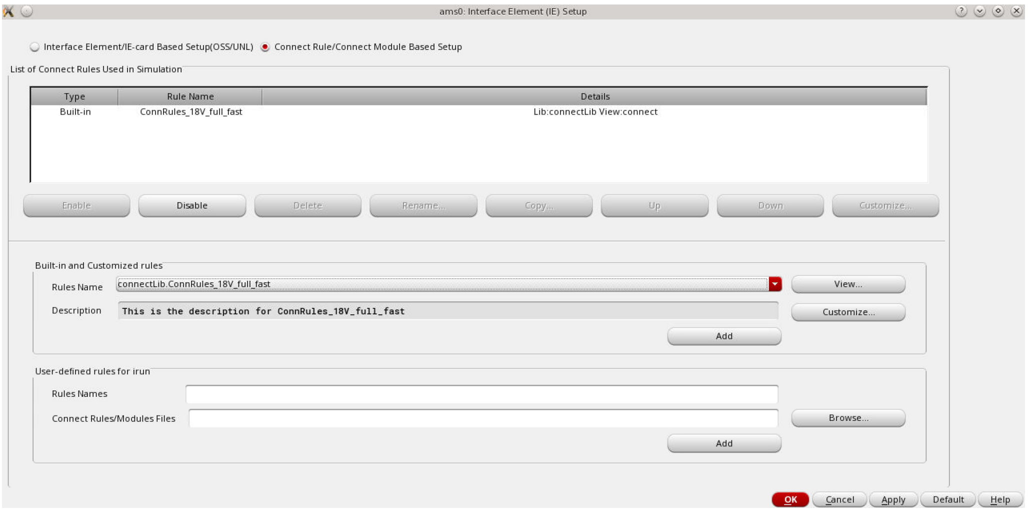
Figure A.4: **Virtuoso® Hierarchy Editor: New Configuration** form.

Listing A.10: Command in cds.lib file.

```
SOFTINCLUDE $AMSHOME/tools/inca/files/cds.lib
```

In the **ADE L** window, **Setup** → **Connect Rules/IE Setup...** is selected and a new window is opened. **Connect Rule/Connect Module Based Setup** is selected. Under the **Built-in and Customized rules** section, there is a list of rules names. The **connectLib.ConnRules_18V_full_fast** rules name is selected and added. Then the form should look like the form in Fig. A.5.

In case the supply voltage is not equal to any of the provided supply voltages (1.8 V, 3 V, 5 V), the connect rules must be customised. The element under **List of Connect Rules Used in Simulation** in the form depicted in Fig. A.5 is clicked and then **Customize...** is selected. A new form appears. In this form, all the modules under **Connect Module Declarations** are selected and a list with all the parameters appears under **Parameters**. The parameter vsup is selected and a new value is specified. In the example of Fig. A.6 the new value is 1.2 V. By clicking **Change**, the value of this parameter changes for all the selected modules. The vthi vtlo must be scaled accordingly as well. The form is closed by clicking **OK**. Note that the name of the customised rules can be changed to something more meaningful, e.g. ConnRules_12V_full_fast.



ams0: Interface Element (IE) Setup

○ Interface Element/IE-card Based Setup(OSS/UNL) ● Connect Rule/Connect Module Based Setup

List of Connect Rules Used in Simulation

Type	Rule Name	Details
Built-in	ConnRules_18V_full_fast	Lib:connectLib View:connect

Built-in and Customized rules

Rules Name: connectLib.ConnRules_18V_full_fast

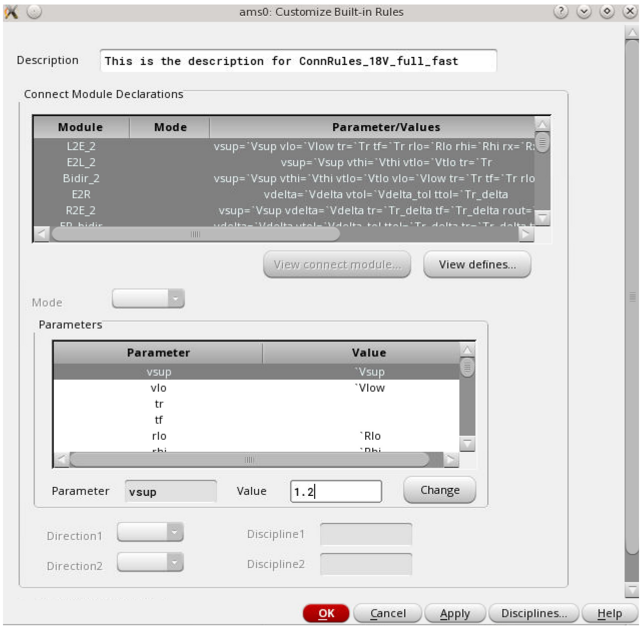
Description: `This is the description for ConnRules_18V_full_fast`

User-defined rules for irun

Rules Names

Connect Rules/Modules Files

Figure A.5: Selecting connect rules.



ams0: Customize Built-in Rules

Description: `This is the description for ConnRules_18V_full_fast`

Connect Module Declarations

Module	Mode	Parameter/Values
L2E_2		vsup=`Vsup vlo=`Vlow tr=`Tr tf=`Tr rlo=`Rlo rhi=`Rhi rx=`R:
E2L_2		vsup=`Vsup vthi=`Vthi vtlo=`Vtlo tr=`Tr
Bidir_2		vsup=`Vsup vthi=`Vthi vtlo=`Vtlo vlo=`Vlow tr=`Tr tf=`Tr rlo
E2R		vdelta=`Vdelta vtol=`Vdelta_tol ttol=`Tr_delta
R2E_2		vsup=`Vsup vdelta=`Vdelta tr=`Tr_delta tf=`Tr_delta rout=

Mode

Parameters

Parameter	Value
vsup	`Vsup
vlo	`Vlow
tr	
tf	
rlo	`Rlo

Parameter: `vsup` Value: `1.2`

Direction1

Direction2

Discipline1

Discipline2

Figure A.6: Customising the supply voltage of connect rules.

The rest of the fields are completed in the same way as in the fully analog simulation, explained in Section A.1. Power is estimated in the same way using the calculator. After the setup is finished, the simulation starts by selecting **Simulation** → **Netlist and Run** in the **ADE L** window. In this mixed-signal simulation, power is only calculated for the custom part which is simulated with Spectre.