AN ABSTRACT OF THE DISSERTATION OF

<u>Jacob Postman</u> for the degree of <u>Doctor of Philosophy</u> in Electrical and Computer Engineering presented on June 11, 2013.

Title: Energy Efficient Communication Across On-Chip Wires in Digital CMOS

Abstract approved: _____

Patrick Yin Chiang

For the past half century, CMOS process scaling has followed Moores law, approximately doubling transistor density every 18 months. While locally routed wires have generally scaled with transistor size, longer wires have scaled at a slower rate and in some cases have grown larger as chip size and complexity have increased. Wires routed for non-local communication now consume a large and increasing portion of the power, thermal and area budgets in CMOS designs. Additionally, dynamic energy expended in driving locally routed wires has become comparable to that expended in logic.

The goal of this research is to investigate methods of reducing the energy required for on-chip communication, primarily through the use of low-voltage swing signaling.

A network-on-chip routing architecture is presented that uses complementary architectural and low-voltage swing signaling techniques to significantly improve the latency, throughput and power of an on-chip network. On-chip signaling circuits are presented that improve the suitability of low-voltage swing signaling for short wire lengths and reduced supply voltages. Finally, a procedure for improving the energy efficiency of wire loads in digital CMOS through the automated insertion of low-voltage swing signaling circuits is presented. [©]Copyright by Jacob Postman June 11, 2013 All Rights Reserved

Energy Efficient Communication Across On-Chip Wires in Digital CMOS

by

Jacob Postman

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Presented June 11, 2013 Commencement June 2014 Doctor of Philosophy dissertation of <u>Jacob Postman</u> presented on June 11, 2013.

APPROVED:

Major Professor, representing Electrical and Computer Engineering

Director of the School of Electrical Engineering and Computer Science

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Jacob Postman, Author

ACKNOWLEDGEMENTS

This work would not have been possible without the support and collaboration of a great many individuals whom I have had the honor of knowing and working with over the past five years.

First and foremost, I would like to thank Professor Patrick Chiang, my major advisor. It is safe to say that I did not quite know what I was getting myself into when I first took his digital design class as an undergraduate. It was his guidance and mentorship that led me to pursue a graduate degree, develop an interest in energy efficient computing, and learn how to navigate the academic landscape. His ability to relentlessly pursue new ideas is an inspiration and his patience, guidance, and curiosity has allowed me to do the same.

Secondly, I would like to thank my graduate committee members, past and present. I am honored and humbled that each of them agreed to participate on my committee. Their input to my education over the past ten years at Oregon State and as members of my committee has been invaluable.

There is a certain mythological quality that becomes associated with individuals so highly regarded as Professor Karti Mayaram. It is both a delight to see him in action and somewhat of a terror to have his efforts directed at oneself, and I am honored to be in such a position as to experience both. Thank you.

I am extremely grateful to Professor Bechir Hamdaoui and Professor Huaping

Liu for taking the time to serve on my committee on relatively short notice. I am indebted to you.

Thank you to Professor Joseph Zaworski for serving as graduate council representative on my committee and for the perspective he has brought to committee meetings.

I would like to thank Professor Ben Lee and Professor Pavan Hanumolu for their participation on my committee. I wish Pavan the absolute best regards as he heads off to his next adventure.

I can not thank Tushar Krishna and Professor Li-Shiuan Peh enough for their collaboration through the first two years of my research. The perspectives, quality of work and quantity of effort that they brought to our research taught me a great deal. Thank you as well to Christopher Edmonds, your work in setting up the initial NoC development environment was instrumental in the success of the project.

Don Heer has been a constant mentor and friend to me from my days as an undergraduate and graduate student at Oregon State. His mentorship as an undergraduate in senior design was a defining time for me and I have valued his continued friendship since then greatly.

Thank you to Roger Traylor, who has been a consistent inspiration throughout my time at Oregon State University. I endeavor to be such an engineer, teacher and human being.

Thank you to Todd Shechter for all the support, often immediately upon request at all hours of the day and night. I have felt many times that nothing would get done without this man. He is the great hidden asset of our department.

Thank you to Professor David Stone who may not even remember me. On the day I received my acceptance letter to the Ph.D. program he gave me advice that has stuck with me in many of the more trying times of graduate school, paraphrased, "Don't try to write your magnum opus. Remember that your Ph.D. is your first work, not your last."

My amazing friends. I am lucky to have such a close group of friends such as yourselves in my life. Aaron Moore, Andre Spycher, Andrew Patterson, Ashley Prater, Beck B. Beck, Brett Gholson, Daniel Donin, Daniel Keese, Douglas Van Bossuyt, Jeff Teigler, Jennie Fairbrother, Joel Baker, Maeve Dempsey, Matt Shuman, Mike Carlsen, Morgan Curtis, Nate Goodson, Ramsi Hawkins, Ray and Claire Anderson, Richie Przybyla, Seth Purkerson, Sharon and Daniel Harada, Stephen Lutz, thank you for helping me to maintain some semblance of sanity, providing encouragement and distraction alike. I hope that any of you that I have forgotten to list here can forgive me.

I have had the privilege of knowing and working with a great group of fellow students at Oregon State. Robert Pawlowski, Joe Crop, Charles Hu, Kangmin Hu, Jiang Tao, Jiao Cheng, Rui Bai, Ben Goska, Tom Ruggeri, Mohsen Nasroulahi, Ashley Diane Mason, Samuel House, Ben Buford, Saeed Pourbagheri, and many others, thank you for the lunches, soccer, and camaraderie that made so much of my time in grad school memorable.

I would like to thank the ARCS Foundation and Leslie and Mark Workman in particular for their generous financial support. As a new graduate student embarking on a long and seemingly uncertain journey, the confidence their support and encouragement instilled in me was and continues to be of extreme benefit to my success.

I would like to thank the National Science Foundation for the opportunities they have afforded me during my time at Oregon State. My experiences in China, made possible by the NSF East Asia Pacific Summer Institute not only provided an unique international research experience, but helped to develop many personal and professional relationships that I will always cherish. Further, the NSF Graduate Research Fellowship made much of this research possible, allowing me to pursue the ideas that I've had over the past few years.

Thank you to my grandparents, aunts, uncles and cousins who have tolerated me working during every trip to visit them and to my sister for her constant support and encouragement.

Thank you to Jenna Moser for putting up with my late nights, erratic schedule, years of living in Corvallis and just me in general for the past seven years.

Finally, and immensely I would like to thank my parents for their unconditional love, support, encouragement and advice. This work for all it is worth is dedicated to them.

TABLE OF CONTENTS

			Page
1	Intro	duction	1
	1.1	On-chip Wire Energy in Digital Integrated Circuits	. 1
	1.2	Networks-on-Chip	. 2
	1.3	LVS transceiver circuits for short on-chip wires	. 4
	1.4	Automated Insertion of Low-voltage Swing Transceivers	. 5
2	SWI	FT: A Token Flow Control Router with Low-voltage Swing Interco	n-
	nect		7
	2.1	Introduction	. 7
	2.2	Motivation	. 12
	2.3	SWIFT Architecture: Bypass Flow Control	. 15
		2.3.1 Related Work on NoC Prototypes	. 15
		2.3.2 Baseline Non-bypass Pipeline Microachitecture	. 16
		2.3.3 Routing with Tokens	. 20
		2.3.4 Flow Control with lookaheads	. 22
		2.3.5 Router Microarchitecture	. 25
	2.4	SWIFT Circuits: Low-swing On-chip Wires	. 27
		2.4.1 Related Work on On-Chip Signaling Techniques	. 30
		2.4.2 Reduced-swing Crossbar	. 34
		2.4.3 Differential Mode Shielding for Crosstalk Reduction	. 36
	2.5	Results	. 39
		2.5.1 The SWIFT Network-on-Chip	. 39
		2.5.2 Baseline NoC	. 40
		2.5.3 Network Performance	. 43
		2.5.4 Power	. 45
		2.5.5 Area	. 50
	2.6	Insights	. 53
		2.6.1 Trade-offs \ldots	. 53
		2.6.2 Technology projections	. 54
	2.7	Conclusions	. 55
		2.7.1 Future Work \ldots	. 56

TABLE OF CONTENTS (Continued)

3	Scala	able V_{dd} Low-swing On-chip Interconnect Circuits	57
	3.1	Introduction	57
	3.2	Overview of On-Chip Links	60
		3.2.1 Conventional Optimally Repeated Buffers	60
		3.2.2 Dual-supply Transmitter	61
		3.2.3 Capacitive Feed-forward Transmitter	63 66
	0.0	5.2.4 Transmission Line based Serial Links	00
	3.3	3.3.1 Operating Principle	07 67
		3.3.2 Transmitter Capacitance C_{TY}	70
		$3.3.3$ Far-End Pre-charge \ldots	71
		3.3.4 Overdriving Signal Swing	72
		3.3.5 Differential Capacitance Imbalancing	73
		3.3.6 Low Voltage Operation	76
	3.4	Receiver Design	77
		3.4.1 Conventional Sense Amplifier Structures	78
	0 F	3.4.2 Degeneration Onset Corrected Receiver Circuit	80
	3.5	Experimental Results	83 84
		$3.5.1$ Hansmitter \ldots	85
	36	Conclusion	87
	5.0		01
4	Auto	omated Insertion of Low-swing On-chip Interconnect Links	90
	4.1	Introduction	90
	4.2	Automated Insertion of LVS Cells	92
		4.2.1 Preparation for Design Evaluation	96
		4.2.2 Replaceable Set Generation	96
		4.2.3 Replaceable Set Analysis	102
	4.3	Low-swing Insertion Procedure	104
	4.4	Results	106
		4.4.1 9-core Network-on-Chip	107
		4.4.2 OpenRISC 1200	109
		4.4.3 Nova H.264	
	4.5	Conclusion	112

TABLE OF CONTENTS (Continued)

		Pa	ıge
	4.5.1	Future work	13
5	Conclusion	1	.17
Bi	bliography	1	18

LIST OF FIGURES

Figure		Page
2.1	Overview of SWIFT NoC versus a baseline NoC and ideal communi- cation. (a) Ideal communication; (b) Baseline NoC; and (c) SWIFT NoC	. 11
2.2	SWIFT NoC Architecture: (a) Routing using tokens (b) Token Dis- tribution (c) Bypass flow-control using lookaheads; and (d) One-cycle router + one-cycle link traversal.	. 21
2.3	Lookahead and Flit payloads.	. 23
2.4	SWIFT router microarchitecture design.	. 24
2.5	SWIFT router non-bypass and bypass pipelines	. 24
2.6	Flow chart for LA-RC.	. 28
2.7	Flow chart for LA-CC.	. 29
2.8	Crossbar and link circuit implementation: (a) Reduced-swing driver (RSD); (b) Bit-slice array crossbar layout; (c) Crossbar bit-slice schematic with link drivers at slice outputs ports; and (d) Clocked sense amplifier receiver (RX).	. 32
2.9	(a) Layout of differential mode link shielding (b) effectiveness of differential mode shielding at reducing crosstalk from full swing ag- gressor logic	. 37
2.10	2x2 SWIFT network prototype overview and die photo overlaid with node 1 layout.	. 40
2.11	Network performance results for fabricated 2x2 chip	. 42
2.12	Network performance results in cycles for 8x8 networks	. 42
2.13	Critical Paths of the SWIFT router and baseline router	. 43
2.14	Network performance results in ns for 8x8 networks	. 46
2.15	Tile Power at (a) high traffic injection (1 packet/NIC/cycle), and (b) low traffic injection (0.03 packets/NIC/cycle) rates	. 47
2.16	Contributions to datapath energy at network saturation	. 48

LIST OF FIGURES (Continued)

Figure		Pa	age
2.17	Circuit contributions to link energy		52
2.18	Post-layout simulated eye at output of 1mm link showing 52% eye opening at 2 GHz.		52
3.1	Data transport vs. operation cost in the Synctium SIMD processor [34]		58
3.2	Analysis of a 2mm x 2mm digital design shows large dynamic load of only a few nets: (a) Distribution of wire lengths; (b) Cumulative distribution of dynamic loading with 65% occurring on nets between 200μ m and 1mm.		59
3.3	(a) Conventional buffered inverter drivers; (b) Dual-supply transmitter [18,40]; and (c) Capacitive feed-forward transmitter [29,37].		62
3.4	Capacitive charge sharing transmitter schematic with tunable volt- age swing and differential signal imbalancing.		64
3.5	a) C_{DAC} TX pre-charge state; b) C_{DAC} transmit state		66
3.6	Wire capacitance connections		67
3.7	Die photo of fabricated chip with capacitive charge sharing trans- mitter based interconnect.		69
3.8	Transmitter signal swing tunability using C_{TX} and energy/bit on a 1mm minimum width wire.		72
3.9	Overdriving signaling voltage V_{swing} beyond the required voltage can improve performance at a small penalty to energy		73
3.10	Eye diagram of output of 1mm wire at 1.66GHz from simulation (a) before and (b) after applying cross-coupled pull-ups, RX side pre-charge and signal overdrive.		74
3.11	Simulated waveforms of differential signal swing tuning with and without a mismatch capacitance applied for asymmetric tuning.		75

LIST OF FIGURES (Continued)

Figure		Page
3.12	Energy per bit (TX, wire, and RX energy) for short interconnect wire transceivers operating at their mean operational signal swing. Zero-one balanced data is applied with a 0.5 transition probability at: a) $V_{dd} = 1.0V$ and b) $V_{dd} = 0.5V$. 76
3.13	9T StrongARM sense amplifier schematic, with common variations shown in gray (b) Dual-tail latch style sense amplifier schematic	. 79
3.14	Proposed receiver circuit with digitally controlled degeneration in- put offset correction and tunable hysteresis.	. 81
3.15	Receiver hysteresis correction across supply voltages	82
3.16	Signal swing on wire across supply voltages as a function of 4-bit C_{DAC} code	. 83
3.17	Capacitive charge sharing transceiver link energy distribution across scaling supply voltage, showing energy optimal operating condition near $V_{dd} = 350 \text{mV}.$. 85
3.18	Receiver input offset variation and correction across V_{dd} with mean offset values annotated. Simulation results are reported from 1000 Monte Carlo runs.	. 86
4.1	Simplified conventional design flow.	. 93
4.2	Proposed digital design flow using automated low-voltage swing cell insertion.	. 94
4.3	Simple conventional net with inverter driver identified for replacement	t. 97
4.4	Conventional net identified for replacement.	. 98
4.5	Simple conventional net with logic driver identified for replacement.	98
4.6	Differential low-swing receiver cell.	. 99
4.7	Two supply voltage differential low-swing driver cell	. 99
4.8	Cell replacement using net splitting on nets identified for replacement	t.101
4.9	Cell replacement without net splitting on nets identified for replace-	. 102

LIST OF FIGURES (Continued)

Figure		Page
4.10 a) ce ne ci re ve u	b) Cells in a replaceable set are identified; b) Existing routing and ells are removed from the design; c) LVS cells are inserted at or the locations of the conventional cells they will replace; d) Spe- ial routes including secondary supply rails are routed; e) LVS wire outes are performed and marked as "don't touch" nets f) Con- rentional routing is performed again for conventional cells in the updated design.	. 105
4.11 a)) Nine core TFC NoC floorplan and; b) placed and routed design.	108
4.12 P	Placed and routed OpenRISC processor core	. 110
4.13 P	Placed and routed Nova H.264 decoder core	. 111
4.14 E	Effect of LVS insertion on maximum operating frequency	. 112

LIST OF TABLES

<u>Table</u>		I	Page
2.1	Comparison of NoC designs	•	14
2.2	Summary of Interconnect Transceivers	•	49
2.3	Area Comparison (Absolute and Percentage)	•	51
3.1	Comparison of Transmitter Circuits	•	77
3.2	Transceiver Comparison Performance Summary		89
4.1	Cell Comparison Summary	•	104
4.2	LVS Insertion Summary		113

Chapter 1 – Introduction

1.1 On-chip Wire Energy in Digital Integrated Circuits

The primary goals of this dissertation are to explore challenges of on-chip communication in digital CMOS designs and develop new techniques suitable to overcome them. Initial research was driven by a desire to find methods of reducing the overhead of on-chip communication. This first manifested in efforts to find more efficient means of passing information between network nodes in an experimental network-on-chip (NoC). A novel on-chip routing architecture, [1], was used to improve router performance while compact low-voltage swing signaling (LVS) circuits were developed to reduce the energy consumed in the wire capacitance dominated loads of router crossbars and core-to-core link data paths.

Use of LVS signaling in the context of this NoC challenged assumptions made in previous on-chip interconnect research that suggested LVS signaling is only useful for long, global on-chip wire routes. Observing this assumption to be invalid led to further investigation to better define under what circumstances LVS signaling could be advantageous. The goal thus became to development even more compact, energy efficient transceivers that target achieving minimal signaling energy and circuit area in order to make the use of LVS signaling on short on-chip wires feasible.

Having demonstrated this feasibility, a remaining challenge was the lack of an efficient method for introducing LVS cells into a conventional design flow without requiring an unreasonable expenditure in manual design effort. This led to the development of an automation procedure for applying low-voltage swing techniques within an existing digital CMOS design flow.

1.2 Networks-on-Chip

While local wires have scaled with transistor size, wires for cross-chip communication have grown larger as chip sizes have increased. Wires routed for non-local communication now consume a large and increasing portion of processor power, thermal and area budgets. This disparity between local and global communication was a driver of the move from single microprocessors to multiple cores that perform computation locally and utilize interconnect circuits for power- and latencyexpensive cross-chip communication [2–4]. As the number of processing nodes increase, the use of ad-hoc interconnects becomes spatially and power prohibitive, severely restricting the scalability of multi-processor designs.

Networks-on-chip (NoCs) have emerged as a processing paradigm to address issues of on-chip communication by providing a framework for scalable multiprocessor networks with a high degree of design regularity. However, NoCs present their own challenges. In recent NoC prototypes, the power, area and latency overhead of the on-chip network and interconnect account for a significant portion of available resources on a chip. For example, 39% of power consumed in each node of a recent Intel NoC prototype [5] is used by the on-chip network.

It is observed that in previous NoC works three primary components of the on-chip networks, data buffers, crossbar switch and core-to-core links, account for the vast majority of network power use. Reducing this overhead in thus necessary for an efficient on-chip network. This is accomplished by implementing a network routing architecture that attempts to minimize data buffer utilization by bypassing their use, helping to ensure that power in the network is used to do the real "work" of moving data across wires from point to point. The second challenge was then to reduce the energy required to move data from point to point which is accomplished by implementing custom crossbar and link blocks that integrate LVS driver and receiver circuits.

Chapter 2 presents the resulting network-on-chip (NoC) which uses complementary routing architecture and low-voltage swing signaling techniques to address the challenges of on-chip communication in the context of an on-chip network. The resulting routing architecture and low-voltage swing crossbar switch and core-to-core links significantly improve the latency, throughput and power of an on-chip network. Insights from this work provide much of the motivation for the development of circuits and automation procedures in the chapters that follow.

1.3 LVS transceiver circuits for short on-chip wires

In experimenting with LVS signaling in the densely routed, but generally "short" wires in an NoC, the assumption made in existing on-chip interconnect signaling literature that long wires were necessary to justify the area, energy, delay and design complexity overheads of LVS signaling circuits was successfully challenged.

Previous on-chip interconnect research focuses on long interconnect wires, addressing design challenges for on-chip signaling that took precedence during the previous two decades. That is that since the mid-1990s, wire delay has been nearly as and often more significant than gate delay, representing a departure from earlier CMOS processes in which wire delay could be generally ignored [2,4]. Largely neglected in earlier interconnect research is the fact that short on-chip wires are also major contributors to design energy consumption. These shorter wires require a different set of design criteria than their long wire counterparts.

From the token flow control NoC it was demonstrated that energy can be reduced using LVS signaling techniques with acceptable area and delay penalties even on fairly short wires. This was particularly true when used in the context of a NoC, where the system architecture is well suited for their use, providing structures with strong regularity and a full pipeline stage for data to traverse crossbar and core-to-core wire routes. The question then became: when is it valuable to use LVS signaling on short wires?

While it was clear that low-voltage signaling can reduce the energy expended in driving wire paths, it also adds significant complexity to the design process, may include latency and area overheads, and introduces reliability concerns into an established design environment with little tolerance for the unknown. Still of concern was the challenge of building low-voltage swing driver and receiver circuits that can be used on short wires without adding significant area or latency bloat. The goal of compact transceiver cells is contrary to the requirement of minimizing device variability which directly effects the reliability and energy efficiency of these circuits. New circuits were necessary to minimize the conflict between circuit foot print, device variation and signaling energy simultaneously in order to build transceiver circuits that are viable choices for use on local intra-core and short core-to-core wires.

Chapter 3 presents transceiver circuits that improve the suitability of lowvoltage swing signaling for short wire lengths and reduced supply voltages. A focus is placed on achieving the minimum energy/bit per mm of wire, compact cell sizes and adaptability to device variation and scaling supply voltages.

1.4 Automated Insertion of Low-voltage Swing Transceivers

Having demonstrated the feasibility of reducing design energy by implementing LVS signaling on short wires, it becomes clear that short wires represent an unexploited opportunity for reducing the energy cost of transporting data on-chip. However, implementation of LVS circuits in the context of a conventional digital design requires significant manual design and verification effort, rendering it impractical. For LVS circuits to useful it is thus necessary to manage the high labor cost of identifying suitable nets and implementing custom low-voltage swing routes.

Chapter 4 presents a design automation methodology for improving the energy efficiency of an arbitrary digital CMOS design through the automated insertion of low-voltage swing signaling circuits. The presented automated insertion procedure operates within the context of a traditional digital design flow and greatly decreases the burden of effort required for a designer to implement LVS signaling.

Chapter 2 – SWIFT: A Token Flow Control Router with Low-voltage Swing Interconnect

2.1 Introduction

Networks-on-chip (NoC) form the on-chip communication backbone between processing cores in many-core processor architectures. Aided by Moores law, increasing core counts have become the answer to meet high performance demands at low power budgets, rather than designing increasingly complex cores. However, for the tens to hundreds of on-chip cores to become a reality, the NoC needs to be highly scalable, and provide low-latency, high-throughput communication in an energyefficient manner. Multi-core research prototypes like MIT's 16-core RAW [6], UT Austin's 40-core TRIPS [7], and Intel's 80-core TeraFLOPS [5] have adopted tilebased homogenous NoC architectures that simplify the interconnect design and verification. The network routers are laid out as a Mesh, and communicate with one another in a packet switched manner. While these designs demonstrated the potential of multi-core architectures, they also exposed the rising contribution of the network towards the total power consumed by the chip. For instance, 39% of the 2.3W power budget of a tile is consumed by the network itself in Intel's TeraFLOPS. If not addressed, this trend will become the major stumbling block for the further scaling of processor counts. This work addresses some of these concerns. NoC routers primarily consist of buffers, arbiters, and a crossbar switch and are interconnected with one another via links. Buffers are required to prevent collisions between packets that share the same links, and are managed by the control path. The crossbar switches and the links form the datapath through which actual data transmission occurs. Prior NoC prototypes have observed that the primary contributers to NoC power are the buffers (31% in RAW [6], 35% in TRIPS [7], 22% in TeraFLOPS [5]), which are typically built out of SRAM or register files; the crossbar switches (30% in RAW [6], 33% in TRIPS [7], 15% in TeraFLOPS [5]); and the core-core links (39% in RAW [6], 31% in TRIPS [7], 17% in TeraFLOPS [5]).

Significant research in NoC router microarchitectures has focused on performance enhancement (low-latency and high-throughput). The MIT RAW [6], and its subsequent commercial version, the Tilera TILEPro64 [8] use multiple physical networks to improve throughput, each interconnected via very simple low-latency routers, presenting one-cycle delay in the router to traffic going straight and twocycles for turning traffic. The UT TRIPS [7] and Intel TeraFLOPS [5] enhance throughput using Virtual Channels $(VC)^1$ [9]. However, VCs increase the arbitration complexity because (i) there are more contenders for the input and output ports of the switch within each router, and (ii) VC arbitration is required in addi-

¹VCs are analogous to turning lanes on highways to avoid traffic leaving different output ports from blocking each other, known as head-of-line blocking.

tion to switch arbitration. This forces all flits² to get buffered, perform successful arbitrations, and then proceed through the switch and the links to the next router, increasing both the delay and the dynamic power consumption within each router. UT TRIPS [7] uses a 3-cycle router pipeline, while Intel TeraFLOPS [5] uses a 5-cycle router pipeline, followed by an additional cycle in the link. Even though these routers could potentially be designed to run slower and finish all tasks within a cycle, flits would still have to be buffered (and perhaps read out in the same cycle upon successful arbitration).

To mitigate the long latencies in these multi-cycle VC routers, recent research in NoCs has suggested ways to perform speculative arbitration [10, 11] or nonspeculative arbitration by sending advanced signals [1,12–14], before the flit arrives, to allow flits to traverse the switch directly, instead of getting buffered. This presents a single-cycle router delay to most flits, and reduces the power consumed in buffer writes/reads. However, none of these designs have been proven in silicon yet.

In this work, we present the SWIFT Network-on-Chip (SWing-reduced Interconnect For a Token-based NoC), which is a novel architecture/circuit co-design that targets power reduction in both the control path (buffers) and the datapath (crossbar and links). On the architecture front, this is the *first* fabricated NoC prototype to target an aggressive one-cycle VC router, with intelligent nonspeculative network flow-control (Token Flow Control [1]) to bypass router buffers. On the circuit front, we identify that datapath traversal is imminent in any router

 $^{^{2}}$ Flits are small fixed-sized sub-units within a packet, equal to the router-to-router link width.

(speculative/non-speculative single or multi-cycle) and custom design a crossbar with low-swing links, a first of its kind, and interface it with low-swing routerto-router links, to provide a reduced-swing datapath. In contrast, previous implementations of low-swing on-chip wires have restricted their use to global signaling [15], [16]. Together, the buffer bypassing flow control, and the reducedswing crossbar and links, provide two-cycle-per-hop on-chip communication from the source to the destination over a reduced-swing datapath. Compared to a state-of-the-art baseline NoC (which we model similar to UT TRIPS [7] and TeraFLOPS [5]) operating at its peak frequency and designed to meet the same saturation throughput, the SWIFT NoC lowers low-load latency by 20% with uniform random traffic. When running both the baseline and SWIFT at the same frequency, the low-load latency savings are 39%. In addition, SWIFT reduces peak control path power by 49%, and peak datapath transmission power by 62%, giving a total power reduction of 38% compared to the baseline running at the same throughput. A high-level overview of the SWIFT NoC is shown in Fig. 2.1.

To experimentally validate the entire NoC co-designed with the proposed microarchitecture and circuit techniques, the SWIFT router is designed in Verilog and implemented using a commercial 90nm, 1.2V CMOS standard cell library. Next, the synthesized router logic is manually integrated with the custom, reduced-swing crossbars and links. While the proposed NoC is designed as an 8x8 2-D mesh, the fabricated test-chip is comprised of a smaller, 2x2 mesh subsection of four routers, verifying the design practicality and validating the network simulation results. At 225MHz, the total power savings of the entire network was measured to be 38%



Figure 2.1: Overview of SWIFT NoC versus a baseline NoC and ideal communication. (a) Ideal communication; (b) Baseline NoC; and (c) SWIFT NoC.

versus a baseline, fully-synthesized, network design operating at the same peak throughput.

The rest of the chapter is organized as follows. Section 2.2 provides the motivation for this work. Section Section 2.3 presents the microarchitecture of the SWIFT router. Section 2.4 describes the reduced-swing crossbar and link circuits. Section 2.5 presents and analyzes our results. Section 2.6 discusses some of our learnings and insights.

2.2 Motivation

The ideal energy-delay of a NoC traversal should be that of just the on-chip links from the source to the destination, as shown in Fig. 2.1a. However, this would require each core to have a direct connection with all other cores which is not scalable. Packet-switched NoCs [9] share links by placing routers at the intersections to manage buffering, arbitration for the crossbar switch, and routing of the flits in order to avoid collisions while the link is occupied, at the cost of increased delay and power, as shown in Fig. 2.1b. In this work, we make two key observations.

First, fabricated NoC prototypes in the past have typically used relatively simple VC router architectures that do not optimize buffer utilization. Data is always buffered without regard to route availability. This dissipates unnecessary energy, and introduces additional latency within a router pipeline. Even if the datapath is available, all packets go through multiple pipeline stages within each router including buffering, routing, switch allocation, and switch (crossbar) traversal; this is then followed by the link (interconnect) traversal. We believe that the router control path can be optimized to utilize the available link bandwidth more efficiently by exploiting adaptive routing and intelligent flow-control, thus enhancing throughput. This architectural enhancement allows many flits to bypass buffering, even at high traffic rates. By obviating the need for buffer reads/writes, fewer buffers are required to sustain a target bandwidth. The result is a reduction in both the router latency and buffer read/write power. This approach has been explored in academic literature [1, 10–13], but has not made it to main stream design.

Second, when buffering is bypassed, the energy consumed by a flit moving through the network will be dominated by both the switch and link traversal. Hence, implementing low-power crossbar and link circuits becomes even more critical for reducing total network power. There have been several recent papers that demonstrate low-swing global links in isolation [15–18], but only [18] has shown integration of low-swing links between cores within a practical SoC application. Previously reported crossbar designs have utilized partial activation [18], segmentation [19], and dynamic voltage scaling [20] to reduce crossbar power, however this is the first work in which low-swing links are utilized within the densely-packed crossbar itself and has recently been extended by automating the generation of low-swing crossbar and link circuits for modular application to NoCs [21] that was outside the scope of this work.

The SWIFT NoC applies both these observations:

(1) We enhance a state-of-the-art router pipeline with the ability to form adaptive bypass paths in the network using tokens. Flits on bypass paths use one-cycle pipelines within the router, while other flits use a 3-stage pipeline. This reduces buffer read/write power in addition to lowering latency.

(2) We custom design the router crossbar and core-to-core links with reducedswing wires to lower data transmission energy.

Characteristic	TILEPro64 [8] (1 sta, 4 dyn nets)	UT TRIPS [7] (OCN)	Intel TeraFLOPS [5]	$Baseline^{\star}$	SWIFT	
Process parameter	rs		1			
Technology	90nm	130nm	65nm	90nm	90nm	
Chip Frequency	700-866 MHz	366 MHz	5GHz	Not Available	400 MHz	
Router Area	Not Available	1.10 mm ²	$0.34 \mathrm{mm}^2$	$0.48^{\dagger} \mathrm{mm}^2$	$0.48 \mathrm{mm}^2$	
Network parameter	ers					
Topology	8x8 mesh	4x10 mesh	8x10 mesh	8x8 mesh	$8x8 \text{ mesh}^{\ddagger}$	
Flit size	32b	138b	39b	64b	64b	
Message Length	1-128 flits	1-5 flits	2 or higher flits	5 flits	5 flits	
Routing	X-Y dimension order	Y-X dimension order	Source	X-Y dimension order	Adaptive	
					(West-first)	
Flow Control	Wormhole	Wormhole	Wormhole	Wormhole	TFC [1]	
		with VCs	with VCs	with VCs		
Buffer Management	Credit-based	Credit-based	On/Off	On/Off	TFC [1]	
Router parameter	S					
Ports	5	6	5	5	5	
VCs per port	0 (5 separate networks)	4	2	2 and 4	2	
Buffers per port	12 (3/dynamic net)	8	32	8 and 16	8	
Crossbar	5x5	6x6	5x5	5x5	5x5	

Table 2.1: Comparison of NoC designs

* Not fabricated, only laid out for comparison purposes.

 † Baseline tile was given same area as SWIFT for place-and-route.

 \ddagger 2x2 mesh for test chip.

2.3 SWIFT Architecture: Bypass Flow Control

Network-on-Chip design primarily involves the following components: Routing, Flow-control and Router Microarchitecture. Routing determines which links the packet traverses from source to destination. Flow-Control determines when the packet can traverse its links. Router Microarchitecture implements the routing and flow-control logic in a pipelined manner and houses the crossbar switch to provide connectivity across different directions. We describe the architectural design of our Baseline NoC as well as the SWIFT NoC in detail, which is based on Token Flow Control (TFC) [1] and provide relevant background.

2.3.1 Related Work on NoC Prototypes

The TILEPro64 [8] from Tilera (inspired by MITs RAW processor [6]) is a 64-core chip that uses 5 separate 8x8 mesh networks. One of these networks is used for transferring pre-defined static traffic, while the remaining four carry variable-length dynamic traffic such as memory, I/O and user-level messages. The TRIPS [7] chip from UT Austin uses two networks, a 5x5 operand network (OPN) to replace operand bypass and L1 Cache buses, and a 4x4 on-chip network (OCN) to replace traditional memory buses. The Intel TeraFLOPS [5] 80-core research chip uses an 8x10 network for memory traffic. Table 2.1 compares the design components for these three prototypes for their multi-flit memory networks. The table also summarizes the design of our state-of-the-art Baseline NoC (designed for comparison purposes similar to UT TRIPS [7] and Intel TeraFLOPS [5]), which will be described later in Section 2.5.2. The three prototypes used text book routers [9] with simple flow control algorithms, as their primary focus was on demonstrating a multi-core chip with a non-bus and non-ring network. In the SWIFT NoC project, we take a step further, and explore a more optimized network design, TFC [1], with reduced-swing circuits in the datapath. We simultaneously address network latency (buffer bypassing, one-cycle router), throughput (adaptive routing, buffer bypassing at all traffic levels using tokens and lookaheads), and power (buffer bypassing, low-swing interconnect circuits, clock gating). The SWIFT NoC optimizations can potentially enhance the simple networks of all these multi-core prototypes.

2.3.2 Baseline Non-bypass Pipeline Microachitecture

A conventional, high-throughput, NoC design is typically packet-switched [9], with routers at the intersection of links to manage buffering, arbitration for the crossbar switch, and routing of the flits. Actions performed by NoC routers are Buffer Write/Read, Routing, Switch Allocation, VC Allocation, and Switch Traversal. These can be pipelined into multiple stages based on implementation decisions, and target frequencies. We implement a highly optimized 3-stage pipeline that builds on the simpler routing architectures in UT TRIPS [7] and Intel TeraFLOPS [5] by leveraging research in shared input buffers [9], lookahead routing [22], seperable allocation [23], and parallel SA/VA [14, 23]. We use this both for our baseline router, and as the non-bypass pipeline in the SWIFT router. These stages are shown in Fig. 2.4. Flits in the SWIFT NoC traverse this pipeline if bypassing fails due to one of the paths shown in the flow chart of Fig. 2.7.

2.3.2.1 Stage 1- Buffer Write (BW)

Incoming flits are written into buffers at each input port, which were implemented with register files generated from the foundry memory generators. These input buffers are organized as a shared pool among multiple VCs [9]. The addresses of the buffers are connected as a linked list. An incoming flit that requires a free buffer obtains the address from the head of the linked list, and every buffer that is freed up appends its address to the tail of the linked list. One buffer is reserved per VC in order to avoid deadlock. Compared to private buffers per VC which can be implemented as a FIFO, our shared buffer design incurs an overhead of storing the read addresses of all flits in the VC state table, but has the advantage of reducing the numbers of buffers required at each port to satisfy buffer turnaround time (minimum number of cycles before which the same buffer can be reused).

2.3.2.2 Stage 1- Switch Allocation-Inport (SA-I)

An input VC is selected from each input port to place a request for the switch. This is implemented using V:1 round robin arbiters at each input port, where V is the number of VCs per port. Round robin arbiters are simple to implement [9] and ensure that every VC gets a chance to send a flit.

2.3.2.3 Stage 2-Switch Allocation-Outport (SA-O)

The winners of SA-I at each input port place requests for their corresponding output ports. As no u-turns are allowed, there can be a maximum of 4 input ports requesting the same output port. These conflicts are resolved using 4:1 matrix arbiters, one for each output port. Matrix arbiters are used for fair allocation of the crossbar output port to all input ports [9]. Separating switch allocation into two phases of simpler arbitration, SA-I and SA-O, is a common approach to satisfy minimum cycle time constraints [23]. Note that a flit may spend multiple cycles in switch allocation due to contention.

2.3.2.4 Stage 2-VC Allocation (VA)

At the end of SA-O, winning head flits are assigned an input VC for their next hop. (Body and Tail flits follow on the same VC). VC allocation in our design is a simple VC selection scheme, based on [14]. Each output port maintains a queue of free VCs at the input port of the next router. A switch request is allowed to be placed for an output port only if the router connected to that output port has at least one free input VC. The winning head flit of a switch output port, at the end of SA-O, picks up the free VC at the head of the queue and leaves. Thus there is no fullfledged arbitration required, simplifying the VC allocation process. If the router receives a signal indicating a free VC from the next router, the corresponding VC id is enqueued at the tail of the queue. VA does not add any extra delay to the critical path since the updating of the queue and the computation of the next free
VC id take place in parallel to SA-O.

2.3.2.5 Stage 2-Buffer Read (BR)

Flits that won SA-I start a pre-emptive read of the buffers, in parallel to SA-O. This is because the register files require all input signals to be ready before the clock edge. If we wait until SA-O declares the winner of the switch output port, BR would have to be pushed to the next cycle, adding latency. The drawback of this is that there are wasted reads from the buffer which would consume power. We address this by biasing SA-I to declare the same input VC as the winner until it succeeds to use the crossbar. This ensures that the same address is read out of BR to avoid additional switching power.

2.3.2.6 Stage 3-Switch Traversal (ST)

The flits that won the switch ports traverse the crossbar switch.

2.3.2.7 Stage 4-Link Traversal (LT)

The flits coming out of the crossbar traverse the links to the next routers.

2.3.3 Routing with Tokens

In the SWIFT NoC, every input port sends a 1-bit token to its neighbor, which is a hint about buffer availability at that port. If the number of free buffers is greater than a threshold (which is three in order to account for flits already in flight), the token is turned ON (by making the wire high), else it is turned OFF. The neighbor broadcasts this token further to its neighbors, along with its own tokens. Flits use these tokens to determine their routes. They try to adapt their routes based on token availability. Fig. 2(a) shows an example of this. The shaded router receives tokens from its N, E and NE neighbors. The incoming flit chooses the East output port over the North output port based on token availability. We implement minimal routing, with a West-first turn rule [9] to avoid deadlocks. Any other adaptive routing algorithm can be used as well.

Each token is forwarded up to 3-hops, via registers at each intermediate router. Tokens are also forwarded up to the network interfaces (NIC) at each router. The number three was fixed based on experiments which can be found in the TFC paper [1]. Intuitively, deeper token neighborhoods do not help much since the information becomes stale with each hop. Moreover, the route is updated at every hop based on the tokens at that router, and the flit only needs to choose between a maximum of two output ports (for minimal routing). Adding more tokens would add more wires and registers without returning much benefit.

For illustration purposes, Fig. 2.2b shows the token distribution relative to the shaded router in a 2-hop neighborhood. 16 tokens enter the shaded router from



Figure 2.2: SWIFT NoC Architecture: (a) Routing using tokens (b) Token Distribution (c) Bypass flow-control using lookaheads; and (d) One-cycle router + one-cycle link traversal.

a 2-hop neighborhood, plus one from the local port. However, West-first routing algorithm allows us to remove tokens from the west neighborhood (except the immediate neighbor) since a packet has to go west irrespective of token availability, reducing the total tokens from (16+1) to (11+1). Similarly, there are a total of (36+1) tokens in a 3-hop neighborhood. Removing the west tokens allows us to reduce this number to (22+1) bits of tokens per router and these act as inputs to

the combinational block that performs route computation.

2.3.4 Flow Control with lookaheads

Conventional flow control mechanisms involve arbitration for the crossbar switch among the buffered flits. Some prior works [8], [1,11–13], [24] propose techniques to allow flits that have not yet arrived to try and pre-allocate the crossbar. This enables them to bypass the buffering stage and proceed directly to the switch upon arrival. This not only lowers traversal latency, but also reduces buffer read/write power. The SWIFT NoC implements such an approach, based on TFC [1], as shown in Fig. 2.2c. TFC [1] has been shown to be better than other approaches like Express Virtual Channels (EVC) [12] as it allows flits to chain together tokens to form arbitrarily long bypass paths with turns, while EVC only allowed bypassing within a dimension upto a maximum of 3-hops. Other approaches to tackle buffer power include adding physical links to bypass intermediate routers [25], or using link repeaters as temporary buffers [8], [24], [26] to reduce buffers within the router. These techniques can enhance energy-delay further at the cost of more involved circuit design.

In the SWIFT NoC, the crossbar is pre-allocated with the help of lookahead signals, which are 14-bit signals sent for each flit, one-cycle before it reaches a router. The implementation of the lookahead generation and traversal to enable a one-cycle advanced arrival will be explained later in Section 2.3.5.

A lookahead is prioritized over locally-buffered flits, such that a local switch

13-9	8	7-5	4	3-1	0	63-3	2-0
Outport	VCid	Y_hops	Y_direction	X_hops	X_direction	Data	Flit_type
Lookahead						Flit	

Figure 2.3: Lookahead and Flit payloads.

allocation is killed if it conflicts with a lookahead. If two or more lookaheads from different input ports arrive and demand the same output port, a switch priority pointer at the output port (which statically prioritizes each input port for an epoch of 20 cycles for fairness) is used to decide the winner and the other lookaheads are killed. The flits corresponding to the killed lookaheads get buffered similar to the conventional case. Since the bypass is not guaranteed, a flit can proceed only if the token from the neighbor is ON (indicating an available buffer).

The lookahead and flit payloads are shown in Fig. 2.3. Lookaheads carry information that would normally be carried by the header fields of each flit: destination coordinates, input VC id, and the output port the corresponding flit wants to go out from. They are thus not strictly an overhead. Lookaheads perform both switch allocation, and route computation.

The SWIFT Flow Control has three major advantages over previous prototypes with simple flow control:

- Lower latency: Bypassing obviates the buffer write, read, and arbitration cycles.
- Fewer buffers: The ability of flits to bypass at all loads keeps the links better utilized while minimizing buffer usage, and reducing buffer turnaround times. Thus, the same throughput can be realized with fewer buffers.



Figure 2.5: SWIFT router non-bypass and bypass pipelines.

• Lower power: Requiring fewer buffers leads to savings in buffer power (dynamic and leakage) and area, while bypassing further saves dynamic switching energy due to a reduction in the number of buffer writes and reads. The SWIFT NoC guarantees that flits within a packet do not get re-ordered. This is ensured by killing an incoming lookahead for a flit at an input port if another flit from the same packet is already buffered. Pt-to-pt ordering is however not guaranteed by SWIFT. This is because lookaheads are prioritized over locally buffered flits, which could result in two flits from the same source to the same destination getting re-ordered if the first one happened to get buffered at some router while the second one succeeded in bypassing that router. Most on-chip network designs use multiple virtual networks to avoid protocol level deadlocks. While request virtual networks often require pt-to-pt ordering for consistency reasons, response virtual networks often do not place this constraint, and TFC can be used within these virtual networks.

A potential network traversal in the SWIFT NoC using tokens and lookaheads is shown in Fig. 2.2d.

2.3.5 Router Microarchitecture

SWIFT tries to present a one-cycle router to the data by performing critical control computations off the critical path. The modifications over a baseline router are highlighted in black in Fig. 2.4. In particular, each SWIFT router consists of 2 pipelines: a non-bypass pipeline which is 3-stages long (and the same as a stateof-the-art baseline), and a bypass pipeline, which is only one-stage and consists of the crossbar traversal. The router pipeline is followed by a one-cycle link traversal.

Fig. 2.5 shows the pipeline followed by the lookaheads to enable them to arrive

a cycle before the flit, and participate in the switch allocation at the next router. All flits try to use the bypass pipeline at all routers. The fallback is the baseline 3-stage non-bypass pipeline.

2.3.5.1 Lookahead Route Compute (LA-RC)

The lookahead of each head flit performs a route compute (LA-RC) to determine the output port at the next router [22]. This is an important component of bypassing because it ensures that all incoming flits at a router already know which output port to request, and whether to potentially proceed straight to ST. We use West-first routing, an adaptive-routing algorithm that is deadlock free [9]. The adaptive-routing unit is a combinational logic block that computes the output port based on the availability of the tokens from 3-hop neighboring routers, rather than use local congestion metrics as indication of traffic. An overview of LA-RC is shown in Fig. 2.6.

2.3.5.2 Lookahead Conflict Check (LA-CC)

The lookahead places a request for the output port in the LA-CC stage, which grants it the output port unless there is a conflict or the output port does not have free VCs/buffers. An overview of LA-CC is shown in Fig. 2.7. LA-CC occurs in parallel to the SA-O stage of the non-bypass pipeline, as shown in Fig. 2.4. A lookahead is given preference over the winners of SA-O, and conflicts between multiple lookaheads are resolved using the switch priority vector described earlier in Section 2.3.4. Muxes connect the input ports of the winning lookaheads directly to the crossbar ports. The corresponding flits that arrive in the next cycle bypass the buffers, as shown in Fig. 2.4. Any flits corresponding to killed lookaheads, meanwhile, get buffered and use the non-bypass pipeline.

2.3.5.3 Lookahead Link Traversal (LA-LT)

While the flit performs its crossbar traversal, its lookahead is generated and sent to the next router. All the fields required by the lookahead, shown in Fig. 2.3, are ready by the end of the previous stage of LA-RC and LA-CC. Fig. 2.5 shows how the lookahead control pipeline stages interact with the flit pipeline stages in order to realize a one-cycle critical datapath within the router.

2.4 SWIFT Circuits: Low-swing On-chip Wires

When flits are able to bypass buffering, SWIFT's pipeline reduces to the one-cycle bypass pipeline comprised of just one-cycle switch traversal (ST), and one-cycle link traversal (LT) that results in two cycles per hop. These two stages correspond to the data movement that flits take through the crossbar switch and through the core-to-core interconnect respectively. Crossbars provide the physical connection between input and output router ports, allowing the flow of data to be directed by routing logic. Core-to-core links provide the communication channel between



Figure 2.6: Flow chart for LA-RC.

adjacent network routers.

Unlike locally connected logic that drive relatively short, locally-routed wires, crossbars and links are primarily composed of tightly-packed, parallel wires that traverse longer distances with close inter-wire coupling. The energy consumed in these components is dominated by the dynamic switching of these wire capacitances rather than transistor gate input capacitances. When flit buffering is bypassed within the router, the energy required to drive these wire capacitances quickly begins to dominate the network power consumption. Low-voltage swing circuit techniques provide an energy-efficient alternative to full-swing digital CMOS signaling that reduces the dynamic energy of interconnect wires without sacrificing



Figure 2.7: Flow chart for LA-CC.

performance.

2.4.1 Related Work on On-Chip Signaling Techniques

Previous works, such as [20], have explored the use of conventional dynamic voltage scaling for reducing crossbar and link power in standard digital logic implementations. Though techniques based on standard logic cells are simpler to implement than custom circuits, they suffer from two major disadvantages. First, in order to reduce the supply voltage, the operating frequency must also be reduced. This limits the savings that can be achieved in signaling energy to times during which the network utilization is low enough that reduced performance can be tolerated. Second, the reduction in voltage signal swing on the interconnect wires is limited to the minimum supply voltage that all attached logic circuits will function at. Thus, custom circuits for low-voltage swing signaling are required in order to reduce wire energy while still maintaining performance.

Custom circuit techniques have targeted improved mW/Gbps on multi-gigabit/second serial links for on-chip wires up to 10mm. In these cases, signal swings as low as 120mV are used on global differential RC wires [15, 16] or transmission line structures [27] either for transporting data directly or for sending control signals to efficiently allocate network buffers and links [28]. However, in order to achieve high performance across long distances, these designs require large transceiver areas of 887μ m² [15] and 1760μ m² [16]. These area overheads make them largely unsuitable for the shorter, wide data busses within NoC core-core links and crossbars.

Alternative energy-efficient, low-swing signaling techniques are needed for the shorter wires and dense transceiver requirements in NoC topologies. Capacitivelycoupled transmitters are explored in [29], achieving speeds of 5-9Gbps across 2mm links. However, the feed-forward capacitor increases transmitter area and requires DC-balanced coding or resistive DC biasing to prevent common-mode drift due to the intrinsic AC-coupling. Current-mode signaling, [17], dissipates constant static current throughout the entire clock period, mitigating its energy-efficiency Figure-of-Merit (FoM). Differential low-voltage swing ($V_{SWING}=0.45V$) is used in [18], but several important differences with our proposed NoC are notable. First, V_{SWING} is 1.5x or more larger than this work, such that receiver offset (due to process variation) and noise coupling from full-swing digital logic below are not problematic. Second, the wire-dominated crossbars in [18] do not utilize low-voltage swing techniques.

To reduce the energy required to drive large capacitive wire loads, reducedvoltage swing signaling was implemented using dual voltage supply, differential, reduced-swing drivers (RSD), Fig. 2.8a, followed by a simple sense-amplifier receiver as shown in Fig. 2.8d. The nominal chip supply voltage is used for the upper voltage supply while the lower supply voltage is generated off-chip. This allows for signal swings to be easily adjusted during testing as the voltage difference between the two supplies and sets the common mode voltage without requiring static power dissipating (except for leakage) in either the driver or receiver. In practice, a voltage supply 0.2V-0.4V below the core logic voltage is often already



Figure 2.8: Crossbar and link circuit implementation: (a) Reduced-swing driver (RSD); (b) Bit-slice array crossbar layout; (c) Crossbar bit-slice schematic with link drivers at slice outputs ports; and (d) Clocked sense amplifier receiver (RX).

available on-chip for the SRAM caches [30] or other components that operate at a reduced supply voltage, and is therefore a small implementation overhead.

Using a second supply offers a number of advantages over single-supply, low swing signaling techniques. First, the energy required to drive the low-swing signal scales quadratically with voltage swing. Second, links are actively driven, making them less susceptible to crosstalk than capacitively-coupled wires, easing constraints on the link routing and any surrounding noisy environment. Finally, the dual supply drivers require no DC biasing of the common mode, compared with feed-forward capacitive coupling as in [29].

While differential signaling approximately doubles the wire capacitance of each bit by introducing a second wire, it removes the necessity of multiple inverter buffer stages for driving long wires and enables the use of reduced voltage swings, resulting in quadratic energy savings in the datapath. Thus, if the energy required to drive a single-ended full swing wire is given by (2.1), then the energy required to drive the differential wire pair at 200mV is approximately given by (2.2).

$$E_{swing=1.2V} = \frac{1}{2}C_{wire}V^2$$
 (2.1)

$$E_{swing=0.2V} = \frac{1}{2} (2C_{wire}) \frac{1}{36} V^2 = \frac{1}{18} E_{swing=1.2V}$$
(2.2)

Hence, reducing the voltage swing from 1.2V to 200mV results in greater than 94% reduction in the energy required to drive the interconnect wire. The link pitch density is limited by the transmitter/receiver layout areas, such that bandwidth/mm is minimally changed for the differential wiring signaling.

The area-efficient sense amplifier receivers, shown in Fig. 2.8d are comprised of near-minimum sized transistors and exhibit approximately 100mV simulated input offset across monte carlo simulations. Occupying $7.8\mu m^2$ and $15.2\mu m^2$ respectively, the same driver and receiver circuits are used in both the crossbar and link designs. Note for comparison, in the technology used, the smallest DFF standard cell available for a full swing implementation occupies $14.8\mu m^2$.

Further reductions in voltage swing requires either larger transistors or offset correction in the receiver in order to overcome the receiver input offset due to process variation. However, at 200mV signal swing, the energy required to drive the wires accounts for only 2% of the energy consumed in the links and crossbar, with the rest being accounted for in the clock distribution, driver input capacitance, sense amplifier and in the case of the crossbar, the port selection signals. Therefore, the resulting increases in receiver area required to further reduce the voltage swing result in diminishing returns for improved energy efficiency.

Datapaths in the crossbar range in length from 150μ m to 450μ m. Differential wires are tightly packed and spaced apart by 0.28μ m, limited not by the minimal wire spacing but by the via size. Link lengths are asymmetric across different ports, with wire routes ranging from 65μ m to 1mm in length. Each of the proposed NoCs routers contains 640 differential pairs and transceivers, making dense wiring, small transceiver sizes, and minimal energy/bit at network operating speed the primary design requirements.

2.4.2 Reduced-swing Crossbar

The simplest and most obvious crossbar layout is to route a grid of vertical and horizontal wires with pass-gates or tri-state buffers at their intersection points, as shown in Fig. 2.8c. While simple, this approach suffers from a number of major disadvantages including poor transistor density, low bandwidth and a bit-to-area relationship.

In practice, higher crossbar speeds and improved density can be achieved in a standard digital synthesis flow using mux-based switches that place buffered muxes throughout the area of the crossbar. For larger crossbars in particular, speed can be further improved by pipelining the crossbar traversal, allowing sections of the wire load to be driven in separate clock cycles. While simple to implement in digital design flows, both approaches introduce additional loading in the form of increased fanout buffering and clock distribution that results in increased power consumption. Furthermore, network latency is also negatively impacted.

The crossbar implemented in our design improves energy efficiency by replacing crossbar wires with low-swing signaling circuits. This approach seeks to drive the large wire capacitances of the crossbar with a reduced voltage swing, without introducing additional buffers or clocked elements. Implemented as a bit-sliced crossbar, each of the 64-bits in each of the five input buses is connected to a one-bit wide, 5-input to 5-output crossbar. An 8x8 grid is then patterned out of 64 of these bit-cell crossbars in order to construct a 64-bit wide, 5x5 crossbar as shown in Fig. 2.8b.

Each crossbar bit-slice consists of 5 Strongarm sense amplifiers receivers (RX), a 5x5 switching matrix (20 single-ended pass-gates), and 5 low-swing transmitters (reduced-swing drivers (RSD)) at each bit-slice output, as shown in Fig. 2.8c. Each of the five reduced-swing differential inputs is driven externally to the input of the crossbar by a RSD, which is connected to the output of the router logic. At the positive clock edge, each of the five low-swing differential inputs is first converted to full-swing logic by the sense amplifier, then driven through a short 6μ m wire via a single-ended pass-gate transistor controlled by the switch arbiter, and finally sent out of the selected output port via the interconnect RSD. In our routing scheme, U-turns are not allowed, so each of the five crossbar input ports can be assigned to one of four possible output ports.

The receiver acts as a sense-amplifier flip-flop with low-swing differential inputs, replacing the flip-flop that would otherwise reside at the output of the crossbartraversal pipeline stage. Like mux-based crossbars, this crossbar topology results in irregular datapaths across the 64b parallel interconnect, requiring that the maximum crossbar speed be bounded by the longest datapath delay through the crossbar.

Full-swing select signals are routed on M1 and M2, differential data signals are routed on minimum width wires on M3-M5, and a separate clock wire is routed on M7 for each port. The clock distribution path is routed to closely match the worst case RC delay of the datapath to match clock skews. The crossbar switch allocator also implements clock gating, activating only the crossbar receive port that is expected to be used for switch/link traversal.

2.4.3 Differential Mode Shielding for Crosstalk Reduction

A major concern for reduced-swing signaling is the increased susceptibility of the low swing signals to crosstalk from a routers full-swing digital logic on lower metal layers. While differential signaling and adjacent wire twisting [31] are effective at rejecting common-mode crosstalk from nearby wires, care must be taken to minimize any asymmetric capacitive coupling from potential aggressor signals.

Complete ground plane shielding below the entire signal path establishes both a well-defined routing environment and provides the best protection against fullswing digital coupling. Unfortunately, this conservative shielding environment also contributes additional capacitance to the already wire-dominated load.

An alternative approach used here is to route shielding on M6 between and in



Figure 2.9: (a) Layout of differential mode link shielding (b) effectiveness of differential mode shielding at reducing crosstalk from full swing aggressor logic.

parallel to the differential pairs on M7 as shown in Fig. 2.9a. In this manner differential mode crosstalk which degrades both timing and signal margin by coupling asymmetrically onto one of the differential wires is significantly reduced. Commonmode crosstalk that couples equally onto both wires is intrinsically rejected by the differential input of the receiver. This approach can improve energy-efficiency by shielding only the differential mode crosstalk while reducing the the per-unit length wire capacitance attributed to wire shielding. Thus, lower capacitance is achieved when compared with full plane ground shielding at the cost of greater commonmode coupling. Hence, the reduced-voltage swing can be decreased close to the minimum level obtained from complete ground shielding.

The cycle alloted to the crossbar and link traversal stages provides sufficient

timing margin for the reduced-swing signals to settle even in the presense of crossstalk induced delay. However, signal integrity of the reduced-swing wires after the value has settled, but immediately before it is sampled by the sense-amp quantizer is a greater concern. Extracted SPICE simulations were used to evaluate the worst-case crossstalk on a 1mm differential link, comparing differential-mode shielding with no signal shielding. A 1mm, full-swing aggressor signal was routed on M5 in parallel to the differential wires of the link on M7, with shielding inserted on M6. The aggressor crossstalk was measured by sweeping the aggressor laterally from the center of the differential pair (where all the crossstalk appears as commonmode), up to a distance of 1.25μ m from the center of the pair. Fig. 2.9b shows a 4.4x reduction in the worst-case aggressor noise coupling using the proposed differential-mode shielding.

Maxwell 2D field-solver was used to more accurately predict the effectiveness of differential-mode shielding at reducing capacitance on the signal line. When modeled with the 90nm process specification this approach shows a 19% reduction in capacitance on the signal wires when compared with a complete grounded plane design. If the signal swing is increased to compensate for the conservative, worstcase differential mode crosstalk estimation of 29mV from the strongly-driven, 1mm parallel aggressor modeled in Fig. 2.9b, the differential mode shielding yields a net energy savings when the signaling voltage swing is below 250mV.

2.5 Results

In this section, we report both the simulated and measured results of the SWIFT NoC prototype, and compare it to a baseline NoC.

2.5.1 The SWIFT Network-on-Chip

The SWIFT NoC parameters are shown in Table 2.1. We chose 8 buffers per port, shared by 2 VCs. This is the minimum number of buffers required per port, with one buffer reserved per VC for deadlock avoidance and six being the buffer turnaround time with on-off signaling between neighboring routers. We used standard-cell libraries provided by ARM Corporation for synthesis. The Place and Route of the router RTL met timing closure at 600MHz. The process technology used and use of standard cells instead of custom layouts, limits our router design from running at GHz speeds, such as in [5]. Note that based on extracted layout simulations, the custom reduced-swing transceivers are designed to operate at 2GHz across 1mm distances with 250mV voltage swing.

We fabricated a 2x2 slice of our 8x8 mesh, as shown in Fig. 2.10. We added on-chip pseudo-random traffic generators at all local 4 network interfaces (L-NIC) and at the 8 unconnected ports at the corners of the mesh, resulting in a total of 12 traffic generating NICs.

In an actual Chip-Multi Processor, a tile consists of both the processing core and a router, with the router accounting for approximately a quarter of the tile area [5]. Since we did not integrate processing cores in this design, we hand placed



Figure 2.10: 2x2 SWIFT network prototype overview and die photo overlaid with node 1 layout.

the routers in order to conserve area. This results in asymptric link lengths in our chip, with drivers sized for the worst-case of 1mm links. A photo of our prototype test-chip overlaid with the layout of node 1 is shown in Fig. 2.10.

Due to the 4mm² network size we used a synchronous clock rather than a globally-asynchronous, locally-synchronous approach as in [5], which was outside the scope of this work. The test chip operates at 400MHz at low load, and 225MHz at high injection rates with a supply of 1.2V. We found that the performance of the test chip was limited from achieving higher clock speeds due to resistive drops in the power routing grid that were not caught prior to fabrication.

2.5.2 Baseline NoC

To characterize the impact of the various features of the SWIFT NoC, we implemented a baseline Virtual Channel (VC) router in the same 90nm technology. A VC router needs to performs the following actions: buffer the incoming flit (Buffer Write or BW), choose the output port (Route Compute or RC), arbitrate and choose an input VC winner for each input port of the switch (Switch Allocation-Inport or SA-I), arbitrate and choose an input port winner for each output port of the switch (Switch Allocation-Outport or SA-O), arbitrate and choose a VC for the next router (VC Allocation or VA), read winning flits out of the buffer (Buffer Read or BR) and finally send the winning flits through the crossbar switch (Switch Traversal or ST) to the link connecting to the next router/NIC.

We design our baseline router similar to the UT TRIPS [7] and Intel TeraFLOPS [5] routers which use VCs. We design a 3-stage router pipeline, the details of which are given in the Appendix. We leverage recent research in shared input buffers [9], lookahead routing [22], separable allocation [23], and VC selection instead of full VC allocation [14] allowing us to optimize the design heavily and perform many of the router actions, which were discussed earlier, in parallel. A target frequency specification of 600MHz or more restricted by 90nm standard cells led us to the 3-stage baseline design. UT TRIPS [7] also uses a 3-stage router pipeline operating at 366MHz. Intel TeraFLOPS [5] uses a 5-stage router pipeline (and an additional stage in the link), but is able to operate at 5GHz³ due to custom blocks instead of standard-cells. The SWIFT NoC is the first NoC prototype demonstrating a one-cycle router pipeline in silicon that bypasses buffering completely to save power. The TileFa TILEPro64 [8] uses 5 separate networks, instead

³Theoretically Intels router could perform all operations within one-cycle for operating frequencies less than 1GHz, but flits would still have to get buffered (and read out the same cycle upon successful arbitration).



Figure 2.11: Network performance results for fabricated 2x2 chip.



Figure 2.12: Network performance results in cycles for 8x8 networks.

of using VCs. As a result, it does not need to perform SA-I or VA. In addition, an XY-routing scheme allows the router to present a one-cycle delay to flits going straight, and two-cycles to flits that are turning. TILEPro64s design philosophy of using physical instead of VCs is a research topic in itself, and thus comparing SWIFT to it quantitatively is beyond the scope of this work.

The non-bypass pipeline in SWIFT is the same as the baseline pipeline, thus allowing us to compare the performance, power and area of the SWIFT and the



Figure 2.13: Critical Paths of the SWIFT router and baseline router.

baseline designs and the impact of our additions (bypass-logic and the reducedswing crossbar).

Once we finalized the baseline router pipeline, we swept the number of VCs and buffers in the baseline such that the peak operating throughput of both the baseline and the SWIFT NoC was the same. This is described in Section V-C. We used two of the configurations for power comparisons, which are described in Section V-D.

2.5.3 Network Performance

Fig. 2.11 demonstrates that the measured and simulated latency curves match, confirming the accuracy and functionality of the chip. The 2x2 network delivers a peak throughput of 113 bits/cycle.

The primary focus of this work was to implement the TFC router in silicon, and integrate reduced-swing circuits in the datapath. Thus, we do not perform a full performance analysis of the design across different traffic patterns. The original TFC paper [1] evaluates TFC for synthetic uniform-random, tornado, bitcomplement and transpose traffic. It also evaluates the impact of TFC with realworld application traces from the SPLASH benchmark suite. In this work, we evaluate the latency-throughput characteristics of the SWIFT NoC with uniform random traffic via on-chip traffic generators to validate the chip functionality. We also use this analysis to set the parameters of an equivalent baseline router (same operating throughput) for a fair comparison of power.

2.5.3.1 Average packet latency (cycles)

We first compare the average packet latencies of the 8x8 SWIFT NoC and the baseline NoC in cycles via RTL simulations. Fig. 2.12 plots the average packet latency as a function of injection rate for SWIFT, and two interesting design points of the baseline: Baseline_2-8 (2VC, 8 buffers) and Baseline_4-16 (4VC, 16 buffers). At low loads, SWIFT provides a 39% latency reduction as compared to the baseline networks. This is due to the almost 100% successful bypasses at low traffic. At higher injection rates Baseline_2-8 saturates at 21% lower throughput. SWIFT leverages adaptive routing via tokens, and faster VC and buffer turnarounds due to bypassing, in order to improve link utilization which translates to higher throughput. Baseline_4-16 matches SWIFT in peak saturation throughput (the point at which the average network latency is three times the no-load latency) in bits/cycle.

2.5.3.2 Average packet latency (ns)

Fig. 2.13 shows that the critical paths of the SWIFT and the baseline routers, which occur during the SA-O stage in both designs, amount to 49 and 36 FO4 delays respectively. We observe that the baseline is 400ps faster, and therefore a dissection of the various components of the critical path provides interesting insights. The primary bottleneck in the SWIFT microarchitecture occurs when SA-O winners need to be quenched, exhibiting an additional 339ps of extra delay. Note that the SWIFT router was designed to perform the SA-O and LA-CC stages in parallel, followed by the removal of SA-O assignments in case they conflicted with the lookahead assignments for the crossbar, in order to maintain higher priority for the lookaheads. In hindsight, if we had allowed both the lookahead and local VC requests to move to the same switch arbiters, relaxing lookahead priority, the critical path would have been significantly reduced.

If we take these critical paths in account, the baseline network can run at a frequency 1.34 times faster than SWIFT. Under this operating condition, Fig. 2.14 shows the performance results of the 8x8 NoCs in nanoseconds, instead of cycles. The SWIFT NoC shows a 20% latency reduction at low-load as compared to the baselines, and similar saturation throughput as Baseline_2-8.

2.5.4 Power

We compare the SWIFT and baseline routers at the same performance (throughput) points for fairness. In Section V-C, we observed that Baseline_4-16 matches



Figure 2.14: Network performance results in ns for 8x8 networks.

SWIFT in saturation throughput if both networks operate at the same frequency. Baseline_2-8 matches SWIFT in saturation throughput if it operates at a higher frequency, or if the networks are operating at low loads. We report power numbers for both Baseline_2-8 and Baseline_4-16 for completeness.

We perform power simulations and measurements at a frequency of 225 MHz and VDD of 1.2V, and the results are shown in Fig. 2.15a and Fig. 2.15b at high and low loads respectively. In both graphs, all 12 traffic generator NICs are injecting traffic. The low-swing drivers were set to 300mV signal swing. Because the L-NIC shares a supply with the router while the crossbar shares a supply with the reduced-swing links, it was not possible to measure each of the blocks separately. Instead, post-layout extracted simulations were performed to obtain an accurate relative breakdown of the power consumption of the different components, which were then compared and validated with chip measurements of the combined blocks.

At high loads, operating at the same frequency, Baseline_4-16 matches SWIFT in performance, but has 49.4% higher buffer (control path) and 62.1% higher cross-



Figure 2.15: Tile Power at (a) high traffic injection (1 packet/NIC/cycle), and (b) low traffic injection (0.03 packets/NIC/cycle) rates.

bar and link (datapath) power. SWIFT (last two bars in Fig. 2.15a) achieves a total power reduction of 38.7% at high injection, with the chip consuming a peak power of 116.5 mW.

At low loads, operating at the same frequency, Baseline_2-8 can match SWIFT in performance, but consumes 24.6% higher power than SWIFT (last two bars in Fig. 2.15b).

Baseline_2-8 and SWIFT have the same VC and buffer resources. SWIFT adds buffer bypassing logic (using tokens and lookaheads), and the low-swing crossbar.



Figure 2.16: Contributions to datapath energy at network saturation.

Thus comparing Baseline_2-8 and the first bar of SWIFT shows us that buffer bypassing reduces power by 28.5% at high loads, and 47.2% at low loads, while the low-swing datapath reduces power by 46.6% at high loads and 28.3% at low loads. These results are intuitive, as buffer write/read bypasses have a much higher impact at lower loads when their success rate is higher, while datapath traversals are higher when there is more traffic.

Lookahead signals allow the crossbar allocation to be determined a cycle prior to traversal, making per-port, cycle-to-cycle clock gating possible. Therefore, clock gating was implemented at each crossbars input port, using the crossbars forwarded clock, reducing the crossbar clock distribution power by 77% and 47%, and sense amplifier power by 73% and 43% at low and high injection, respectively.

The combined average energy-efficiency of the crossbar and link at the network saturation point is measured to be 128fJ/bit, based on chip measurements of the crossbar and link currents, and the count of the received packets. This value is further broken down into component contributions in Fig. 2.16.

2.5.4.1 Link and Crossbar Circuits

	[16]	[29]	Inverter & Flip-flop	Simulated Link	Measured Crossbar & Link
Process	90nm	$90 \mathrm{nm}$	$90 \mathrm{nm}$	90nm	$90 \mathrm{nm}$
Data Rate	4Gbps	9Gbps	1Gbps	2Gbps	$225 \mathrm{MHz}$
Link Length	10mm	$2\mathrm{mm}$	$1\mathrm{mm}$	1mm	$\sim 2 \mathrm{mm}$
TX Area	$1120 \mu m^2$	$\begin{array}{c} \mathrm{C}{=}7.29\mu m^2\\ \mathrm{W}_P{=}1.6\mu m\\ \mathrm{W}_N{=}4\mu \end{array}$	$6.35 \mu m^2$	$7.8\mu m^2$ W _P =2.4µm	$2 \mathrm{x} 7.8 \mu m^2$
RX Area	$1760 \mu m^2$	Not Available	$14.8 \mu m^2$	$15.2 \mu m^2$	$2 \mathrm{x} 15.2 \mu m^2$
Signal Swing	$\sim 200 \mathrm{mV}$	$120 \mathrm{mV}$	1.2V	$250 \mathrm{mV}$	$250 \mathrm{mV}$
Energy/bit	$356 \mathrm{fJ}$	$105 \mathrm{fJ}$	$305 \mathrm{fJ}$	28fJ	$128 \mathrm{fJ}$

 Table 2.2: Summary of Interconnect Transceivers

Fig. 16 shows the distribution of energy consumption in the driver, wire, and receiver of a 1mm link as a function of wire signalling voltage swing. Energy/bit/mm values are averaged across 10,000 cycles of random data at 2GHz.

When not limited by the routers critical path, the reduced-swing transceivers are operational across 1mm wires at 2GHz with a 250mV signal swing (post-layout simulations), achieving a theoretical peak throughput of 640Gbps per crossbar. From post-layout simulations, 28fJ/bit is observed for transmission across 1mm links, including RSD input capacitance, wire energy and sense amplifier. Fig. 17 shows 52% eye closure (240mV) at the sense amplifier input of the 1mm link, representing approximately the worst RC delay-path observed in the fabricated chip. A comparison with previous interconnect works is summarized in Table 2.2.

2.5.4.2 Overheads

The west-first adaptive routing logic used for tokens, the lookahead arbitration logic, and the bypass muxes account for less than 1% of the total power consumed in the router, and are therefore not a serious overhead. This is expected, as the allocators account for only 3% of the total power, consistent with previous NoC prototypes. The control power of the SWIFT NoC is observed to be 37.4% lower than Baseline_4-16, due to fewer buffers and VCs (hence smaller decoders and muxes) required to maintain the same throughput.

The overall control power of SWIFT is approximately 26% of the entire router power, as seen in Fig. 14. This high proportion is primarily due to a large number of flip-flops in the router, many of which were added conservatively to enable the design to meet timing constraints, and could have been avoided by using latches. In addition, the shared buffers require significant state information in order to track the free buffer slots and addresses needed for each flit, adding more flip-flops to the design.

2.5.5 Area

The baseline and SWIFT routers primarily differ in the following hardware components: tokens, lookahead signals with corresponding bypassing logic, buffers, VCs, and crossbar implementation. We estimate the standard-cell area contributions for each of these components and compare them in Table 2.3. For the custom crossbar, we use the combined area of the RSD, transistor switching grid, and sense

Component	SWIFT Area, % of total	Baseline Area, % of total
Tokens	1,235 μm^2 , 0.82%	0
Bypass	10,682 μm^2 , 7.10%	0
Buffers	72,118 μm^2 , 47.94%	81,231 μm^2 , 40.08%
Crossbar	15,800 μm^2 , 10.50%	21,584 μm^2 , 10.64%
Control	$50,596~\mu m^2,~33.63\%$	99,856 μm^2 , 49.27%
Total	$150,431 \ \mu m^2, \ 100\%$	$202,671 \ \mu m^2, \ 100\%$

Table 2.3: Area Comparison (Absolute and Percentage)

amplifier circuits as the metric to compare against the matrix crossbars cell area. The total area of the SWIFT router is 25.7% smaller than the Baseline_4-16. This is an important benefit of the SWIFT design: the 8% extra circuitry, required for implementing tokens and bypassing, results in a 11.2% reduction in buffer area and 49.3% reduction in control area (due to fewer VCs and corresponding smaller decoders and allocators) required to maintain the same peak bandwidth, thereby reducing both area and power.

Note that the SWIFT NoC exhibits some wiring overheads. The 23-bit token signals from the 3-hop neighborhood at each router add 7% extra wires per port compared to the 64-bit datapath. The 14 lookahead bits at each port carry information that is normally included in data flits and are therefore not strictly an overhead. To lessen this overhead, the flit width could have either been shrunk or packets could have been sent using fewer flits, which would further enhance SWIFTs area and performance benefits over the baseline. Finally, while Table 2.3 highlights that the active device area of the reduced-swing custom crossbar is less than that of a synthesized design, differential signaling requires routing twice as



Figure 2.17: Circuit contributions to link energy.



Figure 2.18: Post-layout simulated eye at output of 1mm link showing 52% eye opening at 2 GHz.

many wires as well as potentially requiring an additional metal layer beneath, if shielding is required to mitigate digital crosstalk from below.

2.6 Insights

2.6.1 Trade-offs

The immediate trade-off of TFC versus the baseline that is visible from the results of this work is the critical path. However, we believe that the TFC critical path can be shrunk further by a more optimized LA-CC. A relaxed priority for lookaheads over local requests, or a more optimized priority arbiter would have helped us reduce the 339ps of clear overhead.

The proportion of control power in our router is another aspect that requires additional work. Fig. 2.15a highlighted that about 26% of the router power is in the control, which primarily consists of state within the VCs and the input port. The shared buffer is a major contributor as (1) every VC needs to track where its 5 flits are buffered, and (2) a linked list of free buffers has to be maintained. While we clock-gated the crossbar, all flip flops at the routers were still active irrespective of traffic. Adding clock-gating/power-gating within the router will also help reduce the control power.

In our implementation of TFC [1], we implemented normal tokens. Normal tokens only perform speculative bypass and require VC reservation at every hop to account for the lookahead getting killed and forcing buffering. Guaranteed tokens from the TFC design allow flits to perform a guaranteed bypass and enhance throughput further by enabling flits to bypass intermediate routers that do not have any free buffers. We plan to incorporate this aspect in future work.

2.6.2 Technology projections

We used a 90nm process in this work, which is about 2 generations away from the current state-of-the-art technology at the time of this writing. Moving to a smaller technology node impacts both the microarchitectural aspects of this design as well as the circuits.

In terms of microarchitecture, a newer technology will help shrink the critical path. A synthesis of the same design in IBM 45nm resulted in a critical path of 1ns as opposed to 1.56ns in 90nm. A faster technology node thus enables an intricate pipeline like TFC to be realizable.

Moving to smaller process nodes introduces additional considerations for the interconnect and low-swing circuits. Interconnect energy continues to scale more slowly than logic energy as core-to-core wire lengths remain relatively static and on-chip communication requirements increase, resulting in an increasing benefit for implementing area efficient low-swing interconnect driver and receiver circuits. At the same time, the number of low-swing interconnects may increase dramatically, resulting in the manual implementation of these circuits to require an unreasonable amount of design effort. Automated implementation and validation of these low-swing interconnects will become critical in order to keep up with the pace of digital design flows and is an ongoing area of research.

Finally, device variation is projected to worsen, requiring that sense amplifier circuits either be increased in size relative to logic or implement a compact input offset calibration, [32], in order to provide sufficient sensitivity to low-voltage input
signals without bloating area.

2.7 Conclusions

In this work, we presented a NoC that utilizes low-power architecture and circuit co-design to improve the power, latency and throughput of a NoC. In particular, a token-based smart pipeline by passing scheme, and a reduced-swing crossbar and interconnect together contribute to latency and power improvements in an 8x8 network running uniform random traffic, while requiring half as many buffers as extracted simulations of a baseline NoC using virtual-channel routers. Under uniform random traffic, a reduction of 38% in peak network power is reported when networks are operated at identical frequency conditions, while a 20% reduction in low-load latency is reported when both networks are run at their maximum operating frequencies. Reduced swing circuits achieve 62% power savings in the datapath versus a full-swing, synthesized implementation. Differential mode shielding is also presented as a means to enable protected, reduced-swing signaling over digital logic with less capacitive loading than full ground plane shielding. Many of the architectural and circuit novelties in SWIFT would enhance any NoC router/link design, as SWIFT performs more efficient allocation of network links and buffers, enabling low-power traversal. We hope this work paves the way for more such prototype designs. Demonstrating a SWIFT-like NoC design on a multi-core chip with real application traffic is part of our future work.

2.7.1 Future Work

Network-on-chip architectures will continue to improve, enabling more processing cores and requiring increased data transfer capacity on-chip and improved utilization rates of network crossbar and core-to-core links. As NoC architecture efficiency improves the energy of on-chip wires in these NoCs will become even more critical, accounting for a larger share of the network power.

The regularity of an NoC communication structure presents opportunities for energy improvement that do not necessarily exist in an arbitrary digital design. It is expected that future work will develop more standardized core-to-core interfaces that can be incorporated in a design as macro-cells in the same manner that SRAM and register file IP block generators are currently employed. Strides in this area have already been made in the area of crossbar generation in [21] and continued development of associated cells will allow for further improvements in energy, area and performance of LVS circuit implementations and design incorporation.

Chapter 3 – Scalable V_{dd} Low-swing On-chip Interconnect Circuits

3.1 Introduction

Conventional CMOS technology scaling has continued to improve transistor density, reducing the energy consumed by computation. Unfortunately, the energy and delay required to communicate across on-chip wires has increasingly become a concern [3]. For recent microprocessors, the dynamic energy expended in driving wire capacitance has become comparable to that expended in gate and diffusion capacitances [33]. As a result, the energy cost of transporting data can be greater than that required to perform a computation. For example, in the Synctium SIMD processor [34], a multiply accumulate (MAC) operation on two 16-bit operands requires 200fJ. However, 250fJ is required to transport the resulting 32-bit word to a register, even on relatively short 300μ m wires, Fig. 3.1. Likewise a 16-bit MAC operation in a modern graphic processor may require 1pJ/operation, while moving the result on-chip costs 6pJ/mm/word [35].

Recent publications have attempted to address these signaling limitations of energy and delay for communicating across long on-chip wires [29,36–39]. However,



Figure 3.1: Data transport vs. operation cost in the Synctium SIMD processor [34].

a largely unexplored opportunity for energy savings exists on much shorter wires with relaxed delay constraints. For example, in [40] the crossbar switch contains densely routed datapath signals that are dominated by capacitive loading, but is not in the critical path. Alternatively, extremely energy-constrained sensor applications may use supply voltages operating in near-/sub-threshold [41]. These systems are often delay-bounded by large timing margins required to account for logic delay variation. Due to this large variation and the delay dominance of transistor R_{ON} when operating at low- V_{dd} , the timing contribution from wire delay is minimal [42].

are Despite this dominant research focus on optimizing long on-chip interconnects, the parasitic capacitance of dense short wires is often considerably larger than the gate capacitances of the logic transistors connected to those nets. As a result, these short signal paths are largely overlooked in favor of circuits that improve the energy/delay of only a select few global wires.

In reality, digital designs can exhibit significantly more short, locally-routed



Figure 3.2: Analysis of a 2mm x 2mm digital design shows large dynamic load of only a few nets: (a) Distribution of wire lengths; (b) Cumulative distribution of dynamic loading with 65% occurring on nets between 200μ m and 1mm.

wires than long global routes. Fig. 3.2a plots the distribution of wire lengths for the 2mm x 2mm Synctium SIMD processor [34]. In this design, after analyzing the distribution of wire loads, it is observed that 89% of routed wires in the core are less than 0.2mm in length. However, as shown in Fig. 3.2b, nearly 65% of the design's dynamically-switched capacitive wire load appears on the 7% of nodes with wires between 0.2mm and 1mm in length. These 0.2mm to 1mm intra-core wires thus represent attractive candidates for replacement of conventional full-swing logic with compact low-voltage swing signaling circuits. As a result, the dynamic energy dissipation of the entire digital integrated circuit may be improved. In this work, we present an on-chip low-voltage swing transceiver designed to minimize the energy consumed on short-distance, densely-routed wires. Section 3.2 provides an introduction to recent work in the area of energy efficient and lowlatency on-chip interconnect circuits. Section 3.3 describes and characterizes the proposed charge-sharing transmitter circuit. Section 3.4 presents a compact senseamplifier receiver designed to improve sensitivity without increasing energy per conversion. Section 3.5 presents measured results from a fabricated 65nm chip, and section 3.6 concludes the chapter.

3.2 Overview of On-Chip Links

3.2.1 Conventional Optimally Repeated Buffers

Conventional digital full-swing logic results in the charging and discharging of node capacitances from 0V to V_{dd} , resulting in the average dynamic energy expenditure per transition of (3.1), where C_{load} includes the wire capacitance and the gate capacitance of attached logic cells.

$$E = \frac{1}{2}C_{load}V_{dd}^2\tag{3.1}$$

The wire delay of a single digitally-driven device is proportional to the RC of the wire, which increases quadratically with length. In a conventional signaling scheme, shown in Fig. 3.3a, this is overcome through the insertion of repeater buffers which linearize the relationship of wire delay to wire length [43]. However, the insertion of repeater buffers increases the interconnect area, the energy required to switch the nodes, and the susceptibility to supply noise.

Several previous interconnect studies have targeted improving the energy efficiency and performance of conventional standard-cell logic. These include dynamic voltage and frequency scaling (DVFS) for network-on-chip (NoC) interconnect [44] and charge recycling [45], [46]. Additionally, bus encoding schemes have been proposed to improve signaling performance and energy. These improvements are accomplished by reducing wire transition activity and limiting adjacent aggressor transitions that cause crosstalk-induced link delays [47–50].

3.2.2 Dual-supply Transmitter

Dual-supply transmitters, utilized in [18, 32, 40, 51], are a compact and energyefficient approach for addressing the challenge of generating a low-voltage swing signal. A typical implementation of this transmitter structure is shown in Fig. 3.3b (other variations include using a time-pulsed overdrive voltage [36]). These designs overcome the problem of generating a low-swing voltage on-chip by providing an additional supply voltage off-chip at either $V_{dual} = (V_{dd} - V_{swing})$ or $V_{dual} = (V_{ss} + V_{swing})$. Hence, the energy expended in a transition on a wire is reduced to (3.2), such that the energy used to charge the wire capacitance scales quadratically with voltage swing. This enables a non-linear reduction in signaling energy despite an increase in Miller capacitance caused by the requirement for differential signaling. Even for a nearly doubled capacitance, the reduction of



Figure 3.3: (a) Conventional buffered inverter drivers; (b) Dual-supply transmitter [18,40]; and (c) Capacitive feed-forward transmitter [29,37].

signaling voltage from 1V to 200mV results in an approximately 92% reduction in energy required to drive the wire capacitance, as described by (3.3) and (3.4). Further reductions in interconnect signaling voltage exhibit diminishing returns as the energy required to resolve the signal begins to dominate the wire energy [29, 40].

$$E_{wire} = \frac{1}{2} C_{wire} V_{swing}^2 \tag{3.2}$$

$$E_{swing=1V} = \frac{1}{2}C_{wire}V^2 \tag{3.3}$$

$$E_{swing=0.2V} = \frac{1}{2} (2C_{wire}) \frac{1}{25} V^2 = \frac{2}{25} E_{swing=1V}$$
(3.4)

Unfortunately, the large reduction in wire energy achieved by using dual-supply signaling comes at a design cost. The latency of dual supply driven interconnects is larger than conventional optimally-buffered inverter driven wires and other lowswing signaling techniques. Because of the large delay required to drive them, long wires that would otherwise be strong candidates for low-voltage swing interconnect replacement are often found in the critical path of digital designs. In these cases, low-swing replacement is not appropriate as the energy savings obtained is overshadowed by the exacerbation in delay.

The use of dual-supply circuits requires that a second supply voltage be available from off-die, increasing the system cost and pad count. Once on-chip, this secondary supply voltage consumes significant routing area and bypass capacitance in order to assure sufficient supply noise immunity. Because standard cell power rails cannot be used to provide the lower voltage supply, additional special routes must be used to connect power rings to the dual-supply circuits. This limits the application of low-swing transceiver insertion to well-defined locations such as parallel interfaces between modules and core-core links in networks-on-chip [40].

3.2.3 Capacitive Feed-forward Transmitter

Capacitively-driven links, shown in Fig. 3.3c, operate by AC-coupling a V_{dd} logiclevel signal to the capacitance of the interconnect wire [37,52]. Capacitive coupling



Figure 3.4: Capacitive charge sharing transmitter schematic with tunable voltage swing and differential signal imbalancing.

across a feed-forward capacitor results in a reduced signaling voltage on the wire. Hence, the energy required to drive a wire is limited to the energy required to charge and discharge this coupling capacitor, as well as any static current needed to set the DC bias of the signal wires. This AC coupling also removes any low-frequency components of the signal, resulting in a higher achievable bandwidth. Previous work including [39, 53] have shown energy efficiencies as low as 35.6fJ/mm and 28fJ/mm while reaching data rates of 4Gb/s and 3Gb/s. These energy, latency, and bandwidth characteristics can outperform optimally-buffered inverter driven wires.

In a capacitively-coupled transmitter, the signal swing is set by the ratio of the wire capacitance, C_{wire} , to the coupling capacitor, C_{ffc} , as described by (3.5) and (3.6). DC bias levels are supplied by pull-up and pull-down paths on the wires, providing a well-defined common-mode bias [37], [53]. However, static power consumption from this biasing causes the energy per bit to suffer, particularly for low activity rates. The signaling energy is thus expressed by (3.7).

$$V_{wire} = V_{bias} \pm V_{in} \left(\frac{C_{ffc}}{C_{ffc} + C_{wire}}\right)$$
(3.5)

$$V_{swing} = 2V_{in} \left(\frac{C_{ffc}}{C_{ffc} + C_{wire}}\right)$$
(3.6)

$$E_{ffc} = \alpha \frac{1}{2} V_{dd}^2 \left[C_{ffc} + \frac{C_{wire} C_{ffc}}{C_{wire} + C_{ffc}} \right] + V_{dd} I_{bias} \frac{1}{f}$$
(3.7)

While capacitively coupled transmitters offer low latency, their use requires additional overheads. In order to achieve the highest performance on long wires, significant area is required to implement advanced equalization techniques. In the extreme case of [39], a $1760\mu m^2$ transceiver footprint limits the use of these circuits to only a small number of long, delay sensitive wires.

If wire capacitance is used for AC coupling as in [37], significant capacitor area is required to achieve typical signal swings between 100mV and 200mV. However, if MOS devices are used to provide capacitive coupling as in [29, 38, 39, 52, 53], transistor process variation can imbalance the voltage swing of the differential pair or shift the common mode voltage. Likewise, variation in the DC biasing transistors can shift or separate the DC voltages between wires in a differential pair. This variation sensitivity is of particular concern, when operating at scaled supply voltages in the near-/sub-threshold domain, where: a) the MOS capacitance does not track the wire capacitance and is difficult to predict; and b) device variation is exacerbated.

65



Figure 3.5: a) C_{DAC} TX pre-charge state; b) C_{DAC} transmit state.

3.2.4 Transmission Line Based Serial Links

On-chip serial links exploit the transmission line properties of thick on-chip wires to take advantage of near speed-of-light communications. Point-to-point connections have been demonstrated with data rates upwards of 40Gbps and latencies of approximately 10ps/mm in [54–56].

Unfortunately, wide top-layer wire paths are necessary to achieve the required inductance due to the high-lossiness of on-chip wires, resulting in poor bandwidthdensity. While conventional parallel data busses can provide 1 Gbps/ μ m or more on 10mm wires [37], transmission line structures are typically much less, 0.13



Figure 3.6: Wire capacitance connections.

Gbps/ μ m and 0.57 Gbps/ μ m in [54] and [55] respectively. Additionally, these links require large transceiver areas, such as 2000 μ m² in [55].

3.3 Proposed Capacitive Charge-Sharing Transmitter

In this section, we present a transmitter which enables digitally-tunable low-voltage swing on-chip signaling from a single supply voltage. This is accomplished by incorporating capacitive charge-sharing between the wire capacitance and a capacitor bank at the transmitter to create a programmable differential signal.

3.3.1 Operating Principle

The capacitive charge-sharing transmitter circuit is shown in Fig. 3.4. The transmitter operates by pre-charging the differential wires to the supply voltage when the clock signal is high, as shown in Fig. 3.5a. Concurrently, an internal capacitor, C_{TX} , is pre-charged to ground. On the negative clock edge (Fig. 3.5b), one of the differential pairs is shorted with this grounded TX capacitance. This causes the charge initially stored on C_{wire} to charge-share across the larger capacitance of $C_{TX} + C_{wire}$. This charge sharing between wire and transmitter capacitances causes a voltage to appear across the differential wires.

Using the simple model shown in Fig. 3.5, the output voltage on the active wire creates a differential signal of (3.8) wherein $V_{\overline{out}} = V_{dd}$ and V_{out} is given by (3.9).

$$V_{swing} = V_{\overline{out}} - V_{out} \tag{3.8}$$

$$V_{out} = V_{dd} \times \frac{C_{wire}}{C_{wire} + C_{TX}}$$
(3.9)

In reality, a large coupling capacitance (C_C in Fig. 3.6) exists between the differential pair wires. This coupling results in an effective active wire capacitance of C_{wire} as described by (3.10). Due to the capacitor divider between the wires and the supply, a portion of ΔV_{out} is experienced on the complementary wire as ΔV_{out} in (3.11). Thus, rather than the desired $V_{out} = V_{dd}$ in (3.8), a smaller signal swing results as described by (3.12).

$$C_{wire} = C_w + C_c + \frac{C_c^2 + C_c C_w}{2C_c + C_w}$$
(3.10)

$$\Delta V_{\overline{out}} = \frac{C_c^2}{2C_c + C_w} \Delta V_{out} \tag{3.11}$$

$$V_{\overline{out}} = V_{dd} - \frac{C_c^2}{2C_c + C_w} \left(V_{dd} - V_{out} \right)$$
(3.12)

The cross-coupled pull-up transistors, shown in Fig. 3.4, eliminate any capacitor-



Figure 3.7: Die photo of fabricated chip with capacitive charge sharing transmitter based interconnect.

divided voltage from appearing on the inactive wire by causing it to remain pulled high. However, charge entering the wire from this pull-up path results in a similar capacitance division ΔV_{out} to appear in the reverse direction onto the active wire (3.13). From these two situations, the actual voltage swing appearing at the receiver can be calculated (3.14):

$$\Delta V_{out} \prime = \left[\frac{C_c}{2C_c + C_w + C_{TX}}\right] \left[\frac{C_c}{C_c + C_w}\right] \Delta V_{out}$$
(3.13)

$$V_{swing} = V_{dd} - |\Delta V_{out}| - |\Delta V_{out}|$$
(3.14)

The pre-charge phase of the transmitter results in return-to-zero (RZ) signaling with a transmitted energy per bit of (3.15), which is independent of signaling activity.

$$E_{cap-tx} = \left(\frac{C_{wire}C_{TX}}{C_{wire} + C_{TX}} + C_{rx}\right) V_{dd}^2 \tag{3.15}$$

3.3.2 Transmitter Capacitance C_{TX}

The static capacitance in the transmitter can be replaced with a capacitive DAC, as shown in Fig. 3.4. The ability to adjust the C_{wire}/C_{DAC} ratio allows the signal swing of transmitter cells to be individually optimized and tuned in-situ to each wire route. This is important if there is a need to adapt the operating conditions as the supply voltage is scaled. Additionally, signal swing control can be used to adjust the transmitter swing to tune for actual wire capacitance loading, if the precise wire routing information is not known at the time of cell placement.

In-situ voltage swing tuning can also be used for individual calibration of the signal swing to compensate for the input offset of the associated sense-amplifier receivers. By matching V_{swing} to the input offset of the sense-amplifier receivers, the mean signal swing on a parallel channel interconnect needs only be larger than the mean input offset of the receiver circuits, rather than the worst case receiver offset. This ensures that the signal swing is always sufficiently large without wasting energy through over-margining. In contrast, for a dual-supply link, because the signal swing is set by the difference between two supplies, V_{swing} is shared by all the transceiver pairs, and therefore must be larger than the worst case offset of any of the receivers.

If a design is not expected to change supply voltages, a more efficient device

footprint can be achieved by statically sizing C_{TX} at design time. such that it provides enough signal swing to meet the desired performance and expected receiver offset characteristics. The load capacitance of the routed differential pair wires is extracted from a routed design and the necessary value of C_{TX} is determined based on the desired signal swing using (3.14). By sizing C_{TX} such that the voltage swing is greater than the maximum expected offset voltage, no capacitive DAC is necessary in the transmitter thereby reducing the area footprint and control circuit requirements. However, some energy efficiency is lost because the voltage swing cannot be tuned specifically to match the offset voltage of the associated receiver. In an automated design environment, only a limited number of transmitter cells with different values of C_{TX} are likely necessary in a cell library to cover a wide range of wire lengths.

3.3.3 Far-End Pre-charge

Because pre-charge is performed when the clock is high, signal latency is constrained to only the low clock period of the duty cycle. This places a limit on the maximum operating frequency of the transmitter. To ease this limitation, pre-charge pull-up transistors integrated into the far-end receiver-side can be used to accelerate the pre-charge operation from the opposing end of the wire. This reduces the pre-charge time of the wire by cutting the series resistance in half.



Figure 3.8: Transmitter signal swing tunability using C_{TX} and energy/bit on a 1mm minimum width wire.

3.3.4 Overdriving Signal Swing

The ability to adjust signal swing using C_{TX} as shown in Fig. 3.8 allows latency to be reduced in delay sensitive paths by increasing the signal swing beyond the target voltage. Similar to the over-driving technique used in [36], this results in a significant reduction in the time required to achieve a resolvable signal voltage at the receive end.

Shown in Fig. 3.9, the signal transmission delay on a 1mm minimum width wire, simulated from 10% of the clock transition to 90% of the desired signal value of 250mV, is reduced from 758ps to less than 500ps (250mV signal swing), when C_{TX} is increased in size. If signal resolution is performed after achieving its target signal swing, the charge transfer from the active wire to C_{TX} is limited to that required for the target voltage. This limits the additional energy that would otherwise be consumed using the larger signal swing.

If the overdriven signal is not clocked immediately after reaching the target signal swing, the increased voltage can exacerbate inter-symbol interference (ISI).



Figure 3.9: Overdriving signaling voltage V_{swing} beyond the required voltage can improve performance at a small penalty to energy.

This is of concern if the pre-charge state of the wire is not achieved quickly enough. By using pre-charge pull-up transistors at both the TX and RX cells the pre-charge operation occurs more quickly than the signaling rate. This allows ISI to be limited even with the use of overdriven signaling voltages, provided a large enough portion of the duty cycle is maintained for the pre-charge operation.

When used with cross-coupled pull-ups and RX side pre-charge as described previously, overdriven signaling voltages provides a significant improvement in performance. Shown in Fig. 3.10, signal eye opening at the receiver is improved from 72mV to 232mV at 1.66GHz on a 1mm minimum width wire. Transmitter characteristics are summarized and compared in Table 3.1.

3.3.5 Differential Capacitance Imbalancing

A pair of capacitor mismatch DACs can be applied to the signal wires, as shown in Fig. 3.4. These allow the charge sharing ratios to be adjusted asymmetrically by changing the effective value of C_{wire} . This provides the ability to mismatch



Figure 3.10: Eye diagram of output of 1mm wire at 1.66GHz from simulation (a) before and (b) after applying cross-coupled pull-ups, RX side pre-charge and signal overdrive.

the differential signal swings, which is equivalent to adjusting the RX offset. A transceiver can exploit this property to provide provide offset correction at the transmitter. This is demonstrated in Fig. 3.11, where the center point of the differential signal is shifted based on the capacitance applied to the mismatch DAC.

Applied capacitance mismatch at the transmitter can also be used to compensate for differences in the capacitance of the differentially routed wires, easing constraints on differential routing rules. Relatively small devices can be integrated



Figure 3.11: Simulated waveforms of differential signal swing tuning with and without a mismatch capacitance applied for asymmetric tuning.

into the transmitter to provide correction due to the significantly higher capacitance density of the MOS capacitors compared with the signal wire. This can be done at design time based on calculated capacitance of the routed wires.



Figure 3.12: Energy per bit (TX, wire, and RX energy) for short interconnect wire transceivers operating at their mean operational signal swing. Zero-one balanced data is applied with a 0.5 transition probability at: a) $V_{dd} = 1.0V$ and b) $V_{dd} = 0.5V$.

3.3.6 Low Voltage Operation

Reducing V_{dd} to near-/sub-threshold voltages is appealing for energy optimal computing platforms with low performance requirements. On-chip signaling is particularly attractive for reduced supply voltages, due to the focus on achieving extremely low energy consumption [41].

Gate loading arising from additional devices in the transmitter circuit results in an energy overhead in converting the full swing digital signal to low-voltage. This overhead puts a lower bound on the length of wire for which implementing low-voltage swing signaling produces an improvement in energy efficiency.

Wire capacitance remains constant, when the supply voltage is scaled. However, transistor gate capacitance loading is a function of voltage bias, resulting in a reduced capacitive load at low voltages. Thus, at lower voltages the overhead of the transmitter and receiver circuits is reduced relative to the energy expended in onchip wires. Therefore, the length of wire at which a low-voltage swing transceiver uses less energy than a conventionally driven wire occurs at a shorter wire length at low voltage. These crossover points are plotted in Fig. 3.12.

	Conventional		Feed-forward	Charge Sharing			
	Conver	nionai	Capacitive [29]	Capacitive			
Technology	65nm		90nm	65nm			
	Single Ended		Differential	Differential			
Interconnect	$R_{wire} =$	= 1400Ω	$R_{wire} = 400\Omega$	$\mathbf{R}_{wire} = 1400\Omega$			
	$C_{wire} = 205 fF$		$C_{wire} = 560 fF$	$C_{wire} =$	= 320fF		
Supply	0.5V	1.0V	1.2V	0.5V	1.0V		
Voltage swing	500mV	1V	$120 \mathrm{mV}$	$25-250 \mathrm{mV}$	$50-500 \mathrm{mV}$		
Wire Length	1mm		2mm	1mm			
Energy/bit/mm	30 fJ	$150~{\rm fJ}$	$52.5~{\rm fJ}$	4-24 fJ	20-113 fJ		
Static power	14nA	68nA	$6\mu A$	2.6nA	12nA		
	(leakage)	(leakage)	(bias)	(leakage)	(leakage)		
Data-rate	570Mb/s	$4 \mathrm{Gb/s}$	$9 \mathrm{Gb/s}$	500Mb/s	$1.66 \mathrm{Gb/s}$		

Table 3.1: Comparison of Transmitter Circuits

3.4 Receiver Design

Latch-type sense amplifiers are a fundamental building block of wireline receivers, and have been used widely in recent low-power on-chip interconnect circuits. Their characteristic differential input amplifier stage and regenerative latch result in fast, energy-efficient conversion of low-voltage swings to full digital outputs.

Previous sense-amplifiers for on-chip signaling have targeted long on-chip wires for lengths up to 10mm. These receivers target low conversion latency and low input offset [57], [58]. They often integrate some form of receiver-side equalization to improve link performance [38], [39]. This focus on performance allows for energy and delay improvements on long wires where the energy and area overhead of these circuits can be amortized across a small number of long wires.

Design priorities for short wires, however, are considerably different. In order to achieve energy savings in the entire design, a large number of transceiver circuits are necessary, each of which saves a relatively small amount of energy. Thus, decreasing the area footprint and energy per conversion are the most critical constraints, restricting the use of existing techniques.

3.4.1 Conventional Sense Amplifier Structures

3.4.1.1 Standard 9T/11T strong arm structure

A conventional 9-transistor sense amplifier is shown in Fig. 3.13a, including several common variations of this structure. The sense amplifier acts as a clocked low-to-high converter, providing the interface between the low-voltage swing input and the rail-to-rail digital operation of proceeding blocks. This structure is often used due to its ability to easily integrate into digital designs by providing the functionality of a low-voltage swing input DFF [59]. Additionally, it provides low-power, no static



Figure 3.13: 9T StrongARM sense amplifier schematic, with common variations shown in gray (b) Dual-tail latch style sense amplifier schematic.

power dissipation and high performance when compared with other methods [51].

3.4.1.2 Dual-Tail

A double-tail latch type sense amplifier [57], exhibits improved performance isolation from the common-mode voltage value at the input. The input stage amplification and the non-linear positive feedback regeneration are separated into two stages, thus reducing the device stacking, shown in Fig. 3.13b. Due to the stage isolation, a small current can be set in the tail transistor of the input stage which results in a lower input offset. A large tail current can then still be used in the second stage, allowing low latency during regenerative latching.

3.4.2 Degeneration Offset Corrected Receiver Circuit

The proposed receiver, shown in Fig. 3.14, is based on the conventional 9 transistor sense-amplifier, but utilizes near minimum-sized transistors. This limits the capacitance of internal receiver nodes and allows for a compact area footprint. Unfortunately, minimum-sized devices introduce large variations in the receiver input offset.

In conventional receivers (Fig. 3.13), offset compensation is achieved either by using large devices to minimize variation or by tuning a capacitive or current trimming DAC [60]. These methods negatively impact the receiver footprint and increase capacitive loading on internal receiver nodes, degrading energy per conversion. Offset calibration that is both compact and does not contribute to receiver loading is thus desired to enable acceptable receiver input sensitivity.

In the proposed receiver, the clocked-tail transistor is split to produce a pseudo-



Figure 3.14: Proposed receiver circuit with digitally controlled degeneration input offset correction and tunable hysteresis.

differential structure. Digital trimming is then applied, providing controlled imbalance of the receiver source degeneration through the tail nodes without loading the receiver's internal regeneration nodes.

Receiver offset is calibrated by setting the input signals to the same potential, clocking the receiver and latching the Q and QB outputs to a pair of NMOS in the the offset-correction pull-down network. In this manner, only a single cycle is



Figure 3.15: Receiver hysteresis correction across supply voltages.

required per "bit" of calibration. The capacitive transmitter is particularly well suited to this method of receiver calibration due to its RZ signaling. When the transmitter is in the pre-charge state, receiver inputs are tied together through V_{dd} providing valid calibration inputs.

3.4.2.1 Tunable Hysteresis

Hysteresis in sense amplifier structures increases the minimum resolvable signal swing, and is additive to any input offset that occurs due to device variation. Additionally, sense-amplifier hysteresis can change dramatically as the supply voltage is lowered, particularly near the low-end of the input sensitivity range.

In this work, the same degeneration imbalance control used for offset trimming is incorporporated as 3-bit hysteresis control, using a feedback transistor below a second parallel pull-down network (Fig. 3.14). This tunable hysteresis can be used to correct for large changes in receiver hysteresis across a large dynamic range of supply voltages (Fig. 3.15). Alternatively, this degeneration control can be used to



Figure 3.16: Signal swing on wire across supply voltages as a function of 4-bit C_{DAC} code.

implement a compact and energy-efficient Decision Feedback Equalitation (DFE) by controlling the hysteresis in response to previous received signal value.

3.5 Experimental Results

Three on-chip interconnect systems were fabricated in 65nm CMOS, demonstrating the efficacy of the proposed low-swing building blocks for short to medium length wires (Fig. 3.7). Dynamic energy was measured using a picoammeter, while leakage energy was simulated because current measurements could not be decoupled from the internal leakage current of the power clamps, ESD, and decoupling capacitances. All measurements include the calibration logic/latches, clocking, and signal input buffers that directly drive the transceiver circuit. Transceiver characteristics are summarized and compared with recent works in Table 3.2.

3.5.1 Transmitter

A 1mm link was built implementing the capacitive charge-sharing transmitter and a receiver with ten bits of digital degeneration offset calibration for minimum input signal resolution.

Circuits were characterized for performance and energy-efficiency, with V_{dd} ranging from sub-threshold (0.2V) to super-threshold (1V). At startup, receiver offset was digitally calibrated and a 4-bit programmable capacitive DAC was used to set the signal swing for the operating condition under test (Fig. 3.16). Operating characteristics were collected across supply voltages at the minimal energy operating conditions that were required in order to achieve a BER < 10^{-9} at each supply voltage (Fig. 3.17). In order to provide a well-defined wire capacitance, signal wires are fully ground-shielded. Total transmitter area was a relatively large 122μ m, due in part to overly-conservative large dynamic tuning of the charge sharing capacitor DAC. These can be reduced to half this size with more precisely targeted voltage swing ranges, based upon the post-layout extracted values of wire capacitance.

The energy-optimal operating point was found at $V_{dd} = 350$ mV, with a maximum operating frequency of 5MHz using a 40mV input voltage, resulting in an energy-efficiency of 8.4fJ/bit/mm. At $V_{dd} = 500$ mV, with a 30MHz clock frequency and 75mV voltage swing, the energy per bit was measured to be 10.9fJ/bit/mm.

With a super-threshold supply voltage of $V_{dd} = 1V$, measured performance is limited to 622Mbps, due to small input transistors that are sized for minimal energy at low voltage. Additionally, the use of minimum width wires severely restrict



Figure 3.17: Capacitive charge sharing transceiver link energy distribution across scaling supply voltage, showing energy optimal operating condition near $V_{dd} = 350 \text{mV}$.

the performance at super-threshold supply voltages. However, when targeting minimum energy at low supply voltages, the transistor R_{ON} dominates the link RC bandwidth operation making minimum width wires preferable.

3.5.2 Receiver

1mm and 4mm 16-bit buses were built implementing dual-supply transmitters paired with 2-bit offset corrected receivers. Receivers also incorporate a 1-tap DFE which is digitally tunable through control of the receiver hysteresis in the pull down network (Fig. 3.14). These links were used to characterized the degeneration offset correction in the receivers.

Shown in Fig. 3.18, offset correction improves mean measured receiver offsets



Figure 3.18: Receiver input offset variation and correction across V_{dd} with mean offset values annotated. Simulation results are reported from 1000 Monte Carlo runs.

at by 40% and 42% at $V_{dd} = 1.0V$ and $V_{dd} = 500$ mV respectively. However, worst case receiver offsets improve by only 30% and 9% respectively, limiting the V_{swing} reduction achievable using the dual-supply transmitters. Additional calibration bits offer diminishing returns, with 2-bit calibration achieving 55% of the improvement of 10-bit calibration while requiring only 20% of the area. Each receiver occupies $12\mu m^2$ plus an additional $10\mu m^2$ per latch to store each bit of offset correction.

The hysteretic DFE improves maximum bitrates by 3% and 11% on the 1mm and 4mm links respectively at $V_{dd} = 500$ mV, with negligible impact on energy. At $V_{dd} = 1.0$ V, the DFE pull-down network sized for $V_{dd} = 500$ mV is overwhelmed by the pull-down strength of the calibration network and provides a negligible performance benefit.

3.6 Conclusion

Low-voltage swing signaling shows promising results for reducing the energy required to move data across on-chip wires. Particularly for short wires, a new class of transceiver circuits will be necessary to exploit the energy reduction that can be achieved using low-voltage swing signaling. These transceiver circuits will require a new focus on limited area footprints, energy efficiency and automated replacement of conventional cells for insertion into traditional designs.

On-chip transceiver circuits that seek to meet these objectives were presented for short-range, capacitively-dominated wires. The proposed capacitive chargesharing transmitter presents an alternative to existing low-voltage swing transmitters that require multiple supply voltages or static bias currents. Furthermore, the high degree of digital tunability allows a designer to minimize energy while providing adaptability to process variation and dynamically scaled supply voltages. A sense-amplifier structure that is well suited for short on-chip wires was also presented. These provide small area footprints, tunable hysteresis/DFE, and offset correction.

To achieve an energy savings using low-voltage swing transceivers on local wires, a large number of cell replacements are necessary. The use of EDA tools will thus be needed for the replacement of many thousands of nets in a design to be feasible. The transceiver cells proposed in this work demonstrate that compact LVS transceiver cells can provide an energy savings with relatively small overheads. This will form the basis for future work developing insertion and replacement procedures that consider low-voltage swing cells during post-route optimization and are compatible with existing design flows.

Type	Dual	Capa	citive	Conventional				
туре	Supply	\mathbf{F}	\mathbf{FE}	Full-Swing				
Reference	[40]	[52]	[29]	[29]	Simulated			
Technology	90nm	$90 \mathrm{nm}$	90nm	$90 \mathrm{nm}$	$65 \mathrm{nm}$	$65 \mathrm{nm}$		
Target Application	SoC	High	Perform	SN	SoC			
Frequency (MHz)	100's		1000's	10's	100's			
Wire Length (mm)	1	5	2	2	1			
Supply Voltage (V)	1.2	1.0	1.2	1.2	0.5	1.0		
Data Rate (bps)	300M	2.4G	9G	$5\mathrm{G}$	55M	400M		
Signal Swing (mV)	250	100	120	1200	500	1000		
E/b/mm (fJ)	64	48	52.5	210	30	126		
Transceiver	93	730	N/A	N/A	INV	DFF		
Area (μm^2)	20	130	IN/A	IN/A	2.6	10.7		

 Table 3.2: Transceiver Comparison Performance Summary

Туре	Capacitive-TX		Dual Supply-TX, 2b RX Cal								
	10b	$\mathbf{R}\mathbf{X}$	Cal, No DFE	No DFE	DFE	No DFE	DFE	No DFE	DFE	No DFE	DFE
Reference		This Work									
Technology		65nm									
Target Application	SN	SN	SoC	SN		Sol	C	SN		Sol	3
Frequency (MHz)	1's	10's	100's	10's		100's		10's		100's	
Wire Length (mm)		1		1		1		4		4	
Supply Voltage (V)	0.35	0.5	1.0	0.5		1.0		0.5		1.0	
Data Rate (bps)	5M	30M	622M	70M	72M	800M	805M	45M	50M	200M	200M
Signal Swing (mV)	40	75	230	250	250	250	250	250	250	275	275
E/b/mm (fJ)	8.4	10.9	136	6.5	6.6	37.6	38.4	4.1	4.0	20.2	20.2
Transceiver	TX	RX	Total	TX RX Tota		tal					
Area (μm^2)	122	112	234	5.5 35.1 40.6).6					

SoC - System-on-Chip, SN - Sensor Node/Network

Chapter 4 – Automated Insertion of Low-swing On-chip Interconnect Links

4.1 Introduction

As described in previous chapters, modern digital integrated circuits expend a significant portion of their energy budget through the charging and discharging of wire capacitances. Low-voltage swing signaling (LVS) circuits present an opportunity for reducing this cost of transporting data on-chip. Research exploring the potential of LVS circuits for reducing signaling latency and energy on global on-chip wires have gained traction over the past decade [15–18,29,31,36,39,52,61].

Implementation of LVS circuits in the context of a conventional digital design flow currently requires significant manual design and verification effort and is therefore impractical. Use of LVS circuits for on-chip wire signaling has thus been limited to only the most obvious long, global wire routes in academic designs. Of the research designs in which LVS signaling has been used, all local wires are still driven by conventional logic cells operating at the nominal supply voltage [18, 21, 40].

These short on-chip wires represent an unexploited potential for design energy
reduction. In a 65nm process, replacing conventional logic with LVS transceivers similar to those presented in Chapter 3 with as little as $180\mu \text{m}$ ($V_{dd} = 1\text{V}$) or $120\mu \text{m}$ ($V_{dd} = 0.5\text{V}$) cumulative wire lengths can result in a reduction in signaling energy. Only a small amount of energy can be saved for each of these wires because the vast majority of wires on chip are short local routes, however the shear number of potential wire replacement candidates represents an opportunity for meaningful energy savings. For example, synthesis of the small 2mm x 2mm OpenRISC core results in 249,426 nets of which 16,733 are between $200\mu \text{m}$ and 1mm in length [62]. The wire capacitance alone on this 6.7% of nets accounts for a disproportionate 20% of the total dynamic loading in the digital core. A subset of these nodes terminate in a flip-flop cell performing identical logical functionality to the transceiver pair circuits presented in previous chapters and making them potentially attractive candidates for replacement with LVS cells.

Due to the large number of potential wire replacements, an automated procedure is necessary for the proposed circuits to feasibly be used in a modern digital design. Such a procedure requires identification of routed wires that are viable candidates for replacement based on:

- 1. Wire and cell capacitive loading of the node.
- 2. Logical equivalency of logic cells to available driver and receiver cells.
- 3. Timing requirements of the design and signal paths.
- 4. Minimization of area overhead required for cell replacement/insertion.

In this chapter, an interconnect analysis and insertion procedure is proposed to automate the insertion of LVS cells into a digital design. Nodes which are candidates for insertion of LVS transceivers are identified based on metrics of wire and gate capacitive loading, potential for energy/power reduction, cell functional equivalency to available low-voltage swing cells, area required for cell insertion, and effect on design timing. Section 4.2 describes the conventional digital design flow process and necessary modifications for implementing an energy-reduction LVS cell insertion step. Section 4.3 details the design analysis process, report generation, and procedure for LVS cell replacement/insertion. Section 4.4 presents design details and performance metrics resulting from performing cell insertion on a series of digital CMOS cores. Section 4.5 concludes this chapter, providing a discussion on the feasibility of the proposed automated cell insertion process and recommendations for future work.

4.2 Automated Insertion of LVS Cells

Electronic design automation (EDA) tools have long been necessary in order for digital CMOS designers to manage the development of complex digital integrated circuits with an enormous number of transistors. To be feasible, the introduction of LVS transceiver cells into a digital design requires the inclusion of the LVS cells into some form of EDA design flow. In the proposed LVS cell insertion procedure, a conventional digital design flow is utilized in order to produce an initial synthesized design that can then be analyzed and optimized to provide improved energy efficiency through the use of LVS cells.



Figure 4.1: Simplified conventional design flow.

A typical digital design flow upon which our automation procedure is based is shown in Fig. 4.1. A designer provides a description of how the design should operate using a hardware description language (HDL) such as Verilog. A synthesis tool compiles this HDL into logic gates that perform the functionality described by the designer. A place and route tool is then used to determine efficient placement of these physical cells relative to one another and route wires connecting these nodes. During the place and route stage of the design flow, multiple iterations of both cell placement and wire routing are performed. After wires are routed, the place and route tool is able to accurately determine the effective capacitive loading of routed wires by extracting parasitic resistance and capacitance values of wires in the design. Using this information, the design undergoes a series of optimization iterations to improve routing, cell sizing and placement in order to improve design performance, area, and power.



Figure 4.2: Proposed digital design flow using automated low-voltage swing cell insertion.

A conventional design resulting from this flow is used as a basis for the insertion of LVS cells to produce a more energy efficient, but logically equivalent design. The automated cell insertion procedure operates as an additional post-routing optimization step in the place and route stage of the conventional flow. During this step, nodes are evaluated for their potential as candidates for LVS cell insertion. Where appropriate energy reduction can be achieved, LVS transceiver cells are inserted into the design in place of or in addition to conventional logic cells. This results in a modified design flow as shown in Fig. 4.2 The determination of whether to insert LVS cells on a particular node is dependent upon the wire capacitance, attached cell capacitance, logical function and timing. The base design thus provides the information necessary to determine whether a LVS cell is appropriate for insertion on a node by calculating the effective dynamic energy reduction that can be expected as well as any positive or negative effect on the area and timing of the logic path. Once the conventional design is produced, LVS cell insertion can proceed as follows:

- 1. Logic and buffer tree hierarchies are identified and grouped into sets of nets, representing buffer and/or logic trees that can be functionally replaced.
- 2. Nets and cells that are functionally replaceable are identified, enumerated and used to calculate the expected outcome of LVS cell insertion on the design energy, area and timing.
- Using designer-defined priority metrics of area, energy and timing a determination of which nets are appropriate candidates for LVS cell insertion is made.
- 4. The expected benefits of replacement for each set of identified replaceable nets is reported to the designer for review.
- 5. Design modifications are applied by inserting LVS cells and replacing existing conventional cells where appropriate.

4.2.1 Preparation for Design Evaluation

Analysis to determine the efficacy of LVS cell insertion begins with collection of relevant design and library information. A cell library is read which describes the properties necessary to determine node loading and connectivity including pins, pin function, cell function, power and cell area. Once this cell information is collected, a Verilog netlist of the initial conventional design is used to build two data structures that store the connectivity of each net and each cell respectively.

All cells and pins connected to each net are cross-referenced with information from the imported cell library to build a list of driver (cell output pins) and receiver (cell input pins) cells attached to the net. This information will be used to calculate net capacitive loading and dynamic energy contribution as well as for evaluating net connection hierarchy during replaceable set generation as described in section 4.2.2.

Extracted parasitic device and routing information from the initial design is read from a standardized parasitic exchange format (SPEF) file. The net capacitances contained in the SPEF file are recorded as a property of that net. Likewise, the pin capacitance of each pin from the list of drivers and receivers connected to the net are summed and stored as a property of the net to which it is attached.

4.2.2 Replaceable Set Generation

All wires that are valid candidates for replacement must be identified before the effects of cell replacement and insertion on the design's area, energy, power and delay can be determined. Driver and receiver cells that are identified as being logically equivalent to available LVS cells are grouped together with their associated net(s) and evaluated for replacement efficacy.



Figure 4.3: Simple conventional net with inverter driver identified for replacement.

In previous sections, LVS transceivers are discussed as simple point-to-point connections between a driver and receiver cell which can directly replace conventional logic cells as shown in Fig. 4.3. In many cases however, only a small number of the replaceable nets are analogous point-to-point connections between an inverter and flip-flop that can be directly replaced with the functionally equivalent LVS driver and receiver cells respectively. Instead, many nets consist of a hierarchy or tree of connected nets, drivers, buffers and one or many receiver cells as in Fig. 4.4. It is thus necessary to identify these net hierarchies in order to accurately evaluate a design for energy reduction potential. This is done by first grouping all nets and cells that drive, receiver or buffer each signal into a set. Within each set, nets and cells can then be evaluated to determine if the set as a whole or any subset of the set is replaceable.

Each replaceable set (RS) consists of the propagation path of a signal from the output pin of an originating driver to the input pin of receiver cells. An RS may



Figure 4.4: Conventional net identified for replacement.



Figure 4.5: Simple conventional net with logic driver identified for replacement.

consist of a single CMOS driver cell, a single CMOS receiver cell, and a connecting net wherein each cell may be replaced with a logically equivalent LVS cell as in Fig. 4.3. Alternatively, an LVS driver cell may be inserted immediately following the originating driver cell if no LVS cell with logical equivalence to the driver is available, as in Fig. 4.5. It is therefore necessary that the originating driver cell can either be replaced with an LVS driver cell as in Fig. 4.3, or accept an LVS driver cell at its output as in Fig. 4.5 while receiver cells must be logically equivalent to their LVS counterpart.



Figure 4.6: Differential low-swing receiver cell.



Figure 4.7: Two supply voltage differential low-swing driver cell.

4.2.2.1 Netlist Traversal

Replaceable sets are generated by iterating through each net in the design and identifying each net that is attached only to pins of replaceable receiver cells. For example, a net and its receiver cells are considered replaceable with the LVS receiver cell in Fig. 4.6 if the net is attached only to the D-input pin of flip-flop cells.

If the driver of a net that has been identified as replaceable is not a buffer or inverter cell, then the identification of the net is complete and a set is created that includes the driver cell, the net, and all receiver cells. However, if the driver of a replaceable net is a buffer or inverter, the parent and sibling nets are evaluated. If the driver of the parent net is also a buffer or inverter, the next parent next is evaluated as well. This process continues by traversing the netlist until the top parent net is identified for the set which does not have a buffer or inverter cell as its driver. The driver of this top parent net is considered the "originating driver" of the set.

If buffer/inverter cells appear in the hierarchy of a replaceable set they are removed from the design, connecting the input and output nets directly. In cases where this causes a change in sign between the input and output of the buffer/inverter cell, the signal polarity is corrected by swapping the differential connections at the receiver input.

Once the top parent net is found, a depth first traversal of buffer and inverterdriven children nets is performed in order to determine if each child net is valid for replacement and if so, which set it should be added to. A net may be considered replaceable if it has only inverter, buffer, or D-input pins of flip-flops as receivers. Starting from the top parent net and then iterating through all children nets in a depth-first traversal, each net is checked to determine if it may be considered replaceable. If a child net is replaceable and the current net is not replaceable, then the buffer cell driving the child net is marked as a potential top parent driver cell of a set. If a child net is replaceable and the current net is also replaceable, the buffer cell is marked for removal and the current and child nets can be considered part of the same replaceable set. If a child net is not replaceable, then the current



Figure 4.8: Cell replacement using net splitting on nets identified for replacement.

net is also marked as not replaceable.

During the depth-first traversal, if a net has any receiver cells that are not buffers, inverters, or flip-flop D-pin inputs, the net under evaluation and the associated set is simply marked as not valid for replacement. An example is shown by the NOR gate in Fig. 4.4. In this case, any child nets of the set hierarchy are still marked valid and may be replaced as shown in Fig. 4.9.

Alternatively, components of the net may be separated into two new nets:

- 1. A replaceable set containing a new net and only the replaceable components of the original set; and
- 2. A second new net containing only the non-replaceable components.

If during the depth first net traversal a child net is marked as not replaceable, the parent net is also marked as not replaceable.



Figure 4.9: Cell replacement without net splitting on nets identified for replacement.

4.2.3 Replaceable Set Analysis

After all replaceable nets and cells have been identified and grouped into an RS, analysis of the expected energy contribution is performed. Dynamic switching energy is calculated using the model in (4.1) where E_{TX} and E_{RX} are the average internal switching energies of the driver and receiver cells respectively during a transition and E_{load} is energy expended in driving the capacitive wire and cell load. Values of LVS and common conventional cell area and transition energy (E_{TX} and E_{RX} in (4.1)) are summarized in Table 4.1.

The total load capacitance of the RS which includes all wire and cell pin loads associated with the set is summed to determine C_{load} using parameters extracted from SPEF and standard cell library files. This summed load capacitance is then used to calculate E_{load} as described by (4.2) and (4.3) for conventional and LVS differential signals respectively where V_{swing} is V_{dd} or the LVS signaling voltage.

$$E_{transition} = E_{TX} + E_{load} + E_{RX} \tag{4.1}$$

$$E_{load} = \frac{1}{2} C_{load} V_{swing}^2 \tag{4.2}$$

$$E_{load-differential} = \frac{1}{2} (2C_{load}) V_{swing}^2 \tag{4.3}$$

After the expected switching energy of each set is calculated, a determination is made whether to apply LVS cells for the set. This decision can be made based on the cell area, energy consumption, set path delay, or any combination thereof for the replaceable set candidate under consideration. In the implementation here, energy alone is used to assess set replacement.

Energy required to operate the LVS based sets is calculated and compared with that of the conventional set implementation. A comparison is made of the dynamic energy of the set before and after LVS cell insertion. A set for which LVS insertion is predicted to provide an energy reduction is deemed a successful candidate and marked for replacement. Each set for which LVS insertion reduces set energy is then marked for replacement. Replacement is then executed as described in section 4.3.

4.3 Low-swing Insertion Procedure

Analysis of a conventional design results in the generation of a list of replaceable sets which contains unique cell and net identifiers, their connectivity and connectivity of the LVS cells that are to be applied to the design. This is used to determine the specific modifications to the design that will be performed to insert LVS cells for each replaceable set. These modifications are used to generate a tool command language (TCL) script that executes engineering change order (ECO) commands within the place and route tool that was used in the conventional design flow. These ECO commands are generated for Cadence Encounter[©] and used to create new design files that include LVS cells through the process described in Fig. 4.10.

First, the conventional design is loaded into the place and route tool. Conventional cells and nets that were identified and marked for replacement are deleted from the design along with fill cells, clock routing and design routing. The location of each conventional cell that is removed is recorded and used as an initial placement location for the inserted LVS cell that will replace it. LVS cells, nets and their

	Cell Area (μm)	Transition Energy (fJ)		
0.5X INV	1.2	1.1		
1.0X INV	1.2	1.8		
0.5X DFF	8.0	6.4		
1.0X DFF	8.4	9.6		
Dual Supply TX	6.4	4.8		
Min. Size RX	6.4	4.6		
Increased Drive RX	9.0	8.1		

 Table 4.1: Cell Comparison Summary



Figure 4.10: a) Cells in a replaceable set are identified; b) Existing routing and cells are removed from the design; c) LVS cells are inserted at or near the locations of the conventional cells they will replace; d) Special routes including secondary supply rails are routed; e) LVS wire routes are performed and marked as "don't touch" nets f) Conventional routing is performed again for conventional cells in the updated design.

connections are then added to the design and placed at the saved conventional cell locations. Routing is then performed for the new cells and net connections, first by connecting the low- V_{dd} rail to dual-supply driver cells, then by routing LVS data wires. Wires added during this procedure are marked as "don't touch" such that the placement and routing tool will not modify them during the rest of its run. Finally, with the LVS cells placed and routed, the conventional design flow is resumed and conventional wire routing proceeds as normal.

4.4 Results

To evaluate the effectiveness of the proposed automation procedure, several digital designs were synthesized, placed, and routed in a conventional 65nm CMOS design flow. These were analyzed for their potential for energy reduction and inserted with LVS cells. The effect of this procedure on cell area, energy, and timing was then evaluated.

In these experiments two variations of LVS transceiver pairs were utilized implementing a dual-supply driver, shown in Fig. 4.7 and 9-transistor sense amplifier, shown in Fig. 4.6.

First, a minimum energy, minimum area transceiver pair is used to determine the upper bounds on energy savings that can be achieved through LVS cell insertion without regard for design timing. These compact cells result in a significant increase in path delay due largely to their low output drive strength. However, they also result in a reduction in cell area due to the receiver having a smaller footprint than flip-flop cells available in the conventional cell library.

The second LVS receiver cell provides increased output drive strength, reducing the delay overhead of LVS insertion and helping to demonstrate the effect of a more varied LVS cell library. However, this increased drive strength comes at the cost of increased cell area and energy.

In future work it is expected that a library of LVS cells will be available that can be used to more closely match the timing characteristics of different conventional cells identified for replacement. This will limit the change in design timing that is currently experienced during LVS cell insertion.

Unless otherwise noted, net activity factor is assumed to be the same for all nets in the design. Results are summarized in Table 4.2.

4.4.1 9-core Network-on-Chip

A nine core network-on-chip implementation based on the token flow control routing architecture presented in chapter 2 was synthesized, placed, and routed in a 3mm x 3mm floorplan. Each of the nine NoC routers were placed in an area of 500 μ m x 500 μ m with a cell density of 73% in the center of each 1 mm² tile as shown in Fig. 4.11. The conventional design meets timing at 430 MHz.

Of the 262,569 conventional cells in the 9-core NoC, 79,240, or 30.2%, were identified as being successful candidates for replacement using minimum sized LVS cells. Using the minimum energy LVS cells, 94.7% of functionally replaceable sets were identified as beneficial candidates for replacement that would reduce energy



Figure 4.11: a) Nine core TFC NoC floorplan and; b) placed and routed design.

compared to the conventional cells that they would replace. These represent 14.9% of routed nodes in the NoC.

Using the minimum energy LVS cells, insertion results in a 14.9% energy reduction as well as a 5.8% reduction in cell area. However, the minimum sized LVS cells have insufficient output drive strength. This results in a significantly degraded design timing, increasing the critical path by 138% and resulting in a maximum predicted operating frequency of 180MHz.

Using non-minimum energy LVS cells, insertion results in a smaller 8.5% energy reduction and an increase of 6.9% in cell area. While still limited by the output drive strength of the receiver cells due to the limited selection of LVS cells, insertion of the non-minimum sized receivers result in a design that is able to meet timing at 290 MHz, a significant improvement over the minimum sized LVS cells.

The TFC routing architecture is designed with the intent of maintaining high

utilization of the data path (crossbar and link) and minimizing activity of the network data buffers. Activity factor of core-to-core links on the NoC is thus exceptionally high compared to switching activity in logic and buffer components, with a utilization rate of up to 54% at high loads. This results in the link and cross-bar wires accounting for disproportionately more energy than similar length wires throughout the routing logic itself. In practical application logic and datapath activity factors are thus expected to improve energy reduction further.

4.4.2 OpenRISC 1200

An OpenRISC 1200, 32-bit processor core, [62], was synthesized, placed and routed in the conventional design flow. The core was placed in a 2 mm x 2 mm floorplan with an average cell density of 77%, shown in Fig. 4.12. The OpenRISC 1200 core is able to meet timing at 104 MHz.

The placed and routed design contains 244,110 conventional cells, of which 153,918 or 63% were identified as being successful candidates for replacement using minimum sized LVS cells. This is an exceptionally high percentage of cells that have been identified candidates for replacement and results from synthesis of data and instruction memory blocks in the processor core from standard cells rather than the use of SRAM or register file macros.

Using the minimum energy LVS cells, insertion results in a 19.2% energy reduction as well as a 30.5% reduction in cell area. Use of minimum sized cells results in a very large decrease in design cell area due primarily to the processor memory



Figure 4.12: Placed and routed OpenRISC processor core.

modules being synthesized from standard cells. However, also responsible for the decrease in cell area is the use of minimally sized LVS cells for replacement of larger drive strength (2X and greater) conventional cells, which results in a performance penalty. As was the case in the NoC design, use of only minimum sized LVS cells results in degradation of design timing. Critical path is increased by 13%, resulting in a maximum operating frequency of 92 MHz.

Interestingly, use of non-minimum sized receiver cells results in no improvement in design timing compared with the minimum sized LVS cells. Observing the critical path in this case shows that the OpenRISC 1200 core performance is set by logic rather than wire delay and thus the drive strength of LVS cells is not as significant a factor as in NoC and Nova designs. However, use of non-minimum energy LVS cells does result in reduced savings in energy and cell area of 4.4% and 17.3% respectively.



Figure 4.13: Placed and routed Nova H.264 decoder core.

4.4.3 Nova H.264

A Nova H.264 decoder core, [63], was synthesized, placed, and routed in the conventional design flow. The core was floorplanned in a 1.1 mm x 1.1 mm area with a cell density of 66%, shown in Fig. 4.13.

The placed and routed design contains 106,041 conventional cells, of which 28,704 or 27% were identified as being successful candidates for replacement using minimum sized LVS cells.

The Nova H.264 decoder core is designed to process QCIF resolution video (176x144 pixels) and is intended to be easily extensible to higher resolutions. In the 65nm process, the H.264 decoder core meets timing at 200 MHz. However, because the Nova core is only required to run at 1.5 MHz in order to process QCIF resolution video at 30 frames per second, a large degradation in design timing from the use of only minimum sized LVS cells is acceptable.

Despite a similar percent of functional cell replacement candidates as the NoC design, predicted energy reduction is limited to 9.5% of the design power. This is



Figure 4.14: Effect of LVS insertion on maximum operating frequency.

largely due to the small design area which limits amount of energy reduction that can be achieved for each set.

Because of the extremely low performance requirement of 1.5 MHz, insertion of the minimally sized LVS cells provides an essentially 'free' reduction in design energy while also reducing cell area of the design by 13%. Due to the low operating frequency, energy efficiency of the Nova core could be further improved by operating at reduced supply voltages. While use of non-minimum sized LVS cells results in the Nova core meeting timing at 175 MHz, this results in no increase in performance due to the low target frequency of the core, and provides less savings in energy (2.7%) and cell area (3.9%) than minimum sized LVS cells.

4.5 Conclusion

An automated procedure for implementing LVS signaling across on-chip wires within the context of an existing conventional digital design flow was presented which allows a designer to take advantage of the energy reduction potential of LVS signaling while alleviating the need for custom cell design and integration. The presented design automation methodology demonstrates the feasibility of significant energy savings in arbitrary digital CMOS designs through the insertion of LVS transceiver cells as a post-route optimization step in the conventional design flow. By operating within the context of a traditional digital design flow, the presented procedure greatly decreases the burden of effort required for a designer to implement LVS signaling.

4.5.1 Future work

4.5.1.1 Net switching activity

In the analysis presented here, energy per transition is used at the primary metric guiding the determination of cell replacement. While energy savings per transition can be accurately calculated based on the net wire and gate loading, the switching activity of an individual net can be a determining factor in whether potential area and timing overheads for replacing a net are worth the benefit. For instance,

 Table 4.2: LVS Insertion Summary

	9-core NoC		OpenRISC Processor		Nova H.264	
	Min. Cells	Non-min.	Min. Cells	Non-min.	Min. Cells	Non-min.
		Cells		Cells		Cells
Total Gates	262,569		244,110		106,041	
Percent of Gates Replaced	22.9%	30.2%	63%	63%	27.1%	26.4%
Cell Area Change	-5.8%	+6.9%	-30.5%	-17.3%	-13%	-3.9%
Critical Path Delay Change	+138%	+48%	+13.0%	+13.0%	+82%	+14%
Energy Change	-14.9%	-8.5%	-19.2%	-4.4%	-9.5%	-2.7%

a net with high switching activity may contribute more than its fair share of the total design energy. This may justify the use of additional area for applying offset correction to the inserted transceiver cells in order to maximize the energy reduction on this node. Likewise, a net with low switching activity may contribute so little to the total design power consumption that even a small area overhead is considered an unacceptable expense.

Generation of such data requires that the expected workload of the digital logic be known and analyzed in advance and is beyond the scope of the work presented herein. However, the proposed automation system includes function hooks that allow data resulting from net switching activity analysis to be used during the analysis of cell replacement efficacy if it is available.

4.5.1.2 Expansion of LVS Cell Library

The primary goal of this work is to demonstrate the feasibility of LVS cell insertion and predict the maximum energy savings that can be achieved which only requires a limited set of LVS cells. However, with the development of a larger LVS cell library it becomes possible to evaluate replaceable set candidates on more sophisticated criteria. This would allow for the selection of LVS cells with output drive strengths that are more appropriately matched to the load they are expected to drive, as well as allowing a designer to set cell selection priorities based on trade-offs between design area, performance and signaling energy. Implementation of these additional cells and priority based candidate evaluation would extend the usefulness of the insertion procedure.

Additionally, this work uses only driver and receiver cells for transceiver insertion at flip-flop locations as an exemplary case. However, it is feasible for additional nodes that implement additional logic functionally to benefit from future LVS logic implementations that place a similar focus on energy efficiency. This focus on energy efficiency over performance would represent a departure from the historical use of LVS logic primarily for performance benefit [64,65]. Much of this functionality is already available in existing place and route tools.

4.5.1.3 LVS Cells for Latency Reduction

While this work looks solely at the reduction of energy through the use of LVS cells, a significant body of previous work on on-chip LVS signaling circuits focuses on overcoming the wire delay associated with long, globally routed on-chip wires. Driver and receiver cells presented in this work are well suited for reducing energy on locally routed wires. However, a similar automated insertion procedure would also be beneficial for implementing energy-efficient and low-latency LVS cells on global and semi-global wires. This is expected to be particularly beneficial in larger designs that have many long wire routes, and even more so in situations where these long routes are irregular and thus unfeasible to implement a LVS signaling scheme manually. Other LVS cell structures designed for these varied concerns, such as those described in [61], [37], [51], and [15] would be usable within this framework, with each exhibiting its own trade-offs of area, energy, and timing that would

necessitate additional cell selection criteria beyond the simple energy evaluation performed in this work.

4.5.1.4 Differential Routing Design Rules

Basic support for differential routing and wire shielding is already available within the place and route EDA tool used in these experiments. However, to improve reliability and performance it is desirable to have more advanced routing design rule functionality that would include improved wire shielding, twisting of differential pairs, tolerance limits for pair length and capacitance matching, and automated signal integrity checks to limit parallel routing of nearby wires to limit differential mode noise coupling. It is expected that these additional features should be developed before the LVS cell insertion procedure is used in a fabricated design.

Chapter 5 – Conclusion

Modern digital designs are challenged by the communication demands and energy consumption of on-chip wires as a result of scaling process technology. As CMOS technology scaling continues to improve transistor density, reducing the energy required to transport data on-chip will continue to be of concern to the system designer and is likely to grow in severity.

Low-voltage swing signaling has the potential to provide many advantages over conventional digital signaling across these on-chip wires. Use of various LVS signaling methods presents an opportunity for reducing the energy and/or latency cost of transporting data on-chip. However, overheads associated with implementing LVS signaling in the context of a digital design previously made their use unfeasible.

In chapter 2 it was demonstrated that integration of LVS transceiver cells into wire dominated signal paths in a digital designs can garner a reduction in signaling energy even on relatively short wires. The presented a network-on-chip (NoC) uses complementary routing architecture and low-voltage swing signaling techniques to address the challenges of on-chip communication in the context of an on-chip network. The resulting routing architecture and low-voltage swing crossbar switch and core-to-core links significantly improve the latency, throughput and power of an on-chip network. In chapter 3 transceiver circuits were presented that further explore the potential of LVS signaling on short wires. Transceiver cells were presented that improve the suitability of low-voltage swing signaling for short wire lengths and reduced supply voltages. A focus was placed on achieving the minimum energy/bit per mm of wire, compact cell sizes and adaptability to device variation and scaling supply voltages.

Finally, an automated procedure for implementing LVS signaling across onchip wires within the context of an existing conventional digital design flow was presented in chapter 4. This allows a designer to take advantage of the energy reduction potential of LVS signaling while removing the need for custom cell design and integration. The presented design automation methodology demonstrates the feasibility of significant energy savings in arbitrary digital CMOS designs through the insertion of LVS transceiver cells as a post-route optimization step in the conventional design flow. By operating within the context of a traditional digital design flow, the presented procedure greatly decreases the burden of effort required for a designer to implement LVS signaling.

Bibliography

- [1] A. Kumar, L.-S. Peh, and N. K. Jha, "Token flow control," in *Proc. 41st International Symposium on Microarchitecture*, Nov. 2008.
- [2] M. Bohr, "Interconnect scaling-the real limiter to high performance ulsi," in *Electron Devices Meeting*, 1995. *IEDM* '95., *International*, 1995, pp. 241–244.
- [3] R. Ho, K. Mai, and M. Horowitz, "The future of wires," Proceedings of the IEEE, vol. 89, no. 4, pp. 490 –504, apr 2001.
- [4] R. Ho, "On-chip wires: scaling and efficiency," Ph.D. dissertation, Stanford University, 2003.
- [5] Y. Hoskote *et al.*, "A 5-GHz mesh interconect for a teraflops processor," *IEEE Micro*, vol. 27, no. 5, pp. 51–61, Sep. 2007.
- [6] M. B. Taylor *et al.*, "The Raw microprocessor: A computational fabric for software circuits and general-purpose programs," *IEEE-MICRO*, vol. 22, no. 2, pp. 25–35, Mar./Apr. 2002.
- [7] P. Gratz et al., "On-chip interconnection networks of the TRIPS chip," IEEE Micro, vol. 27, no. 5, pp. 41–50, Sep. 2007.
- [8] D. Wentzlaff *et al.*, "On-chip interconnection architecture of the tile processor," *IEEE Micro*, pp. 15–31, 2007.
- [9] W. Dally and B. Towles, Principles and Practices of Interconnection Networks. Morgan Kaufmann Pub., 2003.
- [10] R. Mullins, A. West, and S. Moore, "Low-latency virtual-channel routers for on-chip networks," in *Proc. of ISCA*, Jun. 2004.
- [11] H. Matsutani, M. Koibuchi, H. Amano, and T. Yoshinaga, "Prediction router: Yet another low latency on-chip router architecture," in *Proc. of HPCA-15*, Feb. 2009.
- [12] A. Kumar, L.-S. Peh, P. Kundu, and N. K. Jha, "Express virtual channels: Towards the ideal interconnection fabric," in *Proc. of ISCA*, Jun. 2007.

- [13] L. Xin and C. sing Choy, "A low-latency noc router with lookahead bypass," in Proc. of ISCAS, Jun. 2010, pp. 3981 – 3984.
- [14] A. Kumar *et al.*, "A 4.6Tbits/s 3.6GHz single-cycle noc router with a novel switch allocator in 65nm CMOS," Oct. 2007.
- [15] J. Bae, J.-Y. Kim, and H.-J. Yoo, "A 0.6pj/b 3gb/s/ch transceiver in 0.18um cmos for 10mm on-chip interconnects," 2008, pp. 2861–2864.
- [16] B. Kim and V. Stojanovic, "A 4gb/s/ch 356fj/b 10mm equalized on-chip interconnect with nonlinear charge-injecting transmit filter and transimpedance receiver in 90nm cmos," Feb. 2009, pp. 66–68.
- [17] D. Schinkel et al., "A 3-gb/s/ch transceiver for 10-mm uninterrupted rclimited global on-chip interconnects," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 1, pp. 297–306, Jan. 2006.
- [18] K. Lee et al., "Low-power network-on-chip for high-performance SoC design," IEEE Trans. VLSI Systems, vol. 14, no. 2, pp. 148–160, Feb. 2006.
- [19] H.-S. Wang, L.-S. Peh, and S. Malik, "Power-driven design of router microarchitectures in on-chip networks," Nov. 2003, pp. 105–116.
- [20] L. Shang, L.-S. Peh, and N. Jha, "Dynamic voltage scaling with links for power optimization of interconnection networks," in *Proc. of HPCA-9*, Feb. 2003, pp. 91 – 102.
- [21] C.-H. Chen, S. Park, T. Krishna, and L.-S. Peh, "A low-swing crossbar and link generator for low-power networks-on-chip," in *Proc. of ICCAD*, Nov. 2011, pp. 779–786.
- [22] M. Galles, "Scalable pipelined interconnect for distributed endpoint routing: The SGI SPIDER chip." in *Proc. of HOTI-4*, Aug. 1996.
- [23] L.-S. Peh and W. J. Dally, "A delay model and speculative architecture for pipelined routers," in *Proc. of HPCA-01*, Jan. 2001, pp. 255–266.
- [24] A. Kodi, A. Louri, and J. Wang, "Design of energy-efficient channel buffers with router bypassing for network-on-chips (nocs)," in *Proc. of ISQED*, Mar. 2009, pp. 826 –832.

- [25] J. Kim, J. Balfour, and W. Dally, "Flattened butterfly topology for on-chip networks," in *Proc. of MICRO-40*, Chicago, IL, Dec. 2007.
- [26] G. Michelogiannakis, J. Balfour, and W. Dally, "Elastic-buffer flow control for on-chip networks," in *Proc. of HPCA-15*, Feb. 2009, pp. 151–162.
- [27] J. Oh, M. Prvulovic, and A. Zajic, "Tlsync: support for multiple fast barriers using on-chip transmission lines," in *Proc. of ISCA*, 2011, pp. 105–116.
- [28] T. Krishna *et al.*, "Noc with near-ideal express virtual channels using globalline communication," in *Proc. of HOTI-16*, 2008, pp. 11–20.
- [29] D. Schinkel et al., "Low-Power, High-Speed Transceivers for Network-on-Chip Communication," *IEEE Trans. VLSI Systems*, vol. 17, no. 1, pp. 12–21, 2009.
- [30] S. Rusu et al., "A 45 nm 8-core enterprise xeon processor," Solid-State Circuits, IEEE Journal of, vol. 45, no. 1, pp. 7–14, Jan. 2010.
- [31] E. Mensink *et al.*, "Optimally-placed twists in global on-chip differential interconnects," in *Proc. of ESSCIRC*, Sep. 2005, pp. 475 – 478.
- [32] J. Postman and P. Chiang, "Energy-Efficient Transceiver Circuits for Short-Range On-chip Interconnects," in *Proc. of CICC*, Sep. 2011.
- [33] N. Magen, A. Kolodny, U. Weiser, and N. Shamir, "Interconnect-power dissipation in a microprocessor," in *Proceedings of the 2004 international workshop* on System level interconnect prediction, ser. SLIP '04. New York, NY, USA: ACM, 2004, pp. 7–13.
- [34] R. Pawlowski et al., "A 530mv 10-lane simd processor with variation resiliency in 45nm soi," in Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International, 2012, pp. 492–494.
- [35] W. Dally, "The end of denial architecture and the rise of throughput computing," in *Design Automation Conference*, 2009. DAC '09. 46th ACM/IEEE, http://videos.dac.com/46th/wedkey/dally.html, 2009, pp. xv-xv.
- [36] R. Ho, K. Mai, and M. Horowitz, "Efficient on-chip global interconnects," in Proc. of Symp. VLSI Circuits. IEEE, 2003, pp. 271–274.
- [37] R. Ho et al., "High speed and low energy capacitively driven on-chip wires," IEEE Journal of Solid State Circuits, vol. 43, no. 1, p. 52, 2008.

- [38] J. Bae, J. Kim, and H. Yoo, "A 0.6 pJ/b 3Gb/s/ch Transceiver in 0.18 um CMOS for 10mm On-chip Interconnects," in *Circuits and Systems, 2008. IS-CAS 2008. IEEE International Symposium on.* IEEE, 2008, pp. 2861–2864.
- [39] B. Kim and V. Stojanovic, "A 4Gb/s/ch 356fJ/b 10mm Equalized Onchip Interconnect with Nonlinear Charge-Injecting Transmit Filter and Transimpedance Receiver in 90nm CMOS." IEEE, 2009, pp. 66–68.
- [40] J. Postman et al., "Swift: A low-power network-on-chip implementing the token flow control router architecture with swing-reduced interconnects," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, vol. PP, no. 99, p. 1, 2012.
- [41] B. Zhai et al., "Energy-Efficient Subthreshold Processor Design," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 17, no. 8, pp. 1127–1137, Aug. 2009.
- [42] S. Hanson et al., "Ultralow-voltage, minimum-energy CMOS," IBM Journal of Research and Development, vol. 50, no. 4.5, pp. 469–490, Jul. 2006.
- [43] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolic, *Digital integrated circuits : a design perspective*, 2nd ed., ser. Prentice Hall electronics and VLSI series. Pearson Education, Jan. 2003. [Online]. Available: http://www.worldcat.org/isbn/0130909963
- [44] L. Shang, L.-S. Peh, and N. K. Jha, "Power-efficient interconnection networks: Dynamic voltage scaling with links," *Computer Architecture Letters*, vol. 1, no. 2, 2002.
- [45] P. P. P. Sotiriadis, "Interconnect Modeling and Optimization in Deep Sub-Micron Technologies by," Ph.D. dissertation, Massachusetts Institute of Technology, 2002.
- [46] B. Bishop and M. Irwin, "Databus charge recovery: practical considerations," Proceedings. 1999 International Symposium on Low Power Electronics and Design (Cat. No.99TH8477), pp. 85–87.
- [47] P. Sotiriadis, A. Wang, and A. Chandrakasan, "Transition pattern coding: An approach to reduce energy in interconnect," in *Solid-State Circuits Conference* (ESSCIRC), Proceedings of the 26th European, no. 1. IEEE, 2000, pp. 348– 351.

- [48] M. Stan and W. Burleson, "Low-power encodings for global communication in CMOS VLSI," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, vol. 5, no. 4, pp. 444–455, 1997.
- [49] K. Lee, S. Lee, and H. Yo, "SILENT: serialized low energy transmission coding for on-chip interconnection networks," *IEEE/ACM International Conference* on Computer Aided Design (ICCAD), pp. 448–451.
- [50] M. R. Scott and R. Amirtharajah, "Pulse width modulation for reduced peak power full-swing on-chip interconnect," *Proceedings of the 14th ACM/IEEE* international symposium on Low power electronics and design - ISLPED '09, p. 213, 2009.
- [51] H. Zhang, V. George, and J. Rabaey, "Low-swing on-chip signaling techniques: effectiveness and robustness," *IEEE Transactions on Very Large Scale Inte*gration (VLSI) Systems, vol. 8, no. 3, pp. 264–272, Jun. 2000.
- [52] J.-S. Seo et al., "High-bandwidth and low-energy on-chip signaling with adaptive pre-emphasis in 90nm CMOS." IEEE, 2010, pp. 182–183.
- [53] E. Mensink *et al.*, "Power efficient gigabit communication over capacitively driven rc-limited on-chip interconnects," *Solid-State Circuits, IEEE Journal* of, vol. 45, no. 2, pp. 447–457, 2010.
- [54] A. P. Jose and K. L. Shepard, "Distributed loss-compensation techniques for energy-efficient low-latency on-chip communications," *IEEE Journal of Solid State Circuits*, vol. 42, pp. 1415–1424, June 2007.
- [55] H. Ito *et al.*, "A bidirectional- and multi-drop-transmission-line interconnect for multipoint-to-multipoint on-chip communications," *Solid-State Circuits*, *IEEE Journal of*, vol. 43, no. 4, pp. 1020–1029, 2008.
- [56] M. Flynn and J. Kang, "Global signaling over lossy transmission lines," in *IEEE/ACM International Conference on Computer-Aided Design*, Nov. 2005, pp. 985–992.
- [57] D. Schinkel et al., "A double-tail latch-type voltage sense amplifier with 18ps setup+hold time," in Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International, feb. 2007, pp. 314 –605.

- [58] B. Goll and H. Zimmermann, "A comparator with reduced delay time in 65nm cmos for supply voltages down to 0.65 v," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 56, no. 11, pp. 810–814, nov. 2009.
- [59] B. Nikolić et al., "Improved sense-amplifier-based flip-flop: design and measurements," Solid-State Circuits, IEEE Journal of, vol. 35, no. 6, pp. 876–884, Jun. 2000.
- [60] K. Hu et al., "A 0.6 mW/Gb/s, 6.4 7.2 Gb/s Serial Link Receiver Using Local Injection-Locked Ring Oscillators in 90 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 45, no. 4, pp. 899–908, 2010.
- [61] E. Mensink et al., "A 0.28pf/b 2gb/s/ch transceiver in 90nm cmos for 10mm on-chip interconnects," *IEEE Solid-State Circuits Conference*, pp. 412–413, February 2007.
- [62] A. Edvardsson *et al.*, "Or1200 openrisc processor," http://opencores.org/or1k/OR1200_OpenRISC_Processor, Jun. 2011.
- [63] K. Xu, "H.264/avc baseline decoder," http://opencores.org/project,nova, Oct. 2009.
- [64] T. Sakurai, H. Kawaguchi, and T. Kuroda, "Low-power cmos design through v/sub th/ control and low-swing circuits," in *Low Power Electronics and De*sign, 1997. Proceedings., 1997 International Symposium on, 1997, pp. 1–6.
- [65] D. Deleganes et al., "Low-voltage-swing logic circuits for a 7ghz x86 integer core," in Solid-State Circuits Conference, 2004. Digest of Technical Papers. ISSCC. 2004 IEEE International, 2004, pp. 154–518 Vol.1.