

# **Design and Analysis of the On-Chip Power Delivery Network for Energy Efficient Systems**

A Dissertation

Presented to

the faculty of the School of Engineering and Applied Science  
University of Virginia

In partial fulfillment

of the requirements for the degree of  
Doctor of Philosophy in Electrical Engineering

By

Kyle Alexander Craig

May 2014



# **Design and Analysis of the On-Chip Power Delivery Network for Energy Efficient Systems**

## **Abstract**

One of the most important metrics in modern digital integrated circuit design is energy efficiency. Energy efficiency is loosely defined as completing an application's workload at the lowest possible energy while still maintaining the desired performance. Energy efficiency is an important metric across the entire application space, from high-end applications to low-end applications. For example, high-end applications are limited by heat constraints in both personal and commercial computing. Low-end applications are not limited by the power wall but are instead constrained by application lifetime due to limited battery capacity.

In this dissertation, we demonstrate two scaling techniques that modify the local on-chip power delivery network to enable voltage scaling and energy efficient operation. The first technique, a programmable resistive power grid, creates a low overhead programmable power grid resistance by modifying the on-chip power delivery network. To achieve a programmable resistance, we break a monolithic power switch into partitions with independent gate control. The second technique, Panoptic Dynamic Voltage Scaling

(PDVS), modifies the power delivery network at the component level by adding multiple PMOS power switches and a discrete set of shared  $V_{DDs}$ . This allows each component to operate at the best  $V_{DD}$  depending on the application requirement.

Next, this dissertation demonstrates architectural and power delivery techniques to enable subthreshold operation in a voltage scalable system. We present a methodology for adapting PDVS architectures (or similar) for subthreshold operation. We propose using an NMOS header power switch with a nominal  $V_{DD}$  gate control to enable subthreshold operation. For designs with variable  $V_{DDs}$ , a transmission gate power switch provides the most robust power switch configuration.

Finally, this dissertation presents an energy efficient research and design infrastructure. This infrastructure leverages a set of scripts for commercial EDA tools and documentation to provide a fast and reliable methodology for power delivery network design space exploration. The scripted infrastructure also enables a fast and reliable methodology for creating large System on Chip (SoC) designs.

# Approval Sheet

This dissertation  
is submitted in partial fulfillment of the requirement  
for the degree of  
Doctor of Philosophy

**Kyle Alexander Craig**

---

AUTHOR

This dissertation has been read and approved by the examining committee:

**Benton Calhoun**

---

Advisor

**Mircea Stan**

---

**Joanne Dugan**

---

**John Lach**

---

**Kevin Skadron**

Accepted for the School of Engineering and Applied Science



Dean, School of Engineering and Applied Science

May

2014

*To my friends and family.*

***Wahoowa!***

# Acknowledgements

As many of us have experienced, graduate school can be stressful, rewarding, nerve wracking, exciting, challenging, and yes, fun. Sometimes it is all these at once. It is nearly impossible to get through alone. I have been very honored to have a lot of amazing people in my life who have supported me, worked with me, and put up with me throughout this journey. My time at the University of Virginia has truly been a life changing experience.

I would like to thank research advisor, Professor Ben Calhoun. It has been a privilege to work with him and learn what it means to be a researcher and an engineer. He is always pushing us to become better and provides us with all the best opportunities. Thank you, Ben. I would also like to thank John Lach. He was my first advisor when I was a teaching assistant during my first semester of graduate school and has been my co-advisor along with Ben. Thank you, John, for helping getting me started on my graduate career. I'd like to thank all the members of my Ph.D. committee, Professor Mircea Stan, Professor Joanne Dugan, Professor John Lach, Professor Kevin Skadron, and Professor Ben Calhoun for all their insight and time.

One of the most enjoyable parts of graduate school was being able to come in everyday and interact with our diverse and fun research group. I've spent the most time interacting with Yousef Shakhsheer (after 9 years I *still* need to look up the spelling of his last name), Alicia Klinefelter (my *first* work spouse), Yanqing Zhang (resident fantasy football expert), Seyi Ayorinde (my *second* work spouse, sorry Alicia), Jim Boley

(Bengroup bro), and Aatmesh Shrivastava (I/O pad hog). We have had countless work discussions, personal discussions, intellectual arguments, as well as non-intellectual arguments (*my bad*), that have all been thoroughly enjoyable and helped me along as both a researcher and a person. I would like to thank the rest of Bengroup past and present: Jiajing Wang, Randy Mann, Satya Nalam, Joe Ryan, Sudhanshu Khanna, Steve Jocke, Peter Beshay, Patricia Gonzalez, Divya Akella, Yu Huang, He Qi, Arijit Banerjee, Abhishek Roy, Farah Yahya, Chris Lukas, Harsh Patel, and Terry Tigner.

I have also had the privilege of working with many amazing people outside of Bengroup. I would like to especially thank Saad Arrabi. We had the opportunity to work together on multiple occasions and had the great (*mis*)fortune of sharing a windowless office, never knowing when we missed a beautiful day. I would also like to thank Nate Roberts from the University of Michigan, who I had the pleasure of collaborating with on multiple tapeouts. I want to thank Ben Boudaoud, we never had the benefit of working closely on a project (which would have undoubtedly led to *hours* of arguing over the best way to layout an inverter) but he has always been there to provide just the right amount of social interaction during stressful tapeouts.

I would also like to thank AMD Research and my manager Steve Kosonocky, for the opportunity to be an intern during the summer of 2010 and spring of 2011. The experience and knowledge I gained in those few short months was invaluable and truly helped focus my research goals.

Throughout my undergraduate and graduate career I had the honor of being a member of the Cavalier Marching Band (CMB). I have had many great experiences with the band

and made countless lifelong friends. My time with the CMB has helped shape me as a person. I would like to thank my entire marching band family, especially my close friends from the baritone section whom I hold dearly.

Last but certainly not least, I would like to thank my close friends and family for all their support in my progress throughout my undergraduate and the pursuit of my Ph.D.

# Contents

Contents .....	viii
List of Figures .....	xi
List of Tables .....	xx
Chapter 1    Introduction.....	1
1.1    Motivation .....	1
1.1.1    The On-Chip Power Delivery Network .....	3
1.2    Design Challenges for the On-Chip Power Delivery Network .....	6
1.2.1    Voltage Scaling .....	7
1.2.2 $V_{DD}$ Granularity .....	7
1.2.3    IR Drop .....	7
1.2.4 $di/dt$ Noise .....	7
1.3    Major Contributions and Organization .....	8
Chapter 2    Optimizing Voltage Scalable Architectures with a Single $V_{DD}$ .....	11
2.1    Background .....	11
2.2    Programmable Resistive Power Grid .....	12
2.2.1    Approach.....	13
2.2.2    Activity Factor .....	16
2.2.3    Energy Savings .....	18
2.2.4    Opportunity for Regulation.....	21
2.2.5    Route Level Macro Model .....	22
2.2.6    Core Level Modeling .....	24
2.2.7    Power Delivery Network Impact .....	26
2.3    Summary & Conclusions .....	30
Chapter 3    Optimizing Voltage Scalable Architectures with Multiple $V_{DD}$ s .....	33
3.1    Background .....	33

3.2	Panoptic Dynamic Voltage Scaling.....	35
3.2.1	Approach.....	36
3.2.2	PDVS Overheads .....	37
3.2.3	Results.....	39
3.2.4	Power Delivery Network Impact .....	43
3.3	Summary & Conclusions .....	46
Chapter 4	Enabling Subthreshold Operation .....	51
4.1	Background .....	51
4.2	PDVS Architecture Enhancements .....	52
4.2.1	Approach.....	53
4.3	A NMOS as a Subthreshold Header Power Switch .....	56
4.3.1	Background.....	56
4.3.2	Approach.....	57
4.3.3	Comparison to Conventional PMOS Power Switch .....	58
4.3.4	Flexible Transmission Gate Power Switch .....	63
4.4	Summary & Conclusions .....	67
Chapter 5	A Scripted Research & Design Infrastructure .....	69
5.1	Background .....	69
5.2	A Scripted Research Platform .....	72
5.2.1	Approach.....	72
5.2.2	Standard Cells for Subthreshold .....	73
5.2.3	Cell Characterization .....	74
5.2.4	Infrastructure.....	74
5.2.5	A Scripted Research Platform Example .....	80
5.3	SoC Design Methodology .....	82
5.3.1	Pad Ring & LEF Generation.....	83
5.3.2	SoC Integration .....	85
5.3.3	Tapeout Results.....	85
5.4	Summary & Conclusions .....	85
Chapter 6	Conclusions.....	89

6.1	Summary of Contributions .....	90
6.1.1	Teams and Individual Contributions.....	92
6.2	Broad Impact of this Work.....	92
6.3	Conclusions and Open Problems.....	93
Appendix A: List of Acronyms.....		99
Appendix B: List of Publications.....		103
Appendix C: Sample Top Level Verilog Code.....		105
Bibliography .....		107

# List of Figures

FIGURE 1-1: SIMPLIFIED DIAGRAM OF THE POWER AND PERFORMANCE DESIGN SPACE. LOW-END PERFORMANCE DEVICES, SUCH AS MEDICAL DEVICES, ARE LIMITED BY BATTERY LIFETIME CONSTRAINTS. HIGH-END PERFORMANCE DEVICES, SUCH AS PERSONAL COMPUTERS AND SERVERS ARE LIMITED BY HEAT DISSIPATION AND COOLING COSTS...	2
FIGURE 1-2: SIMPLIFIED RLC MODEL OF THE POWER DELIVERY NETWORK. THE POWER DELIVERY NETWORK CONSISTS OF VOLTAGE GENERATION, INTRINSIC AND PARASITIC RLC ASSOCIATED WITH THE PCB AND PACKAGE, AND ON-CHIP INTRINSIC AND PARASITIC RLC. FOR THIS WORK, WE WILL FOCUS ON THE ON-CHIP POWER DELIVERY NETWORK SHOWN ON THE RIGHT. ....	3
FIGURE 1-3: DYNAMIC VOLTAGE AND FREQUENCY SCALING. AN OFF-CHIP DC-DC CONVERTER IS USED TO ADJUST SUPPLY VOLTAGE, WHILE A PLL CAN BE USED TO ADJUST FREQUENCY WHEN PERFORMANCE REQUIREMENTS CHANGE. ....	4
FIGURE 1-4: ( <i>LEFT</i> ) POWER GATING WITH PMOS POWER SWITCH ( <i>RIGHT</i> ) POWER GATING WITH NMOS POWER SWITCH. POWER GATING REDUCED IDLE CURRENT BY PLACING EITHER A PMOS OR NMOS IN SERIES WITH A COMPONENT AND $V_{DD}$ OR $V_{SS}$ .....	5
FIGURE 2-1: A LARGE MONOLITHIC POWER SWITCH ( $W^k$ ) IS PARTITIONED INTO $W^k_{N-1}$ INDEPENDENTLY CONTROLLED POWER SWITCHES WITH VARIED WIDTHS. THIS IMPROVES $V_{DD}$ SCALING GRANULARITY WITHOUT THE ADDED DESIGN COMPLEXITY OF MULTIPLE DC-DC CONVERTERS.....	13
FIGURE 2-2: $R_{HEADER}$ VARIES BASED ON THE EFFECTIVE POWER SWITCH RESISTANCE AND AVERAGE CURRENT CONSUMED. BY ENABLING FEWER OF THE INDEPENDENTLY	

CONTROLLED POWER SWITCHES, THE EFFECTIVE POWER SWITCH WIDTH GOES DOWN, INCREASING $R_{\text{HEADER}}$ . .....	14
FIGURE 2-3: SIMULATED VIRTUAL- $V_{\text{DD}}$ OF A 32B KOGGE STONE ADDER DURING SUCCESSIVE ADDER OPERATIONS OVER SEVERAL POWER SWITCH WIDTHS IN A 90NM CMOS TECHNOLOGY. AS THE POWER SWITCH WIDTH SIZE IS DECREASED, THE RESISTANCE IS INCREASED CAUSING THE VIRTUAL- $V_{\text{DD}}$ TO DROOP. THIS ENABLES ENERGY SAVINGS THROUGH LOCAL POWER GRID RESISTANCE CONTROL. ....	15
FIGURE 2-4: VIRTUAL- $V_{\text{DD}}$ VALUES FOR VARYING NUMBER OF ENABLED RING OSCILLATORS FOR DIFFERENT NORMALIZED POWER SWITCH WIDTHS. AS ACTIVITY FACTOR IS INCREASED FOR A GIVEN POWER SWITCH WIDTH THE VIRTUAL- $V_{\text{DD}}$ IS DECREASED....	16
FIGURE 2-5: NORMALIZED PERFORMANCE VALUES FOR VARYING NUMBER OF ENABLED RING OSCILLATORS FOR DIFFERENT NORMALIZED POWER SWITCH WIDTHS. AS EXPECTED, AS ACTIVITY FACTOR IS INCREASED FOR A GIVEN POWER SWITCH WIDTH THE PERFORMANCE ALSO IS DECREASED.....	17
FIGURE 2-6: SIMULATED AND MEASURED DELAY AND ENERGY RESULTS FOR VARYING POWER SWITCH WIDTHS. SILICON RESULTS FROM A 90NM CMOS TEST CHIP CONFIRM THAT VARYING THE RESISTANCE OF THE LOCAL POWER GRID LEADS TO AN ENERGY SAVINGS OF UP TO 30%.....	19
FIGURE 2-7: MEASURED LEAKAGE WITH VARIED POWER GATE WIDTH FROM A 32NM SOI FOUR-CORE X86 PROCESSOR. BY REDUCING THE POWER GATE WIDTH BY DISABLING THE IDLE CURRENT CAN BE GRADUALLY REDUCED FROM 100% TO A LOWEST BOUND OF 10%. .....	20

FIGURE 2-8: ESTIMATED POWER SAVINGS OF APPROXIMATELY 15% FROM A FOUR-CORE X86 PROCESSOR USING A PROGRAMMABLE RESISTIVE POWER GRID. P1@P0 INDICATES THE TOTAL POWER OF CORE(S) THAT ARE RUNNING AT THE P1 FREQUENCY WHILE ANOTHER DIFFERENT CORE IN THE SoC IS RUNNING AT A P0 STATE. ....	21
FIGURE 2-9: AVERAGE AND WORST CASE VIRTUAL- $V_{DD}$ OF A SINGLE RLM DURING A TDP BENCHMARK ACROSS POWER SWITCH WIDTH. THE FLAT AREA OF THE CURVE IS DUE TO LOW IMPEDANCE SIZING TO MAINTAIN HIGH SPEED OPERATION. IN THIS EXAMPLE, TO ACHIEVE VARIABLE POWER GATE REGULATION POWER GATE PARTITIONING IS NEEDED IN THE RANGE OF $\sim 0.2\%$ OF TOTAL WIDTH. ....	23
FIGURE 2-10: APACHE REDHAWK SIMULATION SET FOR BULLDOZER CORE. A SIMPLIFIED PACKAGE MODEL WAS INCLUDED TO CAPTURE ALL INTRINSIC AND PARASITIC RLCs. EACH INDIVIDUAL RLM WAS MODELED AS A CAPACITANCE AND TIME DEPENDENT CURRENT SOURCE. ....	24
FIGURE 2-11: (TOP) $V_{DD}$ AND CURRENT RESPONE FROM REDHAWK (BOTTOM) $V_{DD}$ AND VIRTUAL- $V_{SS}$ PROFILE OVER TIME FOR DIFFERENT NORMALIZED POWER SWITCH WIDTH. ALL RLMS ARE SUPER IMPOSED ONTO THE PLOT. THE NEGLIGIBLE VARIANCE BETWEEN RLMS SHOWS THE POWER GRID IS ROBUST, AND THE POWER GATE RESISTANCE IS THE DOMINATE FACTOR IN THE REGULATION. ....	25
FIGURE 2-12: MODIFIED RLC MODEL USED FOR THIS ANALYSIS. WE SIMPLIFY THE MODEL AND LUMP THE PACKAGE AND CHIP RESISTANCE TOGETHER. $C_{CHIP}$ INCLUDE THE LOCAL CHIP CAPACITANCE AS WELL AS THE DECOUPLING CAPACITANCE. WE ALSO INCLUDED A PACKAGE INDUCTANCE. ....	27

FIGURE 2-13: SCHEMATIC AND TIMING DIAGRAM OF POWER GATE PARTITIONED INTO THREE PARTS ( $W_1$ , $W_2$ , AND $W_3$ ). $W_{TOTAL}$ IS FIXED WHILE $W_1$ AND $W_2$ WILL BE VARIED. $W_3$ IS EQUIVALENT TO $W_{TOTAL}-W_1-W_2$ .....	28
FIGURE 2-14: MAXIMUM VOLTAGE SUPPLY NOISE HOLDING $W_1$ CONSTANT AT $10\mu\text{M}$ AND SWEEPING $W_2/W_3$ AND HOLDING $W_2$ CONSTANT AT $15\mu\text{M}$ AND SWEEPING $W_1/W_3$ . $W_{TOTAL}=60\mu\text{M}$ AND $D_1=D_2=3\text{ NS}$ .....	28
FIGURE 2-15: NOISE ON THE VOLTAGE SUPPLY RAIL FROM SWITCHING WITH THE OPTIMALLY SIZED POWER GATES AND WITH A SINGLE POWER GATE. THE MAXIMUM VOLTAGE SUPPLY NOISE FROM $140\text{ mV}$ TO $40\text{ mV}$ – A 3.5X REDUCTION.....	29
FIGURE 3-1: THE PDVS ARCHITECTURE. MULTIPLE POWER SWITCHES ARE USED FOR EACH COMPONENT CONNECTING EACH TO CHIP-WIDE $V_{DDs}$ WHICH ENABLES FINE SPATIAL AND TEMPORAL GRANULARITY DVFS GRANULARITY. LEVEL CONVERTERS ARE USED TO PREVENT SHORT CIRCUIT CURRENT BETWEEN COMPONENTS AT DIFFERENT $V_{DDs}$ .	35
FIGURE 3-2: BLOCK DIAGRAM OF THE PDVS DATA FLOW PROCESSOR. SRAMS AND CONTROL SERVE FOUR DATA PATHS FOR DIRECT COMPARISON OF PDVS WITH $SV_{DD}$ & $MV_{DD}$ . . THE DATA PATH CONSISTS OF FOUR BAUGH-WOOLEY MULTIPLIERS, FOUR KOGGE-STONE ADDERS, A PROGRAMMABLE CROSSBAR, AND LEVEL CONVERTERS TO PREVENT SHORT-CIRCUIT CURRENT. ....	36
FIGURE 3-3: (A) SIMULATED LEVEL CONVERSION OVERHEAD VARYING $V_{DDL}$ FOR BOTH THE ADDER AND MULTIPLIER. (B) SIMULATED VIRTUAL- $V_{DD}$ SWITCHING OVERHEAD VARYING $V_{DDL}$ FOR BOTH THE ADDER AND MULTIPLIER. WITH THESE OVERHEADS WE ACHIEVE A BREAKEVEN CYCLE OF $<4$ AND $<1$ FOR THE ADDER AND MULTIPLIER, RESPECTIVELY. ....	38

FIGURE 3-4: (A) ORIGINAL IMAGE FED INTO CHIP (B) POST PROCESSED IMAGE FROM THE CHIP (C) NORMALIZED MEASURED INSTANTANEOUS POWER FROM OUR DEMONSTRATION. ALL AT $V_{DDH}$ HAS ALL $V_{DDs}$ AT THE NOMINAL 1.2V, $V_{DDH}/V_{DDL}$ ARE SET AT 1.2V AND 0.7V RESPECTIVELY.....	39
FIGURE 3-5: AVERAGE MEASURED POWER (w/ OVERHEADS) VS. WORKLOAD ACROSS 7 DIFFERENT DFGs. PDVS GIVES LOWER ENERGY FOR VARYING WORKLOADS THAN $MV_{DD}$ BY OPERATING COMPONENTS AT LOWER $V_{DDs}$ WHEN POSSIBLE WHILE $MV_{DD}$ COMPONENTS ARE HARD-TIED TO HIGHER VOLTAGES. PDVS POWER SWITCHES ENABLE $V_{DD}$ DITHERING (RAPID SWITCHING BETWEEN TWO $V_{DD}$ RATE PAIRS) TO APPROXIMATE IDEAL DVS. ....	41
FIGURE 3-6: MEASURED ENERGY BENEFIT (INCLUDING OVERHEAD) OF PDVS & $MV_{DD}$ NORMALIZED TO $SV_{DD}$ . PDVS SHOWS UP TO 80% AND 43% ENERGY SAVINGS OVER $SV_{DD}$ AND $MV_{DD}$ , RESPECTIVELY.....	42
FIGURE 3-7: AREA SAVINGS OF PDVS OVER $MV_{DD}$ FOR THE SAME ENERGY CONSTRAINT. PDVS SAVES UP TO 65% AREA OVER $MV_{DD}$ , SINCE INDIVIDUAL COMPONENTS ARE REUSED AT DIFFERENT VOLTAGES WHILE $MV_{DD}$ REQUIRES MULTIPLE COPIES OF EACH COMPONENT AT DIFFERENT VOLTAGES. ....	42
FIGURE 3-8: MODIFIED SIMPLE RLC MODEL USED FOR NOISE ANALYSIS. WE SIMPLIFY THE MODEL AND LUMP THE PACKAGE AND CHIP RESISTANCE TOGETHER. $C_{CHIP}$ INCLUDE THE LOCAL CHIP CAPACITANCE AS WELL AS DECOUPLING CAPACITANCE. WE ALSO INCLUDE A PACKAGE INDUCTANCE. WE ONLY CONSIDER THE NOISE SEEN ON $V_{DDH}$ .....	44
FIGURE 3-9: ANNOTATED DIE PHOTO AND CHIP SUMMARY .....	48

FIGURE 4-1: REGISTER BANK AND CROSS BAR MODIFIED FOR SUBTHRESHOLD OPERATION.

A NOMINAL AND SUBTHRESHOLD POWER SWITCH IS ADDED TO EACH PUTTING THE ENTIRE DATA PATH INTO SUBTHRESHOLD. .... 53

FIGURE 4-2: (*TOP*) LEVEL CONVERTER CAPABLE OF CONVERTING FROM SUBTHRESHOLD INPUT VOLTAGES ( $V_{SUBVT}$ ) TO NOMINAL VOLTAGES ( $V_{DDH}$ ). (*BOTTOM*) THE BYPASS STRUCTURE ALLOWS THE USE OF THE SUBTHRESHOLD LEVEL CONVERTERS ONLY WHEN OPERATING IN SUBTHRESHOLD. .... 54

FIGURE 4-3: (*LEFT*) SIMULATED DELAY & ENERGY OF AN ADDER AT 0.3V. CIRCUIT & POWER SWITCH BULK CONNECTIONS ARE TIED TO  $V_{DDH}$  (H) OR TO VIRTUAL- $V_{DD}$  (V), E.G.  $V_V$  = ADDER AND POWER SWITCH BULKS TIED TO VIRTUAL- $V_{DD}$ . (*RIGHT*) BODY CONNECTIONS OF THE SUBTHRESHOLD DATA PATH WITH  $V_{DDH}/V_{DDM}$  POWER SWITCH TIED  $V_{DDH}$  AND  $V_{SUBVT}$  TIED TO VIRTUAL- $V_{DD}$ . .... 54

FIGURE 4-4: VIRTUAL- $V_{DD}$  DURING  $V_{DD}$  SWITCHING.  $V_{DDH/M/L}$  = 1V, 0.5V, 0.25V WITH TRANSITION TIMES OF 5NS, 200NS, & 2NS. VERIFIED FUNCTIONALITY WITH HARDWARE MEASUREMENTS ..... 55

FIGURE 4-5: (*LEFT*) CONVENTIONAL PMOS POWER SWITCH WITH THE  $V_{SUBVT}$  POWER SWITCH BODY TIED TO VIRTUAL- $V_{DD}$  (*RIGHT*) PROPOSED NMOS POWER SWITCH WITH THE  $V_{SUBVT}$  POWER SWITCH BODY CONNECTION TIED TO  $V_{SS}$ . .... 57

FIGURE 4-6: SIMULATED VIRTUAL- $V_{DD}$  AT 0.3V FOR THE PMOS AND NMOS FOR THE HIGHEST AND LOWEST ACTIVITY FACTOR. AN ACTIVITY FACTOR OF 1.0 CORRESPONDS TO 10 RING OSCILLATORS ENABLED IN PARALLEL WHILE 0.1 CORRESPONDS TO ONLY 1 RING OSCILLATOR ENABLED. THE NMOS IS ABLE TO KEEP VIRTUAL- $V_{DD}$  AT THE TARGET 0.3V DUE TO THE NMOS BEING IN THE LINEAR OPERATING REGION. .... 59

FIGURE 4-7: SIMULATED FREQUENCY AT 0.3V FOR THE CONVENTIONAL PMOS AND PROPOSED NMOS. USING THE TRADITIONAL SIZING METHODOLOGY AND A TARGET DELAY DEGRADATION OF 10%, THE REQUIRED NMOS SIZE IS APPROXIMATELY 280X SMALLER THAN A PMOS FOR THE SAME TARGET DEGRADATION AT THE SAME WORST CASE ACTIVITY FACTOR. ....	60
FIGURE 4-8: SIMULATED ENERGY PER OPERATION AT 0.3V FOR THE CONVENTIONAL PMOS AND PROPOSED NMOS. THE DECREASE IN ENERGY AS POWER SWITCH SIZE IS REDUCED IS DUE VIRTUAL- $V_{DD}$ DROP DUE TO THE DEVICE RESISTANCE. WITH THE PMOS THE ENERGY STARTS TO INCREASE AT THE LOWER WIDTHS DUE LEAKAGE BECOMING THE DOMINATE FACTOR BECAUSE OF LOWER OPERATING FREQUENCIES. ....	61
FIGURE 4-9: SIMULATED AND MEASURED FREQUENCY AT 0.3V WITH AN ACTIVITY FACTOR OF 0.1 THE NMOS HAS A WORST CASE MEASURED FREQUENCY DEGRADATION OF ONLY 3%, WHILE THE SMALLEST PMOS AVAILABLE IN HARDWARE HAS A WORST CASE FREQUENCY DEGRADATION OF 50% .....	62
FIGURE 4-10: SIMULATED AND MEASURED FREQUENCY AT 0.3V WITH AN ACTIVITY FACTOR OF 1.0. THE NMOS HAS A WORST CASE MEASURED FREQUENCY DEGRADATION OF 13%, WHILE THE MINIMUM PMOS HAS A WORST CASE FREQUENCY DEGRADATION OF 84%.....	63
FIGURE 4-11: TRANSMISSION GATE POWER SWITCH ARCHITECTURE. BOTH PMOS AND NMOS TRANSISTORS ARE USED WITH COMPLIMENTARY CONTROL SIGNALS WHEN $V_{DDL}$ IS NEAR- OR SUBTHRESHOLD THE NMOS WILL BE THE DOMINATE DEVICE. OTHERWISE THE PMOS WILL BE THE DOMINATE DEVICE. ....	64

FIGURE 4-12: MEASURED ENERGY DELAY CURVE OF A NMOS, PMOS AND TRANSMISSION GATE BASED POWER SWITCH. FOR THE SAME DELAY CONSTRAINT THE TRANSMISSION GATE IS 21% LOWER ENERGY, WHILE FOR THE SAME ENERGY CONSTRAINT THE TRANSMISSION GATE IS 33% WHEN COMPARED TO THE NMOS AND PMOS RESPECTIVELY. ....	65
FIGURE 4-13: MEASURED ENERGY DELAY CURVE OF A NMOS, PMOS AND TRANSMISSION GATE BASED POWER SWITCH. THE TRANSMISSION GATE IS THE PARETO OPTIMAL CURVE OF THE NMOS AND PMOS HAVING THE LOWEST EDP ACROSS ALL $V_{DD}$ S. ....	66
FIGURE 5-1: GENERIC FLOW DIAGRAM OF THE SYNTHESIS AND PLACE & ROUTE FLOW. SYNTHESIS IS THE PROCESS OF TRANSLATING BEHAVIORAL RTL INTO A GATE LEVEL NETLIST. PLACE AND ROUTE IS THE PROCESS OF TRANSLATING THE GATE LEVEL NETLIST INTO PHYSICAL DESIGN. ....	70
FIGURE 5-2: ( <i>LEFT</i> ) SYNTHESIS FLOW DIAGRAM OF A CONVENTIONAL SCRIPTED RC FLOW. ( <i>RIGHT</i> ) FLOW MODIFIED FOR ADVANCED POWER DELIVERY NETWORK TECHNIQUES THROUGH THE USE OF CPF AND MANUAL .LIB FILES FOR POWER GATES. THROUGH PARAMETERIZATION AND STANDARD CELL LIBRARY CHARACTERIZATION BOTH ARE CAPABLE OF NOMINAL AND SUBTHRESHOLD SYNTHESIS. ....	75
FIGURE 5-3: ( <i>LEFT</i> ) SAMPLE LAYOUT OF A CMOS STANDARD CELL INVERTER. ALL STANDARD CELLS A BUILT WITH THE SAME PITCH. ( <i>MIDDLE</i> ) PMOS POWER SWITCH DESIGNED AT THE SAME STANDARD CELL PITCH FOR EASY INTEGRATION INTO ENCOUNTER ( <i>RIGHT</i> ) NMOS HEADER POWER SWITCH DESIGNED AT THE SAME STANDARD CELL PITCH FOR EASY INTEGRATION INTO ENCOUNTER .....	78

FIGURE 5-4: (*TOP*) PDVS IMPLEMENTATION WITH A PMOS POWER SWITCH FOR  $V_{DDH}$  AND AN NMOS FOR  $V_{DDL}$  (*BOTTOM*) DEPICTION OF THE OUTPUT OF THE ENCOUNTER INFRASTRUCTURE FOR PDVS. THERE ARE 4 POWER RINGS FOR  $V_{DDH}$ ,  $V_{SUBVT}$ , VIRTUAL- $V_{DD}$  AND  $V_{SS}$ . THE TOP ROW OF POWER SWITCHES SHOWS THE PMOS CONNECTED TO  $V_{DDH}$ , WHILE THE BOTTOM IS THE NMOS CONNECTED TO  $V_{SUBVT}$ . THE COMPONENT IS CONNECTED TO VIRTUAL- $V_{DD}$  AND  $V_{SS}$ . ..... 79

FIGURE 5-5: BLOCK DIAGRAM OF THE CASE STUDY. THERE ARE EIGHT BEHAVIORALLY IDENTICAL DUTs, EACH WITH AN ALU, MAC, AND RO. THERE ARE TWO LFSRS TO GENERATE QUASI RANDOM DATA. EACH DUT HAS A DIFFERENT POWER DELIVERY NETWORK CONFIGURATION..... 82

FIGURE 5-6: ENCOUNTER OUTPUT. EACH DUT HAS A DIFFERENT POWER DELIVERY NETWORK CONFIGURATION. DUT1 INCLUDES A PROGRAMMABLE RESISTIVE POWER GRID AND NMOS HEADER POWER SWITCH CONNECTED TO  $V_{DD}$ . DUT7 INCLUDES PDVS, NMOS HEADER POWER SWITCH CONNECT TO  $V_{SUBVT}$ , AND A PROGRAMMABLE RESISTIVE GRID CONNECTED TO  $V_{DDH}$ . ..... 83

FIGURE 5-7: PAD RING GENERATION INFRASTRUCTURE. A USER DEFINED SPREADSHEET IS TRANSLATED INTO SCHEMATIC AND LAYOUT..... 84

FIGURE 5-8: SoC GENERATED WITH THE SCRIPTED RESEARCH AND DESIGN INFRASTRUCTURE. THIS EXAMPLE CONTAINS BOTH CUSTOM (POWER MANAGEMENT, ANALOG FRONT END, RADIOS AND MEMORIES) AND OVER 15 DIGITAL BLOCKS..... 86

# List of Tables

TABLE 3-1: COMPARISON OF PEAK POWER SUPPLY NOISE TRANSITIONING FROM $V_{DDL}$ TO $V_{DDH}$ WITH AND WITHOUT SVS .....	45
TABLE 3-2: PDVS CHIP SUMMARY .....	48
TABLE 3-3: PDVS COMPARISON TO THE START OF THE ART .....	49
TABLE 5-1: RECENT TAPEOUT SCHEDULE: GREEN = USED INFRASTRUCTURE AS PRIMARY DESIGN/RESEARCH TOOL. YELLOW = USED SOME PART OF THE INFRASTRUCTURE .....	87

# Chapter 1

## Introduction

### 1.1 Motivation

In modern digital integrated circuit design, one of the primary focuses of research is on energy efficient systems. In this dissertation, energy efficiency is broadly defined as completing an application's workload at the lowest possible energy. For this work, the applications are broken up into two general categories based on performance: high-end and low-end (Figure 1-1). A few examples of high-end applications are: servers, desktop computers and laptop computers. Examples of low-end applications are: portable media players, smart phones, tablets, and bio-medical devices.

Both high-end and low-end applications have energy constraints for different reasons. For example, high-end applications are limited by heat constraints in both personal and commercial computing. As the transistor feature size continues to scale, allowing for more transistors on chip, the power density has become prohibitively high, leading to a power wall. The power wall refers to a system's power limit due to thermal constraints. Historically, this led to the shift away from increasing clock frequencies for single-core

processors to slower, multi-core processors that leverage parallelism to increase performance. Heat is also a large concern for data centers. From 1995 to 2005 the worldwide cost of powering and cooling data center servers increased from \$10.3 billion to \$26.1 billion [1]. On the other hand, low-end applications are limited by life-time concerns associated with battery and battery-less operation. In both application spaces there is a demand for ICs with high energy efficiency.

The strongest design knob for improving energy efficiency is the supply voltage,  $V_{DD}$ . Energy per operation is defined in terms of dynamic (equation (1-1)) energy and leakage energy (equation (1-2)), where  $C_L(V_{DD})$  is the circuit's load capacitance as a function of  $V_{DD}$ ,  $I_L(V_{DD})$  is the leakage current as a function of  $V_{DD}$ , and  $t_{op}(V_{DD})$  is the delay of the circuit as a function of  $V_{DD}$ .

$$E_{op_{dynamic}}(V_{DD}) = C_L(V_{DD}) * V_{DD}^2 \quad (1-1)$$

$$E_{op_{leakage}}(V_{DD}) = V_{DD} * I_L(V_{DD}) * t_{op}(V_{DD}) \quad (1-2)$$

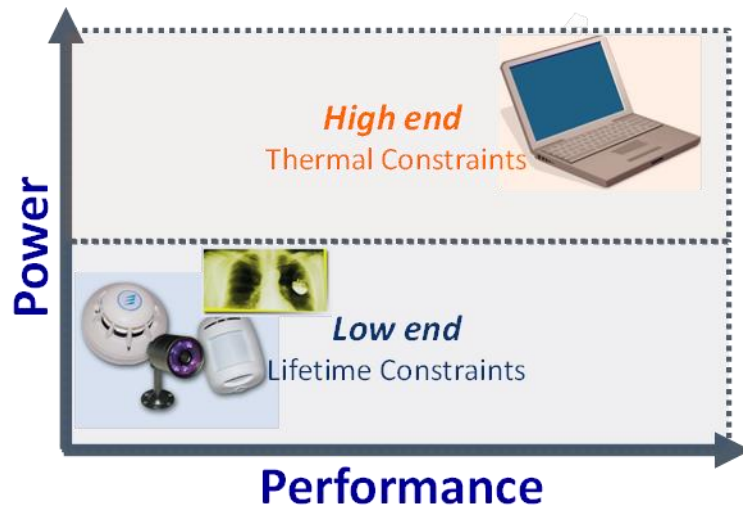


Figure 1-1: Simplified diagram of the power and performance design space. Low-end performance devices, such as medical devices, are limited by battery lifetime constraints. High-end performance devices, such as personal computers and servers are limited by heat dissipation and cooling costs.

Dynamic energy is the energy consumed by the charging of internal load capacitances, caused by transistor switching. Leakage energy is the energy that is consumed through transistors that are “off” and not switching. Leakage occurs due to a transistor not being an ideal device, thus having a small amount of quiescent current. Reducing supply voltage can lead to a greater than quadratic savings in energy per operation when considering both the dynamic and leakage components of energy. Two conventional power delivery network techniques to reduce dynamic and leakage energy are dynamic voltage and frequency scaling and power gating, while an emerging technique is subthreshold operation.

### 1.1.1 The On-Chip Power Delivery Network

The power delivery network is a large system consisting of many different components.

Figure 1-2 below shows a simplified model of the power delivery network. In many commercial systems the voltage generation is done through DC-DC converters on the printed circuit board (PCB). There are intrinsic and parasitic resistances, inductances, and capacitances (RLC) associated with the PCB, the package, and the chip itself. For the

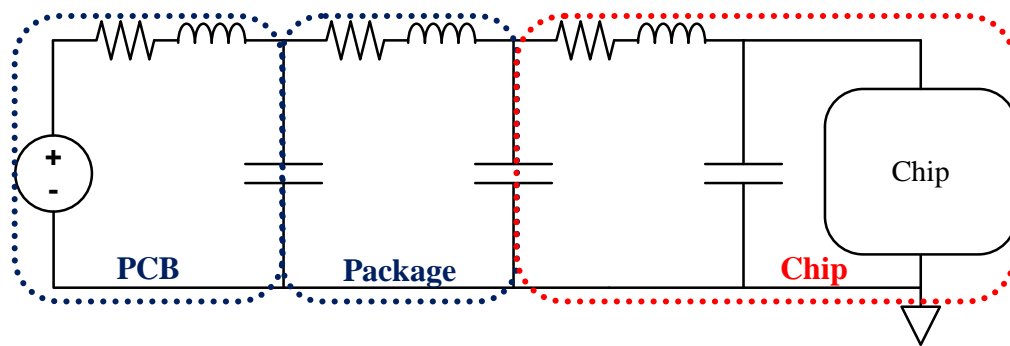


Figure 1-2: Simplified RLC model of the power delivery network. The power delivery network consists of voltage generation, intrinsic and parasitic RLC associated with the PCB and package, and on-chip intrinsic and parasitic RLC. For this work, we will focus on the on-chip power delivery network shown on the right.

purpose of this dissertation, we will focus on modifications to the on-chip power delivery network, highlighted in red, to enable energy efficient design. To evaluate the impact of our energy efficient design on the entire power delivery network, we will use a simplified model similar to Figure 1-2.

### ***Dynamic Voltage and Frequency Scaling***

As previously mentioned, the strongest design knob for improving energy efficiency is supply voltage. However performance is proportional to  $V_{DD}$ , and reducing the supply voltage saves energy at the cost of performance. Dynamic voltage and frequency scaling is a technique that leverages the greater than quadratic energy savings, by lowering  $V_{DD}$ , whenever performance requirements allow [2]. As shown in Figure 1-3, a conventional DVFS implementation uses an off-chip DC-DC converter to scale  $V_{DD}$  when performance requirements change. The method for scaling frequency is outside the scope of this work however a typical method is to use on-chip clock generation, such as with

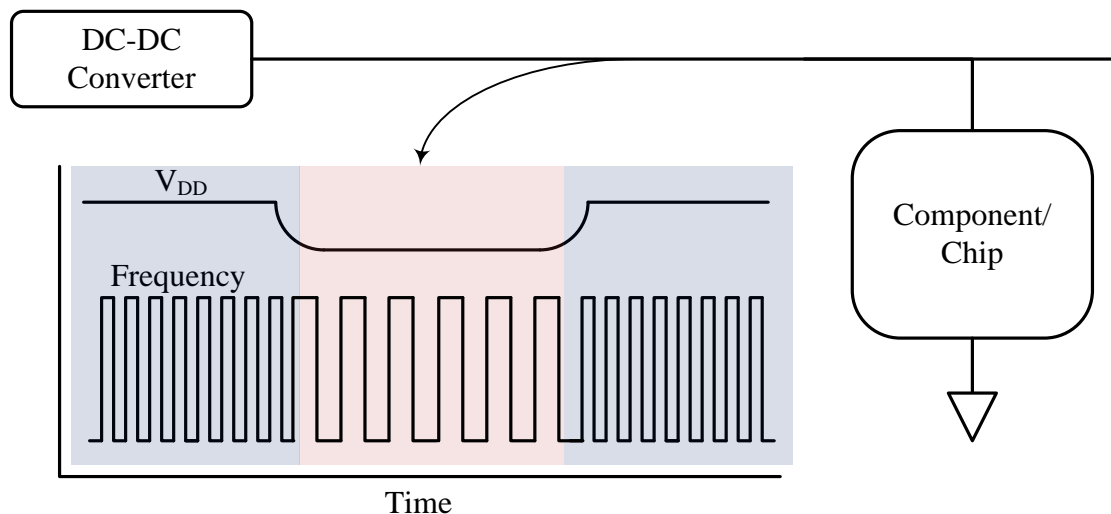


Figure 1-3: Dynamic Voltage and Frequency Scaling. An off-chip DC-DC converter is used to adjust supply voltage, while a PLL can be used to adjust frequency when performance requirements change.

phase locked loops (PLLs), to control the frequency scaling. DVFS improves energy efficiency by adjusting to the lowest  $V_{DD}$ , and the slowest frequency allowable for a given application. DVFS has been shown to have energy reductions up to 4.5x for various high-performance applications [2]. In this dissertation the terms DVFS and Dynamic Voltage Scaling (DVS) are used interchangeably.

### ***Power gating***

Modern commercial processors consist of multiple, complex cores, where each has many different components required to execute general purpose applications. For a given application, it is unlikely that all the cores or components will be used at once. Instead some components will be idle and leaking, reducing the overall energy efficiency. A conventional technique for minimizing leakage and improving energy efficiency is to power gate the core (or component) [3]. Power gating is shown in Figure 1-4. A PMOS power switch transistor is placed in series in the power delivery network between the  $V_{DD}$  and component, or a NMOS power switch transistor is placed in series between the

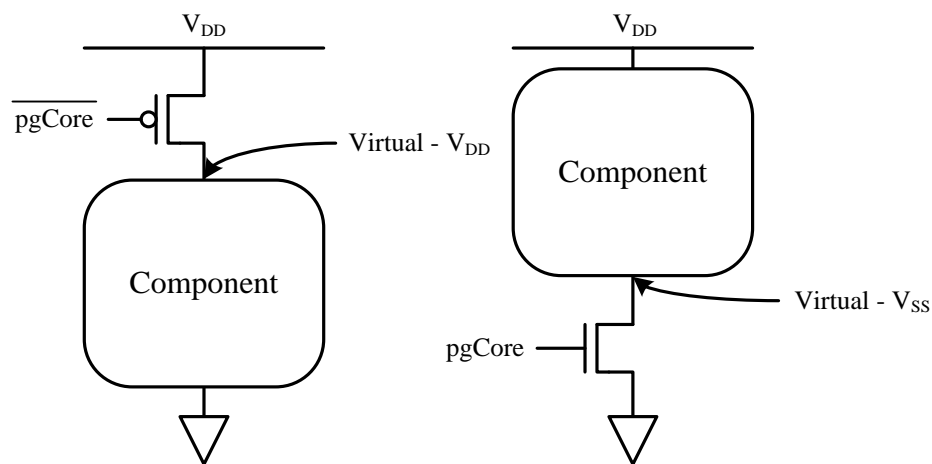


Figure 1-4: (left) Power gating with PMOS power switch (right) power gating with NMOS power switch. Power gating reduced idle current by placing either a PMOS or NMOS in series with a component and  $V_{DD}$  or  $V_{SS}$ .

ground and core. This creates an intermediate node known as virtual- $V_{DD}$  (or virtual- $V_{SS}$ ) that acts as the effective supply to the core. When the power gate is turned off, the voltage on the virtual- $V_{DD}$  node will collapse and reduce the voltage across the core, thus reducing the leakage from the component.

### ***Subthreshold Operation***

A major focus of energy efficient design in research is to run the entire circuit with a supply voltage below the threshold voltage of a single transistor. This is known as subthreshold operation [4][5]. The threshold voltage ( $V_T$ ) is defined as the transistor gate to source voltage differential ( $V_{GS}$ ) that is required to invert the channel of the transistor. When  $V_{GS}$  is below  $V_T$ , the transistor is “off”, and when  $V_{GS}$  is above  $V_T$  the transistor is “on”. Even though the transistor is considered “off” there is still enough current to charge and discharge internal capacitances and differentiate between the logical 1’s and 0’s in order to perform digital operations. As shown in equations (1-1) and (1-2), energy per operation has a more than quadratic dependency on  $V_{DD}$ . However, in the subthreshold mode of operation performance degradation has an exponential dependency on  $V_{DD}$ . For this reason, subthreshold is best suited for low-end performance applications.

## **1.2 Design Challenges for the On-Chip Power Delivery Network**

There are four key design challenges that arise when the on-chip power delivery network is optimized to improve energy efficiency. The first two design challenges, voltage scaling and  $V_{DD}$  granularity, are introduced with traditional DVFS. The second two design challenges, IR drop and  $di/dt$  noise, are generally associated with power gating. The design challenges reduce the opportunities to scale  $V_{DD}$ , thus reducing energy efficiency.

### 1.2.1 Voltage Scaling

In systems that use DFVS, DC-DC converters are commonly used to scale the chip  $V_{DD}$ . One of the biggest challenges associated with using DC-DC converters is the time required to scale voltage for an entire chip. This time can often be in the order of tens to a hundred microseconds and limits the opportunity to implement voltage scaling [6].

### 1.2.2 $V_{DD}$ Granularity

In multi-core designs, a small number of DC-DC converters and  $V_{DD}$ s are often used to power all the components in the design, with most cores being supplied by the same  $V_{DD}$ . This coarse  $V_{DD}$  granularity further limits the opportunity to use DVFS since voltage scaling can only occur when all cores in a design have the same voltage requirements.

### 1.2.3 IR Drop

In systems that use power gating, the IR drop across the PMOS or NMOS power switch is a concern. During normal operation, when the power gate is on, we want the virtual- $V_{DD}$  voltage to be the same as the  $V_{DD}$  voltage. However, since a power switch is a transistor, there is a finite resistance and IR voltage drop associated with it. If a power switch is sized too small for the given workload (i.e., current), the effective resistance of the power switch will be large. This results in a large voltage drop across the power switch, leading to the virtual- $V_{DD}$  voltage to be lower than  $V_{DD}$ . A lower virtual- $V_{DD}$  voltage reduces performance and, in the worst case, can lead to a breakdown of system functionality.

### 1.2.4 $di/dt$ Noise

When a power gated component is reconnected to  $V_{DD}$  (i.e., the power switch is turned back on), there is a large rush current that is associated with returning the collapsed

virtual- $V_{DD}$  to the nominal  $V_{DD}$  value. Since the power delivery network consists of many intrinsic resistances, inductances, and capacitances, the rush current creates a ringing noise throughout the power delivery network. Power delivery network noise will reduce system performance by requiring a reduction in the maximum frequency, or in the worst case, this noise will cause a breakdown of functionality.

### 1.3 Major Contributions and Organization

Energy efficiency is the major focus in modern digital design and is often achieved with voltage scaling. However,  $V_{DD}$  scaling with off-chip DC-DC converters limits the  $V_{DD}$  scaling opportunities. New on-chip power delivery network optimizations need to be developed to continue to improve energy efficiency. The impact of these optimizations also needs to be evaluated in the context of the entire power delivery network. This dissertation discusses modifications to the power delivery network to improve energy efficiency.

#### *Optimizing Voltage Scalable Architectures with a Single $V_{DD}$*

In Chapter 2, we propose a power delivery network modification in systems that have a shared common  $V_{DD}$ . This modification is called a programmable resistive power grid. This technique modifies the power delivery network by breaking a monolithic power switch into partitions with independent gate control enabling a low overhead programmable power grid resistance. The programmable resistance provides improved energy efficiency through dynamic energy savings and leakage reduction with data retention. This technique enables fine-grained voltage scaling without the use of the DC-DC converters.

### ***Optimizing Voltage Scalable Architectures with Multiple $V_{DD}$ s***

In Chapter 3, we consider systems that have multiple  $V_{DD}$ s that can be leveraged to improve energy efficiency. We demonstrate the first processor implementing Panoptic Dynamic Voltage Scaling (PDVS). PDVS is a power delivery network modification that extends DVS to finer granularity and removes the need for DC-DC converters for voltage scaling. PDVS modifies the power delivery network by adding multiple PMOS power switches at the component level. This allows each component to select the best  $V_{DD}$  from among a discrete set of shared  $V_{DD}$  rails depending on the application requirement.

### ***Enabling Subthreshold Operation***

In Chapter 4, we demonstrate architecture techniques that can be used to modify a voltage scalable system to operate in subthreshold for low performance applications. We present a methodology for adapting PDVS architectures for subthreshold operation. For DVFS designs using multiple PMOS power switches to select between multiple voltage rails, or for designs with power gating, we propose using an NMOS header power switch with a nominal  $V_{DD}$  gate control for the subthreshold voltage. For designs with a variable  $V_{DD}$ , we propose a transmission gate power switch, which provides the most robust power switch configuration for post manufacturing flexibility.

### ***A Scripted Research and Design Infrastructure***

The culmination of this work led to an energy efficient design infrastructure enabling fast and reliable power delivery network systems. The current EDA tools are generally very cumbersome, this chapter provides an easy methodology for implementing standard energy efficient design techniques (e.g., power gating, clock gating, multi- $V_{DD}$ ) as well as our proposed power delivery network techniques (e.g., PDVS, NMOS power switch,

transmission gate, and programmable resistive power grid). Finally we demonstrate a design and methodology infrastructure for creating a large scale SoC.

# Chapter 2

## Optimizing Voltage Scalable

## Architectures with a Single $V_{DD}$

### 2.1 Background

Many systems across this broad design space have applications that have a varying workload requirement. To improve energy efficiency in systems with varying workloads, we use power delivery modifications such as DVFS. When timing slack exists in a given application, DVFS adjusts the supply voltage and frequency to match an applications workload, thereby providing quadratic energy savings at these lower workload requirements (Figure 1-3). However, DVFS often uses off-chip DC-DC converters to scale  $V_{DD}$  of an entire chip. These DC-DC converters are relatively slow and limit the ability to scale  $V_{DD}$ .

In this chapter, we are focusing on systems with a single  $V_{DD}$ . One example of these systems is a commercial multi-core SoC, which has all cores sharing a common  $V_{DD}$ . During low workloads, two conditions need to be satisfied in order to scale voltage.

Firstly, all cores in the design need a similar workload requirement. Secondly, all cores in the system need the same  $V_{DD}$  requirement. Not satisfying these conditions limits the ability to scale voltage and reduces the energy efficiency. We propose a power delivery network modification that allows for voltage scaling without relying on DC-DC converters in single  $V_{DD}$  systems and improves  $V_{DD}$  granularity by providing independent component level voltage scaling. A programmable resistive power grid partitions a monolithic power switch into parallel independently controlled power switches with varying widths to enable fine-grained voltage scaling through power switch resistance control.

## 2.2 Programmable Resistive Power Grid

Power gating and dynamic voltage and frequency scaling are two common solutions to reduce leakage energy during standby mode and to improve energy efficiency. Power gating modifies the on-chip power delivery network by placing large power switches in series with the power supply or ground to collapse the virtual supply node to reduce leakage during idle mode. One disadvantage of power gating is that data stored in registers is lost. A variety of approaches to deal with this problem include putting registers on a separate supply, using high- $V_T$  balloon registers in parallel with core registers (or other alternative dual- $V_T$  register circuits). All of these incur overhead and added design complexity. DVFS during active mode saves power by lowering the frequency and voltage together when timing slack exists. Applying DVFS to multiple blocks requires multiple DC-DC converters that adjust the local voltage levels or alternative schemes that allow local  $V_{DD}$  selection from among multiple regulated supplies. Again, the overhead of these approaches can be substantial.

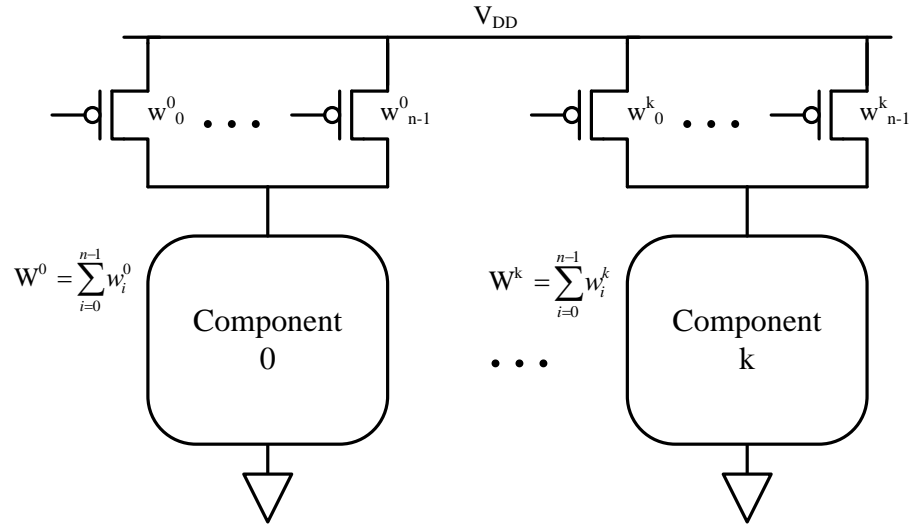


Figure 2-1: A large monolithic power switch ( $W^k$ ) is partitioned into  $W_{n-1}^k$  independently controlled power switches with varied widths. This improves  $V_{DD}$  scaling granularity without the added design complexity of multiple DC-DC converters.

We designed a programmable resistive power grid for providing dynamic system level flexibility by partitioning large, monolithic power gating transistors into parallel, independently controllable power switches with different widths, as shown in Figure 2-1. We can leverage this functionality to implement fine-grained DVFS at the component level and enable component level low leakage standby modes with data retention.

### 2.2.1 Approach

Parallel, independently controlled power switches are utilized as a controlled resistance to efficiently provide local voltage control without adding new DC-DC regulators, changing the voltage output of the existing regulators, or adding metal routing complexity. These power switches provide the effective voltage by utilizing the IR voltage drop across the power switch transistor as a controlled resistor. The virtual supply rail (virtual- $V_{DD}$ ) decreases voltage as the effective power switch width decreases. As seen in Figure 2-2, there is no feedback loop to adjust the output voltage. Rather, the voltage rail is allowed

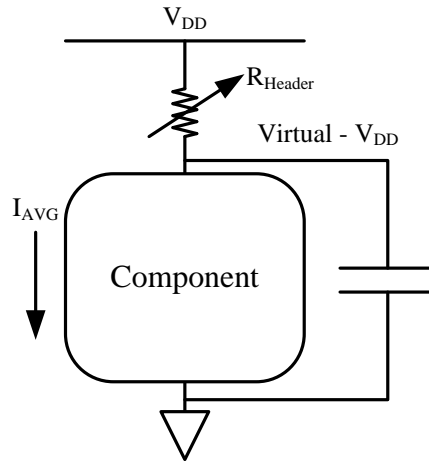


Figure 2-2:  $R_{Header}$  varies based on the effective power switch resistance and average current consumed. By enabling fewer of the independently controlled power switches, the effective power switch width goes down, increasing  $R_{Header}$ .

to settle during operation, thus providing an output voltage on the voltage rail for the circuit that is proportional to the circuit current load and the power switch effective resistance. This virtual rail voltage not only depends on the size of the power switch, but also on the activity factor of the block and extrinsic and intrinsic decoupling capacitance on the virtual- $V_{DD}$ . The virtual- $V_{DD}$  voltage sets the delay and energy of the component. The component is therefore not constrained by available voltage rails; it can operate at lower voltages than the rest of the system without changing the entire chip  $V_{DD}$ , and it avoids the high overhead of added DC-DC converters. In addition, a low energy standby mode is provided that reduces leakage current in idle blocks but enables data retention and incurs lower overheads when returning to normal operation than does full power gating. A similar leakage reduction mode was used to give state retention modes in SRAM arrays [7] for reducing idle power.

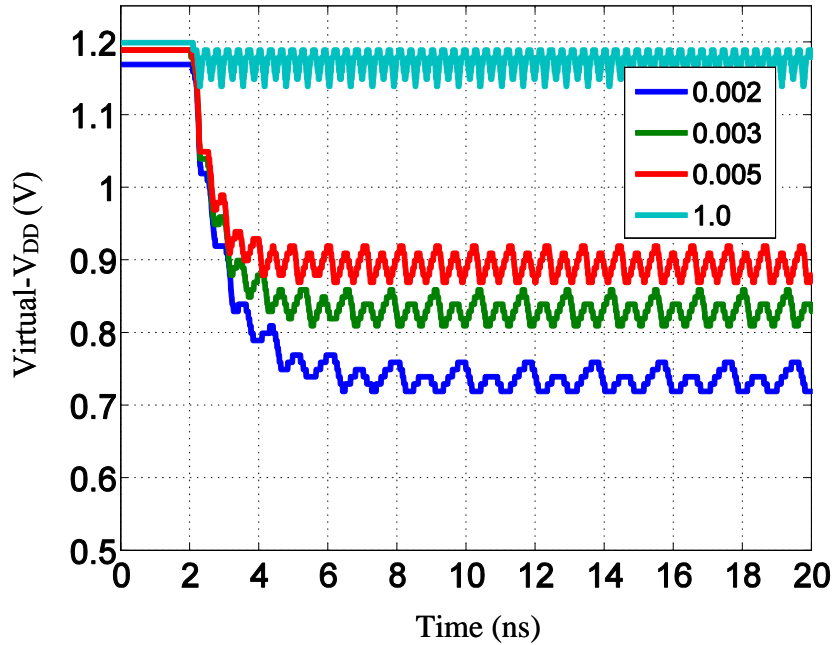


Figure 2-3: Simulated virtual- $V_{DD}$  of a 32b Kogge Stone adder during successive adder operations over several power switch widths in a 90nm CMOS technology. As the power switch width size is decreased, the resistance is increased causing the virtual- $V_{DD}$  to droop. This enables energy savings through local power grid resistance control.

Assuming frequent operation and no idling, the virtual rail will drop to a voltage below the nominal  $V_{DD}$  due to the controlled resistance. Figure 2-3 shows a 90nm CMOS simulation of the virtual- $V_{DD}$  voltage of successive operations of a 32b Kogge Stone adder for different percentages of enabled PMOS power switch widths, i.e., the width of enabled power switches divided by the total summed width of power switches. Included in the simulation was the extracted virtual- $V_{DD}$  capacitance with an ideal  $V_{DD}$  applied to the circuit. Notice that the virtual- $V_{DD}$  settles near a certain voltage for each power switch width. In this operation, the energy is reduced by up to 37% in simulation with a maximum increase of  $\sim 2.8x$  in delay for frequent operations when compared to circuit operation tied directly to  $V_{DD}$  with no power switch. This effectively implements a light

weight DVS mechanism with no additional circuits required (except control), which can be used in lieu of or in addition to conventional DVS methods.

### 2.2.2 Activity Factor

Although the effective  $V_{DD}$  of the block is set by the unregulated virtual- $V_{DD}$  voltage, analysis of the block under maximum current conditions can allow us to select the power switch partition width to set the worst case circuit performance. When the circuit consumes lower amounts of current than expected, virtual- $V_{DD}$  will not droop as far. This will cut into the dynamic energy savings, but the scheme still saves energy compared to having the full power switch on. Since the overhead for implementing this scheme is so low, the savings essentially come for free, since the power switches are already utilized in

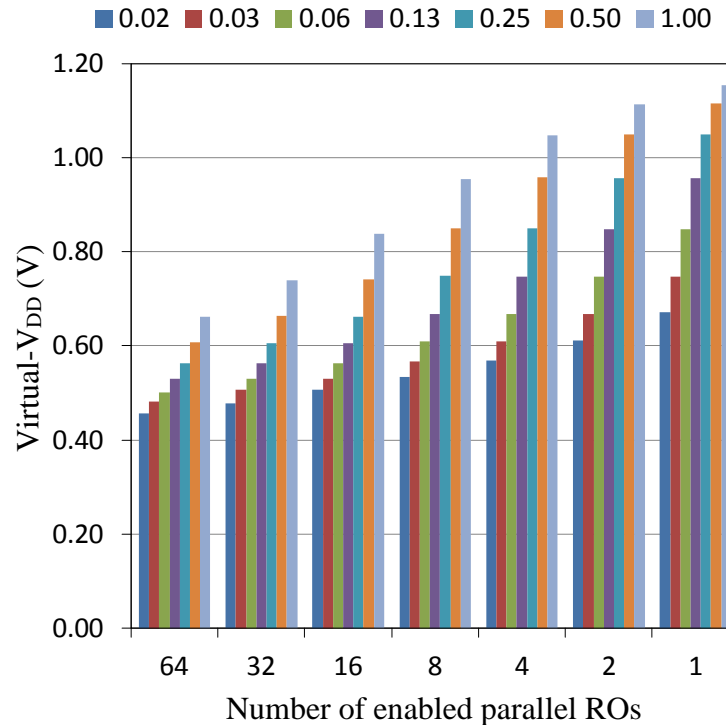


Figure 2-4: Virtual- $V_{DD}$  values for varying number of enabled ring oscillators for different normalized power switch widths. As activity factor is increased for a given power switch width the virtual- $V_{DD}$  is decreased.

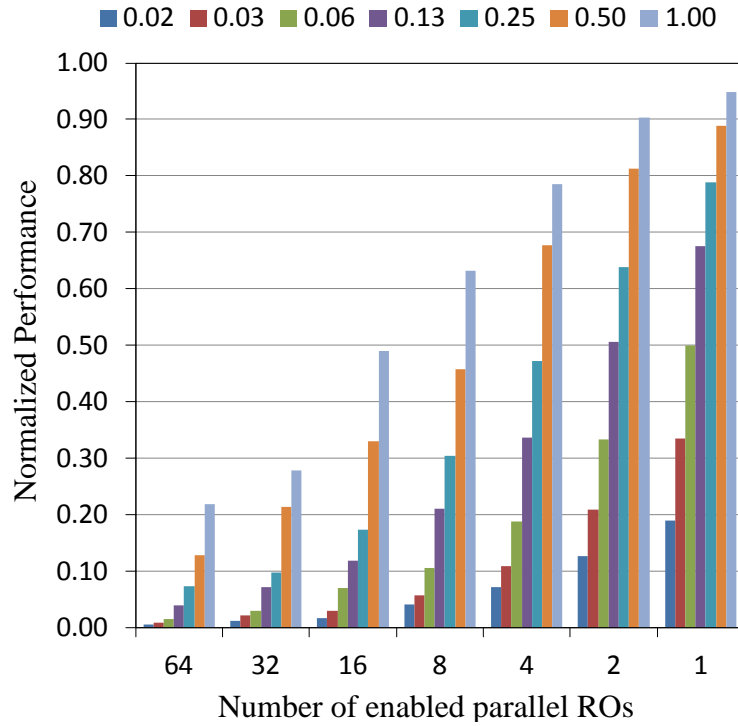


Figure 2-5: Normalized performance values for varying number of enabled ring oscillators for different normalized power switch widths. As expected, as activity factor is increased for a given power switch width the performance also is decreased.

these designs to reduce leakage.

To model the effect of activity factor, a similar methodology presented in [8] was used. Sixty-four five stage ring oscillators (RO) were simulated in parallel with the ability to disable different ROs via enable signals. With this setup, 64 enabled ROs correspond to the highest activity factor of 1.0, 32 enabled ROs correspond to an activity factor of 0.5, and so on. Each of the different activity factors were simulated with a varied percentage of enabled power switch width. Figure 2-4 shows the impact of activity factor and power switch width on virtual- $V_{DD}$ . The x-axis is the number of enabled parallel ROs while each bar represents a different amount of total power switch width enabled. The power switch partitions were sized for the 1-RO case, and the same sizes were used for all other RO cases. For the 1-RO case, we are able to see a large potential range of virtual- $V_{DD}$  values

(~0.6- 1.15V), however as the activity factor increases, the virtual- $V_{DD}$  range shrinks. For the high RO cases, larger power switch partition widths are required to enable a large virtual- $V_{DD}$  range. In Figure 2-5, RO performance is shown with varying activity factor and power switch width. The RO performance is normalized to an RO without any power gates at the nominal  $V_{DD}$ . As expected the performance is highly dependent on activity factor and enabled power switch width and shows similar trends as virtual- $V_{DD}$ .

The traditional power switch sizing methodology still applies, meaning that the total power switch size needs to be based on the worst-case activity factor and target performance requirement. The power switch width partitioning is highly dependent on the activity factor. In order to achieve a flexible power grid resistance, the power switch width partitioning should be sized for the range of activity factors expected and the performance range desired based on application. This methodology involves running simulations with varying power switch widths for different activity factors to characterize the virtual- $V_{DD}$  drop, performance degradation and energy savings. This characterization is accomplished with low level circuit simulations or high-level model techniques as discussed in section 2.2.5 and 2.2.6.

### 2.2.3 Energy Savings

To highlight the potential energy savings from a programmable resistive power grid, simple modifications can be made to equations (1-1) and (1-2).

$$E_{op\_dynamic}(V_{DD}, VV_{DD}) = C_{eff}(VV_{DD}) * V_{DD} * VV_{DD} \quad (2-1)$$

$$E_{op\_leakage}(V_{DD}, VV_{DD}) = V_{DD} * I_L(VV_{DD}) * t_{op}(VV_{DD}) \quad (2-2)$$

As virtual- $V_{DD}$  ( $V_{DD}$ ) decreases, we expect a greater than linear dynamic energy reduction due to the linear reduction of  $V_{DD} * V_{DD}$ , plus a less than linear reduction in  $C_{eff}$ , due to device source and drain parasitic junction capacitance being dependent on virtual- $V_{DD}$ . Finally, as virtual- $V_{DD}$  decreases,  $I_L$  will decrease due to the exponential relationship between virtual- $V_{DD}$  and the amount of DIBL (drain-induced barrier lowering).

To highlight the potential benefits of leveraging this droop for energy efficiency, we connected a ring oscillator to variable-weighted power gates and allowed the virtual- $V_{DD}$  to settle for each power gate width. Figure 2-6 shows the normalized energy and delay versus the normalized power gate width. As expected, reducing the power gate width decreases the energy and increases the delay. These RO results were confirmed through

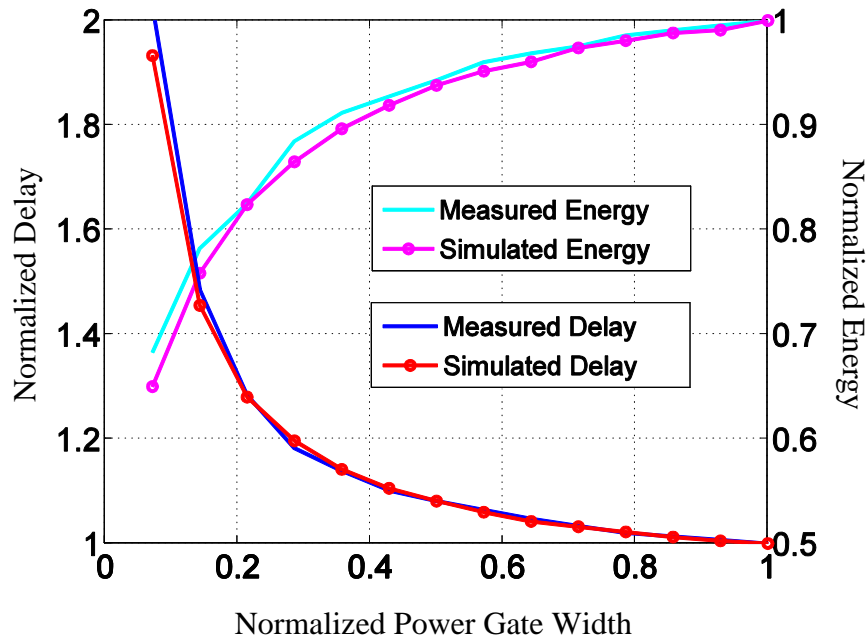


Figure 2-6: Simulated and measured delay and energy results for varying power switch widths. Silicon results from a 90nm CMOS test chip confirm that varying the resistance of the local power grid leads to an energy savings of up to 30%.

silicon measurements using a 90 nm commercial bulk technology with four 97 stage ROs in parallel consisting of inverters and delay cells to simulate a high current and activity load. The normalized measured values match the simulated values and show an energy savings of over 30% in silicon.

Many systems and blocks within systems spend large amounts of time idling. Additionally, blocks such as register files or memory may need to retain their data, which is not supported by most power gating schemes. A programmable power grid provides a low energy solution that enables data retention with reduced idle leakage current. Leakage current is reduced through lowering power gate width by dropping the voltage across the active devices which reduces the amount of DIBL, thus reducing device

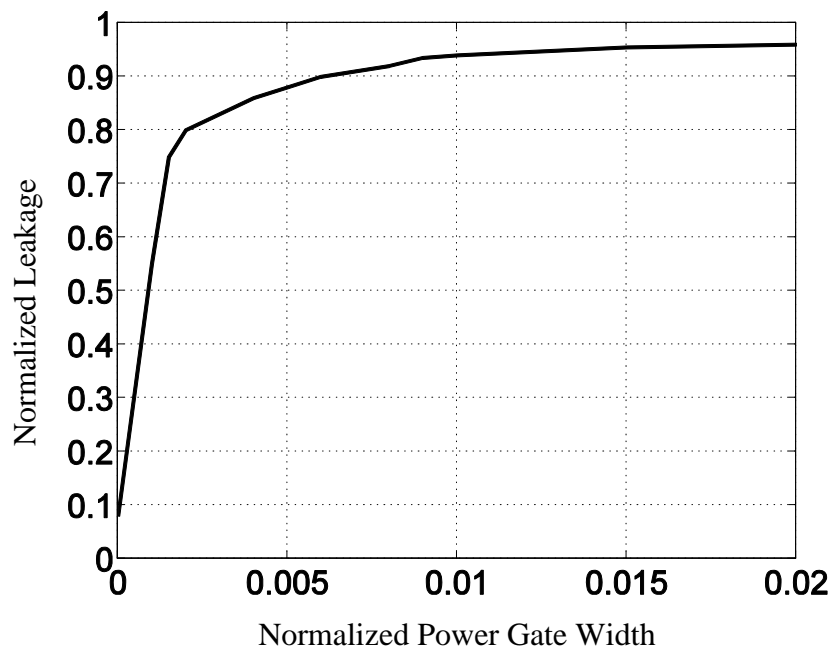


Figure 2-7: Measured leakage with varied power gate width from a 32nm SOI four-core x86 processor. By reducing the power gate width by disabling the idle current can be gradually reduced from 100% to a lowest bound of 10%.

leakage. In Figure 2-7, the leakage current is measured for a 32nm SOI four-core x86 processor SoC chip which has a variable-weighted footer based power gate ring around the core [9] on silicon hardware. By changing the power gate width by disabling distributed sections of the power switch ring via configuration bits, the idle current can be gradually reduced from 100% to a lowest bound of 10%.

### 2.2.4 Opportunity for Regulation

In order to assess the benefit of a programmable resistive power grid, we investigated the opportunity for dynamic grid voltage control and reduction in a commercial x86 four core processor SoC using typical P-state (power state) occupancy data. A P-state defines a voltage/frequency pair independently for each core in the processor, where P0 is the

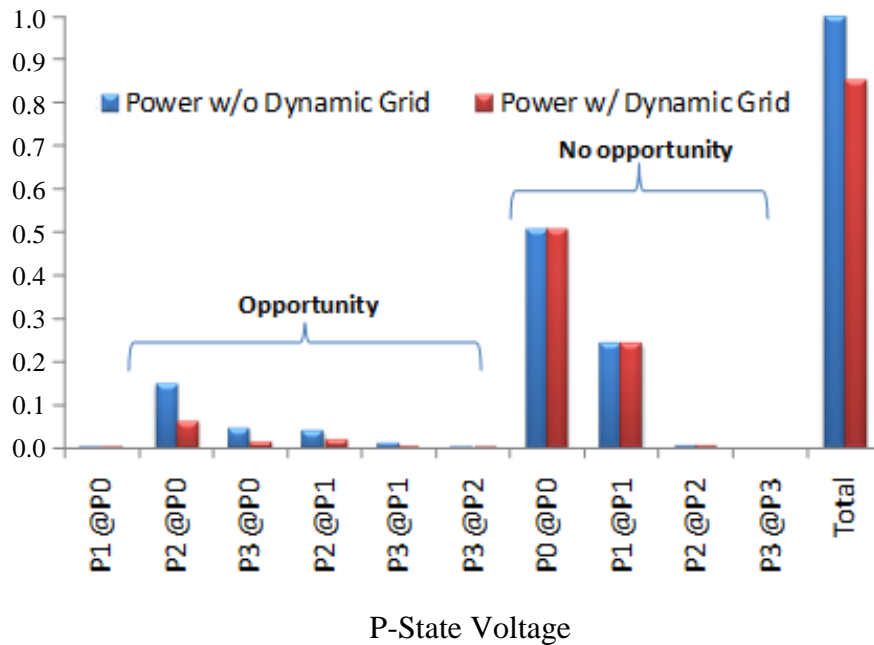


Figure 2-8: Estimated power savings of approximately 15% from a four-core x86 processor using a programmable resistive power grid. P1@P0 indicates the total power of core(s) that are running at the P1 frequency while another different core in the SoC is running at a P0 state.

fastest state and P3 is the slowest. Since all cores share a common  $V_{DD}$ , the lowest core P-state with the fastest frequency sets the operating  $V_{DD}$  for all cores, leading to non-optimal  $V_{DD}$  selection for cores with higher numbered P-states. The cores running at a higher  $V_{DD}$  than required are only able to use frequency scaling in the absence of a programmable resistive grid. Figure 2-8 shows the opportunity for power savings using a programmable resistive power grid. In the figure, the label P1@P0 indicates the total power of core(s) that are running at the P1 frequency while another different core in the SoC is running at a P0 state imposing a higher  $V_{DD}$  requirement. During the SysMark trace, the core-wise P-state occupancy is determined by the operating system. By including a programmable resistive grid at each core, the cores running at a higher P-state will run at their near optimal  $V_{DD}$  during periods of high activity. The figure shows up to ~15% power savings opportunity by using the programmable resistive grid technique. Power and performance results can vary depending on the P-state frequency, voltage settings, and the profile of system activity. By allowing individual core-wise voltage settings, the system has more flexibility to differentiate high performance workloads from lower performance workloads, which can allow opportunity for additional performance boosting when a single core is running at a low P-state.

### 2.2.5 Route Level Macro Model

A commercial power integrity tool, Apache Redhawk [10], was used to model the effectiveness of variable-weighted power switches as a controlled power supply resistance in a large system. In order to demonstrate the feasibility of using a controlled power supply resistance to create local effective voltages, Redhawk was used for a single power gated Route Level Macro (RLM) in a commercial 32nm core. An ideal  $V_{DD}$  and

$V_{SS}$  were applied to the power grids, which were modeled from metal 11 (M11) to M9 and M11 to M1 respectively. The virtual- $V_{DD}$  grid, modeled from M8 to M2, was measured as the power switch width and was varied in the Redhawk simulation. The internal net activity was generated from benchmark simulations of a thermal design point (TDP) benchmark. Figure 2-9 shows the average virtual- $V_{DD}$  and worst-case virtual- $V_{DD}$  observed for the TDP benchmark. The relatively flat part of the curves is due to the very low impedance sizing required for the maximum power switch size to enable high frequency operation at the maximum  $V_{DD}$ . To achieve programmable resistive grid regulation and maintain high frequency operation, fine-grained power switch partitioning with small widths is needed in the sub 0.2% range of the current total power switch

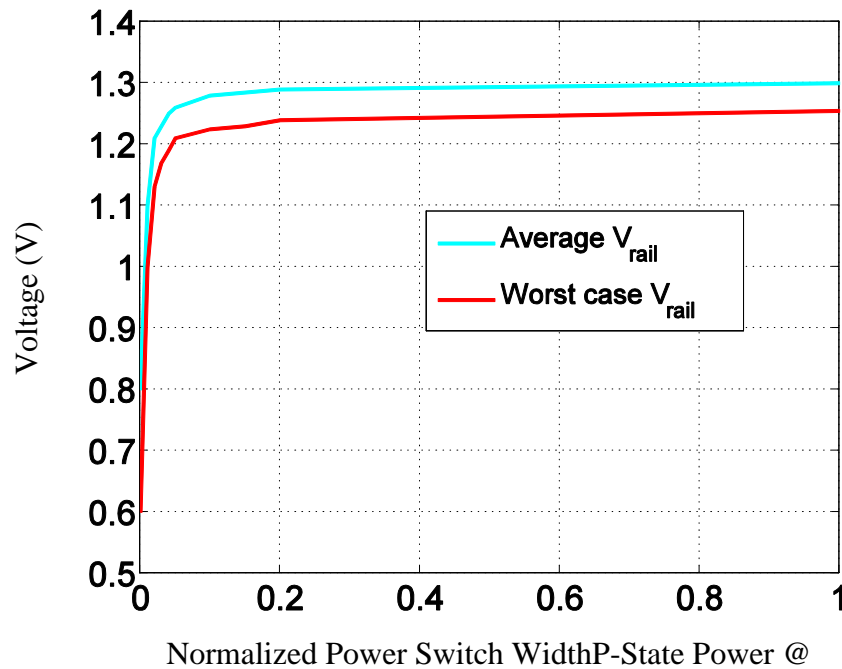


Figure 2-9: Average and worst case virtual- $V_{DD}$  of a single RLM during a TDP benchmark across power switch width. The flat area of the curve is due to low impedance sizing to maintain high speed operation. In this example, to achieve variable power gate regulation power gate partitioning is needed in the range of ~0.2% of total width.

width.

### 2.2.6 Core Level Modeling

Since the application of the programmable resistive power grid requires characterization of the power supply resistance, we proposed a design flow, using a commercial power integrity tool, for applying the approach to arbitrary digital designs. We used this design flow to model a full commercial processor using the proposed method for implementing a programmable resistive power grid. Apache Redhawk was used to model the effectiveness of variable-weighted power gates as a controlled power supply resistance in a large system. Redhawk was used to model the AMD Bulldozer core [11], which has a similar footer-based power switch ring structure seen in [9]. For this simulation, to prevent the simulation time from being prohibitively large, each Route Level Macro (RLM) (roughly 50 in total) in the Bulldozer core was modeled as a time-dependent current source and capacitance model, with the exception of the L1 cache and two RLMs

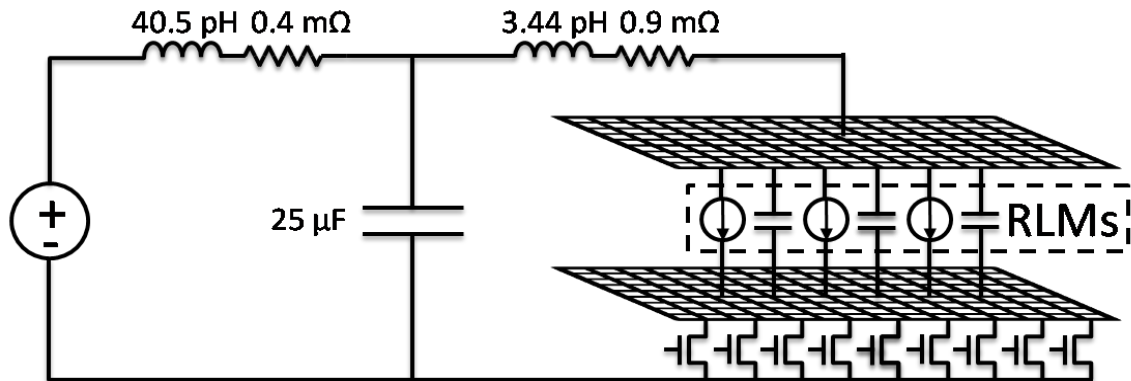


Figure 2-10: Apache Redhawk simulation set for Bulldozer core. A simplified package model was included to capture all intrinsic and parasitic RLCs. Each individual RLM was modeled as a capacitance and time dependent current source.

without available data. These current profiles were generated from simulations of the Double-precision General Matrix Multiply (DGEMM) benchmark. A simplified package model was included to capture the real RLC effects seen on hardware. Figure 2-10 shows a simplified diagram of the simulation setup.

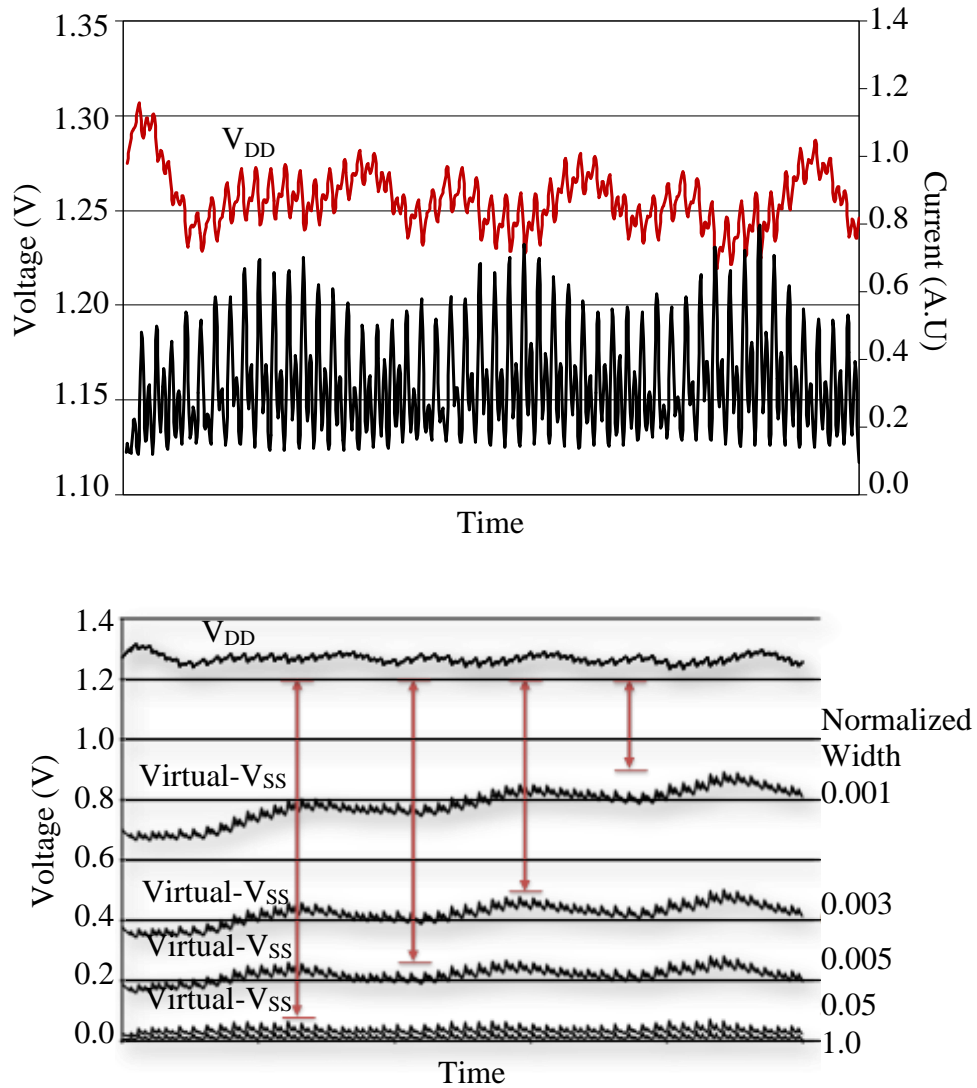


Figure 2-11: (top)  $V_{DD}$  and current response from Redhawk (bottom)  $V_{DD}$  and virtual- $V_{SS}$  profile over time for different normalized power switch width. All RLMs are super imposed onto the plot. The negligible variance between RLMs shows the power grid is robust, and the power gate resistance is the dominate factor in the regulation.

### 2.2.7 Power Delivery Network Impact

#### *Local IR Drop*

Figure 2-11 shows the average  $V_{DD}$  response to the applied current models in time, showing that our current model profile was functioning correctly. Figure 2-11 shows the  $V_{DD}$  and virtual- $V_{SS}$  voltage profile over time for different normalized power switch widths. Notice that at 5.07%, the virtual- $V_{SS}$  is only slightly above the 100% case, which is expected for a power gate designed for a high performance core. This figure also includes every RLM's  $V_{DD}$  and virtual- $V_{SS}$  superimposed into a single graph. The negligible variance in  $V_{DD}$  and virtual- $V_{SS}$  between RLMs across the chip is due to the robust power grid with low resistance. This shows that the dominant factor in the virtual- $V_{SS}$  droop is caused by the controlled power supply resistance of the power switches. Through variable-weighted power switches, we are able to achieve a programmable resistive power grid capable of a wide range of virtual- $V_{SS}$  supplied to the core.

#### *di/dt Noise Reduction*

A major design challenge for the resistive grid is managing  $di/dt$  noise associated with voltage scaling. In a power gated system,  $di/dt$  noise occurs when returning a power gated component to the full  $V_{DD}$  after being idle. Returning the virtual- $V_{DD}$  to the full  $V_{DD}$  generates a sudden rush current. The rush current will cause noise due to the RLC components in the power delivery network. One method of turning on a power switch to reduce noise is to slowly ramp the gate voltage, limiting the amount of current that can pass through the transistor. However this is impractical as the virtual- $V_{DD}$  charge will be large. We want to reduce noise while still being able to maintain fast local  $V_{DD}$  regulation. For the purpose of analysis, in this section we assume a power delivery

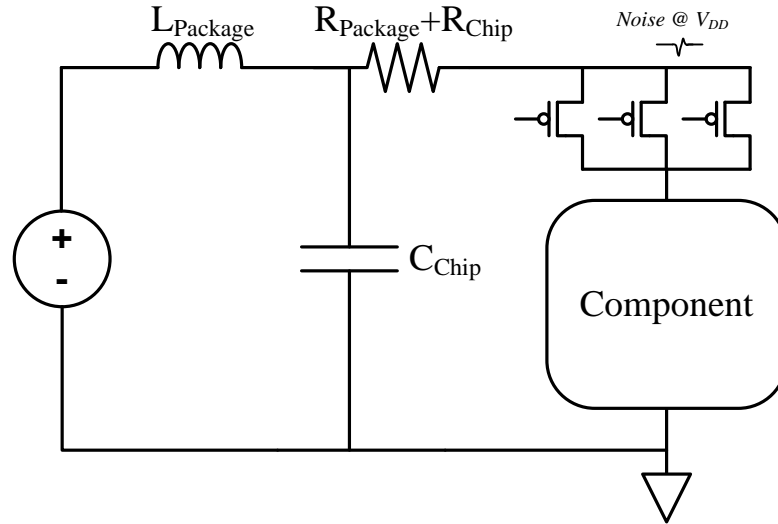


Figure 2-12: Modified RLC model used for this analysis. We simplify the model and lump the package and chip resistance together.  $C_{Chip}$  include the local chip capacitance as well as the decoupling capacitance. We also included a package inductance.

network configuration that is shown in Figure 2-12. We simplify the model and lump the package and chip resistance together.  $C_{Chip}$  includes the local chip capacitance as well as the decoupling capacitance. We also included a package inductance. This model is used to evaluate the local noise seen on the  $V_{DD}$  rail.

In [13] it is demonstrated how staggered power switch turn-on can be used to minimize  $di/dt$  noise. [13] divides the power gate into 48 equally sized power switches with uniform delay between device turn-on. This dissertation expands on this idea, but explores the noise reduction when the power switches are not equally sized. For simplicity sake, we assume a programmable resistive power grid configuration with three power switches, sized  $W_1$ ,  $W_2$  and  $W_3$  and two delays between power gate turn-on,  $D_1$  and  $D_2$  (Figure 2-12). We fix the total power switch width,  $W_{Total}$  to be a constant and explore the design trade-off as we vary  $W_1$  and  $W_2$  and we assume the worst case turn-on

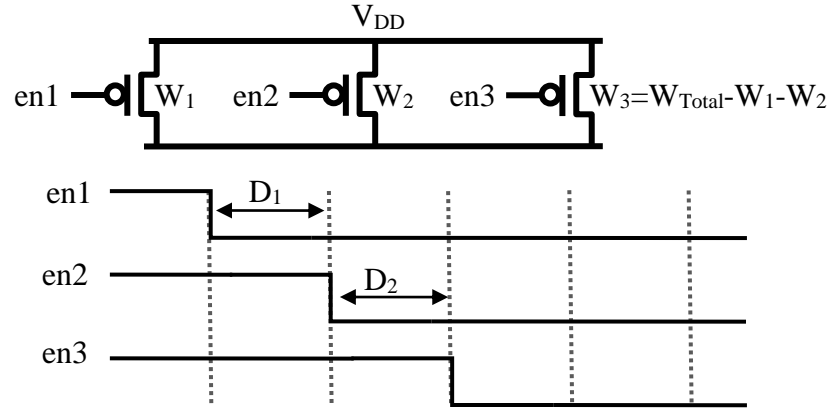


Figure 2-13: Schematic and timing diagram of power gate partitioned into three parts ( $W_1$ ,  $W_2$ , and  $W_3$ ).  $W_{Total}$  is fixed while  $W_1$  and  $W_2$  will be varied.  $W_3$  is equivalent to  $W_{Total} - W_1 - W_2$ .

from power gating (i.e., virtual- $V_{DD} \sim 0V$ ). With the turn-on time and noise trade-off, the optimal noise scenario is to have all power switches contribute an equal amount of noise. Since  $W_1$  has the largest voltage potential, it has the largest potential rush current.

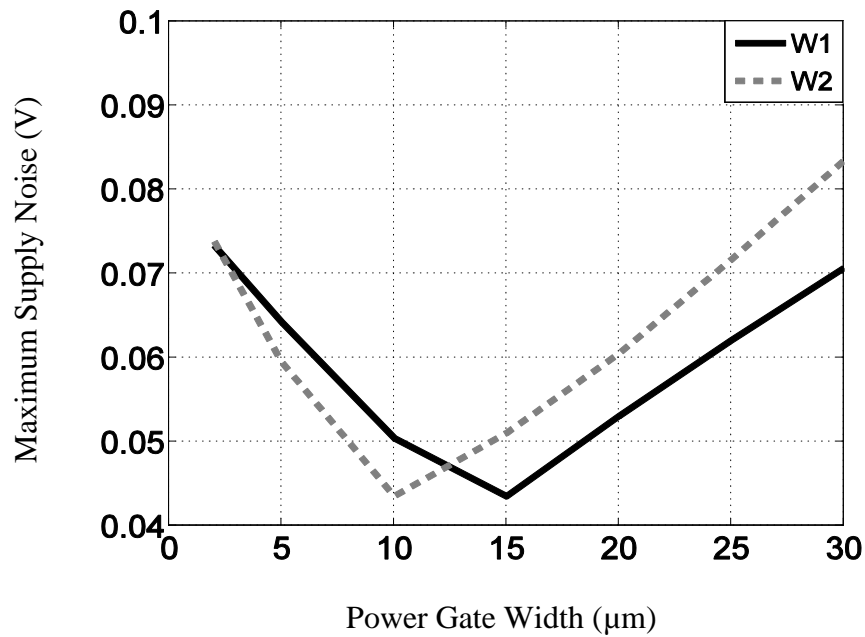


Figure 2-14: Maximum voltage supply noise holding  $W_1$  constant at  $10\mu m$  and sweeping  $W_2/W_3$  and holding  $W_2$  constant at  $15\mu m$  and sweeping  $W_1/W_3$ .  $W_{Total} = 60\mu m$  and  $D_1 = D_2 = 3$  ns.

$W_2$  has the second largest potential rush current, and  $W_3$  has the smallest. To achieve equal noise, per power switch the sizing relationship should be  $W_1 < W_2 < W_3$ . Figure 2-14 shows the impact of  $W_1$  and  $W_2$  on noise for a given total power switch width of  $60\text{ }\mu\text{m}$ . The plot shows that there is an optimal point for both  $W_1$  and  $W_2$  to minimize noise. As predicted, the optimal  $W_1$  is less than the optimal  $W_2$ . Thus, uniform division of a power gate into parts is not the best solution to minimize noise.

Figure 2-15 shows the  $V_{DDH}$  noise reduction benefit of the optimally configured weighted power switches compared to a same-size monolithic power switch. The total power switch size is  $60\text{ }\mu\text{m}$  with the ratio  $W_1:W_2:W_3 = 1:1.25:3.25$  and the total turn on delay for the weighted power switches ( $D_1+D_2$ ) is  $6\text{ ns}$ . This approach reduced the maximum voltage supply noise from  $140\text{ mV}$  to  $40\text{ mV}$  – a  $3.5\text{X}$  reduction. As mentioned above,

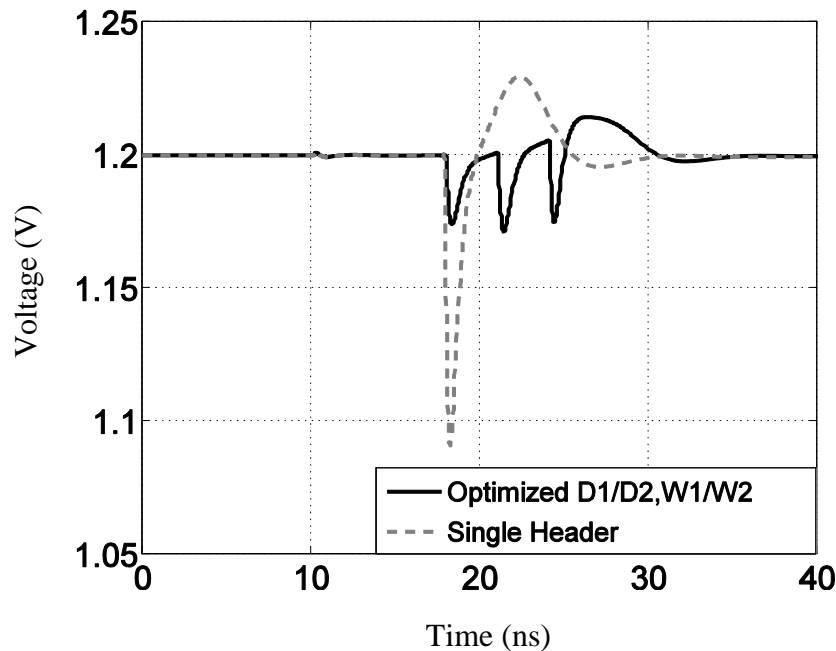


Figure 2-15: Noise on the voltage supply rail from switching with the optimally sized power gates and with a single power gate. The maximum voltage supply noise from  $140\text{ mV}$  to  $40\text{ mV}$  – a  $3.5\text{X}$  reduction.

this sizing makes the noise contribution of each variable power switch almost the same, leading to optimal noise on  $V_{DD}$  given some additional delay for charging the virtual rail up to the nominal voltage.

### 2.3 Summary & Conclusions

In this chapter, we demonstrated work that allows for voltage scaling without relying on DC-DC converters, and provides independent core  $V_{DD}$  scaling in multi-core designs. Since different applications have varying workload requirements, an energy efficient solution such as DVFS is required for optimal energy efficiency. However, DVFS often uses off-chip DC-DC converters to scale  $V_{DD}$  of the entire chip. These DC-DC converters are relatively slow and limit the ability to scale  $V_{DD}$ . With multi-core designs, all cores generally share a common  $V_{DD}$ .  $V_{DD}$  scaling can only happen when all cores in the design have a workload requirement that allows for a voltage change.

A programmable resistive power grid partitions a monolithic power switch into parallel independently controlled power gates with different widths to enable fine-grained  $V_{DD}$  scaling. With this technique, we demonstrated how to address the major design challenges in a power delivery network while improving energy efficiency.

To address the first design challenge of voltage scaling, we demonstrated how a resistive power grid is used to scale  $V_{DD}$  and reduce energy during active mode and to limit leakage current through measurements from a 90nm test chip and a 32nm x86 processor, respectively. The voltage scaling is achieved without relying on off-chip DC-DC converters. Instead a controlled power grid resistance through variable power switch widths can be used to provide a large number of virtual- $V_{DD}$ s to a block during active

operation. Through extensive simulation and modeling, we can select the correct amount of power switch width needed for various desired operating modes.

To address the second design challenge of  $V_{DD}$  granularity, we showed how the architecture could be used to scale the virtual- $V_{DD}$  at the component level, independent of the other components. We discussed the opportunity for using the programmable resistive power grid in a commercial x86 four-core SoC.

To address the third design challenge of IR drop, we showed how a commercial power integrity tool can be used to model a resistive power grid in a commercial SoC. With our model, we showed that the IR drop was controlled globally by the power switch resistance, and local IR drop across the design was not an issue.

Finally, to address the fourth design challenge of  $di/dt$  noise, we demonstrated a technique to minimize noise by using a controlled turn-on methodology with unequal power gate sizes.

Using variable-weighted power switches as a programmable resistive power grid is a low cost solution for providing component level voltage scaling without the use of DC-DC converters. We demonstrated dynamic energy savings of 30%, and leakage reduction of 90%.



# Chapter 3

## Optimizing Voltage Scalable

## Architectures with Multiple $V_{DD}$ s

### 3.1 Background

As mentioned in the previous chapters, DVFS is the conventional solution for adjusting energy consumption based on varying workload requirements. When timing slack exists in a given application, DVFS adjusts the supply voltage and frequency to match a circuit's workload, providing quadratic energy savings at these lower workload requirements (Figure 1-3). Traditionally, DVFS implementations suffer from coarse spatial and temporal granularities. Spatial granularity is the ability to assign different components in a design to different voltages. Most recent DVFS implementations are limited to a spatial granularity at the microprocessor core level to entire chip [7][13][14][15]. Temporal granularity refers to the speed at which the  $V_{DD}$  to a component can change. DVFS techniques generally rely on DC-DC converters to adjust  $V_{DD}$ . These off-chip DC-DC converters traditionally limited temporal granularity, since

they take tens to hundreds of  $\mu\text{secs}$  to adjust the  $V_{DD}$  [6]. The coarse spatial and temporal granularity of traditional DVFS limits the energy efficiency of these systems.

Further, these coarse grained DVFS blocks are typically supplied a voltage that is generated directly by a DC-DC converter. Practical cost considerations of DC-DC converters, such as on-chip area and off-chip passives, can limit the number of blocks that can be supplied with separate supply voltages. More recent work on integrated DC-DC converters shows significant speedups in switching time, for example a  $>1V$  transition is achieved in roughly 20ns in [16], but including dedicated DC-DC converters for each block is still impractical for designs with fine-grained spatial granularity that have many distinct power regions. To maximize energy efficiency for varying workload requirements, DVFS would ideally support voltage control across a broad range for multiple blocks with fine-grained spatial and temporal granularity.

In the previous chapter, we presented a power delivery system modification to improve energy efficiency in a system with a single common  $V_{DD}$  and without DC-DC converters for scaling, or the inclusion of extra DC-DC converters. In this chapter, we present a technique that is used in systems that already have multiple  $V_{DDs}$  available. The multiple  $V_{DDs}$  can be generated from multiple DC-DC converters or multiple output DC-DC converters, but in either situation we do not require the DC-DC converters for scaling  $V_{DD}$ . Panoptic Dynamic Voltage Scaling is a power delivery network modification that leverages multiple PMOS power switches at the component level to enable fine-grained  $V_{DD}$  scaling without the use of DC-DC converters for voltage scaling.

### 3.2 Panoptic Dynamic Voltage Scaling

To meet these needs for improving energy efficiency, we implement a method called Panoptic (“all-inclusive”) Dynamic Voltage scaling (PDVS) [17]. To improve spatial and temporal granularity, PDVS uses multiple PMOS power switches at the component level, Figure 3-1, to provide a local  $V_{DD}$  (virtual- $V_{DD}$ ) from a discrete set of chip-wide shared  $V_{DDs}$  (e.g.,  $V_{DDH}$ ,  $V_{DDM}$ ,  $V_{DDL}$ ). This allows for an individual component’s virtual- $V_{DD}$  to be set independently from any other component as well as allowing for fast local  $V_{DD}$  switching. The use of voltage dithering [18], or using a division of operations across two voltage/frequency points to approximate an effective intermediate operating point, further enables the approach to closely approximate an ideal energy/performance trade-off across

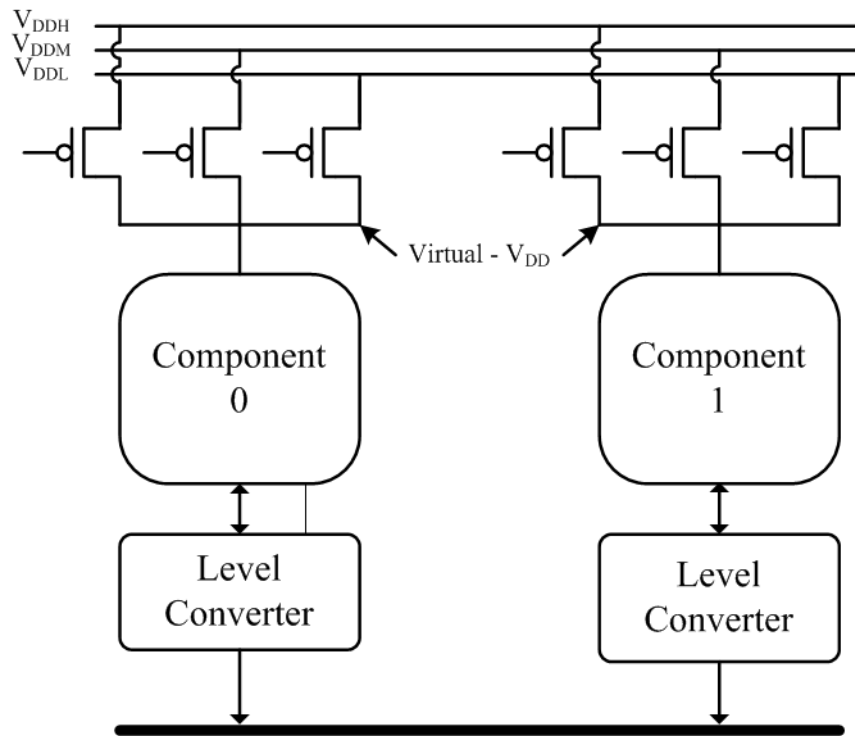


Figure 3-1: The PDVS architecture. Multiple power switches are used for each component connecting each to chip-wide  $V_{DDs}$  which enables fine spatial and temporal granularity DVFS granularity. Level converters are used to prevent short circuit current between components at different  $V_{DDs}$ .

a broad range.

### 3.2.1 Approach

To explore the full benefits of PDVS, we designed a 32b data flow processor, Figure 3-2, capable of executing arbitrary data flow graphs (DFGs) at 1 GHz at 1.2V. We used the PDVS architecture to implement the data path of the processor. The data path consists of four Baugh-Wooley multipliers and four Kogge-Stone adders. Each of these components uses three PMOS power switches tied to the three  $V_{DDs}$  ( $V_{DDH}$ ,  $V_{DDM}$ , and  $V_{DDL}$ ) that are common throughout the processor. The processor includes a programmable crossbar that feeds input registers of the data path components either directly from the data path, the register bank, or the memory. To prevent short-circuit current from blocks operating below the nominal  $V_{DD}$ , level converters (LCs) are used at the output of each multiplier and adder to up-convert their outputs to the  $V_{DDH}$  level that is used at the register file.

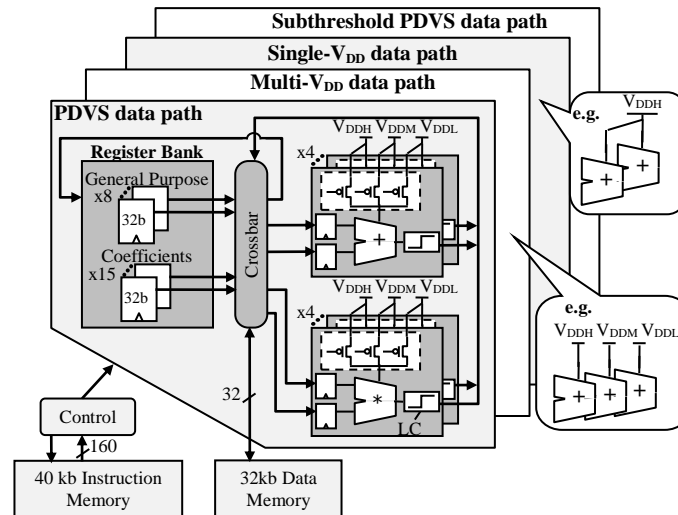


Figure 3-2: Block diagram of the PDVS data flow processor. SRAMs and control serve four data paths for direct comparison of PDVS with  $SV_{DD}$  &  $MV_{DD}$ . The data path consists of four Baugh-Wooley multipliers, four Kogge-Stone adders, a programmable crossbar, and level converters to prevent short-circuit current.

In order to provide a fair hardware comparison, we include three additional data paths on the chip that are functionally identical to the PDVS data path, but that use different power delivery network options: single- $V_{DD}$  ( $SV_{DD}$ ), multi- $V_{DD}$  ( $MV_{DD}$ ) [14][15], and a subthreshold optimized PDVS data path discussed in section 4.2. In the  $SV_{DD}$  data path, the four multipliers and adders all share the same  $V_{DD}$ . In the  $MV_{DD}$  data path, the four multipliers and adders are permanently tied to either  $V_{DDH}$ ,  $V_{DDM}$ , or  $V_{DDL}$ , and operations can be scheduled for execution on any of these components based on the workload requirements. The processor has a 32kb data memory and a 40kb instruction memory that are shared for all of the data paths. The control word for controlling the data flow (and power switch control where applicable) of the various data paths is 160b for this test chip.

### 3.2.2 PDVS Overheads

There are overheads associated with the PDVS architecture compared to  $SV_{DD}$  and  $MV_{DD}$ . The primary overheads are the area, energy, and delay overheads associated with the inclusion of LCs and the power switches associated with PDVS compared to  $SV_{DD}$  and  $MV_{DD}$ . The adder and multiplier have 2.4% and 1.7% power switch area overhead, and 11.4% and 2.1% level converter (LC) area overhead, respectively. From simulations, we see the LCs have a 32.0% and 2.0% LC delay overhead, and 8.0% and 0.3% LC energy overhead for converting from 0.8V to 1.2V (Figure 3-3a) relative to a single addition or multiplication operation in  $SV_{DD}$ . From simulations, we also see the power switches have a 35% and 12% delay overhead, and 215% and 10% energy overhead switching the virtual- $V_{DD}$  from 0.8V to 1.2 (Figure 3-3b) relative to a single addition or multiplication operation in  $SV_{DD}$ . Although the overheads may appear large at first, they

actually are minor in the overall timing and energy budget, since the multiplier dominates DFG delay and energy. Using the power switches to switch the components' virtual- $V_{DD}$  from a lower  $V_{DD}$  to a higher  $V_{DD}$  (i.e.,  $V_{DDL}$  to  $V_{DDH}$ ) incurs both energy and delay overheads. For the adder and multiplier, this energy overhead leads to breakeven times of  $<4$  and  $<1$  operations for the adder and multipliers, respectively. This means that the multiplier can switch to the low voltage, execute just one instruction, and then switch back to the high voltage and save energy relative to executing that one instruction at the high voltage. However, the adder must execute four consecutive instructions at the low voltage in order to overcome the energy overhead of  $V_{DD}$  switching. As mentioned previously, since the multiplier dominates the DFG energy, the energy benefit of PDVS overwhelms the overheads.

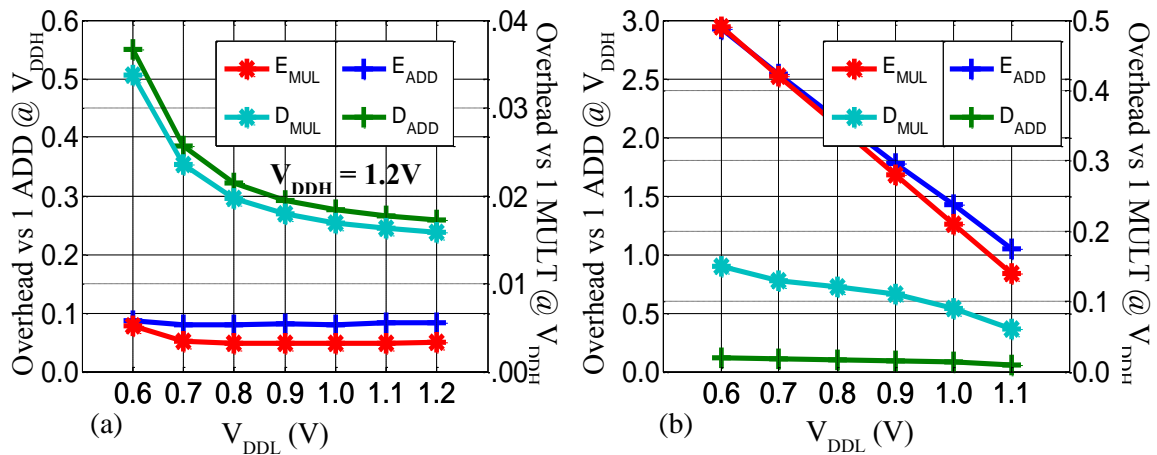


Figure 3-3: (a) Simulated level conversion overhead varying  $V_{DDL}$  for both the adder and multiplier. (b) Simulated virtual- $V_{DD}$  switching overhead varying  $V_{DDL}$  for both the adder and multiplier. With these overheads we achieve a breakeven cycle of  $<4$  and  $<1$  for the adder and multiplier, respectively.

### 3.2.3 Results

#### *Approach*

In order to compare post-fabricated results with our design, we set up a test platform that generates inputs and compares outputs and allows us to run the same benchmarks on a VHDL model, Spectre netlist, and physical hardware. The benchmarks are developed in VHDL, and custom scripts translate them into a Spectre stimulus file and a VHDL state machine for the Spectre simulation and hardware testing, respectively. Only the Spectre simulation and test chip are used to measure energy. We use a custom synthesis script to

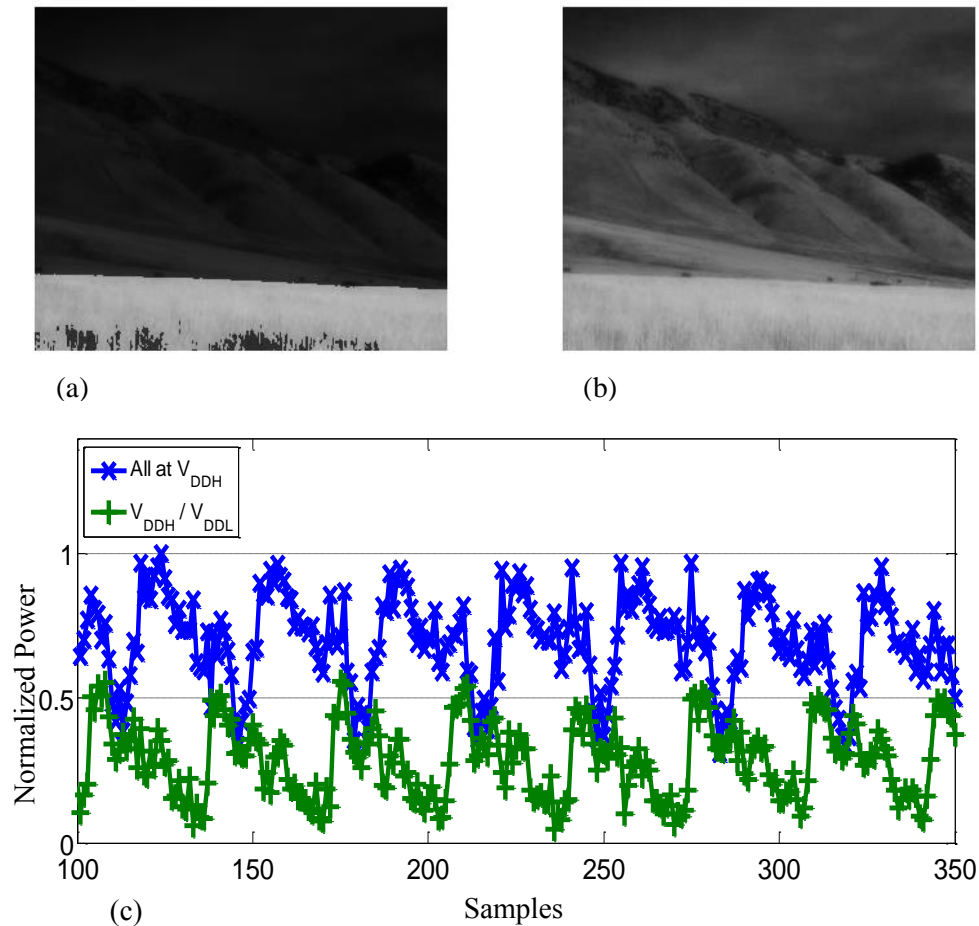


Figure 3-4: (a) Original image fed into chip (b) post processed image from the chip (c) normalized measured instantaneous power from our demonstration. All at  $V_{DDH}$  has all  $V_{DDs}$  at the nominal 1.2V,  $V_{DDH}/V_{DDL}$  are set at 1.2V and 0.7V respectively

map the benchmark DFGs to the architecture and use MATLAB to create the 160b instruction words. We designed two custom PCBs to test and measure the four different data paths on our test chip. As previously mentioned, our comparison is between three energy efficient topologies:  $SV_{DD}$ ,  $MV_{DD}$  and PDVS. To achieve a fair comparison, all of the measured data in this section comes from a single data path (PDVS) implementing various techniques to emulate the other two energy efficient topologies. To emulate  $SV_{DD}$ , we power the three rails using the same voltage source and enable all the power switches to minimize resistance across the power switch. To emulate  $MV_{DD}$ , we assign a different voltage to each rail and make sure each component is powered by only one of these rails at all times (i.e., no power switch switching is allowed).

### ***Demonstration***

As a demonstration of the benefits of PDVS, we implemented a video processing application that brightens dimly lit areas of a frame. The workload of this application varies according to the number of dark pixels of each frame. The number of dark pixels that need to be brightened can easily be calculated from each image. With this information, we can compute the workload needed to achieve the required application rate, e.g., 24 frames per second (FPS) or 30 FPS. For the sake of simplicity, we created a program that brightens pixels by multiplying the pixels below a threshold value by a specific constant. The multiplication can be done either at  $V_{DDH}$  or  $V_{DDL}$  (i.e. fast or slow). With the knowledge of the total number of pixels that need brightening, we calculate the number of multiplications that will be scheduled at  $V_{DDH}$  and the multiplications scheduled at  $V_{DDL}$ .

Since our test chip was a custom processor designed to demonstrate the benefits of PDVS, it does not contain any operating system that enables video data input, so we feed each frame as an image to the input data. We can see in the Figure 3-4a/b the demo images before and after the processing that was executed on our test chip. The graph in Figure 3-4c shows the measured instantaneous power consumption during the demo for all  $V_{DDs}$  set to the  $V_{DDH}$  value (1.2V) and with  $V_{DDH}$  at 1.2V and  $V_{DDL}$  at 0.7V. To obtain this data, we used a lower demo frequency to accommodate data transfer to/from the chip and removed leakage. With this example, we are able to see a 40% average power reduction by using PDVS.

### ***Measured Results***

Our test chip was fabricated in a commercial 90nm bulk CMOS process. The average of

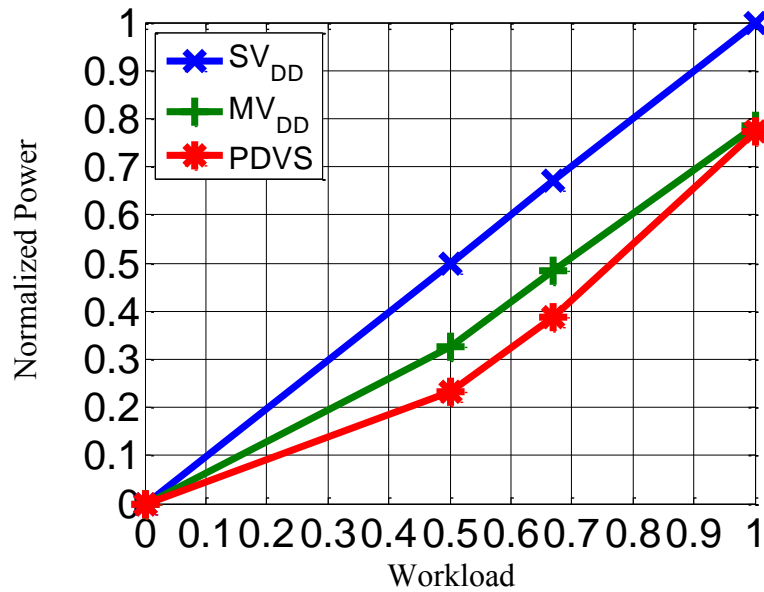


Figure 3-5: Average measured power (w/ overheads) vs. workload across 7 different DFGs. PDVS gives lower energy for varying workloads than MV<sub>DD</sub> by operating components at lower  $V_{DDs}$  when possible while MV<sub>DD</sub> components are hard-tied to higher voltages. PDVS power switches enable  $V_{DD}$  dithering (rapid switching between two  $V_{DD}$  rate pairs) to approximate ideal DVS.

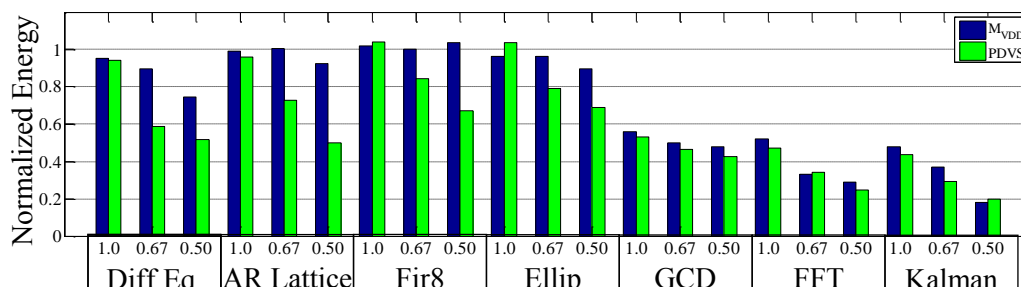


Figure 3-6: Measured energy benefit (including overhead) of  $PDVS$  &  $MV_{DD}$  normalized to  $SV_{DD}$ .  $PDVS$  shows up to 80% and 43% energy savings over  $SV_{DD}$  and  $MV_{DD}$ , respectively

seven measured DFG energies, shown in Figure 3-5, demonstrates  $PDVS$  savings across various workloads. Given the same area constraint,  $PDVS$  gives lower energy for varying workloads than  $MV_{DD}$  by operating components at lower  $V_{DD}$ s when possible while  $MV_{DD}$  components are hard-tied to higher voltages.  $PDVS$  power switches enable  $V_{DD}$  dithering (rapid switching between two  $V_{DD}$  rate pairs) to approximate ideal DVS, providing the energy vs. rate profile that lies on the line between these points. With a nominal workload of 1, the  $PDVS$  and  $MV_{DD}$  curves have slightly lower energy than

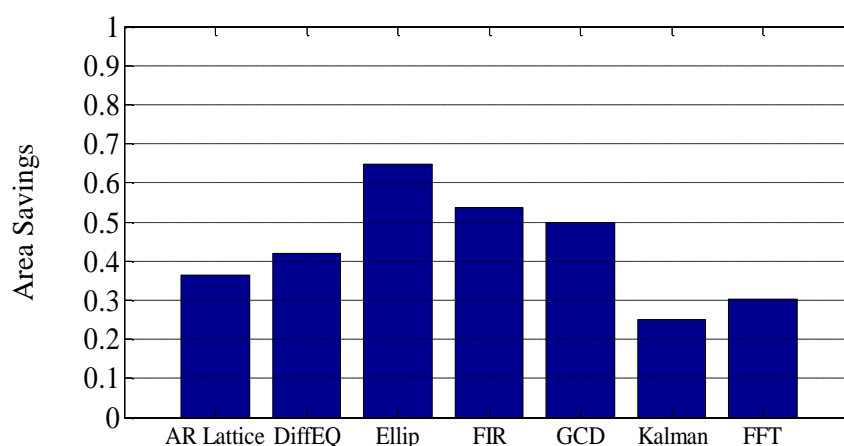


Figure 3-7: Area savings of  $PDVS$  over  $MV_{DD}$  for the same energy constraint.  $PDVS$  saves up to 65% area over  $MV_{DD}$ , since individual components are reused at different voltages while  $MV_{DD}$  requires multiple copies of each component at different voltages.

$SV_{DD}$  since timing slack was removed by running some components at lower  $V_{DDs}$ . Figure 3-6 shows results for the seven benchmark DFGs we ran on all the data paths to demonstrate PDVS's benefits for various rates. As the workload rate decreases, the energy benefits increase due to the timing constraint being relaxed. For the given DFGs, the PDVS data path shows up to 80% and 43% energy savings over  $SV_{DD}$  and  $MV_{DD}$ , respectively. The largest energy savings were in the FFT DFG. This is because the FFT heavily uses the multiplier, which dominates the energy budget. Given unlimited area,  $MV_{DD}$  can theoretically provide the same energy as PDVS for a given DFG by having the exact number of components needed at a given  $V_{DD}$  to implement a given DFG across workloads. To illustrate this point, assume a trivial DFG requires three multipliers at  $V_{DDH}$  for a workload of 1, two multipliers at  $V_{DDM}$  and one at  $V_{DDL}$  for a workload of 0.66, and three multipliers at  $V_{DDL}$  for a workload of 0.5. PDVS could achieve all workloads with only three multipliers, while  $MV_{DD}$  would require eight. For our non-trivial DFGs given the same energy constraint, PDVS saves up to 65% area (Figure 3-7) over  $MV_{DD}$ , since it allows individual components to be reused at different voltages while  $MV_{DD}$  would require multiple copies of each component at different voltages.

### 3.2.4 Power Delivery Network Impact

#### *Noise Analysis*

As is the case to power gating and a programmable resistive power grid,  $di/dt$  noise in the power delivery network is a concern for PDVS. Switching from a power gated state (i.e., virtual- $V_{DD}=0$ ) or from  $V_{DDL}$  to  $V_{DDH}$  will generate rush current which creates noise in the power delivery network. This noise generated when one component is switched to  $V_{DDH}$ , will impact the other blocks that are operating at  $V_{DDH}$ . In the previous chapter, we

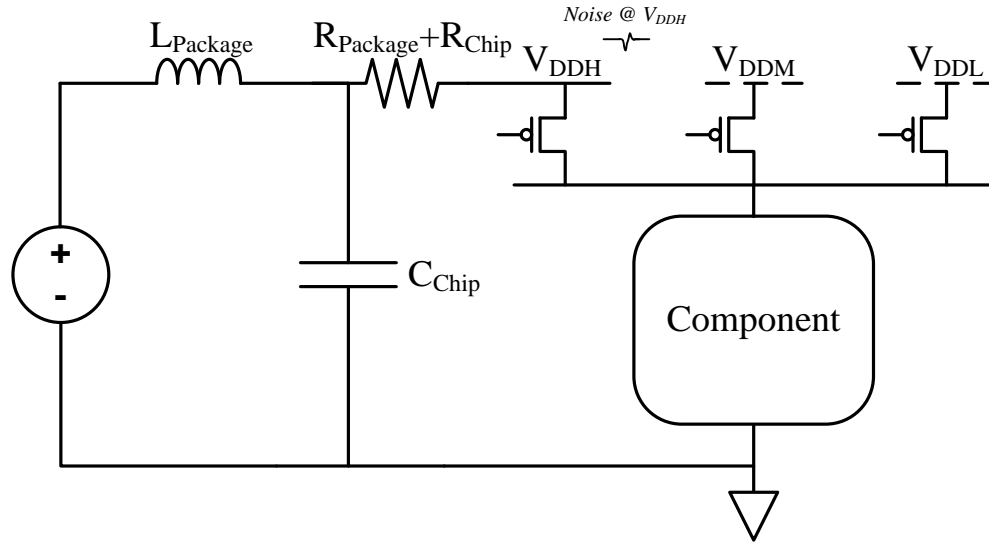


Figure 3-8: Modified simple RLC model used for noise analysis. We simplify the model and lump the package and chip resistance together.  $C_{Chip}$  include the local chip capacitance as well as decoupling capacitance. We also include a package inductance. We only consider the noise seen on  $V_{DDH}$ .

discussed how a programmable resistive power grid reduces  $di/dt$  noise. Another similar technique takes fingered power switches and introduces a time delay between different parts of the same power switch leading to a gradual turn-on. Those techniques are all generally applicable to PDVS; however we can leverage the PDVS architecture to reduce noise without the additional overheads associated with other schemes.

In [20] we presented a novel power switch methodology for reducing  $V_{DD}$  switching energy called Stepped Supply Voltage Scaling (SVS). With SVS, energy is saved by switching from  $V_{DDL}$  to  $V_{DDM}$  to  $V_{DDH}$  instead of switching from  $V_{DDL}$  directly to  $V_{DDH}$ . While switching from  $V_{DDL}=0.3$  to  $V_{DDH}=1.2$ , SVS saves up to 45% energy. SVS is also used as an effective technique to lower power supply noise. By stepping to  $V_{DDM}$  when transitioning from  $V_{DDL}$  (or a power gated state) to  $V_{DDH}$ , SVS not only has inherent

staged turn-on, but SVS also lowers noise because the rush current being drawn from the supply is broken up into two steps.

Figure 3-8 shows the modified power delivery network we simulated, which is similar to the noise analysis done in the previous chapter. We simplify the model and lump the package and chip resistance together.  $C_{Chip}$  includes the local chip capacitance as well as decoupling capacitance. We also include a package inductance. In this example, we only consider the noise seen on  $V_{DDH}$ . We evaluated the noise benefit of SVS by simulating a 32b Kogge Stone adder along with characteristic power delivery network package inductance of 10nH, rail and package resistance of 20ohm, and on-chip decoupling capacitance of 10pF. Parasitic virtual- $V_{DD}$  and power switch capacitances were included in the simulation. Peak to peak noise values with and without SVS are reported in Table 3-1. The table shows noise values on  $V_{DDH}$  when the adder transitions from  $V_{DDL}$  to  $V_{DDH}$  with one step at  $V_{DDM}$ . The value of  $V_{DDM}$  in each case was taken to be halfway between  $V_{DDL}$  and  $V_{DDH}$ . We compare SVS with the conventional methodology of a direct transition  $V_{DDL}$  directly to  $V_{DDH}$ . This conventional case is referred to as “Without SVS”. SVS helps reduce power supply noise by over 40%.

Table 3-1: Comparison of peak power supply noise transitioning from  $V_{DDL}$  to  $V_{DDH}$  with and without SVS

$V_{DDL}$	<b>With SVS</b> $V_{DDL}$ to $V_{DDM}$ to $V_{DDH}$	<b>Without SVS</b> $V_{DDL}$ to $V_{DDH}$
0.3V	80 mV	137 mV
0.6V	55 mV	105 mV
0.9V	33 mV	58 mV

### ***DC-DC Converter Efficiency***

Unlike traditional DVS techniques, PDVS does not require the use of DC-DC converters to scale the voltage of an entire chip or core. As discussed, instead PDVS uses PMOS power switches to connect each block to different voltages. These voltages are assumed to be generated by independent DC-DC converters that have low efficiency for low current loads which could impact breakeven times due to energy inefficiencies. The authors of [21] created a system-level DC-DC converter model and compared PDVS to a traditional DVS scheme. The authors found that when considering the full power delivery network, PDVS has a breakeven time  $\sim 30\times$  faster. As the system the PDVS architecture is applied to is scaled up, the breakeven time and energy savings will improve due to the overall current load on the DC-DC converters increasing in these larger systems, improving the DC-DC converter efficiency.

### **3.3 Summary & Conclusions**

In the previous chapter, we demonstrated a power delivery modification to systems that only have one common  $V_{DD}$ . However, in this chapter we demonstrated a power delivery network modification that can be used in systems that have multiple  $V_{DDs}$  available. PDVS leverages multiple PMOS power switches at the component level to select from the multiple  $V_{DDs}$  available to enable fine-grained  $V_{DD}$  scaling. Many systems across the broad design space have applications that require high performance. However, due the varying nature of their applications, the workload requirements remain below this upper limit for the majority of their lifetime. Since different applications have varying workload requirements an energy efficient solution, such as dynamic voltage and frequency scaling

(DVFS), is needed. Traditionally, DVFS implementations suffer from coarse spatial and temporal granularities. DVFS techniques generally rely on DC-DC converters to adjust  $V_{DD}$ . These off-chip DC-DC converters traditionally limited temporal granularity, since they take tens to hundreds of  $\mu$ secs to adjust the  $V_{DD}$ . The coarse spatial and temporal granularity of traditional DVFS limits the energy efficiency of these systems. Further, these coarse grained DVFS blocks are typically supplied a voltage that is generated directly by a DC-DC converter. To maximize energy efficiency for varying workload requirements, DVFS would ideally support voltage control across a broad range for multiple blocks with fine-grained spatial and temporal granularity.

To address the first design challenge of voltage scaling, we presented the PDVS architecture. Unlike traditional DVS implementations, PDVS does not use dedicated DC-DC converters; instead it modifies the local on-chip power delivery network with PMOS power switches to select from a discrete set of voltages depending on the applications' workload.

To address the second design challenge,  $V_{DD}$  granularity, we demonstrated the first processor implementing PDVS. This processor demonstrates single clock cycle  $V_{DD}$ -switching at the component level and implements integrated  $V_{DD}$  dithering for near optimal energy scalability.

Finally to address the design challenge of  $di/dt$  noise, we showed how SVS could be used to reduce noise by transitioning from  $V_{DDL}$  to  $V_{DDM}$  to  $V_{DDH}$  instead of just  $V_{DDL}$  to  $V_{DDH}$ . This technique showed a 40% reduction in peak-to-peak noise.

Through spatial and temporal granularity, PDVS is able to improve energy efficiency over the conventional architectures,  $SV_{DD}$  and  $MV_{DD}$ . Measured energy savings across seven benchmark DFGs show an energy savings up to 80% and 43% over  $SV_{DD}$  and  $MV_{DD}$ . Our 32b data flow processor was designed and fabricated in a conventional 90nm CMOS process (Figure 3-9). Table 3-2 shows our test chip summary and Table 3-3 compares PDVS to the state of the art.

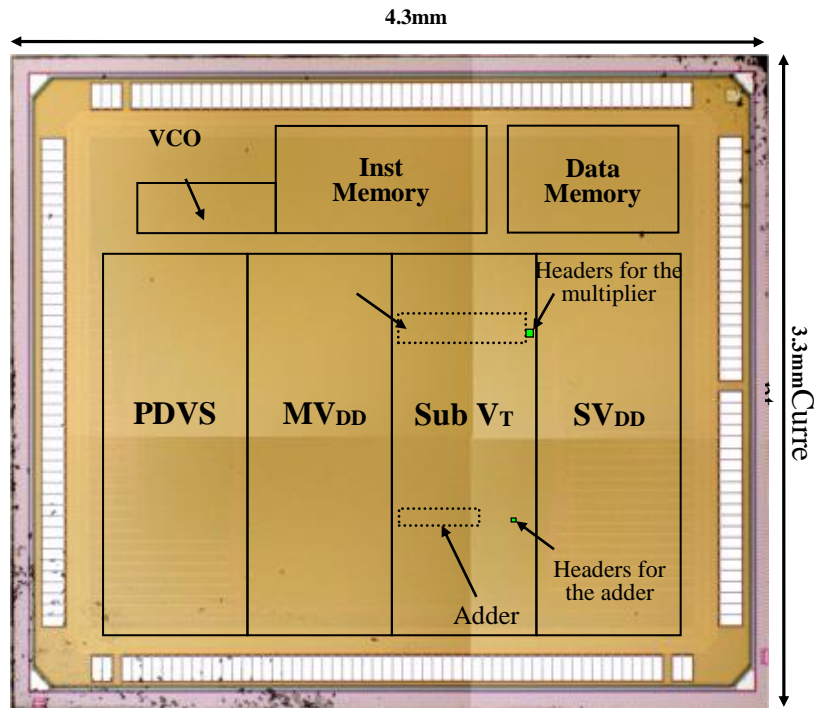


Figure 3-9: Annotated die photo and chip summary

Table 3-2: PDVS Chip Summary

<i>Feature</i>	<i>This Chip</i>
Process	90nm CMOS Bulk w/ Dual $V_T$
Area	4.3mm x 3.3mm
Transistor Count	~2 million
$V_{DD}$	250mV – 1.2V
Memory	40kb & 32kb

Table 3-3: PDVS Comparison to the start of the art

<i>Feature</i>	<i>[7]</i>	<i>[13]</i>	<i>[14]</i>	<i>This work</i>
$V_{DD}$ Granularity	6 cores	1 core	1 core	Add, Multiplier
Speed of $V_{DD}$ change	$>10\mu s$ (e.g. [6])	2-5ns	$>10\mu s$ (e.g. [6])	1ns
$V_{DD}$ dithering	No	No	No	Yes
Subthreshold operation	No	No	No	Yes



# Chapter 4

## Enabling Subthreshold Operation

### 4.1 Background

In previous chapters we have discussed techniques to improve energy efficiency by improving voltage scalable architectures through moving voltage scaling to the on-chip power delivery network for high-end and low-end applications. However, in this chapter we improve energy efficiency by providing methodologies for enabling subthreshold operation for low-end applications.

As previously mentioned a major focus in research has been subthreshold digital operation. That is to use a  $V_{DD}$  for the entire circuit that is below the threshold voltage of the device. The threshold voltage ( $V_T$ ) is defined as the transistor gate to source voltage differential ( $V_{GS}$ ) that allows the transistor to begin conducting current. When  $V_{GS}$  is below  $V_T$ , the transistor is “off”, however there is still enough current to perform digital operations. Operating in subthreshold is enticing for energy constrained systems, such as portable devices, since energy has a more than quadratic dependency on  $V_{DD}$ . Operating in subthreshold drastically reduces energy consumption. However, in the subthreshold

mode of operation delay increases exponentially, not linearly, with  $V_{DD}$ . For this reason, subthreshold is best suited for low-end performance applications where high speed operation is not necessary. One common application for subthreshold operation is body sensor nodes. Subthreshold sensor nodes have been shown to operate with very little energy consumption while processing electrocardiogram (ECG) signals [22], and operating off of harvested energy [23]. Designing for subthreshold and super-threshold operation is challenging, the authors in [24] explore many of the design considerations for operating across a wide voltage range. Instead of having design that can operate in subthreshold and nominal voltage rangers, the authors in [25] use multiple cores: one designed for nominal operation, and two designed for subthreshold. Memory design in subthreshold also has many design challenges [26][27]; however subthreshold memory design was not a focus of this dissertation. For the context of this dissertation we will focus on architectural organizations as well as power delivery network optimizations to enable subthreshold in the context of PDVS. These same concepts are generally applicable to all architectures however.

## 4.2 PDVS Architecture Enhancements

As discussed previously, PDVS extends DVFS to finer  $V_{DD}$  granularity and removes the need for DC-DC converter voltage scaling. PDVS, previously shown in Figure 3-1, modifies the power delivery network by adding multiple PMOS power switches at the component level. This allows each component to select the best  $V_{DD}$  from among a discrete set of shared  $V_{DD}$  rails depending on the application requirement. PDVS was initially constrained to have  $V_{DDH}$ ,  $V_{DDM}$  and  $V_{DDL}$  all be super threshold voltages. Simply lowering  $V_{DDL}$  to  $V_{SUBVT}$  would not work. First, as was shown in Figure 3-2, the

PDVS architecture has level converters that are used to convert from  $V_{DDL/M}$  to  $V_{DDH}$ . However, these level converters only function with super threshold  $V_{DD}$ s. Second, PDVS provides fast  $V_{DD}$  scaling at the component level between  $V_{DD}$ s, but as discussed subthreshold hold operation is exponentially related to  $V_{DD}$  and is much slower than super threshold operation. This implies that subthreshold operation should be considered a mode change that stays in the subthreshold mode for extended periods of time. Finally, for the best energy efficiency, all processor components (crossbar, register bank) need to be in subthreshold, not just adder and multipliers.

#### 4.2.1 Approach

In order to achieve subthreshold operation as well as maintain super-threshold operation, design changes were made to optimize the data path. In our traditional PDVS implementation, power switches were placed on the arithmetic components, but the crossbar and register bank were hard-tied to  $V_{DDH}$ . For every component, short circuit current was avoided by level converting from the virtual- $V_{DD}$  up to the nominal  $V_{DD}$  after every operation. However, in the subthreshold data path, leaving the crossbar and register

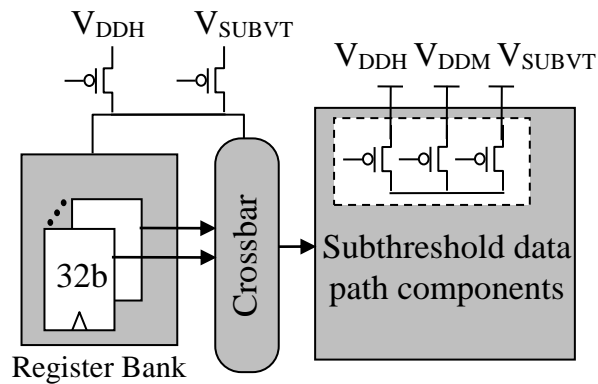


Figure 4-1: Register bank and cross bar modified for subthreshold operation. A nominal and subthreshold power switch is added to each putting the entire data path into subthreshold.



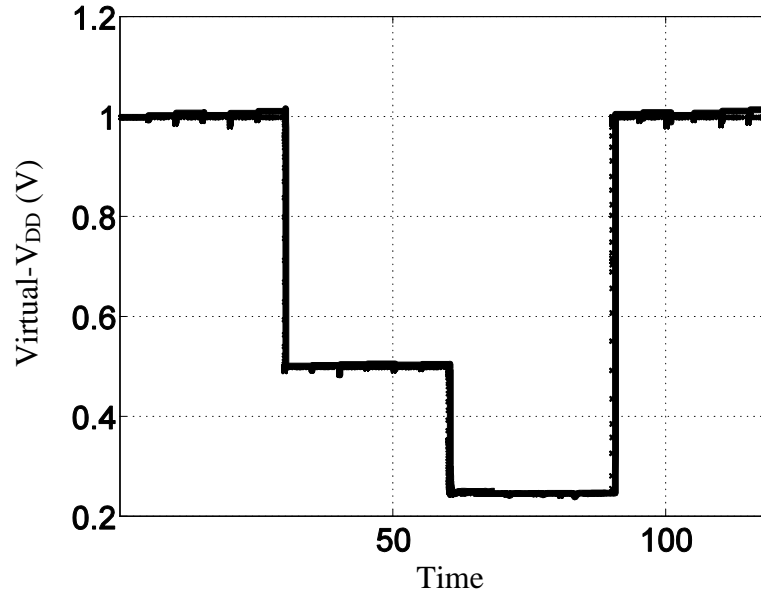


Figure 4-4: Virtual- $V_{DD}$  during  $V_{DD}$  switching.  $V_{DDH/ML}$  = 1V, 0.5V, 0.25V with transition times of 5ns, 200ns, & 2ns. Verified functionality with hardware measurements

register bank, and arithmetic components are all operating at  $V_{SUBVT}$ ; we bypass the level converters that are normally used for super-threshold operation at the output of the arithmetic components to avoid unnecessary energy and delay penalties. For communication with the super-threshold memories, a special designed level converter capable of converting from subthreshold up to 1.2V was used [28]. We use a similar bypass scheme as described for the super-threshold level converters, for the subthreshold level converters (Figure 4-2). We only utilize the subthreshold level converters while operating in subthreshold, avoiding any unnecessary delay and energy overheads while operating in super-threshold. Special consideration needs to be taken when deciding how to tie the bulk connections of the  $V_{SUBVT}$  power switch and the component. For ease, in the PDVS data path, all the power switch bulk connections and component bulk connections were tied to  $V_{DDH}$ . In the subthreshold data path, leaving the bulk connection at  $V_{DDH}$  would lead to reverse biasing of the power switch and component

[29]. Instead, by tying the bulk to the virtual- $V_{DD}$ , we are able to decrease the energy per operation by 20% when compared to  $V_{DDH}$  (Figure 4-3). Sub-threshold operation was simulated and was functionally verified in hardware for  $V_{DDH/ML}$  values of 1V, 0.5V and 0.25V, respectively, to demonstrate virtual  $V_{DD}$  switching capability across a broad voltage range (Figure 4-4).

### 4.3 A NMOS as a Subthreshold Header Power Switch

#### 4.3.1 Background

As discussed previously, PDVS uses multiple voltage supplies ( $V_{DDH}$ ,  $V_{DDM}$ ,  $V_{DDL}$ ) with PMOS power switches to select the appropriate  $V_{DD}$  for different fine-grained blocks/components in the design depending on local application requirements. PDVS supports DVFS, power gating for leakage reduction, and subthreshold operation when appropriate. The introduction of a power switch device in the power delivery network creates an IR drop across the power switch resulting in a reduced virtual- $V_{DD}$  value and performance degradation. Power switch sizing is critical to maintain energy efficiency. An undersized power switch results in exponential performance degradation, however an oversized power switch results in increased leakage and increased area overhead. We presented PMOS power switch methodologies in the previous section for enabling subthreshold operation. In this section, we investigate the benefit of using a NMOS transistor as a header power switch instead of a PMOS in a PDVS architecture.

Power switch sizing methodologies have been examined in depth to support techniques like multi-threshold CMOS (MTCMOS) and power gating [30]-[34]. To compare our NMOS power switch against the conventional PMOS we use a common power switch sizing methodology that sets the power switch size such that the critical path meets

acceptable delay degradation from the nominal case (i.e., no power switch). The allowable degradation is a design choice chosen by the system designer.

### 4.3.2 Approach

Figure 4-5 shows the conventional subthreshold power switch architecture as well as the proposed NMOS power switch architecture. The conventional architecture uses a PMOS power switch with the body tied to virtual- $V_{DD}$  to avoid reverse body bias [29]. Since the PDVS architecture has multiple supplies, the power switch control signals are full swing, up to  $V_{DDH}$ . For the conventional PMOS power switch this provides a strong turn off of the power switch. In the proposed alternative, an NMOS power switch with its body tied to ground is used as the power switch between the subthreshold rail and the component. During subthreshold operation (i.e., only  $V_{SUBVT}$  enabled) the conventional PMOS power switch has a  $|V_{GS}| = V_{SUBVT}$ , however the proposed NMOS power switch has a

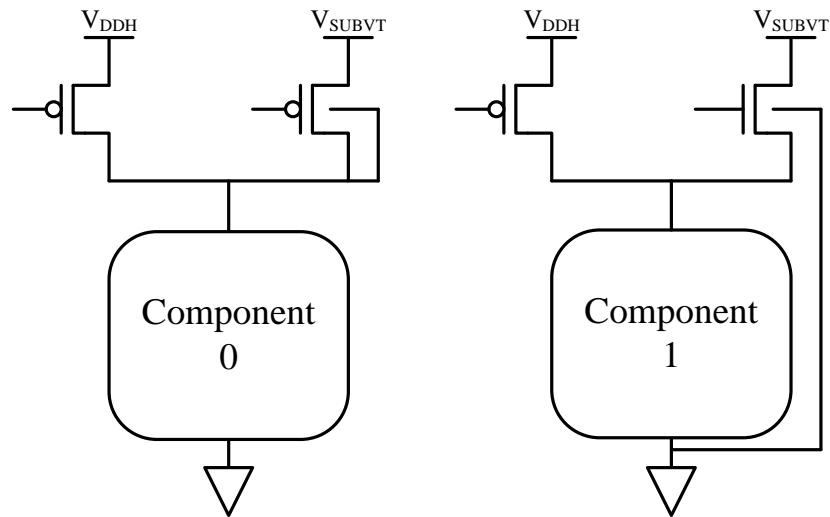


Figure 4-5: (*left*) conventional PMOS power switch with the  $V_{SUBVT}$  power switch body tied to virtual- $V_{DD}$  (*right*) proposed NMOS power switch with the  $V_{SUBVT}$  power switch body connection tied to  $V_{SS}$ .

$|V_{GS}| = V_{DDH} - V_{SUBVT}$ . The higher  $|V_{GS}|$  of the NMOS device provides a much higher current than the PMOS, since the NMOS is in the linear region of operation while the PMOS is in the cutoff/subthreshold region of operation. The higher current from the NMOS device provides a more stable virtual- $V_{DD}$  for a much smaller transistor.

### 4.3.3 Comparison to Conventional PMOS Power Switch

#### *Approach*

We use a commercial 130nm bulk process to simulate, measure, and compared the conventional PMOS and proposed NMOS subthreshold power switch topologies. To provide a flexible, representative circuit load, we used ten 27-stage ring oscillators (ROs) in parallel, and each RO was capable of being enabled independently. To simplify the comparison, each block of parallel ROs only had two header power switches as shown in Figure 4-5. In simulation, we swept the widths of the power switches to examine the impact of size on power switch behavior, and in the test chip we describe later, we included programmable sized power switch for flexible hardware measurements.

#### *Simulation Results*

Figure 4-6 demonstrates the impact of power switch width on virtual- $V_{DD}$  for two different activity factors. An activity factor of 1.0 corresponds to all 10 ring oscillators enabled in while 0.1 corresponds to only 1 ring oscillator enabled. These two activity factors represent the upper and lower bounds of this design.  $V_{SUBVT}$  was set to 0.3V, well below the threshold voltage ( $V_T$ ) in the technology. Across the wide range of sizes used, the NMOS is able to keep virtual- $V_{DD}$  at the target 0.3V due to the NMOS being in the linear operating region. The PMOS, however, is unable to keep virtual- $V_{DD}$  at the target

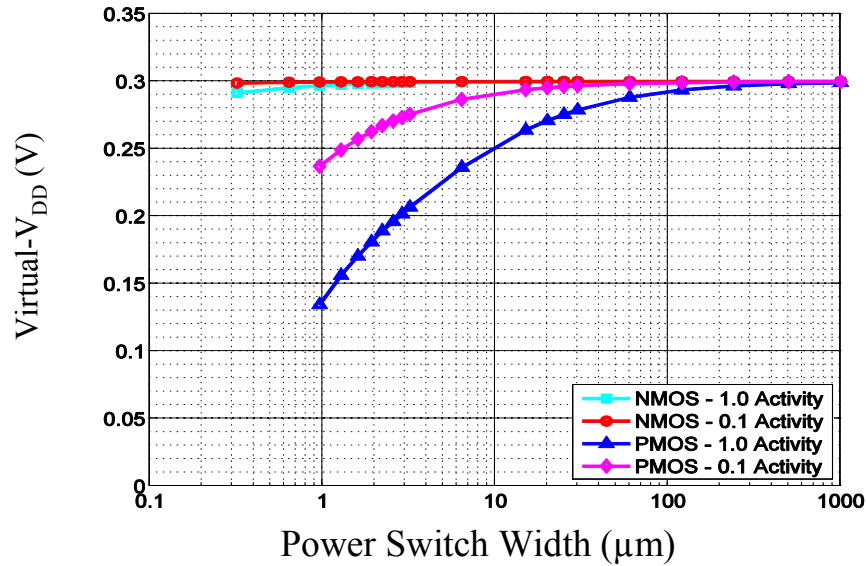


Figure 4-6: Simulated Virtual- $V_{DD}$  at 0.3V for the PMOS and NMOS for the highest and lowest activity factor. An activity factor of 1.0 corresponds to 10 ring oscillators enabled in parallel while 0.1 corresponds to only 1 ring oscillator enabled. The NMOS is able to keep virtual- $V_{DD}$  at the target 0.3V due to the NMOS being in the linear operating region.

0.3V for small widths since it is in the subthreshold operating region. It is necessary to keep the virtual- $V_{DD}$  near the target  $V_{DD}$  because frequency depends exponentially on the virtual- $V_{DD}$  voltage in subthreshold, so voltage droop leads to huge performance degradation.

The impact of the power switch width on oscillator frequency is shown in Figure 4-7. The frequency has been normalized to the frequency at 0.3V without power switches. With near minimum sizing at the lowest activity factor the NMOS has a worst case frequency degradation of only 3%, while the minimum PMOS has a worst case frequency degradation of 88%. At the highest activity factor with near minimum sizing the NMOS has a worst case frequency degradation of 16%, while the smallest PMOS has a worst case frequency degradation of 93%. Using the traditional sizing methodology and a target delay degradation of 10%, the required NMOS size is approximately 280x smaller than a

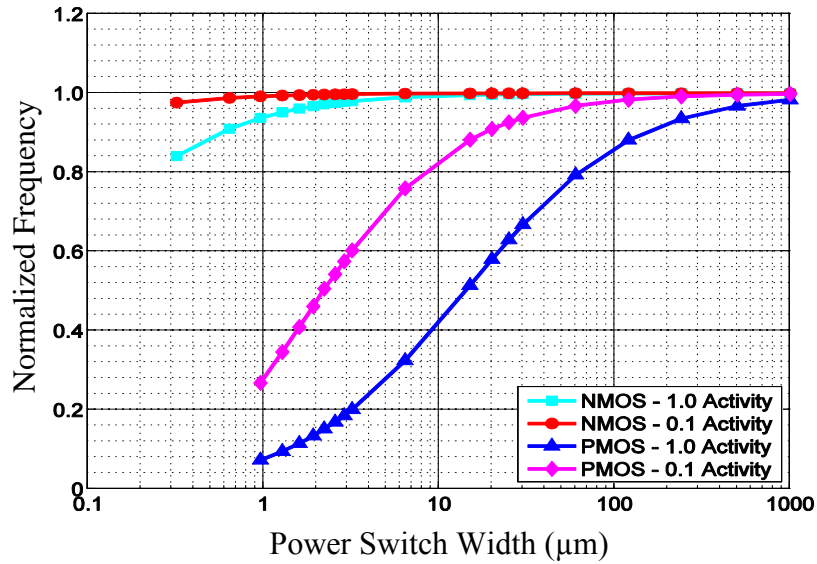


Figure 4-7: Simulated Frequency at 0.3V for the conventional PMOS and proposed NMOS. Using the traditional sizing methodology and a target delay degradation of 10%, the required NMOS size is approximately *280x smaller* than a PMOS for the same target degradation at the same worst case activity factor.

PMOS for the same target degradation at the same worst case activity factor, with sizes of 640nm and 180μm respectively.

The total energy per operation while operating at  $V_{SUBVT}$  is defined by the following equation:

$$E_{op} = E_{DYNV_{SUBVT}} + E_{LEAKV_{SUBVT}} + E_{LEAKV_{DDH}} \quad (4-1)$$

This energy equation includes the overheads of the PDVS architecture associated with having multiple  $V_{DD}$ s and power switch devices. Simulated energy per operation versus power switch width for an activity of 1.0 is shown in Figure 4-8. For both NMOS and PMOS, the energy is normalized to the same value, the energy per operation with no power switches. The shift in energy above the nominal for each of these designs is due to overheads inherent in the PDVS architecture. Specifically, the increase in energy comes from  $E_{LEAK}$  through the off  $V_{DDH}$  power switch while the  $V_{SUBVT}$  power switch is on. This

$V_{DDH}$  leakage contributes to an energy overhead of 3% and exists whether the NMOS or PMOS power switch is used. The decrease in energy as power switch size is reduced is due virtual- $V_{DD}$  drop, which lowers the dynamic energy. However for the PMOS design, the energy starts to increase at the lower widths due leakage becoming the dominate factor because of lower operating frequencies. At these lower widths, the virtual- $V_{DD}$  becomes much lower, resulting in initially lower dynamic and total energy due to the IR drop. As the width is reduced further the frequency becomes prohibitively slow resulting in  $E_{LEAK}$  through the  $V_{DDH}$  and  $V_{SUBVT}$  power switches becoming dominate causing an increase in total energy. The use of an NMOS power switch does not adversely increase the overheads associated with the PDVS architecture or have a higher energy per operation than a PMOS power switch. Gate leakage for this technology was not a concern; at the largest power switch size of 1mm, the gate leakage energy was only

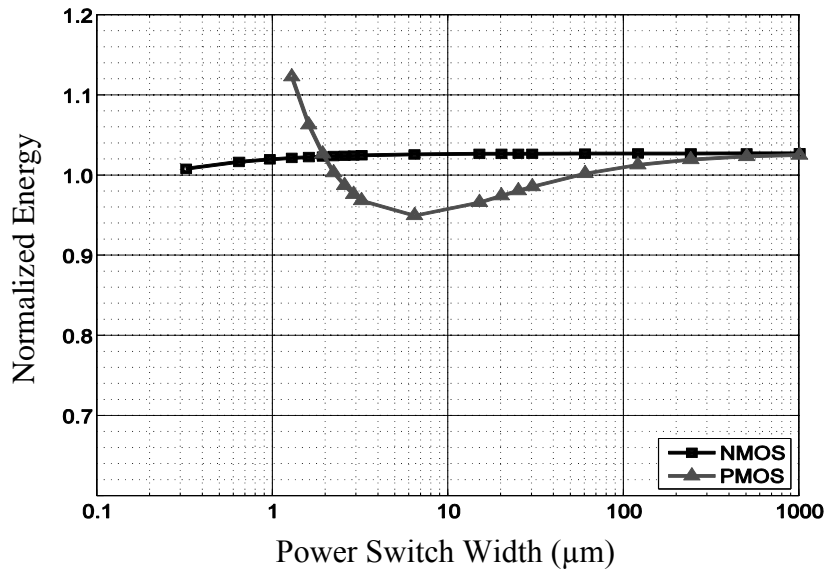


Figure 4-8: Simulated energy per operation at 0.3V for the conventional PMOS and proposed NMOS. The decrease in energy as power switch size is reduced is due virtual- $V_{DD}$  drop due to the device resistance. With the PMOS the energy starts to increase at the lower widths due leakage becoming the dominate factor because of lower operating frequencies.

0.04% of the total energy.

### ***Measured Results***

A test chip was fabricated in a 130nm bulk commercial process to verify the simulation results. Figure 4-9 shows the normalized frequency with an activity of 0.1, and Figure 4-10 shows the normalized frequency at the activity of 1.0. The simulated data were normalized as described earlier, while the measured data were normalized to the largest power switch value for NMOS and PMOS at the largest power switch width possible. Even though the range of width values for the measured data is not as large as the simulated, the same trend is observed for both activity factors. At the lowest activity factor of 0.1, the NMOS has a worst case measured frequency degradation of only 3%, while the smallest PMOS available in hardware has a worst case frequency degradation of 50%. At the highest activity factor with near minimum sizing, the NMOS has a worst case measured frequency degradation of 13%, while the minimum PMOS has a worst

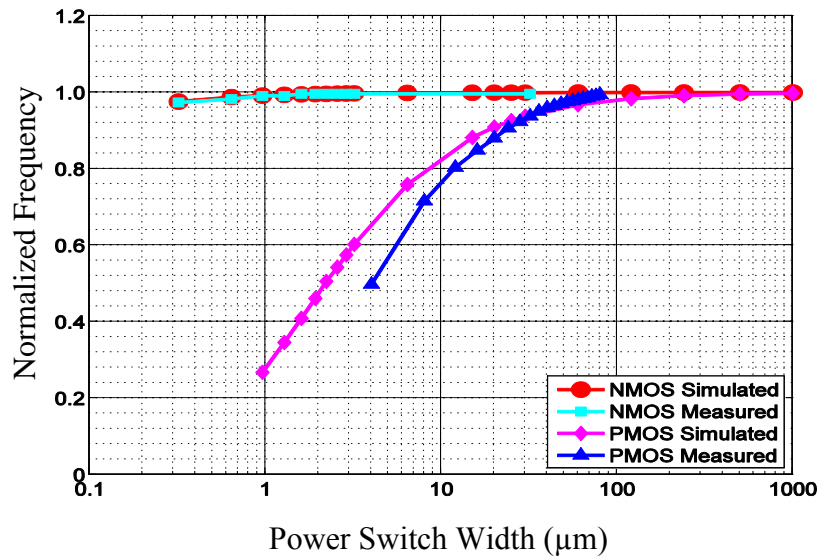


Figure 4-9: Simulated and measured frequency at 0.3V with an activity factor of 0.1. The NMOS has a worst case measured frequency degradation of only 3%, while the smallest PMOS available in hardware has a worst case frequency degradation of 50%.

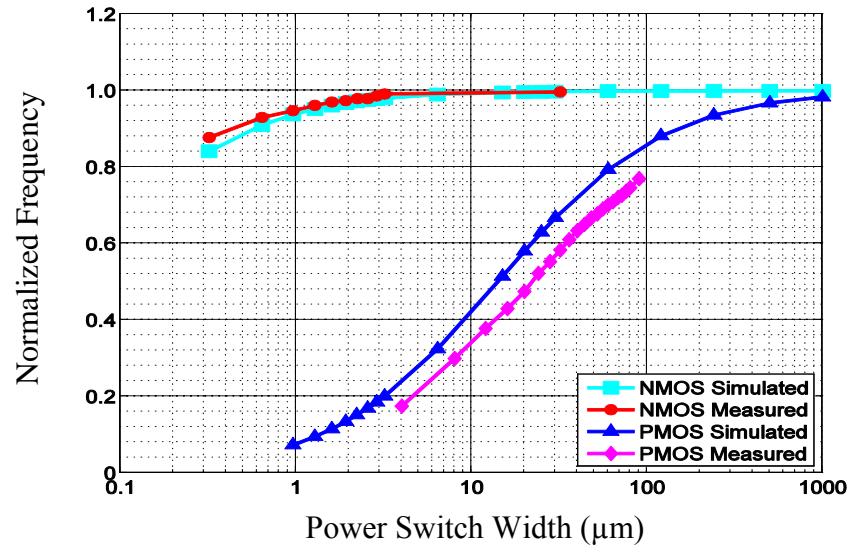


Figure 4-10: Simulated and measured frequency at 0.3V with an activity factor of 1.0. The NMOS has a worst case measured frequency degradation of 13%, while the minimum PMOS has a worst case frequency degradation of 84%

case frequency degradation of 84%. The power switch size range in hardware was not large enough to meet the 10% frequency degradation at the highest activity for the PMOS. Using the traditional sizing methodology and a target degradation of 23% (the best achievable by the PMOS in hardware), the required NMOS size is approximately *280x smaller* than a PMOS for the same target degradation at the same worst case activity factor, with values of 320nm and 90μm respectively.

#### 4.3.4 Flexible Transmission Gate Power Switch

If the  $V_{DDL}$  rail is always kept at a subthreshold voltage, using an NMOS power switch is optimal since a near minimum sized transistor would provide the target frequency requirement. However, if the rail connecting the power switch to the component needs to be a flexible and encompass a wide range of  $V_{DDS}$ , the NMOS fails as a power switch above  $V_T$  due to the  $V_T$  drop seen across the NMOS. For designs that require a wide range of voltages on the  $V_{DDL}$  rail, we propose to use a transmission gate architecture

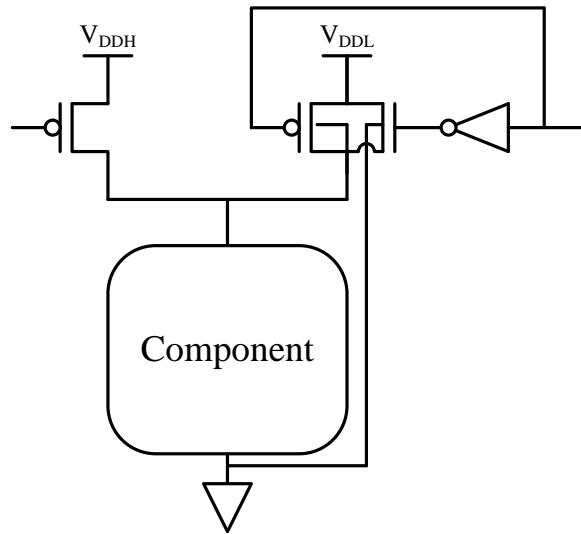


Figure 4-11: Transmission gate power switch architecture. Both PMOS and NMOS transistors are used with complimentary control signals. When  $V_{DDL}$  is near- or subthreshold the NMOS will be the dominate device. Otherwise the PMOS will be the dominate device.

shown in Figure 4-11. When  $V_{DDL}$  is near- or subthreshold, the NMOS will be the dominate device. Conversely, when in voltages above  $V_T$ , the PMOS will be the dominate device.

Both PMOS and NMOS devices should be sized independently for a given frequency degradation. This will lead to the transmission gate power switch having asymmetric sizing. The PMOS device would be relatively large to meet the frequency target at high voltage; in our design it is in the 10s-100s of  $\mu\text{m}$  depending on the frequency target. However, the NMOS device would be in the  $<1 \mu\text{m}$  range and still provide the target frequency for the component with the supply at low voltages. For our test chip, the NMOS actually provides a frequency degradation target better than designed.

### Measured Results

Measured results were taken from the test chip verifying the ability to use a transmission gate for a power switch for a wide range on the supply voltage rail. The  $V_{DDL}$  rail voltage was varied from 0.3V to 1.2V. The transmission gate PMOS device was limited to a max width of  $40\mu\text{m}$  resulting in a frequency degradation of 30% resulting in a PMOS size of  $40\mu\text{m}$  and a much-smaller NMOS size of 320nm.

Figure 4-12 shows the measured Energy-Delay (ED) curve for an NMOS-only, PMOS-only, and transmission gate power switch. The NMOS-only ED curve reaches a frequency limitation due to the  $V_T$  drop across the NMOS above about 0.6V. The PMOS-only curve becomes slower than the NMOS at about 0.4V due to the virtual- $V_{DD}$  drop causing the increase in delay. For the same delay constraint the transmission gate is 21% lower energy, while for the same energy constraint the transmission gate is 33% when

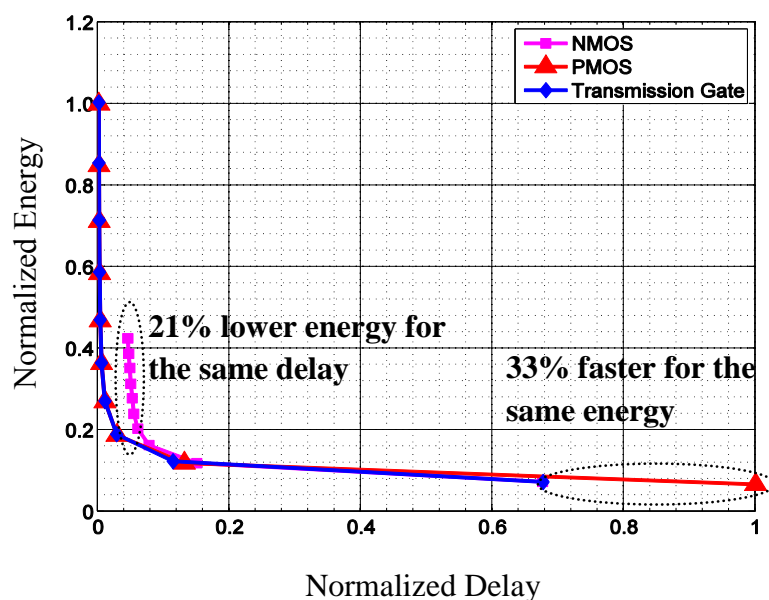


Figure 4-12: Measured energy delay curve of a NMOS, PMOS and transmission gate based power switch. For the same delay constraint the transmission gate is 21% lower energy, while for the same energy constraint the transmission gate is 33% faster when compared to the NMOS and PMOS respectively.

compared to the NMOS and PMOS respectively. Finally, the transmission gate merges the best from the NMOS and PMOS and is able to provide the lowest delay and the lowest energy. In Figure 4-13: Measured energy delay curve of a NMOS, PMOS and transmission gate based power switch. The transmission gate is the pareto optimal curve of the NMOS and PMOS having the lowest EDP across all  $V_{DDs}$ , the measured energy delay product (EDP) across the range of  $V_{SUBVT}$  is shown for the same three cases described above. Since EDP is a measure of energy efficiency, it is desirable to have a lower EDP. As expected, the NMOS has a lower EDP at lower  $V_{DDs}$ , but suffers from a higher EDP at higher  $V_{DDs}$ . Conversely, the PMOS has a lower EDP at higher  $V_{DDs}$ , but suffers a higher EDP at lower  $V_{DDs}$ . The transmission gate is the pareto optimal curve of the NMOS and PMOS having the lowest EDP across all  $V_{DDs}$ .

Traditional methodologies of physical power switch design generally distribute the power

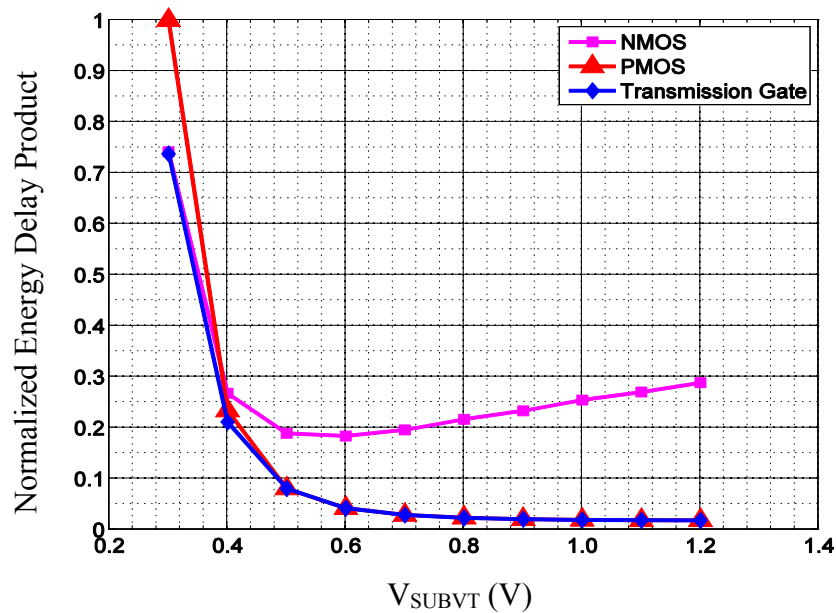


Figure 4-13: Measured energy delay curve of a NMOS, PMOS and transmission gate based power switch. The transmission gate is the pareto optimal curve of the NMOS and PMOS having the lowest EDP across all  $V_{DDs}$ .

switch as standard cell rows, columns, or as rings around design [33][34]. These help the electrical properties of the power delivery system. Physical implementation of the NMOS device in these methodologies would lead to using above minimum sizing to prevent power delivery system electrical problems, potentially slightly increasing the area associated with the NMOS power switch. This would slightly alter the presented results by decreasing the NMOS area savings prevent compared to the PMOS. This would be especially true for smaller designs where a minimum NMOS could be used, but would need to be increased for the power delivery system. However, for larger designs which require larger than minimum NMOS sizing the impact of potentially needing to increase the physical size would be reduced.

#### **4.4 Summary & Conclusions**

In this chapter we investigated methodologies for enabling subthreshold operation for improved energy efficiency. That is to use a  $V_{DD}$  for the entire circuit that is below the threshold voltage of the device. Operating in subthreshold is enticing for energy constrained systems, such as portable devices since energy has a more than quadratic dependency on  $V_{DD}$ . Operating in subthreshold drastically reduces energy consumption. One common application for subthreshold operation is body sensor nodes.

We demonstrated architectural organizations as well as power delivery network optimizations to enable subthreshold in the context of PDVS. These same concepts are generally applicable to all architectures however. We demonstrated the best practices for modifying architectures to enable operation in both nominal and subthreshold modes of operation. We also demonstrated the best body connection for PMOS or transmission gate based power switches for the low voltages. We demonstrated the use of an NMOS

transistor as a header power switch for low voltages. Simulations and measurements show that an NMOS device with a nominal gate swing ( $0 - V_{DDH}$ ) is able to provide the target frequency degradation with a size over 280X smaller than a PMOS. For flexible designs that have a wide range of  $V_{DD}$  rail values we proposed using an imbalanced transmission gate power switch with a near minimum sized NMOS device in parallel with a large PMOS. The transmission gate will provide the targeted frequency degradation at the nominal  $V_{DD}$  and provide a better than target frequency in subthreshold with minimal additional area.

# **Chapter 5**

## **A Scripted Research & Design**

### **Infrastructure**

#### **5.1 Background**

Previous chapters discussed two techniques that improve energy efficiency through voltage scaling by modifying the power delivery network to move the voltage scaling to the on-chip power delivery network. We discussed architectural and power delivery network optimizations for enabling subthreshold operation. In this chapter we introduce a research and design platform that is capable of implementing the aforementioned techniques and optimizations, enabling quick design exploration, as well as design and implementation of a full subthreshold system on chip (SoC).

One of the major challenges faced in our research group is a lack of infrastructure support for energy efficient design exploration and SoCs design. The current synthesis and place and route flow is optimized for high performance nominal voltage designs, not for

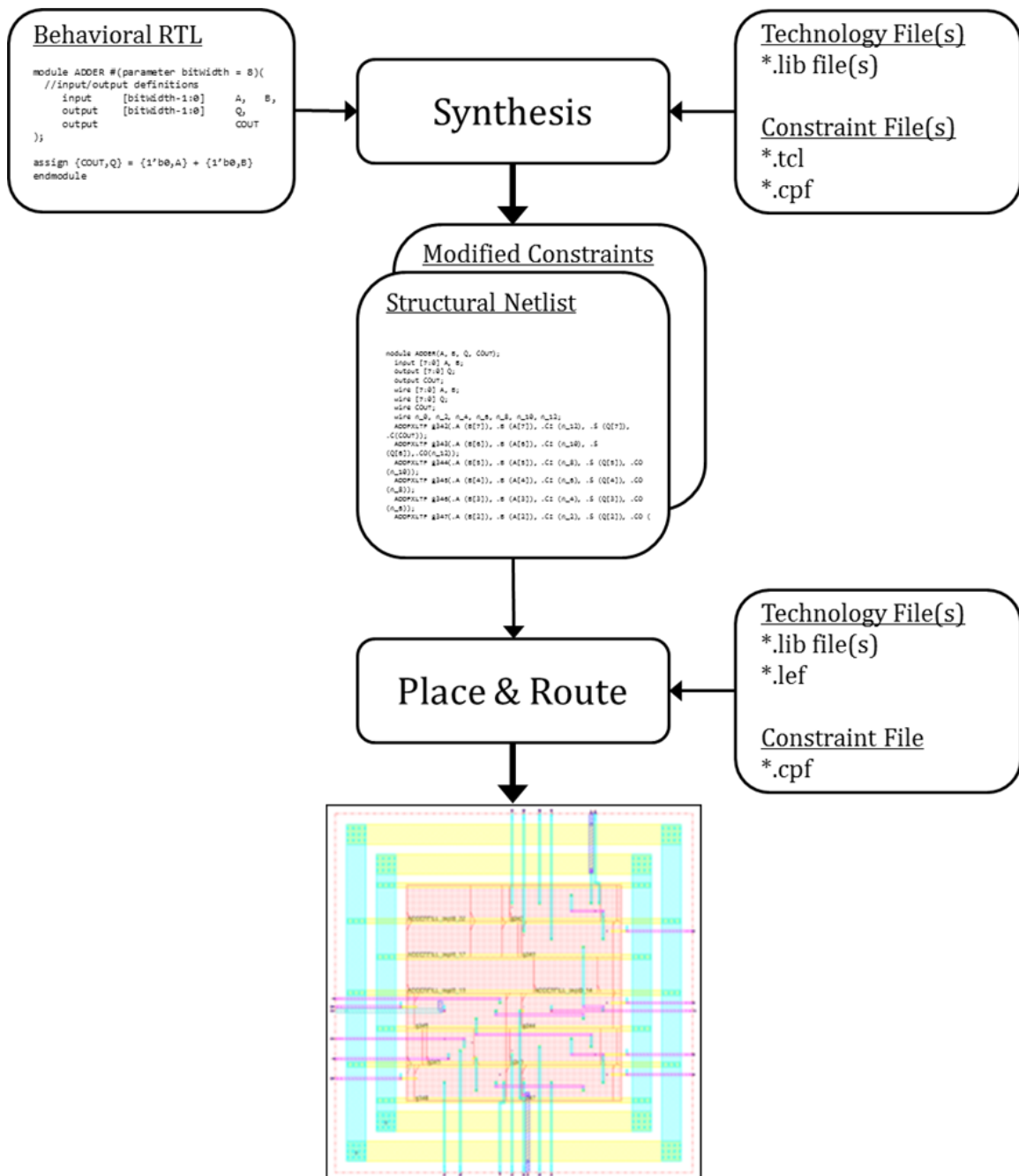


Figure 5-1: Generic flow diagram of the synthesis and place & route flow. Synthesis is the process of translating behavioral RTL into a gate level netlist. Place and route is the process of translating the gate level netlist into physical design.

subthreshold design. The flow supports multiple  $V_{DD}$  domains, power gating and clock gating. For academic research, a more robust infrastructure is needed to explore the design trade-offs of energy efficient power delivery networks and to build complete SoCs. Many corporations, such as Intel and Advanced Micro Devices (AMD), have their own infrastructure and tool flows, however a direct comparison to those tools is not feasible since they are proprietary and not readily available. Figure 5-1 shows a generic flow diagram for synthesis and place and route flow. There are two main steps for this flow: synthesis and place and route.

### ***Synthesis***

Synthesis is the process of translating behavioral register transfer level (RTL) into a gate level netlist consisting of logical gates, while meeting specific design constraints. Behavioral RTL is the functional definition of the design written in a hardware description language (HDL). The behavioral RTL can be as basic as a simple arithmetic component, such as a multiply accumulate (MAC), or as complicated as a multi-core processor. The two most common HDL languages used by our research group, and industry, are Verilog and VHDL. For synthesis to work, technology files and constraint files are needed. The technology files contain the timing and power information of the technology specific standard cells. The constraint files contain specific synthesis run options and the user generated circuit operating conditions, such as number of clocks, clock frequency, target area, target power, and power domain planning. Finally, the output from synthesis is the gate level netlist and a modified constraints file. The gate level netlist is a structural representation of the behavioral RTL mapped to a specific

technology standard cell library. The modified constraints file contains the user generated conditions, as well as constraints based on synthesis run options.

### ***Place & Route***

Place and route (P&R) is the process of translating a gate level netlist into a physical layout that meets timing closure (target frequency, hold time and setup time checks). The gate level netlist and modified constraint files from synthesis are most often used as the inputs to P&R, but for simple/custom created designs user generated versions of both these files can be used. The same technology files used for synthesis are also used in P&R with the addition of the technologies and standard cells library exchange files (LEF). The LEF contains the physical representation of the technology and standard cells. This includes metal and via definitions, as well as an abstract view of the standard cells (metal only view).

## **5.2 A Scripted Research Platform**

### **5.2.1 Approach**

As mentioned, the current EDA flow has limited energy efficient design options, and therefore is not immediately useful for energy efficient design space exploration. Specifically the tool allows for power gating, clock gating and multi- $V_{DD}$ . The flow has been modified to enable PDVS, a programmable resistive power grid, a NMOS header power switch, and a transmission gate power switch. In order to implement these energy efficient design techniques, we created a parameterized scripted infrastructure. The design exploration infrastructure uses the Cadence Digital Implementation System [35]. For synthesis and place and route we use Cadence RTL Compiler (RC) and Cadence

Encounter, respectively. The infrastructure is mostly written in the tool command language (TCL) language for running the Cadence design tools.

### 5.2.2 Standard Cells for Subthreshold

In order to utilize the EDA flows, we rely on using standard cells for the given technology we are designing in. Standard cells are the basic building blocks of synthesized digital systems, containing logic cells (e.g., inverter, nand/nor, and/or, etc.), sequential cells (e.g., D flip-flops, latches), and high impedance cells (e.g., tristate inverters, tristate buffers). A standard cell library can be made from different logic families. For nominal voltage designs two commonly used logic families are CMOS and dynamic logic.

The behavior of these logic families is well known for nominal voltage operation which makes choosing a standard cell library for nominal voltage a simple task. However special consideration needs to be given when choosing the proper logic family and standard cells for subthreshold operation. Due to device current being exponentially proportional to the threshold voltage, subthreshold logic families and standard cells are more sensitive to variation than their nominal voltage counterparts, and therefore more susceptible to failure [36]. When deciding the appropriate logic family to use for subthreshold operation, the primary metrics are robustness, speed, area, and power. Standard cell design is outside the scope of this dissertation, however in [37] the author presents a comparative study of ten different logic families and their suitability for subthreshold operation. The different logic families compared in this study were array logic, adiabatic logic, bootstrap logic, dynamic logic, Pseudo-NMOS, differential cascade voltage switch logic (DCVS), dynamic threshold CMOS (DTCMOS), dual mode logic,

transmission gate logic, and CMOS [38]-[46]. The outcome of this study concludes that the most energy efficient and robust logic families for subthreshold operation are DTCMOS and a trimmed down standard CMOS library. Once a logic family and standard cells are chosen they need to be incorporated into the EDA synthesis and place and route flow.

### **5.2.3 Cell Characterization**

For both synthesis and place and route, characterization information from standard cells is needed regardless of the logic family being used. The most common characterization format is the liberty format (.lib). The .lib file is an open industry standard originally developed by Synopsis, and is now widely used across industry and academia [47]. This file contains the logical functional description of the standard cell as well other metrics, such as load, cell delay, and cell power. In [48] a comparative study was done between using nominal voltage standard cells and standard cells characterized for subthreshold operation and found that for timing critical design the highest energy efficiency is achieved when using standard cells characterized for subthreshold. Characterization of a standard cell library into the .lib format can be done with commercially available EDA tools, such as the Encounter Library Characterizer (ELC). With the standard cell logic family chosen (e.g., DTCMOS or trimmed down CMOS library) and ELC, we created a standard cell library optimized for subthreshold to enable energy efficient synthesized and placed and routed designs.

### **5.2.4 Infrastructure**

The primary infrastructure developed for academic design space exploration consists of TCL scripts designed for use with RC for synthesis and Encounter for place and route.

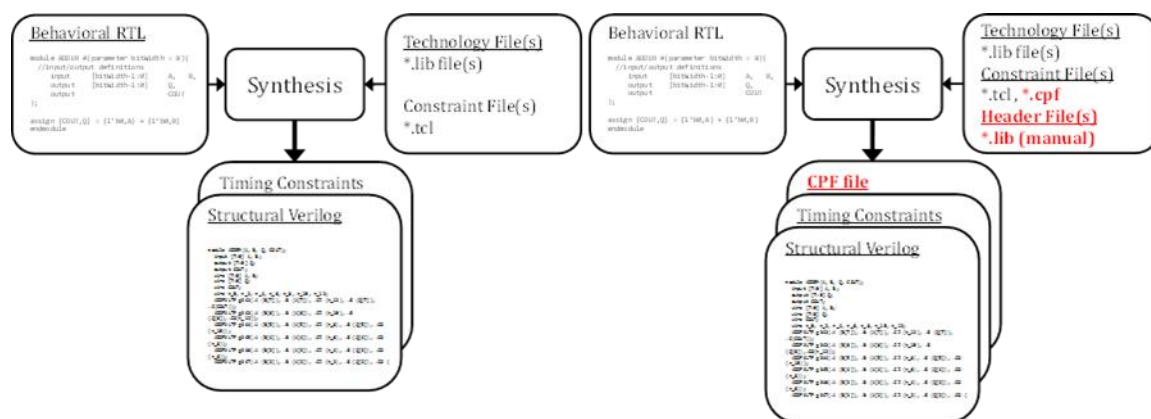


Figure 5-2: (left) Synthesis flow diagram of a conventional scripted RC flow. (right) Flow modified for advanced power delivery network techniques through the use of CPF and manual .lib files for power gates. Through parameterization and standard cell library characterization both are capable of nominal and subthreshold synthesis.

The scripts have also been adapted to be able to implement the different energy efficient design techniques presented in this dissertation.

## Synthesis

As discussed previously, synthesis is the translation of behavioral RTL into a gate level netlist. Since behavioral RTL is just a functional description of a specific design, it does not contain  $V_{DD}$  information. With a design that only contains a single  $V_{DD}$  the standard synthesis flow can be used with the standard cells characterized for subthreshold. However, to implement the energy efficient power delivery network techniques demonstrated in this work, we need to assign  $V_{DD}$  information to the design. This is achieved through the common power format (CPF). CPF is a Cadence format that is used to assign power domains and power gate information to an entire design, or sub portions of the design. For our energy efficient design techniques we designed power gates and created a .lib file for each power switch to be defined in the CPF methodology. To enable energy efficient design exploration, the created infrastructure, shown in Figure 5-2,

includes parameterized TCL scripts with documentation and is capable of both nominal and subthreshold synthesis with and without the use of CPF.

### ***Place & Route***

After generating the gate level netlist from synthesis we now can use Encounter to perform place and route to obtain a physical design with timing closure. There are six primary steps to place and route:

- *Floor planning*
- *Power planning*
- Placement
- Clock tree synthesis
- Routing
- Verification

Each one of these steps is a separate parameterized TCL script in our scripted infrastructure. Most of the end user modifications for different designs are in the tool setup and the following place and route steps: floor planning and power planning.

To generate designs with PDVS, a resistive power grid, NMOS header power switch, or a transmission gate power switch, special care is taken with the floor planning and power planning. Through floor planning we partition our RTL into different domains. The simplest way to do so is through hierarchical RTL and selecting power domains based on different sub components in the hierarchy. The power domain partitioning is achieved through the use of the CPF and floor planning options within Encounter. Each domain is independent of one another and capable of having any, or none, of the energy efficient techniques. The number of power domains is only limited by the RTL. We can generate

one power domain, or multiple. Through the scripted flow multiple iterations of a design can be created with different power configurations, enabling design space exploration.

Power planning is the next step in the P&R infrastructure. Through power planning we insert power switches into the design as well as create the power and ground rings and power grids. The type and size of the power switch inserted into the design is up to the end user, but the infrastructure supports variable width power switches to enable the resistive power grid, multiple  $V_{DDs}$  and power switches to enable PDVS, and NMOS power switch to enable subthreshold operation. The gate level netlist that is used in P&R is created of standard cells that are provided for a specific process design kit (PDK). These standard cells are generally all one height, and are placed in rows. In order to easily add power switches into our flow we designed power switches that are the same height as the standard cells.

Figure 5-3 shows a sample layout of a standard cell inverter, PMOS power switch for  $V_{DDH}$ , and a NMOS header power switch for subthreshold. The height of the cell, or standard cell pitch, is defined as the middle point of the  $V_{SS}$  rail connection to the  $V_{DD}$  rail connections. For the power switches designed, the  $V_{SS}$  to virtual- $V_{DD}$  pitch is made to be the same as the standard cells. When standard cells are placed in the design  $V_{DD}$  and  $V_{SS}$  are connected together by the rails seen in the standard cells. When considering a design, such as PDVS the component is tied to virtual- $V_{DD}$  and not the actual  $V_{DD}$ . The power switches designed for implementation in P&R are therefore designed to have the virtual  $V_{DD}$  net connect to the  $V_{DD}$  net of the standard cells. As discussed in previous chapters, the  $V_{DDH}$  PMOS power switch is designed to have the body connected to  $V_{DDH}$ , while the NMOS header power switch is designed to have the body connected to  $V_{SS}$ . To

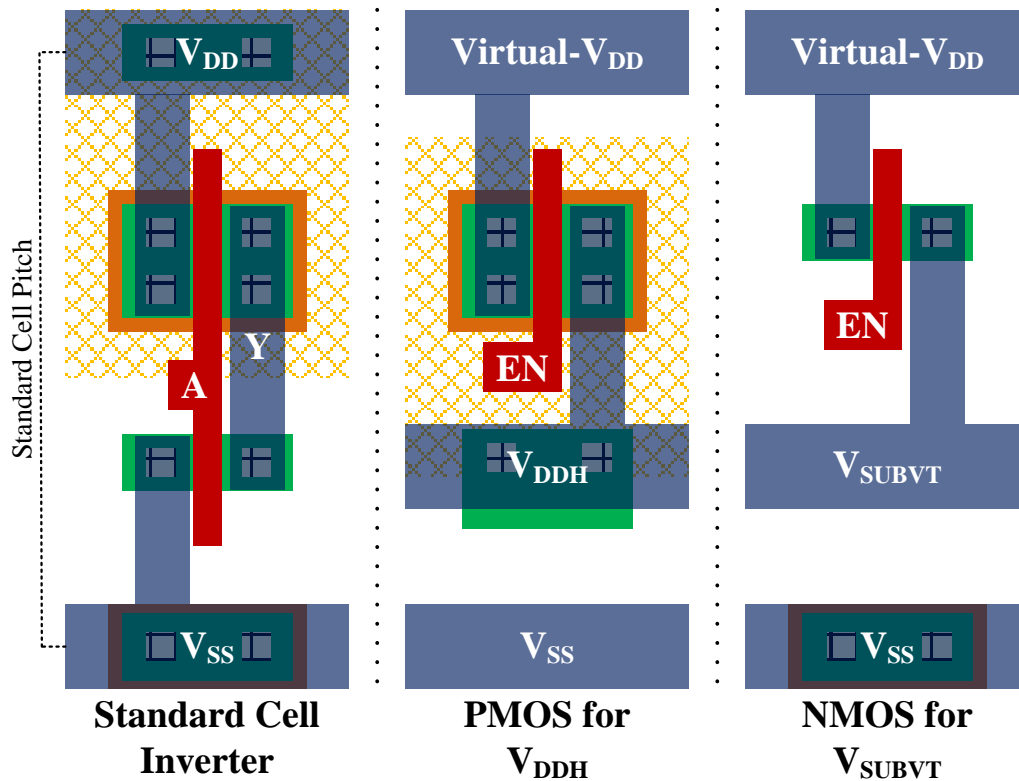


Figure 5-3: (left) Sample layout of a CMOS standard cell inverter. All standard cells are built with the same pitch. (middle) PMOS power switch designed at the same standard cell pitch for easy integration into Encounter (right) NMOS header power switch designed at the same standard cell pitch for easy integration into Encounter

use the power switches in the synthesis and place and route flow a .lib file and LEF file were created for all the designed power switches.

In Figure 5-4 we show a simplified PDVS implementation with only two voltages,  $V_{DDH}$  and  $V_{SUBVT}$  and a component that can be assumed to have been synthesized and therefore is a gate level netlist (or a cluster of standard cells). The second part of Figure 5-4 shows an artistic depiction of how the PDVS implementation would look if done through the scripted infrastructure in Encounter. For simplicity sake in this example there was only one power domain. There are four power rails routed throughout the design,  $V_{DDH}$ ,

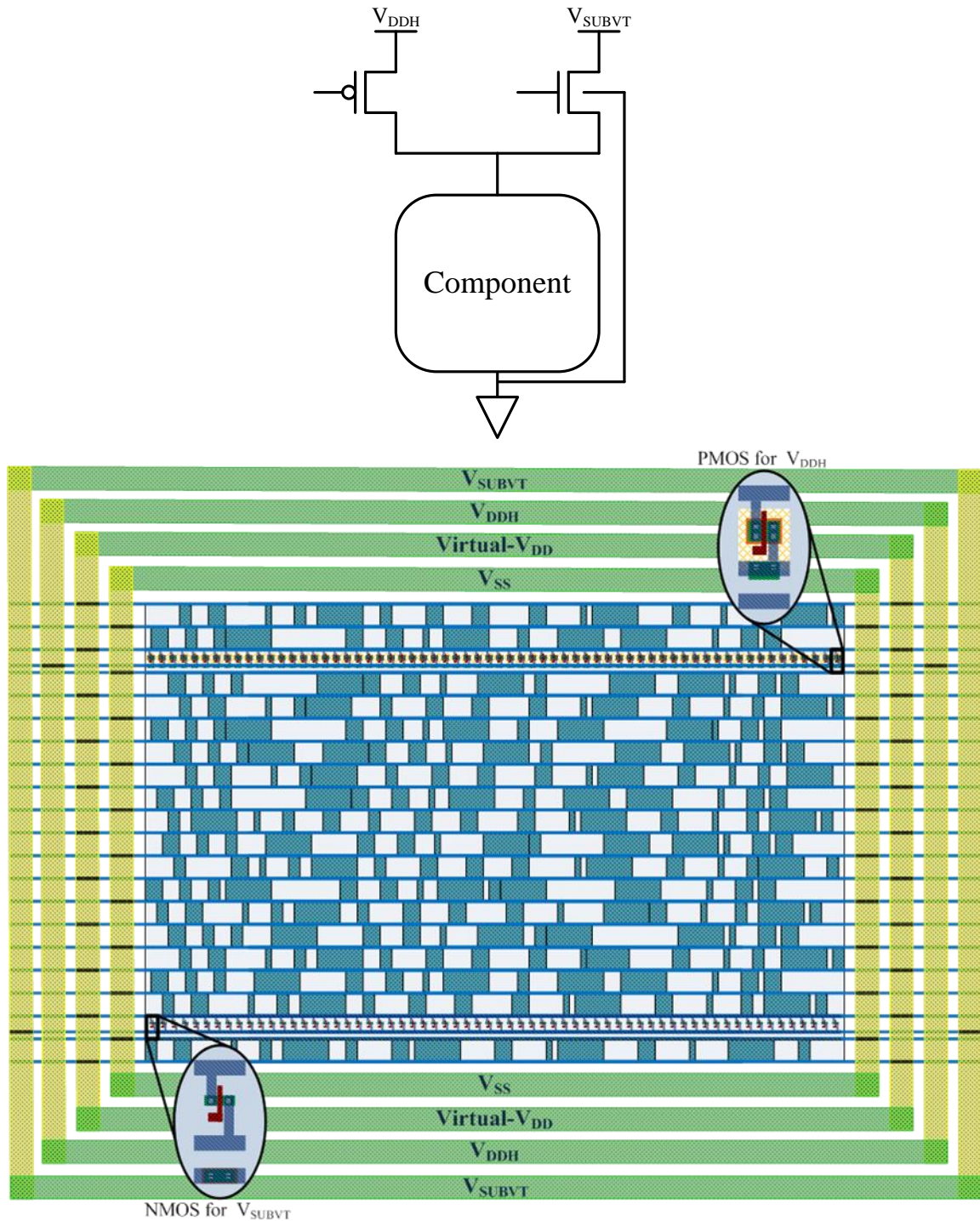


Figure 5-4: (top) PDVS implementation with a PMOS power switch for  $V_{DDH}$  and an NMOS for  $V_{DDL}$  (bottom) Depiction of the output of the Encouter infrastructure for PDVS. There are 4 power rings for  $V_{DDH}$ ,  $V_{SUBVT}$ , Virtual- $V_{DD}$  and  $V_{SS}$ . The top row of power switches shows the PMOS connected to  $V_{DDH}$ , while the bottom is the NMOS connected to  $V_{SUBVT}$ . The component is connected to Virtual- $V_{DD}$  and  $V_{SS}$ .

$V_{SUBVT}$ , virtual- $V_{DD}$  and  $V_{SS}$ . All these power rails are needed throughout the design; therefore we created a power ring for each. The dark rectangles represent the different standard cells that comprise the component. Since the component is tied to virtual- $V_{DD}$ , every other row in the design is connected to virtual- $V_{DD}$  and  $V_{SS}$ . In the example, there are two rows for power switch headers. The power switch row close to the bottom consists of the NMOS header power switch repeated throughout the entire row. As can be seen, the NMOS is connected to the  $V_{SUBVT}$  ring, virtual- $V_{DD}$  ring, and  $V_{SS}$  ring. Similarly, the power switch row at the top of the design consists of the PMOS power switch repeated throughout the entire row. The PMOS is connected to the  $V_{DDH}$  ring, virtual- $V_{DD}$  ring, and  $V_{SS}$  ring. For each power switch row, only power switches associated with the same  $V_{DD}$  can be used. With multiple power switches designed at variable widths, we use the same infrastructure to create a programmable resistive power grid.

### 5.2.5 A Scripted Research Platform Example

To show the potential for the scripted research platform we present one case study. For this case study we used the IBM 130nm PDK. This example uses the scripted infrastructure to create multiple power domains, each with a different on-chip power delivery network implantation. Figure 5-5 shows the block diagram used. There are eight behaviorally identical designs under test (DUTs), each with an arithmetic logic unit (ALU), multiply accumulate (MAC), and a ring oscillator (RO). There are two linear feedback shift registers (LFSRs) to generate quasi-random data. Each DUT has a different power delivery network configuration. Each power delivery network either implements PDVS, a resistive power grid, an NMOS header power switch, and

transmission gate power switch, or some combination of all four. For each power delivery network the amount of metal allocated for the grid, as well as decoupling capacitance were also varied. The power delivery network configuration for each DUT was as follows:

<b>DUT0</b>	SVDD (no power grid modifications)
<b>DUT1</b>	Resistive grid, NMOS header power switch, metal allocation 0
<b>DUT2</b>	Resistive grid, NMOS header power switch, metal allocation 1
<b>DUT3</b>	Resistive grid, NMOS header power switch, metal allocation 2
<b>DUT4</b>	Resistive grid, NMOS header power switch, metal allocation 0, decoupling capacitance
<b>DUT5</b>	Resistive grid, NMOS header power switch, metal allocation 2, decoupling capacitance
<b>DUT6</b>	Resistive grid, NMOS header power switch, PDVS, decoupling capacitance
<b>DUT7</b>	Resistive grid, NMOS header power switch, PDVS

Figure 5-6 shows the output from Encounter when using the scripted infrastructure. All eight DUTs generated, each with their own power delivery network as describe above. DUT1 and DUT7 are enlarged to show highlight the differences in the power delivery network. DUT1 has two power switch rows, each with NMOS and programmable PMOS devices connected to the same  $V_{DD}$ . DUT7 has four power switch rows. Two of the rows have NMOS devices connected to  $V_{SUBVT}$  while the other two rows are have programmable PMOS devices connected  $V_{DDH}$ . This case study was fabricated in the 130nm processes with results and a future publication pending.

### 5.3 SoC Design Methodology

The first part of the design infrastructure showed how to use a scripted synthesis and P&R flow for research and design space exploration. The second part of our research and design infrastructure focuses on design of large SoCs in an academic setting. Large SoCs include more than just synthesizable RTL. There are potentially many different blocks: digital and analog, synthesizable and custom. Some typical blocks from wireless sensor nodes are pad rings, power management, analog front ends, memories, and wireless communication [23]. In this section we present a design methodology for incorporating all

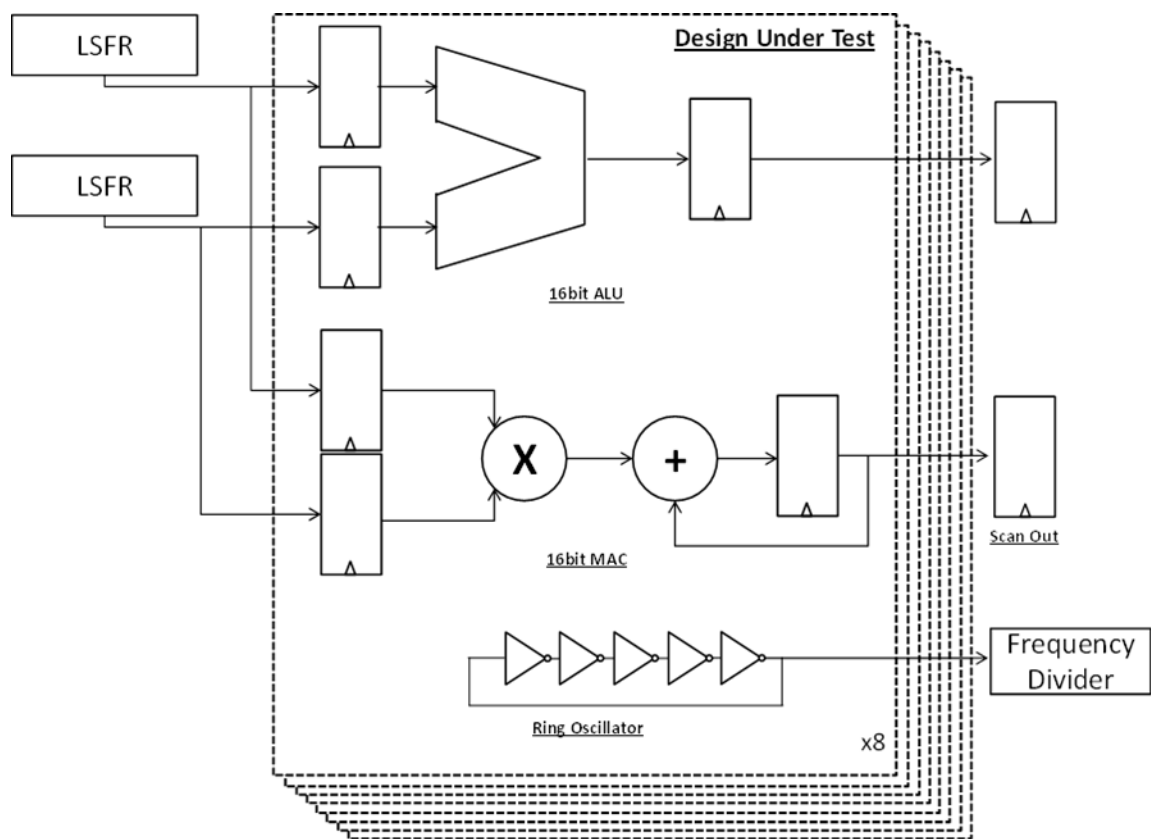


Figure 5-5: Block diagram of the case study. There are eight behaviorally identical DUTs, each with an ALU, MAC, and RO. There are two LFSRs to generate quasi random data. Each DUT has a different power delivery network configuration.

these components into a large SoCs using Encounter.

### 5.3.1 Pad Ring & LEF Generation

Common to most of academic SoCs is a pad ring. The pad ring is the interface boundary of the SoC and the rest of the world. To facilitate easy pad ring generation we created a scripted infrastructure that takes a user defined spreadsheet definition of a pad ring, and

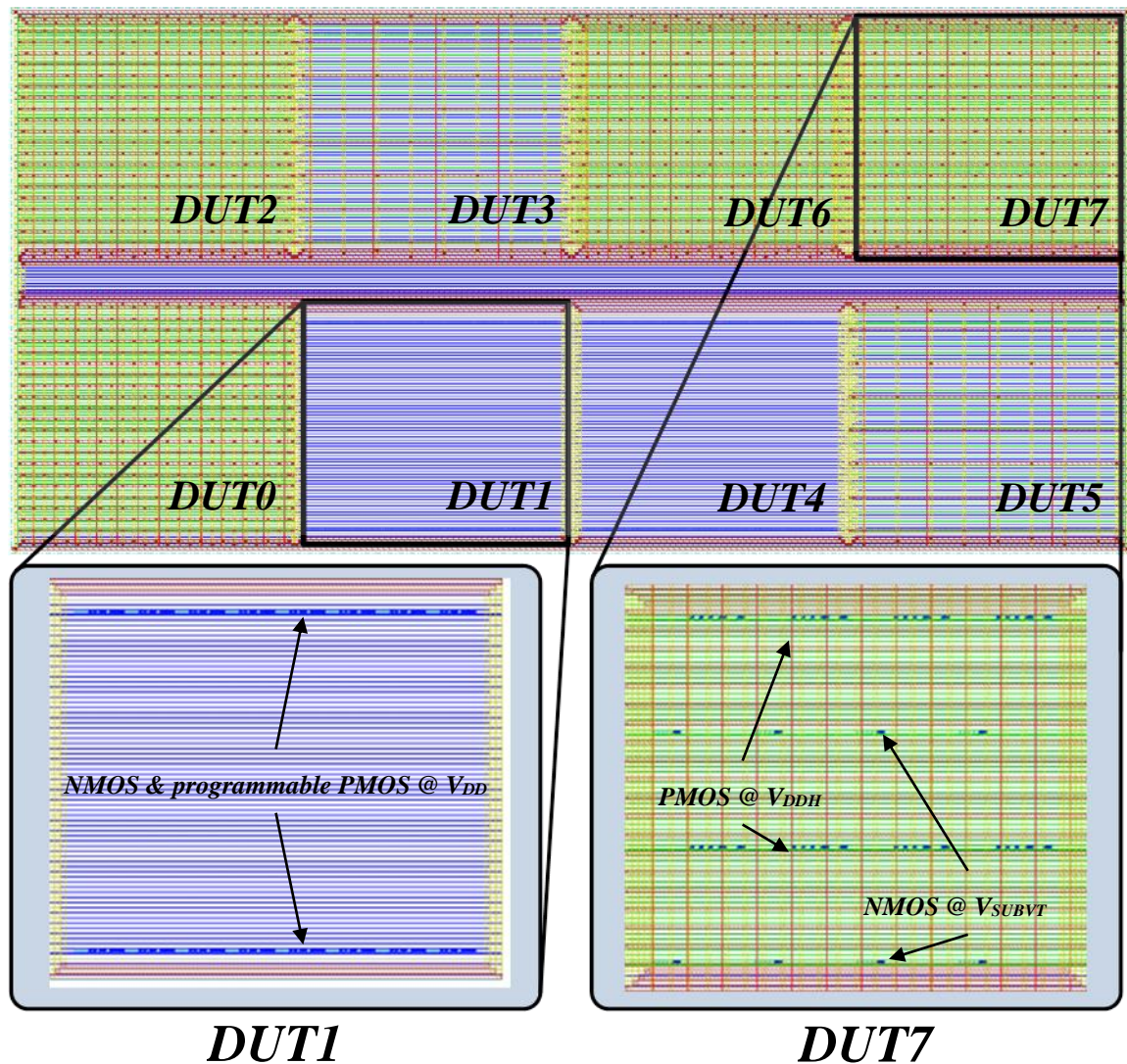


Figure 5-6: Encounter output. Each DUT has a different power delivery network configuration. DUT1 includes a programmable resistive power grid and NMOS header power switch connected to  $V_{DD}$ . DUT7 includes PDVS, NMOS header power switch connect to  $V_{SUBVT}$ , and a programmable resistive grid connected to  $V_{DDH}$ .



with Cadence Abstract generator.

### 5.3.2 SoC Integration

To put the entire design together all the components (custom and digital) need to have a LEF and a structural RTL is created to connect all the components together. For the custom blocks the LEF is generated from the Abstract generator scripts. For the digital components, a LEF is created with the P&R infrastructure. A sample Verilog top level file is shown in Appendix C. It describes the top level connections of the entire SoC. All the components need an empty module definition that matches the LEF pin definition. Figure 5-8 shows an example of a SoC generated with the scripted research and design infrastructure. This example contains custom block (power management, analog front end, radios, and memories) and over than 15 digital blocks.

### 5.3.3 Tapeout Results

To show the benefit of the research and design infrastructure we show in Table 5-1 the major tapeouts from 2009-2014 from the RLPVLSI research group. In the table, the green color code signifies the primary research and design was done using developed infrastructure. The yellow indicates part of the infrastructure was used, such as only P&R or pad ring generation. As can be seen in the table, as the infrastructure was being developed the frequency of tapeouts increased, many of them using all or some part of the scripted research and design infrastructure.

## 5.4 Summary & Conclusions

In previous chapters we have discussed two techniques that improve voltage scalable architectures and discussed architectural and power switch optimizations for enabling subthreshold operation. In this chapter we demonstrated a research and design platform

capable of implementing the aforementioned techniques and optimizations enabling quick design exploration as well as integrate a full SoCs

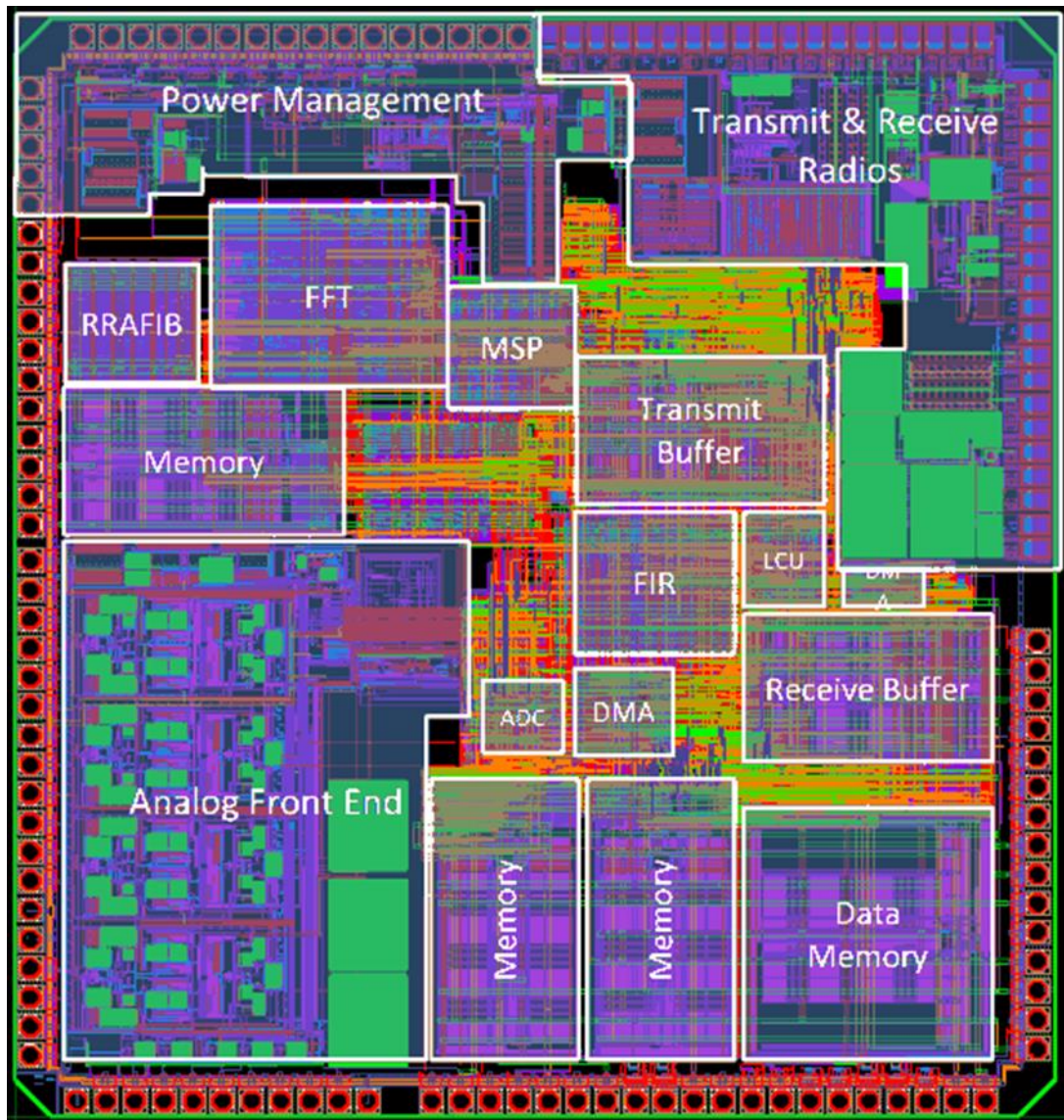


Figure 5-8: SoC generated with the scripted research and design infrastructure. This example contains both custom (power management, analog front end, radios and memories) and over 15 digital blocks.

Table 5-1: Recent tapeout schedule: Green = used infrastructure as primary design/research tool. Yellow = used some part of the infrastructure

Name	PDK	Foundry	Date	SoC Design	Synthesis/ P& R
Collaborative Design 2	IBM 130	Mosis	Feb. 2014		
ADC Test Chip	IBM 130	Mosis	Nov. 2013		
FGPA Test Chip	IBM 130	Mosis	Nov. 2013		
Multi Project Design 3 (bug fix)**	IBM 130	Mosis	Nov. 2013		
Assist Revision 1**	IBM 130	Mosis	Nov. 2013		
MSP Test Chip	MITLL 90	MITLL	Jun. 2013		
PsiKick Design**	IBM 130	Mosis	Nov. 2013		
Assist Revision 0**	IBM 130	Mosis	Feb. 2013		
Multi Project Design 3**	IBM 130	Mosis	Feb. 2013		
Multi Project Design 2**	IBM 130	Mosis	Aug. 2011		
Multi Project Design 1*	IBM 130	Adesto	Nov. 2011		
DC-DC converter	TSMC 65		Sept. 2011		
Collaborative SRAM	IBM 65		July 2011		
AMD Internship**	GF	GF	May 2011		
Body Sensor Node	IBM 130	Mosis	Nov. 2010		
Mouse House	ST 130nm		Apr. 2010		
PDVS Test Chip**	IBM 90	MOSIS	Nov. 2009		
Collaborative Design 1*	MITLL180	MITLL	Jun. 2009		

\*= on tapeout, \*\*= was lead SoC Designer



# Chapter 6

## Conclusions

In order to improve energy efficiency, this dissertation presented two techniques to improve voltage scaling and  $V_{DD}$  granularity. The first technique, a programmable resistive power grid, is a low cost solution for systems that have a shared common  $V_{DD}$  and a single DC-DC converter. The second technique, Panoptic Dynamic Voltage Scaling, is best suited for systems that have multiple  $V_{DDS}$  available throughout the power delivery network. Both of these techniques enable local voltage scaling without relying on DC-DC converters for scaling.

Next, to improve energy efficiency in low-end applications, we demonstrated how to enable architectures for subthreshold operation, as well as proposed using an NMOS header power switch with a nominal  $V_{DD}$  gate control. For designs with flexible  $V_{DDS}$ , a transmission gate power switch provides the most robust power switch configuration.

Finally, we demonstrated a scripted research and design infrastructure for energy efficient design space exploration and large SoC creation.

## 6.1 Summary of Contributions

### *Optimizing Voltage Scalable Architectures with a Single $V_{DD}$*

- Demonstrated a programmable resistive power grid for improved  $V_{DD}$  granularity without relying on DC-DC converter for voltage scaling.
- Demonstrated a 30% energy savings using a programmable resistive grid in silicon compared to no voltage scaling, and a 90% reduction in leakage current in a 32nm SOI four-core x86 processor SoC.
- Discussed opportunity for using this scheme in a four-core x86 processor with an estimated energy savings of up to 15%.
- Created a model for implementing and evaluating a programmable resistive grid in a large SoC.
- Discussed how to use a programmable resistive power grid to minimize di/dt noise when scaling  $V_{DD}$ . Showed a 3.5x reduction in noise compared to scaling voltage without noise minimizing techniques.

### *Optimizing Voltage Scalable Architectures with Multiple $V_{DD}$ s*

- Demonstrated the first processor using PDVS in silicon.
- Demonstrated PDVS's measured energy savings of up to 80% and 43% over single- $V_{DD}$  and multi- $V_{DD}$ .
- Demonstrated PDVS's area savings of up to 65% compared to multi- $V_{DD}$  for energy constrained systems.

- Discussed how to use PDVS to reduce  $di/dt$  noise. Showed how using stepped supply voltage scaling can reduce noise during a voltage transition to  $V_{DDH}$  by up to 40%.

### ***Enabling Subthreshold Operation***

- Discussed architectural enhancements, in the context of PDVS, to enable subthreshold operation in nominal voltage scaling designs.
- Verified functional operation of PDVS data path in subthreshold in simulation and in silicon.
- Demonstrated measured benefits of using a NMOS with a full VDD gate control as a header power switch for subthreshold operation. Showed a 280x reduction in power switch size when compared to a conventional PMOS for a given performance requirement.
- Demonstrated measured benefit of using a transmission gate power switch for design with variable requirements on the low voltage rail.

### ***A Scripted Research and Design Infrastructure***

- Presented a scripted research infrastructure capable of implementing PDVS, a programmable resistive power grid, and a NMOS header power switch.
- Presented a scripted design infrastructure for generating large scale System on Chips.

### **6.1.1 Teams and Individual Contributions**

Much of this work was not an individual effort; instead it was a collaborative effort. The work in Chapter 1 was a collaborative effort between UVA and AMD, where I spent a summer and spring semester as an intern. During my internship I developed the model and results discussed in Section 2.2.5, 2.2.6, & 2.2.7. The UVA team consisted of Yousef Shakhsheer, Sudhanshu Khanna, Saad Arrabi, and me. I was involved with the design, simulation, and measured results collected by the UVA team.

The work in Chapter 3 and in Chapter 4 section 4.2 was also a collaborative team effort. The PDVS team consisted of Yousef Shakhsheer, Sudhanshu Khanna, Saad Arrabi, and me under the guidance of Benton Calhoun and John Lach. I was primarily responsible for investigating different design knobs, the design and layout of the 32b Kogge Stone adder, the layout of the subthreshold level converters, the implementation of each data path (PDVS,  $S_{VDD}$ ,  $M_{VDD}$ , and subthreshold) in schematic and layout, and top level SoC integration. The architecture was developed and refined as a group.

Finally, the work in Chapter 5 could not have been done without RLPVLSI group as beta testers during various tapeouts over the past few years. Their feedback and insights will continue to help develop and refine the scripted infrastructure.

## **6.2 Broad Impact of this Work**

A programmable resistive power grid was demonstrated with fine  $V_{DD}$  granularity, such as an adder, as well as in a multi-core x86 SoC. Similarly, PDVS was demonstrated as another voltage scaling option capable of component level voltage scaling. Like a programmable grid, PDVS could be applied at the core level to improve energy

efficiency. These techniques have the potential to improve energy efficiency across a range of applications. They can be applied in high performance servers and personal computer to reduce cooling costs associated with data centers and avoid dark silicon problems associate with the power wall. The can also be adopted in battery powered mobile electronics, such as cell phone or tablets to improve battery lifetime.

Subthreshold operation has many applications in the low performance spectrum. For example, in body sensor nodes (BSNs), it has been shown that an ECG can operate off of harvested energy from the environment. This effectively gets rid of the lifetime concern for BSNs provided that the energy harvesting techniques can provide enough energy to run the system. This capability has the ability to drastically change the medical field. These capabilities are not just limited to medically sensors, but could be used in a wide variety of wireless applications, such as natural environment monitor, home security, etc.

### **6.3 Conclusions and Open Problems**

This dissertation presented a programmable resistive power grid and PDVS to address the design challenges of voltage scaling without relying on DC-DC converter and  $V_{DD}$  granularity. We demonstrated how to enable subthreshold operation for architectures conventionally operating in the nominal  $V_{DD}$  range. We proposed the use of an NMOS header power switch with a nominal  $V_{DD}$  gate control as a subthreshold power switch or power gate. For designs with a flexible  $V_{DD}$  requirement, a transmission gate power switch provides the most robust power switch configuration. Finally, we demonstrated a scripted research and design infrastructure for design space exploration and creating SoCs.

A programmable resistive power grid explored the benefits of splitting a monolithic power switch into parallel, independently controlled, variable-weighted power gates to provide programmable post-fabrication power grid resistance. This power gate topology creates energy saving opportunities by providing adjustable localized voltages during active modes and reducing leakage current in idle blocks while retaining data. Measurements showed over 30% active energy savings per operation and 90% savings in idle current with data retention. A modeling flow for a resistive power grid was also developed that demonstrates the effectiveness of this approach in a Bulldozer processor core. A noise minimization technique was also shown.

However, there are still open questions with this technique. The currently implementation of this technique has no monitoring or feedback control of the virtual- $V_{DD}$ . In the event of a sudden change in performance/current draw, it is possible it collapse the virtual- $V_{DD}$  further than expected and reduce performance, or fail to meet system requirements. We explored the potential benefits of this scheme using standard P-state occupancy, but to utilize power grid modification, the control scheme for P-state control would also need to be modified. We presented a noise minimization technique, and for the purpose of this work explored the worst case noise reduction from returning a power gated block to the nominal  $V_{DD}$ . Further work should explore the noise impact when multiple components are turning on and/or scaling  $V_{DD}$ .

Through spatial and temporal granularity, PDVS is able to improve energy efficiency over the conventional architectures  $SV_{DD}$  and  $MV_{DD}$ . Measured energy savings across seven benchmark DFGs showed an energy savings up to 80% and 43% over  $SV_{DD}$  and  $MV_{DD}$ , respectively. For given energy constraint, PDVS had an energy savings of 65%

compared to  $MV_{DD}$ . Our 32bit data flow processor was designed and fabricated in a conventional 90nm CMOS process. Finally to address the design challenge of  $di/dt$  noise, we showed how SVS could be used to reduce noise by transitioning from  $V_{DDL}$  to  $V_{DDM}$  to  $V_{DDH}$  instead of just  $V_{DDL}$  to  $V_{DDH}$ . This technique showed a 40% reduction in peak to peak noise.

There are still many interesting open questions associated with PDVS. The 32b data flow processor achieved voltage scaling by assigning voltage values with specific instructions at program time. There was no active monitoring or feedback to control when the voltage scaling would occur, it was assigned a priori. The next step for PDVS would be to incorporate the PDVS architecture into a conventional DVS scheduling methodology, such as P-state occupancy for large multi-core SoCs. For demonstration purposes, PDVS was applied to custom created processor architecture. Another future step for PDVS would be to apply the PDVS power delivery network modification to a conventional pipeline commercial processor. When considering a multi-core SoC, an interesting open question is the  $V_{DD}$  granularity to use, such as RLM level granularity. A traditional power gated design has three power/ground nets ( $V_{DD}$ , virtual- $V_{DD}$ , and  $V_{SS}$ ), while PDVS has five ( $V_{DDH}$ ,  $V_{DDM}$ ,  $V_{DDL}$ , virtual- $V_{DD}$ , and  $V_{SS}$ ). One final issue is the physical impact of the metal allocation of PDVS compared to a power gated design and how to minimize the impact of PDVS. One potential avenue to explore with the PDVS metal allocation is the ratio of metal given for each voltage level. For example, it is possible that the metal allocation is not even and  $V_{DDH}$  and virtual- $V_{DD}$  would require the most metal,  $V_{DDM}$  the second most and  $V_{SUBVT}$  the least. A study exploring this intuition and trade-off would be interesting future work.

Finally, we demonstrated architectural organizations and power delivery network optimizations to enable subthreshold in the context of PDVS. These same concepts are generally applicable to all designs. We demonstrated the best practices for modifying architectures to be able to operate in both nominal and subthreshold modes of operation. We demonstrated the best body connection for PMOS or transmission gate based power switches for the low voltages. We demonstrated the use of an NMOS transistor as a power switch for subthreshold voltages. Simulations and measurements show that an NMOS device with a nominal gate swing ( $0 - V_{DDH}$ ) is able to provide the target frequency degradation with a size over 280X smaller than a PMOS. For flexible designs that have a wide range of  $V_{DD}$  rail values we proposed using an imbalanced transmission gate power switch with a near minimum sized NMOS device in parallel with a large PMOS. The transmission gate will provide the targeted frequency degradation at the nominal  $V_{DD}$  and provide a better than target frequency in subthreshold with minimal additional area.

One of the primary assumptions with an NMOS or transmission gate power switch is the ability to have a nominal gate swing ( $0 - V_{DDH}$ ). This assumption is valid in the context of PDVS, or other multiple voltage systems, but one open question would be to compare the NMOS header power switch capability across a range of gate voltages. Similarly, we proposed to use a transmission gate power switch for flexible design with a wide range to potential  $V_{DD}$  values. Our analysis assumed a fixed width power gate size. Another open question would be the NMOS header power switch performance at near threshold across a range of power switch sizes. One final open question would be an electron migration study with an NMOS header power switch. With the reduced power switch size, the

current in the power delivery network will be forced to funnel through the small NMOS devices which potentially presents reliability problems. A study on how the electron migration impact, and reduction techniques would be interesting. One possible solution to this problem would be simply to distribute the NMOS header power switch across the design. This would reduce the area savings, but would allow the current to flow evenly through the power delivery network.

The culmination of all the power delivery network energy efficient design techniques came in a research and design platform capable of implementing all the aforementioned optimizations. The research and design tool is a scripted infrastructure enabling quick design exploration as well as full subthreshold SoCs integration. This infrastructure was developed to aid and improve our research group's ability to do research, including design large scale SoCs.

As with any infrastructure or tool, this is a constant ongoing process and will be constantly revised and updated by future students. The current tool flow was developed primarily using the Cadence design environment. One future direction would be to incorporate other EDA tools as well. For example, we used Cadence RTL Compiler for synthesis and Encounter for place and route. However, in industry it is very common to use Synopsis Design Compiler for synthesis. Adding support for DC and having both tools incorporated could be beneficial to our research groups. For power delivery network analysis there are two commercial tools that could be incorporated into our infrastructure. The first is Apache Redhawk (used for the Bulldozer modeling), and the second is Encounter Power System (EPS). Preliminary work has been done with incorporating EPS into the flow, but further work would incorporate it fully.



# Appendix A: List of Acronyms

<b>ALU</b>	Arithmetic Logic Unit
<b>AMD</b>	Advanced Micro Devices
<b>BSN</b>	Body Sensor Node
<b>CMOS</b>	Complementary Metal Oxide Semiconductor
<b>CPF</b>	Common Power Format
<b>DC</b>	Direct Current
<b>DCVS</b>	Differential Cascade Voltage Switch
<b>DFG</b>	Data Flow Graph
<b>DGEMM</b>	Double-precision General Matrix Multiply
<b>DIBL</b>	Drain Induced Barrier Loading
<b>DTCMOS</b>	Dynamic Threshold CMOS
<b>DUT</b>	Design Under Test
<b>DVFS</b>	Dynamic Voltage and Frequency Scaling
<b>DVS</b>	Dynamic Voltage Scaling
<b>ECG</b>	Electrocardiogram
<b>ED</b>	Energy-Delay

<b>EDA</b>	Electronic Design Automation
<b>EDP</b>	Energy Delay Product
<b>ELC</b>	Encounter Library Characterizer
<b>EPS</b>	Encounter Power System
<b>FFT</b>	Fast Fourier Transfer
<b>FIR</b>	Finite Impulse Response
<b>FPS</b>	Frames Per Second
<b>GCD</b>	Greatest Common Divisor
<b>GF</b>	Global Foundries
<b>HDL</b>	Hardware Description Language
<b>IC</b>	Integrated Circuit
<b>LC</b>	Level Converter
<b>LEF</b>	Library Exchange File
<b>LFSR</b>	Linear Feedback Shift Register
<b>LIB</b>	Liberty
<b>MAC</b>	Multiply Accumulate
<b>MITLL</b>	MIT Lincoln Labs

<b>MTCMOS</b>	Multi-threshold CMOS
<b>MV<sub>DD</sub></b>	Multi-V <sub>DD</sub>
<b>NMOS</b>	N-type Metal Oxide Semiconductor
<b>P&amp;R</b>	Place and Route
<b>PCB</b>	Printed Circuit Board
<b>PDK</b>	Process Design Kit
<b>PDVS</b>	Panoptic Dynamic Voltage Scaling
<b>PLL</b>	Phase Lock Loop
<b>PMOS</b>	P-type Metal Oxide Semiconductor
<b>P-state</b>	Processor State
<b>RC</b>	RTL Compiler
<b>RLC</b>	Resistor Inductor Capacitor
<b>RLM</b>	Route Level Macro
<b>RO</b>	Ring Oscillator
<b>RTL</b>	Register Transfer Logic
<b>SoC</b>	System on a Chip
<b>SOI</b>	Silicon on Insulator

<b>SV<sub>DD</sub></b>	Single-V <sub>DD</sub>
<b>SVS</b>	Stepped Supply Voltage Scaling
<b>TCL</b>	Tool Command Language
<b>TDP</b>	Thermal Design Point
<b>V<sub>DD</sub></b>	Positive Supply Voltage
<b>V<sub>GS</sub></b>	Gate to Source Voltage
<b>V<sub>SS</sub></b>	Negative Supply Voltage, Ground
<b>V<sub>T</sub></b>	Threshold Voltage

# Appendix B: List of Publications

- [KAC1] **Craig, K.**, S. Arrabi, Y. Shakhsheer, S. Khanna, et al., "A 32b 90nm Processor Implementing Panoptic DVS Achieving Energy Efficient Operation from Sub-threshold to High Performance" *Journal of Solid State Circuits*, 02/2014.
- [KAC2] Calhoun, B.H., K. **Craig**, "Flexible on-chip power delivery for energy efficient heterogeneous systems," *Design Automation Conference (DAC)*, 2013
- [KAC3] **Craig, K.**, Y. Shakhsheer, S. Khanna, S. Arrabi, J. Lach, B. H. Calhoun, and S. Kosonocky, "A Programmable Resistive Power Grid for Post-Fabrication Flexibility and Energy Tradeoffs", *International Symposium on Low Power Electronics and Design*, 2012.
- [KAC4] **Craig, K.**, Y. Shakhsheer, and B. H. Calhoun, "Optimal Power Switch Design for Dynamic Voltage Scaling from High Performance to Subthreshold Operation", *International Symposium on Low Power Electronics and Design*, 2012.
- [KAC5] Shakhsheer, Y., S. Khanna, **K. Craig**, S. Arrabi, J. Lach, and B. H. Calhoun, "A 90nm Data Flow Processor Demonstrating Fine-grained DVS for Energy Efficient Operation from 0.25V to 1.2V", *Custom Integrated Circuits Conference*, San Jose, 2011.
- [KAC6] **Craig, K.**, Y. Shakhsheer, and B. H. Calhoun, "Optimal Power Switch Design for Panoptic Dynamic Voltage Scaling Enabling Subthreshold Operation", *International Subthreshold Microelectronics Conference*, 09/2011.

- [KAC7] Khanna, S., **K. Craig**, Y. Shakhshier, S. Arrabi, J. Lach, and B. Calhoun, "Stepped Supply Voltage Switching for Energy Constrained Systems", *ISQED*, 2011.
- [KAC8] Calhoun, B. H., Y. Zhang, S. Khanna, **K. Craig**, Y. Shakhshier, J. Lach, "A Sub-Threshold FPGA: Energy-Efficient Reconfigurable Logic." *GOMACTech*, 03/2011.
- [KAC9] Calhoun, B. H., S. Arrabi, S. Khanna, Y. Shakhshier, **K. Craig**, J. Ryan, and J. Lach, "REESES: Rapid Efficient Energy Scalable ElectronicS", *GOMACTech*, 03/2010.

# Appendix C: Sample Top Level Verilog Code

```
module exampleTop(
//Digital pad ring IO (not used for routing)
    inout    PAD_BLOCKA_IN,    PAD_BLOCKA_OUT1,    PAD_BLOCKA_OUT2,
    inout    PAD_BLOCKB_IN1,    PAD_BLOCKB_IN2,    PAD_BLOCKB_OUT1,

//VDD/VSS IO
    inout    VDDA,            VDDDB,            VSS
);

//Define wires (good practice)
wire    PAD_BLOCKA_IN,    PAD_BLOCKA_OUT1,    PAD_BLOCKA_OUT2;
wire    PAD_BLOCKB_IN1,    PAD_BLOCKB_IN2,    PAD_BLOCKB_OUT1;
wire    VDDA,            VDDDB,            VSS;

//Wires bewteen blocka and the padring
wire    blocka_in,        blocka_out1,        blocka_out2;

//Wires bewteen blocka and the padring
wire    blockn_in1,        blockb_in2,        blockb_out;

//Wires between blocka and blockb
wire    blocka_to_b2,        blocka_to_b2,        blockb_to_a1,        blockb_to_a2;

BLOCKA instA (
//Connections to padring
    .BLOCKA_IN    (blocka_in),
    .BLOCKA_OUT1  (blocka_out1),
    .BLOCKA_OUT2  (blocka_out2),
//Connections between blocks
    .TO_BLOCKB1   (blocka_to_b1),
    .TO_BLOCKB2   (blocka_to_b2),
    .FROM_BLOCKB1 (blockb_to_a1),
    .FROM_BLOCKB2 (blockb_to_a2),
//VDD/VSS
    .VDD          (VDDA),
    .VSS          (VSS)
);

BLOCKB instB (
//Connections to padring
    .BLOCKB_IN1   (blockb_in1),
    .BLOCKB_IN2   (blockb_in2),
    .BLOCKB_OUT    (blockb_out),
//Connections between blocks
    .FROM_BLOCKA1 (blocka_to_b1),
    .FROM_BLOCKA2 (blocka_to_b2),
    .TO_BLOCKA1   (blockb_to_a1),
    .TO_BLOCKA2   (blockb_to_a2),
    .VDD          (VDDDB),
    .VSS          (VSS)
);
```

```

);

padRing padRingInst (
//Block A internal pins
    .BLOCKA_IN   (blocka_in),
    .BLOCKA_OUT1 (blocka_out1),
    .BLOCKA_OUT2 (blocka_out2),
//Block B internal pins
    .BLOCKB_IN1  (blockb_in1),
    .BLOCKB_IN2  (blockb_in2),
    .BLOCKB_OUT  (blockb_out),
//VDD/VSS
    .VDDA        (VDDA),
    .VDDB        (VDDB),
    .VSS         (VSS),
//External Pad connections (left as floating wires)
    .PAD_BLOCKA_IN (PAD_BLOCKA_IN),
    .PAD_BLOCKA_OUT1(PAD_BLOCKA_OUT1),
    .PAD_BLOCKA_OUT2(PAD_BLOCKA_OUT2),
    .PAD_BLOCKB_IN1 (PAD_BLOCKB_IN1 ),
    .PAD_BLOCKB_IN2 (PAD_BLOCKB_IN2 ),
    .PAD_BLOCKB_OUT1(PAD_BLOCKB_OUT1)
);
end module //end top

module padRing (
//External connections
    input  PAD_BLOCKA_IN, PAD_BLOCKB_IN1, PAD_BLOCKB_IN1,
    output PAD_BLOCKA_OUT1, PAD_BLOCKA_OUT2, PAD_BLOCKB_OUT,
//Interntal connections
    output BLOCKA_IN,   BLOCKB_IN1,   BLOCKB_IN1,
    input  BLOCKA_OUT1,  BLOCKA_OUT2,  BLOCKB_OUT,
    inout  VDDA,        VDDB,        VSS
);
end module

module BLOCKA (
    input  BLOCKA_IN,   FROM_BLOCKB1,   FROM_BLOCKB2,
    output BLOCKA_OUT1,  BLOCKA_OUT2,   TO_BLOCKB1,   TO_BLOCKB2,
    inout  VDD          VSS
);
end module //end blocka

moduel BLOCKB (
    input  BLOCKB_IN1,  BLOCKB_IN2,   FROM_BLOCKA1,   FROM_BLOCKA2,
    output BLOCKB_OUT,  TO_BLOCKA1,   TO_BLOCKA2,
    inout  VDD          VSS
);
end module //end blockb

```

# Bibliography

- [1] IDC., "Worldwide Server Power and Cooling Expense 2006-2010 Forecast," 2006, <http://www.mm4m.net/library/IDCPowerCoolingForecast.pdf>
- [2] Burd, T., Pering, T., Stratakos, A., Brodersen, R., "A dynamic voltage scaled microprocessor system," *ISSCC*, pp.294-295, 2000.
- [3] Gutta, S.R., et al., "A low-power integrated x86-64 and graphics processor for mobile computing devices," *ISSCC*, pp.270-272, 2011.
- [4] Soeleman, H., Roy, K., "Ultra-low power digital subthreshold logic circuits," *Low Power Electronics and Design*, pp.94-96, 1999.
- [5] Wang, A., Chandrakasan, A.P., Kosonocky, S., "Optimal supply and threshold scaling for subthreshold CMOS circuits," *Computer Society Annual Symposium on VLSI*, pp.5-9, 2002.
- [6] Zheng, C., and D. Ma, "A 10MHz 92.1%-Efficiency Green-Mode Automatic Reconfigurable Switching Converter with Adaptively Compensated Single-Bound Hysteresis Control," *ISSCC*, pp.204-205, 2010.
- [7] Zhang, K., et al., "SRAM design on 65-nm CMOS technology with dynamic sleep transistor for leakage reduction," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 4, pp. 895-901, 2005.
- [8] Wang, A, et al., *Sub-threshold Voltage Circuit Design for Ultra-Low Power Systems*. Springer, New York, NY
- [9] Jotwani, R., et al., "An x86-64 Core Implemented in 32nm SOI CMOS," *ISSCC*, pp. 106-107, 2010.

- [10] Apache Design, Inc., *Company Web Site*, <http://www.apache-da.com/products/redhawk>
- [11] Fischer, T., et al., "Design Solutions for the Bulldozer 32nm SOI 2-Core Processor Module in an 8-Core CPU," *ISSCC*, pp. 78-79, 2011.
- [12] Howard, J., et al., "A 48-Core IA-32 Message-Passing Processor with DVFS in 45nm CMOS," *ISSCC*, pp. 22-33, 2010.
- [13] Truong, D., et al, "A 167-processor 65 nm Computational Platform with Per-Processor Dynamic Supply Voltage and Dynamic Clock Frequency Scaling," *Symposium on VLSI Circuits*, pp.22-23, 2008.
- [14] Nam, B., et al., "A 52.4mW 3D Graphics Processor with 141Mvertices/s Vertex Shader and 3 Power Domains of Dynamic Voltage and Frequency Scaling," *ISSCC*, pp.278-603, 2007.
- [15] Gammie, G., et al., "A 45nm 3.5G Baseband-and-Multimedia Application Processor using Adaptive Body-Bias and Ultra-Low-Power Techniques," *International Solid-State Circuits Conference*, 2008.
- [16] Kim, W., D.M. Brooks, W. Gu-Yeon, "A fully-integrated 3-level DC/DC converter for nanosecond-scale DVS with fast shunt regulation," *International Solid-State Circuits Conference Digest of Technical Papers*, pp.268-270, 2011.
- [17] Putic, M., et al., "Panoptic DVS: A Fine-Grained Dynamic Voltage Scaling Framework for Energy Scalable CMOS Design," *ICCD*, pp.491-497, 2009.
- [18] Gutnik V., A.P. Chandrakasan, "Embedded Power Supply for Low-Power DSP," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, , vol.5, no.4, pp.425-435, 1997.

- [19] Kim, S., et al, "Understanding and minimizing ground bounce during mode transition of power gating structures," *International Symposium on Low Power Electronics and Design* , 2003.
- [20] Khanna, S., et al., "Stepped Supply Voltage Switching for Energy Constrained Systems", *International Symposium on Quality Electronic Design*, 2011.
- [21] Shrivastava A., Calhoun B.H., "A DC-DC Converter Efficiency Model for System Level Analysis in Ultra Low Power Applications," *Journal of Low Power Electronics and Applications*, pp. 215-232, 2013.
- [22] Ashouei, M., et al., "A voltage-scalable biomedical signal processor running ECG using 13pJ/cycle at 1MHz and 0.4V," *International Solid-State Circuits Conference*, pp.332-334, 2011.
- [23] Zhang, Y., et al., "A Batteryless 19  $\mu$  W MICS/ISM-Band Energy Harvesting Body Sensor Node SoC for ExG Applications," *Journal of Solid-State Circuits*, pp.199-213, 2013.
- [24] Ruhl, G., et al., "IA-32 Processor with a Wide-Voltage-Operating Range in 32-nm CMOS," *Micro*, pp.28-36, 2013.
- [25] Lutkemeier, S., et al., "A 65 nm 32 b Subthreshold Processor With 9T Multi-Vt SRAM and Adaptive Supply Voltage Control," *Journal of Solid-State Circuits*, pp.8-19, 2013.
- [26] Calhoun, B.H., Chandrakasan, A.P., "A 256-kb 65-nm Sub-threshold SRAM Design for Ultra-Low-Voltage Operation," *Journal of Solid-State Circuits*, pp.680-688, 2007.
- [27] Qazi, M., Sinangil, M.E., Chandrakasan, A.P., "Challenges and Directions for

- Low-Voltage SRAM," *Design & Test of Computers*, pp.32-43, 2011.
- [28] Wooters, S., et al., "An Energy-Efficient Subthreshold Level Converter in 130-nm CMOS," *IEEE Trans. Circuits Syst. II*, vol.57, no.4, pp.290-294, 2010.
- [29] Calhoun, B.H., et al., "Flexible Circuits and Architectures for Ultralow Power," *Proceedings of the IEEE*, vol.98, no.2, pp.267-282, 2010.
- [30] Mutoh, S., et al., "1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS," *IEEE Journal of Solid-State Circuits*
- [31] Shi, K., Howard, D. "Challenges in sleep transistor design and implementation in low-power designs," *Design Automation Conference*, 2006.
- [32] Kao, J., Chandrakasan, A.P., Antoniadis, D., "Transistor Sizing Issues And Tool For Multi-threshold Cmos Technology," *DAC*, 1997.
- [33] Shi, K., Howard, D., "Sleep Transistor Design and Implementation - Simple Concepts Yet Challenges To Be Optimum," *VLSI-DAT*, April 2006.
- [34] Pakbaznia, E., Pedram, M., "Coarse-Grain MTCMOS Sleep Transistor Sizing Using Delay Budgeting," *DATE*, 2008.
- [35] Cadence Design Systems, Inc., *Company Web Site*, [www.cadence.com](http://www.cadence.com)
- [36] Kwong, J., Chandrakasan, A.P., "Variation-driven Device Sizing for Minimum Energy Sub-threshold Circuits", *Proceedings of the 2006 International Symposium on Low Power Electronics and Designs*, pp. 8-13, 2006.
- [37] Zhang, Y., "Synthesis Based Design Techniques for Robust, Energy, Efficient Subthreshold Circuits", Ph.D. Dissertation, *University of Virginia*
- [38] Takahashi, O., Aoki, N., Silbermah, J., Dhong, S., "1 GHz Logic Circuits with Sense Amplifiers", *Symposium on VLSI Circuits*, pp. 110-111, June 1998.

- [39] Dickinson, A. G., Denker, J. S. , “Adiabatic Dynamic Logic”, *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 282-285, May 1994.
- [40] Choe, S.Y., Rigby, G.A., “A 1V Bootstrapped CMOS Digital Logic Family”, *Proceedings of the 23rd European Solid-State Circuits Conference*, pp. 352-355, Sept. 1997.
- [41] Tseng, Y.-K., Wu, C.-Y., “A New True-Single-Phase-Clocking BiCMOS Dynamic Pipelined Logic Family for High-Speed, Low-Voltage Pipelined System Applications”, *Journal of Solid-State Circuits*, pp. 68-79, Jan. 1999.
- [42] Heller, L., et. al., “Cascode Voltage Switch Logic: A Differential CMOS Logic Family”, *International Solid-State Circuits Conference*, pp. 16-17, Feb. 1984.
- [43] Albuquerque, E., Silva, M., “A New Low-Noise Logic Family for Mixed-Signal Integrated Circuits”, *IEEE Transactions on Circuits and Systems I*, pp. 1498-1500, vol. 46, issue 12, Dec. 1999.
- [44] Soeleman, H., Roy, K., Paul, B.C., “Robust Subthreshold Logic for Ultra-Low Power Operation”, *IEEE Transactions on VLSI Systems*, pp. 90-99, vol. 9, issue 1, Feb. 2001.
- [45] Kaizerman, A., Fisher, S., Fish, A., “Subthreshold Dual Mode Logic”, *IEEE Transactions on VLSI Systems*, pp. 979-983, vol. 21, issue 5, May 2013.
- [46] Reynders, N., Dehaene, W., “Variation-Resilient Sub-threshold Circuit Solutions for Ultra-Low-Power Digital Signal Processors with 10MHz Clock Frequency”, *2012 Proceedings of the ESSCIRC*, pp. 474-477, Sept. 2012.
- [47] Synopsis Inc., *Liberty Library Modeling*,  
<http://www.synopsys.com/community/interoperability/pages/libertylibmodel.aspx>

- [48] Meinerzhagen, P., Andersson, O., Sherazi, Y., Burg, A., Rodrigues, J., "Synthesis strategies for sub-VT systems," *European Conference on Circuit Theory and Design (ECCTD)*, 2011, pp.552,555, Aug. 2011