UNIVERSITY OF SOUTHAMPTON

FACULTY OF PHYSICAL AND APPLIED SCIENCES

Electronics and Computer Science

Leakage Power Minimisation Techniques for Embedded Processors

by

Jatin Nawnit Mistry

Thesis for the degree of Doctor of Philosophy

February 2013

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL AND APPLIED SCIENCES

Electronics and Computer Science

Doctor of Philosophy

LEAKAGE POWER MINIMISATION TECHNIQUES FOR EMBEDDED PROCESSORS

by Jatin Nawnit Mistry

Leakage power is a growing concern in modern technology nodes. In some current and emerging applications, speed performance is uncritical but many of these applications rely on untethered power making energy a primary constraint. Leakage power minimisation is therefore key to maximising energy efficiency for these applications. This thesis proposes two new leakage power minimisation techniques to improve the energy efficiency of embedded processors. The first technique, called sub-clock power gating, can be used to reduce leakage power during the active mode. The technique capitalises on the observation that there can be large combinational idle time within the clock period in low performance applications and therefore power gates it. Sub-clock power gating is the first study into the application of power gating within the clock period, and simulation results on post layout netlists using a 90nm technology library show 3.5x, 2x and 1.3x improvement in energy efficiency for three test cases: 16-bit multiplier, ARM Cortex-M0 and Event Processor at a given performance point. To reduce the energy cost associated with moving between the sleep and active mode of operation, a second technique called symmetric virtual rail clamping is proposed. Rather than shutting down completely during sleep mode, the proposed technique uses a pair of NMOS and PMOS transistors at the head and foot of the power gated logic to lower the supply voltage by $2V_{th}$. This reduces the energy needed to recharge the supply rails and eliminates signal glitching energy cost during wake-up. Experimental results from a 65nm test chip shows application of symmetric virtual rail clamping in sub-clock power gating improves energy efficiency, extending its applicable clock frequency range by 400x.

The physical layout of power gating requires dedicated techniques and this thesis proposes dRail, a new physical layout technique for power gating. Unlike the traditional voltage area approach, dRail allows both power gated and non-power gated cells to be placed together in the physical layout to reduce area and routing overheads. Results from a post layout netlist of an ARM Cortex-M0 with sub-clock power gating shows standard cell area and signal routing are improved by 3% and 19% respectively. Sub-clock power gating, symmetric virtual rail clamping and dRail are incorporated into power gating design flows and are compatible with commercial EDA tools and gate libraries.

Contents

Declaration of Authorship xv

Acknowledgements xvi

1 Introduction 1
- 1.1 Power in Digital Circuits . . . 2
  - 1.1.1 Dynamic Power . . . 2
  - 1.1.2 Leakage Power . . . 4
- 1.2 Dynamic Power Reduction . . . 6
  - 1.2.1 Clock Gating . . . 6
  - 1.2.2 Glitching, Input Reordering and Gate Sizing . . . 7
  - 1.2.3 Voltage and Frequency Scaling . . . 8
- 1.3 Technology Scaling and Implications on Power . . . 8
- 1.4 Leakage Power Reduction . . . 10
  - 1.4.1 Power Gating . . . 11
    - 1.4.1.1 Physical Implementation . . . 15
  - 1.4.2 Minimum Energy Computation . . . 18
- 1.5 Applications . . . 22
- 1.6 Thesis Organisation . . . 23
- 1.7 Contributions . . . 24

2 Literature Survey 27
- 2.1 Design Time: Transistor and Gate Level Techniques . . . 28
- 2.2 Runtime: Standby Mode Leakage Techniques . . . 31
  - 2.2.1 Power Gating . . . 31
    - 2.2.1.1 Header Vs Footer . . . 32
    - 2.2.1.2 Power Gating Alternatives . . . 34
  - 2.2.2 Natural Transistor Stacks . . . 38
  - 2.2.3 Body Biasing . . . 39
- 2.3 Runtime: Active Mode Leakage Techniques . . . 40
  - 2.3.1 Power Gating . . . 41
  - 2.3.2 Adaptive Body Biasing . . . 44
  - 2.3.3 Subthreshold . . . 46
- 2.4 Physical Layout . . . 47
- 2.5 Objectives . . . 49
- 2.6 Concluding Remarks . . . 51

3 Active Mode Sub-Clock Power Gating 53

3.1 Motivation . . . 54
3.2 Proposed Sub-Clock Power Gating Technique . . . 57
3.2.1 Sub-Clock Power Gating Architecture . . . 57
3.2.2 Design Flow . . . 60
3.3 Simulation Results . . . 60
3.3.1 Case Study 1: 16-bit Multiplier . . . 64
3.3.2 Case Study 2: ARM Cortex-M0 . . . 70
3.3.3 Case Study 3: Event Processor . . . 76
3.4 Comparative Analysis with Subthreshold . . . 79
3.5 Concluding Remarks . . . 82

4 Symmetric Virtual Rail Clamping for Sub-Clock Power Gating 83
4.1 Wake-Up Energy Cost . . . 84
4.1.1 Power Gating Techniques . . . 84
4.1.2 Power Gating Techniques Comparison . . . 88
4.2 Sub-Clock Power Gating with Symmetric Virtual Rail Clamping . . . 89
4.3 Implementation . . . 90
4.3.1 Silicon Design Flow . . . 93
4.3.1.1 Design Preparation for Sub-Clock Power Gating . . . 93
4.3.1.2 Layout: Design Planning . . . 96
4.3.1.3 Verification . . . 98
4.3.2 Test Chip Overview . . . 100
4.4 Experimental Results . . . 101
4.4.1 Symmetric Virtual Rail Clamping Vs Shut Down Power Gating . . . 103
4.4.2 Effect of Duty Cycle . . . 107
4.4.3 Sub-Clock Power Gating with Symmetric Virtual Rail Clamping Analysis . . . 108
4.4.4 Ground Bounce Analysis . . . 111
4.5 Concluding Remarks . . . 113

5 dRail: A Physical Layout Technique for Power Gating 115
5.1 Motivation . . . 116
5.2 Proposed dRail Technique . . . 120
5.2.1 dRail Layout . . . 121
5.2.2 dRail Design Flow . . . 123
5.2.3 Design Considerations . . . 125
5.3 Experimental Results . . . 126
5.3.1 Case Study 1: ARM Cortex-M0 with SCPG . . . 128
5.3.2 Case Study 2: ARM Cortex-A5 Data Engine . . . 133
5.3.2.1 dRail Vs Voltage Area . . . 135
5.3.2.2 Bounded dRail . . . 138
5.4 Concluding Remarks . . . 142

6 Conclusion and Future Work 145
6.1 Thesis Contributions . . . 145
6.2 Future Work Directions . . . 149
6.2.1 Improved Sub-Clock Power Gating . . . 149

6.2.2 Further Applications of Symmetric Virtual Rail Clamping 150
6.2.3 Physical Layout for Body Biasing 150

A Microprocessor Details **153**
A.1 ARM Cortex-M0 153

B Benchmarks and Simulation **157**
B.1 Dhrystone Benchmark 157
B.2 Energy Harvester Tuning Program 159
B.3 HSpice Simulation 164
B.3.1 Prerequisites 165
B.3.2 Extract RC netlist 167
B.3.3 Simulation Vectors 167
B.3.4 Netlist Simulation with HSpice 169

C Scripts **173**
C.1 Test Chip ARM Cortex-M0 UPF 173
C.2 dRail LEF Modification Script 177

References **179**

List of Figures

1.1 Dynamic power charging and discharging of load capacitance . . . 3
1.2 Subthreshold, band-to-band tunneling and gate leakage currents in MOSFETs . . . 5
1.3 Recirculation multiplexer to clock gate (based on [16]) . . . 7
1.4 ITRS projection for dynamic and leakage power dissipation per device [6] 10
1.5 Digital circuit execution schedule with no power gating . . . 12
1.6 Digital circuit execution schedule from Fig. 1.5 using power gating in idle time (based on [3]) . . . 12
1.7 Example of coarse grain power gating (based on [3]) . . . 13
1.8 In-rush current using simultaneous power gate switch on and staggered switch on [45] . . . 14
1.9 Typical clamp low and clamp high isolation gates . . . 15
1.10 Control of signals during power down and power up . . . 15
1.11 Physical design flow for power gating (based on [3]) . . . 16
1.12 Example of sleep transistor connection in physical layout (based on [3]) . 18
1.13 Sleep transistor insertion methods for a voltage area [40] . . . 19
1.14 Delay of an inverter against V_{dd} (130nm Technology) [37] . . . 20
1.15 Simulation of a 50 stage inverter chain (130nm process): Top - Power as a function of V_{dd}, Bottom - Energy per operation as a function of V_{dd} [37] 21

2.1 Opportunities for leakage power reduction . . . 27
2.2 Path balancing during design time (based on [73]) . . . 28
2.3 Stack Effect (based on [88]) . . . 30
2.4 Example of zig-zag power gating [95] . . . 32
2.5 Example of fine grain power gating with an *NAND* gate [40] . . . 33
2.6 Example of retention register (based on [3]) . . . 35
2.7 Virtual Rail Clamping using a MOSFET [103] (a) RUN/IDLE mode for normal operation, (b) COLD mode for sleep with full shut down and (c) PARK mode for sleep with state retention . . . 36
2.8 Multiple sleep mode power gating using a bias generator [105] . . . 37
2.9 Drowsy power gating of cache line [108] . . . 38
2.10 Reverse body biasing [32] . . . 40
2.11 Power gating of individual executional units [121] . . . 41
2.12 Grouping of logic gates into power domains controlled by clock enable signals [128] . . . 43
2.13 Adaptive body biasing scheme using feedback [135] . . . 45
2.14 Physical layout techniques for power gating . . . 48

3.1 Idle time within the clock period from reduced clock frequency 54
3.2 Increased power consumption at a given low performance target due to leakage power dissipation at fixed V_{dd} 56
3.3 Sub-clock power gating technique 58
3.4 Isolation control circuit 59
3.5 Sub-clock power gating timing 59
3.6 Design flow of the sub-clock power gating technique 61
3.7 Experimental flow for generation of sub-clock power gating power results . 62
3.8 Effective PMOS power gating transistor width against IR drop and ground bounce in the 16-bit parallel multiplier 63
3.9 Effective PMOS power gating transistor width against sleep current in the 16-bit parallel multiplier 64
3.10 Example of a 4x4 sum of partial products parallel multiplier 65
3.11 Example of how the multiplier circuit is mapped into the SCPG technique 66
3.12 Example of how modules in Verilog are mapped into the power domain definitions in the UPF 67
3.13 16-bit parallel multiplier, V_{dd}=0.6V 69
3.14 Complete ARM Cortex-M0 block diagram [160], implemented blocks highlighted 71
3.15 Example of how the Cortex-M0 Core block is mapped into the proposed SCPG technique 72
3.16 Switching probability of the Cortex-M0 for each set of 10 vectors from Dhrystone benchmark 73
3.17 Cortex-M0, V_{dd}=0.6V 74
3.18 Architecture of the Event Processor [63] 76
3.19 Event processor state machine, V_{dd}=0.6V 78
3.20 Supply voltage Vs energy per operation, 16-bit parallel multiplier 79
3.21 Supply voltage Vs energy per operation, Cortex-M0 81

4.1 An inverter with (a) single rail clamping [103] (b) symmetric virtual rail clamping 85
4.2 V_{dd} reduction against time for three power gating techniques 86
4.3 Output deviation of 4-input NOR gate from Synopsys 90nm library with virtual rail clamping (VRC) and symmetric virtual rail clamping (SVRC) with reduced supply voltage 87
4.4 Sub-clock power gating technique with symmetric virtual rail clamping . . 89
4.5 Combinational logic timing of sub-clock power gating technique with symmetric virtual rail clamping 90
4.6 Power intent of sub-clock Cortex-M0 microprocessor 92
4.7 How the Perl script identifies gates and records their connections 94
4.8 Voltage area and power gate placement in sub-clock Cortex-M0 97
4.9 Simulated VV_{dd} and VV_{ss} in Cortex-M0 with symmetric virtual rail clamping 98
4.10 Expanded verification flow for the silicon fabrication 99
4.11 Simulated transient behaviour of isolation enable signal *ISOLATE*. Left - entering sleep, Right - exiting sleep 99
4.12 Final Layout of test chip and sub-clock ARM Cortex-M0 101
4.13 Clock modulator circuit 101

4.14 Testboard for experimental measurement . . . 102
4.15 Measured VV_{dd} and VV_{ss} behaviour in sub-clock power gating using symmetric virtual rail clamping . . . 103
4.16 Measured VV_{dd} and VV_{ss} behaviour in sub-clock power gating using shut down power gating . . . 104
4.17 Measured VV_{dd} charge-up and evaluation time in SCPG with shut down power gating . . . 104
4.18 Measured effective V_{dd} reduction against time . . . 105
4.19 Measured Cortex-M0 power with power gating disabled, proposed SCPG with symmetric virtual rail clamping and SCPG with shut down power gating . . . 106
4.20 Normalised measured power of ARM Cortex-M0 microprocessor with 10kHz clock at varying duty cycle in SCPG mode, V_{dd}=0.7V . . . 108
4.21 Dhrystone - Measured power of ARM Cortex-M0 at varying clock frequency, V_{dd}=0.7V . . . 109
4.22 Tuning Program - Measured power of ARM Cortex-M0 at varying clock frequency, V_{dd}=0.7V . . . 111
4.23 Measured ground bounce on the always-on V_{ss} supply rail . . . 112
4.24 Measured charge-up time with varied number of active power gates (PGs) in proposed SCPG with symmetric virtual rail clamping . . . 113

5.1 D-type flip-flop standard cell in TSMC 65nm ARM Artisan™ library [155] 116
5.2 Conventional standard cell placement and power delivery with no power gating . . . 117
5.3 Example of standard cell separation and power delivery with power gating and voltage area . . . 118
5.4 Different combinational and sequential voltage area locations for Cortex-M0 from Chapter 4 . . . 119
5.5 Shrinking of V_{dd} and V_{ss} pins to stop power and ground sharing . . . 121
5.6 Power routing and hook-up in the dRail layout for a single VV_{dd} . . . 122
5.7 Power gating physical design flow for (a) traditional voltage area (b) dRail 124
5.8 Spreading of standard cells due to inclusion of routing channel . . . 125
5.9 Changes to LEF definition of an inverter logic gate in the TSMC 65nm ARM Artisan™ Library [155] for dRail technique . . . 127
5.10 D-type flip-flop from Fig. 5.1 modified for dRail . . . 128
5.11 Site row spacing and Metal1 Metal2 rail creation . . . 130
5.12 Power hook-up in the dRail layout . . . 131
5.13 Physical layout of ARM Cortex-M0 with sub-clock power gating using traditional voltage area layout (left) and dRail (right) . . . 132
5.14 Distribution of signal routing in sub-clock power gated Cortex-M0 using voltage area and dRail . . . 133
5.15 Top level block diagram of an ARM Cortex-A5 processor core [165] . . . 134
5.16 Floorplan of A5 with interaction of Data Engine and Data Processing Unit (a) no power gating (b) DE power gated with voltage area (c) DE power gated with dRail . . . 136
5.17 Floorplan of A5 with interaction of Data Engine and Data Processing Unit (a) DE power gated with partial dRail (b) DE power gated with dRail on interface . . . 139

5.18 Normalised total area with cell area and PG area overhead shown 141
5.19 Normalised total routing length 141
5.20 Normalised dynamic and leakage power at 400MHz 141

A.1 Block diagram of the Cortex-M0 processor 154

List of Tables

3.1 Worst case capacitive loaded gate delays . . . 55
3.2 Power and energy per operation of sub-clock power gated multiplier, V_{dd}=0.6V . . . 68
3.3 Power and energy per operation of sub-clock power gated Cortex-M0, V_{dd}=0.6V . . . 74
3.4 Power and energy per operation of sub-clock power gated Event Processor, V_{dd}=0.6V . . . 77
3.5 Comparison of sub-clock power gating relative to subthreshold operation performance points, 16bit Multiplier . . . 80
3.6 Comparison of sub-clock power gating relative to subthreshold operation performance points, Cortex-M0 . . . 81

4.1 Ring oscillator wake-up energy, leakage saving and wake-up time . . . 88
4.2 Control signals to power gates and corresponding mode of operation . . . 91
4.3 Dhrystone - Average measured power and energy in three modes of operation, V_{dd}=0.7V . . . 105
4.4 Dhrystone - Average measured power and energy over five test chips with power gating disabled (No-PG) & sub-clock power gating (SCPG) . . . 109
4.5 Tuning Program - Average measured power and energy over five test chips with power gating disabled (No-PG) & sub-clock power gating (SCPG) . . . 111

5.1 Area comparison of voltage area and dRail layout, Cortex-M0 . . . 132
5.2 Distribution of signal routing in voltage area and dRail layout, Cortex-M0 133
5.3 Sub-clock power gated Cortex-M0 average power of voltage area and dRail layouts, V_{dd}=0.7V . . . 133
5.4 Area, routing and power in no power gating and power gating with voltage area [3], and proposed dRail, Cortex-A5 . . . 137
5.5 Area, routing and power in no power gating and power gating with voltage area [3], partial dRail, and interface dRail, Cortex-A5 . . . 140

B.1 Statistics of statements and operands in Dhrystone benchmark . . . 158

Declaration of Authorship

I, Jatin Nawnit Mistry, declare that this thesis entitled *Leakage Power Minimisation Techniques for Embedded Processors* and the work presented in it are both my own, and have been generated by me as the result of my own original research. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University;
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- Where I have consulted the published work of others, this is always clearly attributed;
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- Parts of this work have been published as listed in Section 1.7

Signed: _____ **Date:** _____

Acknowledgements

I would like to express my sincerest gratitude to Professor Bashir M. Al-Hashimi for his supervision and guidance throughout my Ph.D. I have learnt tremendous amounts from him and without his support and encouragement this work would not have been possible. I would also like to extend my gratitude to my industrial advisory team Professor David Flynn, James Myers and Stephen Hill for their invaluable support and technical discussions throughout my Ph.D. My thanks also go to my second supervisor Iain McNally, internal examiner Dr Geoff Merrett and visiting friend of Professor David Flynn, Dr Harry Oldham for their insightful discussions.

I wish to thank the Engineering and Physical Sciences Research Council (EPSRC) for supporting my work by means of scholarship, the ARM-ECS Research Center in the School of Electronics and Computer Science, University of Southampton for providing state of the art research facilities and ARM Ltd. for allowing me to spend 4 months of my Ph.D working at their offices. My special thanks also go to John Biggs, Anand Savanth, Karthik Sivashankar (ARM) and Matthew Swabey (Purdue University) for their help in fabricating and testing the silicon test chip used in this thesis.

The constructive discussions I shared with my colleagues and friends and invaluable support I have received from them throughout my Ph.D has also contributed to its successful completion. These people include, but is not limited to: Sheng Yang, Shida Zhong, Dr Mustafa Imran-Ali, Dr Saqib Khursheed, Dr Rishad Shafik, Jedrzej Kufel, Luis Maeda-Nunez, Dr Alex Weddell, Dr Richard Lowe, Dan Reid, Dr Harry Rose, Celia Yeung, Ben Waller, Dr Amit Acharyya, Taihai Chen, Alex Wood, Kier Dugan, Tom Redman and Jon Storey. My thanks go to them all.

Finally, I would like to thank my parents, Nawnit Mistry and Narmada Mistry, my brothers Ilesh Mistry and Jayesh Mistry, my sisters-in-law Neeta Mistry and Taejal Mistry, and my sister Bhavika Mistry, in addition to my incredible extended family for their continuous love, support and understanding throughout my Ph.D.

Chapter 1

Introduction

Technology scaling has been the driving force in the microelectronics industry enabling increased integration, cheaper devices, and increased performance with each new generation of CMOS. This is because higher performance and lower cost are two important design goals of many digital integrated circuits [1]. Historically, dynamic power has dominated the power consumption of digital integrated circuits and has been the main focus of power reduction for many years. However, as technology scaling has continued, leakage power has become a cause for concern in power dissipation prompting a variety of leakage reduction techniques to be developed. In some current and emerging applications there is a shift from performance driven design goals to power and energy constraints and as leakage continues to grow in dominance it presents a major obstacle in achieving these targets. This thesis describes new leakage power minimisation techniques, their physical layout and their validations for digital designs in the context of low performance energy constrained applications.

This chapter gives an overview of low power design in digital circuits and provides preliminary information for the subsequent thesis chapters. The major components of power dissipation in digital integrated circuits (IC) are described in Section 1.1. A summary of established dynamic power reduction techniques are outlined in Section 1.2. Section 1.3 discusses the impact of technology scaling on leakage power dissipation and Section 1.4 summarises some effective techniques for improving energy efficiency in digital circuits which are relevant to the work reported in this thesis. Section 1.5 gives examples of energy constrained applications where low performance microprocessors are used. The contribution of each chapter is summarised in Section 1.6 and finally the list of publications generated from the research in this thesis is given in Section 1.7.

1.1 Power in Digital Circuits

In order to design energy-efficient digital circuits it is important to understand the different sources of power dissipation, of which there are mainly two: dynamic power and leakage power [2]. Dynamic power is caused by switching activity and is dissipated whenever the digital circuit is doing useful work whereas leakage power is dissipated whenever the digital circuit is switched on regardless of whether useful work is being performed. This means that when a digital circuit is in the *active mode* of operation i.e. doing useful work, total power is contributed to by both dynamic and leakage power, however when the digital circuit is in *idle mode* leakage power is the only contributor of power dissipation. The total power of a digital circuit is therefore given by:

$$P_{total} = P_{dyn} + P_{leak} \tag{1.1}$$

1.1.1 Dynamic Power

The dynamic power in Eqn. 1.1 can be split into two components:

$$P_{dyn} = P_{switching} + P_{sc} \approx P_{switching} \tag{1.2}$$

The P_{sc} term in this equation refers to the short circuit power that is consumed when a logic gate switches due to a direct path from V_{dd} (power) to V_{ss} (ground). This can be understood with the example inverter in Fig. 1.1. As the input IN makes a $0 \rightarrow 1$ or $1 \rightarrow 0$ transition, for a short period of time both the NMOS and PMOS transistors may both be conducting and a short circuit current I_{sc} will flow. The short circuit power dissipation in most circuits can be considered negligible as long as the input signal ramp time is kept short [3]. For this reason dynamic power can be approximated by $P_{switching}$. $P_{switching}$ is due to the charging of load capacitances during switching activity and the inverter in Fig. 1.1 can be used again to understand the energy that is drawn from the power supply. C_L represents the physical load capacitance at the output node of the inverter. First consider the input IN to be set at 1 such that the PMOS transistor is off, the NMOS transistor is on and C_L is fully discharged. Now consider a $1 \rightarrow 0$ transition at the input node. The PMOS transistor connects C_L to V_{dd} which is charged through the PMOS transistor until the output OUT reaches V_{dd} at time T resulting in a $0 \rightarrow 1$ transition at the output. The power dissipation of this transition is given by:

$$P_{switching} = V_{dd}.I_{charge} = V_{dd}.C_L\frac{dV_{out}}{dt} \tag{1.3}$$

Therefore the energy drawn from the power supply in this transition is given by:

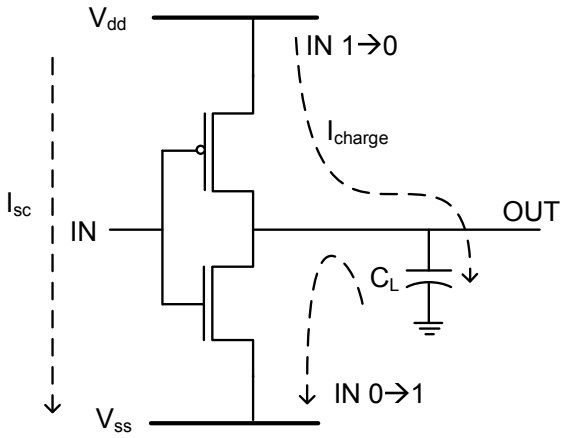


Figure 1.1: Dynamic power charging and discharging of load capacitance

$$E_{0\to1} = \int_0^T P_{switching}dt = V_{dd}.\int_0^T I_{charge}dt = V_{dd}.C_L.\int_0^{V_{dd}} dV_{out} = C_L V_{dd}^2 \tag{1.4}$$

It should be noted that the energy stored in the capacitor C_L is $\frac{1}{2}C_L.V_{dd}^2$, as half of the energy is dissipated in the PMOS transistor [2]. Furthermore during a $0 \to 1$ transition at the input, the charge stored in C_L is discharged through the NMOS transistor resulting in no energy being drawn from the power supply. Although the example given here is for a simple inverter the discussion is true for more complex gates [2, 4]. Therefore, over a number of clock cycles N_c, the switching energy dissipation of an entire digital circuit can be given by:

$$E_{switching} = N_c.V_{dd}^2.C_{eff} \tag{1.5}$$

Where C_{eff} is the lumped average capacitance that is switched in the digital circuit given by the product of switching probability α and total load capacitance C_L [2]. Assuming a clock frequency of f, the average switching power of the digital circuit can then be given by:

$$P_{switching} = \frac{E_{switching}.f}{N_c} = f.V_{dd}^2.C_{eff} \tag{1.6}$$

It can be seen from the equations of switching energy (Eqn. 1.5) and switching power (Eqn. 1.6) that, assuming a constant C_{eff} determined by the design of the digital circuit, and fixed number of clock cycles N_c, then energy is proportional to the square of the supply voltage V_{dd} whereas power is proportional to the product of f and square of V_{dd}. Therefore, although reducing the clock frequency can reduce power, the energy required remains the same. It may be thought that a simple fix for improved energy consumption

is reduction in supply voltage, but there is a cost. The propagation delay of a gate can be approximated as [5]

$$T_{prop} \propto \frac{C_L V_{dd}}{(V_{dd} - V_{th})^k} \tag{1.7}$$

where V_{th} is the threshold voltage of the transistors and k is a technology dependent parameter used to model short channel effects and normally has a value between 1-2. Eqn. 1.7 shows that as V_{dd} is reduced the propagation delay increases and forces the reduction of the operating frequency.

1.1.2 Leakage Power

Leakage power unlike dynamic power is present in the digital circuit at all times and is not a function of the useful work done. This means that if a circuit is *idle* - powered but not doing useful work - it still dissipates leakage power. Leakage power is the result of parasitic current flows within the device and is dominated by mainly three sources [6]: subthreshold leakage current, band-to-band tunneling current and gate leakage current

$$P_{leak} = V_{dd}.(I_{sub} + I_{BTBT} + I_{gate}) \tag{1.8}$$

These three sources of current are represented diagrammatically in Fig. 1.2. Gate leakage is current that flows directly between the gate and the substrate due to tunneling of charge carriers through the gate oxide [7]. The current is a result of the electric field that is present across the oxide which increases with reduction in the thickness of the gate oxide, T_{ox} [6]. Gate leakage current is present at all times in a MOSFET regardless of the state of the device - conducting or not. Band-to-band tunneling (BTBT), also known as PN junction reverse-bias current, is due to the fact that the drain and source to substrate junctions are effectively reverse biased diodes in a MOSFET [2]. PN junction reverse-bias leakage is a function of the junction area and the doping concentration, causing the tunneling current to increase with heavier doping of the N and P regions [6]. Like gate leakage, BTBT is present in the device when the transistor is both conducting or not. Unlike the previous two leakage current sources which are present at all times, subthreshold leakage is the flow of minority carriers from source to drain due to the partial formation of a conduction channel when a MOSFET is cut-off ($V_{gs} < V_{th}$) and can be approximated by [8]

$$I_{sub} = \mu C_{ox} V_t^2 \frac{W}{L} e^{1.8} e^{\frac{V_{gs} - V_{th}}{nV_t}} [1 - e^{-\frac{V_{ds}}{V_t}}] \tag{1.9}$$

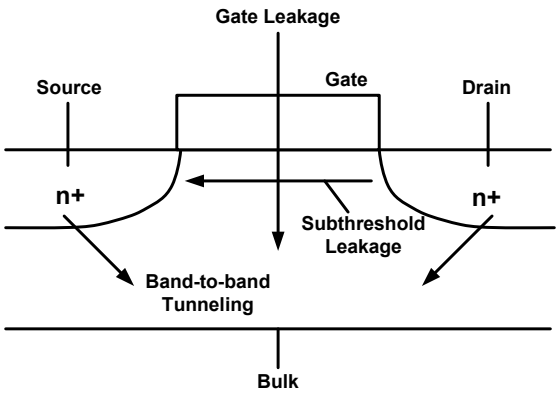


Figure 1.2: Subthreshold, band-to-band tunneling and gate leakage currents in MOSFETs

Where μ is the carrier mobility, C_{ox} is the oxide capacitance, V_t is the thermal voltage $\frac{KT}{q} = 25.9\text{mV}$ at room temperature, W is the width of the transistor, L is the length of the transistor, V_{gs} is the gate to source voltage, V_{ds} is the drain to source voltage, V_{th} is the threshold voltage and n is a function of the fabrication process. From Eqn. 1.9 it can be seen that lowering V_{ds}, achieved by lowering the supply voltage, affects the term $[1 - e^{-\frac{V_{ds}}{V_t}}]$ and when $V_{ds} \ll V_t$ this term approaches 1, significantly reducing the subthreshold leakage. Similarly, as can be seen, the subthreshold leakage current is exponentially dependent on V_{th}. Increasing V_{th} lowers the subthreshold leakage current and therefore MOSFET gates with higher threshold voltages exhibit less subthreshold leakage current [9]. The threshold voltage can be controlled in mainly two ways, the first is through the manufacturing process, the second is by changing the body-to-source voltage (V_{bs}) of a transistor and is known as the body effect [2]. Traditionally the body of a transistor is at the same potential as its source but can be raised or lowered to affect the threshold voltage [4]. This effect causes a negative V_{bs} to increase the threshold voltage whereas a positive V_{bs} lowers the threshold voltage. Both techniques will be discussed further in Chapter 2.

Taking into consideration the leakage power dissipation of a digital circuit the total energy consumed in a digital circuit over a number of cycles N_c with clock period t_p can now be expressed as:

$$E_{total} = E_{dyn} + E_{leak} = N_c.V_{dd}^2.C_{eff} + N_c.V_{dd}.I_{leak}.t_p \tag{1.10}$$

What is interesting to note with this equation is that the previous discussion with dynamic power (Section 1.1.1), where it was observed that a lower clock frequency equates to lower power but identical energy, now has a new dynamic. A lower clock frequency would lower dynamic power as per Eqn. 1.6, but would increase the clock period t_p. From Eqn. 1.10 we see that this would result in more energy lost to leakage leading to lower energy efficiency. This observation will be exploited in Chapter 3.

1.2 Dynamic Power Reduction

Dynamic power has dominated the power consumption of digital circuits for many years [2]. Although, power reduction in digital circuits can be achieved through a number of application specific changes including architecture changes and instruction set modification, these improvements are constrained to a particular processor [10]. In this thesis the focus is instead on general purpose techniques i.e. ones that can be applied to any digital circuit. Examining Eqn. 1.6, it can be seen that by reducing the effective load capacitance C_{eff} by reducing the switching activity α and load capacitances C_L, or the operating voltage V_{dd}, energy consumed to dynamic power can be lowered. Lowering V_{dd}, however, increases propagation delay, Eqn. 1.7, and therefore techniques to reduce dynamic power ideally attempt to affect these variables without hindering performance. In this section a brief overview of some of the most widely adopted dynamic power reduction techniques is given.

1.2.1 Clock Gating

The dynamic power consumption of a processor is primarily dominated by the clock distribution [11] also known as the clock tree. It has been reported that approximately 32% of the dynamic power dissipation in the Alpha 21264 Microprocessor is due to the global clock network [12]. This large dynamic power comes from the high activity and the sizable capacitive fan-out load of the clock tree. Additionally, although registers may retain the same state over two or more clock cycles, internal switching of the gate from the toggling of the clock adds to the overall dynamic power consumption. This observation of registers not being updated at every clock edge led to clock gating, which was introduced to eliminate the switching of parts of the clock tree where registers did not need to be updated. The result of this is a set of partitioned regions in the clock tree where the clock can be enabled depending on the requirement to update registers within sub-sections of a digital circuit [13]. The simplicity of clock gating enables it to be used in all types of sequential digital logic. R. Gonzalez et al. report a 33% saving in global clock network power dissipation when using clock gating [14] in a microprocessor and S. Huda et al. report over 50% clock power saving when the technique was used in FPGAs [15].

The technique is achieved by adding a logic cell, called a clock gate, into the clock tree allowing a control signal to enable or disable the clock as required. Careful planning of the insertion of clock gating has been proven to even reduce the area of a digital circuit. This is because a digital circuit without clock gating would require recirculation multiplexers to retain the state of registers, and the addition of a single clock gate can remove a number of these multiplexers, demonstrated in Fig. 1.3, leading to greater savings in power. In a case-study by K. C. Pokhrel, it is demonstrated that a circuit

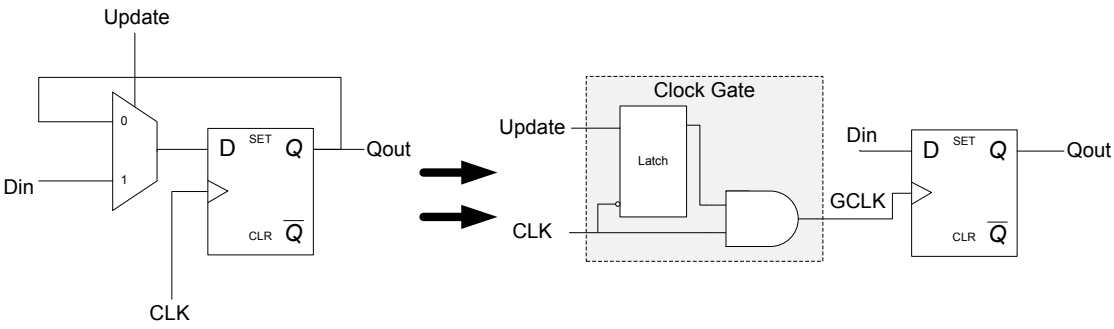


Figure 1.3: Recirculation multiplexer to clock gate (based on [16])

with clock gating is 20% smaller than the same circuit without clock gating [16]. Clock gating is now well supported in industry standard EDA (Electronic Design Automation) tools such as Synopsys Design Compiler making the insertion of clock gating automatic [17].

1.2.2 Glitching, Input Reordering and Gate Sizing

Part of the dynamic power dissipation of CMOS circuits can be accounted to glitching due to imbalanced paths to a gate's inputs [18]. As an example, consider a 2-input NAND gate with inputs '10', if delay causes the second input to change to '1' before the first input changes to '0', the output momentarily takes the value of '0'. This glitch dissipates unnecessary dynamic power due to the recharging of the output load capacitance. By using path balancing, buffers can be added to ensure inputs to gates arrive at the same time to reduce the number of glitches in the circuit by up to 61.5% [19]. The maximum saving is limited by the ability to balance both rising transitions and falling transitions to all inputs which can be difficult.

Since the dynamic power dissipation is also governed by the probability of switching load capacitances, C_{eff} dynamic power can be minimised by reordering inputs/gates to reduce the probability of switching. By carefully reordering the inputs such that the critical signal is closest to the output, dynamic power dissipation due to spurious intermediate transitions, can be minimised [20]. Probability can also be reduced through an appropriate state encoding strategy to minimise switched capacitance between state transitions [21]. If the probability cannot be reduced then the load capacitance could be reduced by using minimum size gates on non-critical paths [2]. Across a set of test circuits, Y. Huang et al. show that using an effective method of gate sizing, total power can be reduced by an average of 18.5% [22].

1.2.3 Voltage and Frequency Scaling

Dynamic Voltage and Frequency Scaling (DVFS) targets both the supply voltage and the operational clock frequency. The technique capitalises on the times when a system does not need to be operated at its maximum frequency due to workload demands [23]. At these times the voltage can be scaled down to reduce dynamic power and improve energy efficiency, Eqn. 1.5, but results in reduced performance due to the increase in delay, Eqn. 1.7, forcing the reduction of clock frequency too. T.D. Burd et al. report energy efficiency of a processor system being improved by up to 10x when using DVFS [23]. The advantage with DVFS is the ability to switch between high performance, high energy and low performance, low energy states depending on the current workload. Its efficiency at reducing dynamic power dissipation has prompted many DVFS designs and algorithms on a variety of different systems [24–26].

An alternative to DVFS is to statically scale the supply voltage of a system. By using a reduced supply voltage, the dynamic power dissipation is reduced, energy consumption is improved but performance is penalised (Eqn. 1.5 to 1.7). This is effective if a system's performance is not a primary concern as it cannot be increased back to nominal performance as is done with DVFS. This technique can be extended for use in multi-supply/multi-voltage operation [27], where different parts of a system on chip may require different levels of performance enabling a number of different supply voltages to be utilised. By using multiple supplies, *voltage islands* can be created to partition sub-sections of the system that are off critical paths, thereby saving dynamic power [3, 28]. Since the sub-section chosen is off a critical path, the path can cope with the increase in delay associated with the reduced supply voltage without affecting overall system performance [3]. This technique is most common in Systems on Chip (SoCs) that require caches to be as fast as possible whereas the CPU and the rest of the SoC can be operated at a reduced voltage whilst still meeting timing constraints [3, 27].

1.3 Technology Scaling and Implications on Power

Over the last 50 years the industry has followed a trend in scaling where geometry dimensions decrease by 30% every two to three years [2, 29]. The main reason behind this scaling trend is to reduce the cost of IC fabrication [1]. With a 30% decrease in process size, area is reduced by 50% (0.7x0.7=0.49) meaning double the number of transistors can fit in the same area on a silicon wafer, significantly reducing manufacturing cost. Additionally, a 30% reduction in device geometry leads to a 30% reduction in gate delay leading to $\frac{1}{0.7}$ improvement in integrated circuit performance [29]. There are two types of scaling that have taken place in CMOS technology: constant voltage scaling and constant field scaling. In the former case, only the device geometries, length, width and gate oxide thickness are shrunk while the supply voltage remains the same. The advantage

of this is that it provides compatibility with older circuit technologies but suffers an increase in the electric field across the channel of the MOSFET which can lead to velocity saturation, mobility degradation and lower breakdown voltages [2]. Nevertheless, constant voltage scaling was preferred at first and was used down to the 0.5μm process [29]. The alternative constant field scaling is based on 'Dennard's Scaling Law' [30] which states the performance of a transistor can be improved if the critical parameters of a device are scaled by a given factor. These parameters include the length, width and gate oxide but also the supply voltage V_{dd} and transistor threshold voltage V_{th}. By maintaining a constant electric field across the channel, the problems seen with constant voltage scaling are avoided [2] but also has the secondary advantage of reduced dynamic power consumption. If a 30% reduction in geometry and supply voltage is assumed, dynamic power should reduce by 50% with each new technology node [29]. This can be calculated by substituting the reduced geometry and supply voltage into Eqn. 1.6:

$$P_{dyn} = C_{eff}.f.V_{dd}^2 = 0.7 \times (0.7)^2 \times (1/0.7) \approx 0.5$$

Constant electric field scaling has thus resulted in smaller, faster and lower power devices. However, as each new generation of technology is introduced, the component of subthreshold leakage current has increased and this is because of the necessity to reduce the threshold voltage (V_{th}) as part of the technology scaling [6, 31]. This can be seen from Eqn. 1.9, which shows that as the threshold voltage is reduced, the subthreshold leakage current of the transistor increases exponentially. Maintaining a higher threshold voltage can help to limit the subthreshold leakage power dissipation but results in an increase in gate propagation delay (T_{prop}), seen in Eqn. 1.7. As V_{th} is increased in this equation the lower denominator reduces in size resulting in a rise in propagation delay. As explained in Chapter 2, Section 2.1, however, multiple threshold voltage gates are common in digital IC design to help reduce leakage power dissipation.

In addition to the rise in subthreshold leakage, as transistor geometries have reduced to 90nm and below, gate leakage current has become a greater source of leakage power due to the aggressive reduction in the gate oxide thickness, T_{ox} [7]. In sub-90nm technology nodes, the oxide thickness is only a few atoms thick which has caused a sharp rise in gate leakage current [2, 9]. At sub-65nm process technologies the gate leakage is measured to be as dominant as subthreshold leakage [32]. Recent introduction of high-k dielectric materials to provide better insulation between the gate and the channel has seen a reduction in gate leakage [9, 33]. Intel introduced high-k material in their 45nm technology generation and latest research shows up to 1000x reduction in gate leakage current [34]. Although gate leakage has currently been reduced with high-k dielectric, it has only been suppressed and will continue to increase again as technology scaling continues. With every technology generation, band-to-band tunneling (see Section 1.1.2) is also on the rise. As the channel length gets shorter with technology scaling there is an increase in a phenomenon known as the short channel effect [2, 6, 32]. The depletion regions surrounding the source and drain of a transistor extend into the channel and

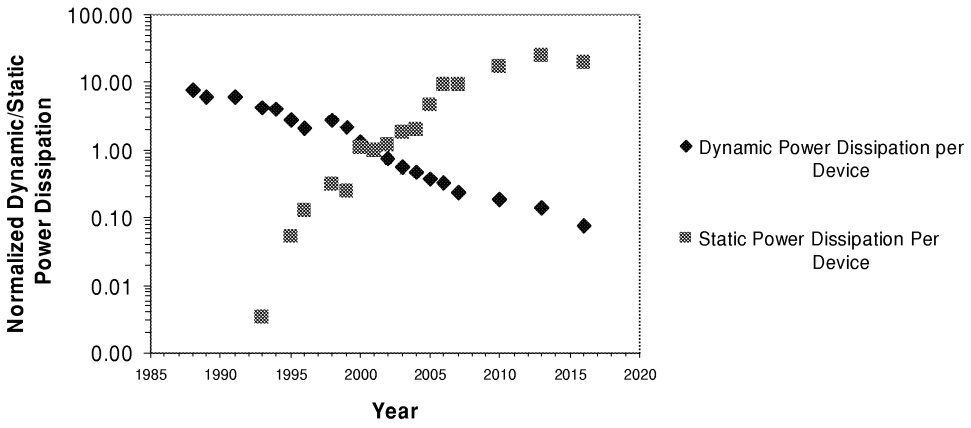


Figure 1.4: ITRS projection for dynamic and leakage power dissipation per device [6]

with shorter channel length their interaction increases, reducing the threshold voltage and increasing subthreshold leakage current [2, 6]. To control this effect a process called halo doping is used around the source and drain and within the channel to increase the doping concentrations in these regions. While it helps to control subthreshold leakage with a scaled channel length, it increases band-to-band tunneling leakage between the source/drain and substrate as BTBT leakage is a function of doping concentration [6]. It has been shown that BTBT is increasing with every generation and matches the magnitude of both the gate and subthreshold leakage components at sub-65nm process technologies [32].

The continued scaling trend projected by the ITRS [35] shows that the aggressive scaling of transistors to increase integration, reduce cost and improve performance will continue to have a positive impact on dynamic power consumption but at the cost of increased leakage power dissipation per device as shown in Fig. 1.4. For this reason, whilst leakage is a problem now it has become a major concern for future technologies [9].

1.4 Leakage Power Reduction

With the growing dominance of leakage power dissipation in digital circuits, significant research effort is being put into reducing it within digital circuits [6]. There are many techniques for reducing leakage power within a microprocessor which will be covered in Chapter 2. However, there are two techniques which are of particular interest in this thesis: power gating and subthreshold operation. Power gating is considered to be the most effective and practical technique of reducing leakage power [2–4] and will be used in Chapters 3, 4 and 5. The subthreshold technique on the other hand is an effective technique for reducing both dynamic and leakage power [36] and trades performance for energy efficiency [37] making it well suited for low performance, energy constrained applications (Section 1.5).

1.4.1 Power Gating

Power gating is a leakage power reduction technique that has gained increased popularity over the last decade and is commonly used in many processors today such as the commercially popular Nvidia Tegra processors, which are based on the ARM Cortex-A9 microprocessor [38]. The fundamental goal of power gating is to enable two modes of operation: an *active mode* of operation with which a digital circuit can continue execution as normal and a low leakage *sleep mode* of operation [3]. In its most common form, power gating achieves the low leakage sleep mode of operation by cutting off the power to the processor [39] and can therefore be referred to as 'shut down power gating'. Other forms of power gating exist which extend this idea but will be discussed in Chapter 2. For clarification, when power gating is mentioned in this thesis, shut down power gating will be assumed unless otherwise stated. Fig. 1.5 shows an execution schedule of a typical processor where two tasks are separated by a period of time where no work is done. During this period of time, the clocks may be stopped but the circuit is still powered and dissipates leakage power wasting energy. Power gating capitalises on these periods of idle time to cut off the power to the processor to reduce the leakage power. Fig. 1.6 shows the same execution schedule employing power gating. The processor begins in the *active mode* and is doing useful work. When the idle period is entered, a *sleep* signal enables the power gating technique to cut off the power and put the processor into the low leakage *sleep mode*. The processor then remains in this low leakage state until it is needed again. To resume execution of the next task, a *wake* signal restores power to the processor so that it is ready before execution begins.

Fig. 1.7, shows a conceptual view of how power gating is implemented in a digital circuit. A PMOS *power gating transistor*, also referred to as a *header* or *sleep transistor*, is placed in series with a collection of logic gates and provides the power to the entire block. The source of the power gating transistor is connected to the true V_{dd} and the drain side becomes the effective power supply rail to the logic block and is referred to as the virtual V_{dd} (VV_{dd}). A PMOS transistor is shown here but an NMOS *footer* transistor is equally viable for power gating as will be shown in Chapter 2, Section 2.2.1. Notice that the entire processor does not need to be power gated and the power gated block can interface with a subset of always-on logic. The state of the PMOS transistor is controlled with a *Sleep* control signal. By switching the power gating transistor off, the virtual rail discharges resulting in leakage current being reduced to that of the power gating transistor [3]. This method of power gating is referred to as *coarse-grain* power gating and the collection of logic gates is often referred to as a *power gated block* or *power domain*. As mentioned in Section 1.3, threshold voltages have been scaled lower with every technology generation and benefit from improved performance, however, a transistor with a higher threshold voltage exhibits lower subthreshold leakage power. To capitalise on this, the shut down power gating technique uses a high threshold voltage transistor as the power gating transistor, to further suppress sleep mode leakage,

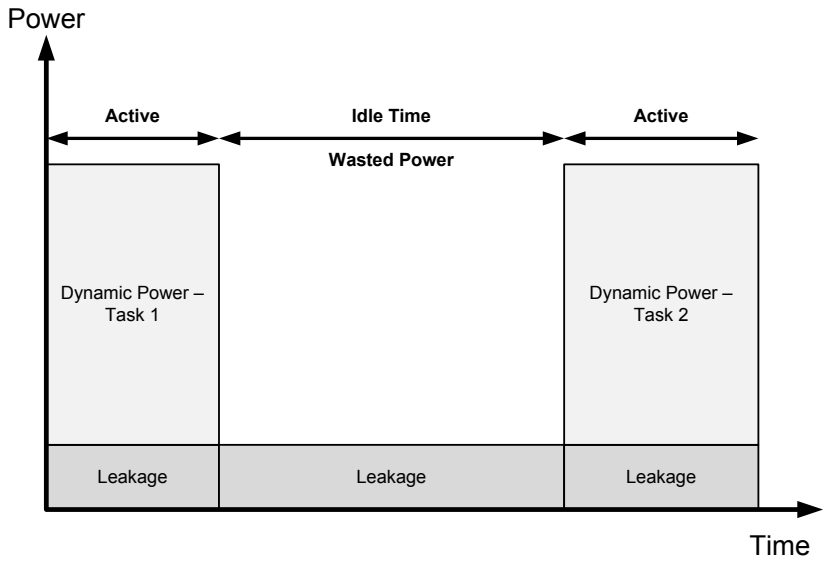


Figure 1.5: Digital circuit execution schedule with no power gating

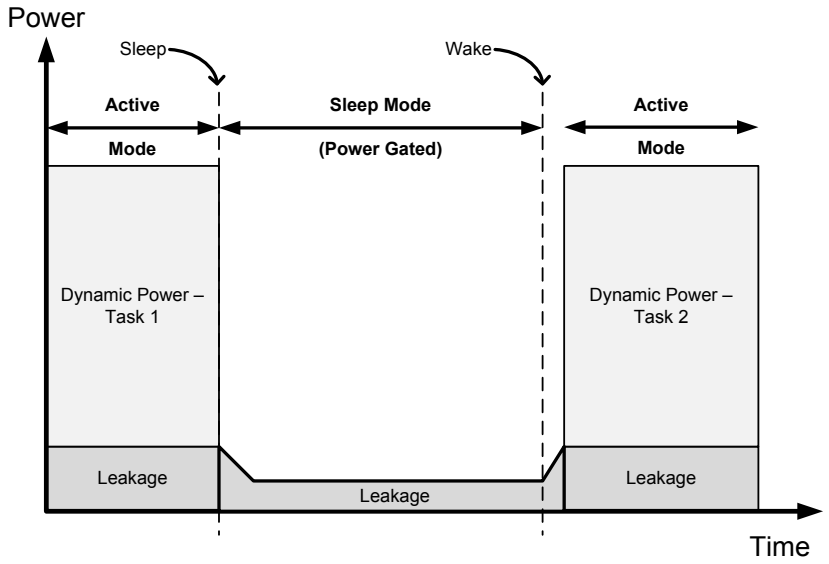


Figure 1.6: Digital circuit execution schedule from Fig. 1.5 using power gating in idle time (based on [3])

whereas low threshold voltage transistors are used in the power gated block to maintain performance. Employing low and high threshold voltages in a single design is referred to as Multi-Threshold CMOS (MTCMOS) [3, 39]. Using this technique Mutoh et al. showed a 600x reduction in leakage current when comparing active mode leakage to sleep mode leakage of a power gated block of logic [39] and up to 25x reduction in leakage power was shown in the ARM926EJ-S [3].

In practice, the power gating transistor shown in Fig. 1.7 is not a single transistor and is instead a series of distributed transistors in the physical layout [3, 40]. The inclusion of the power gating transistors introduces a small IR drop as the transistors can be modeled as resistors when the circuit is active [5]. As can be seen from the equation

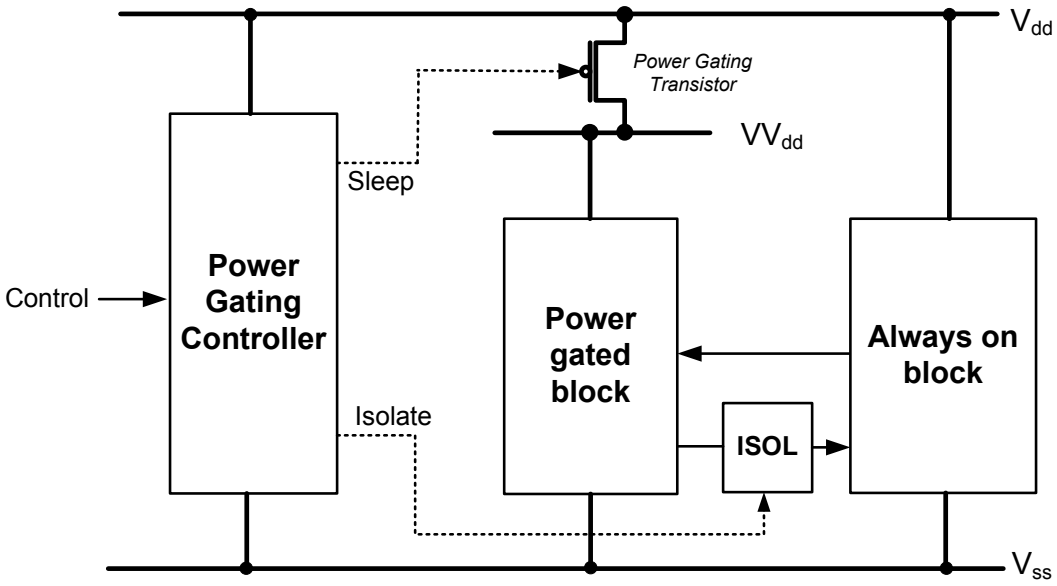
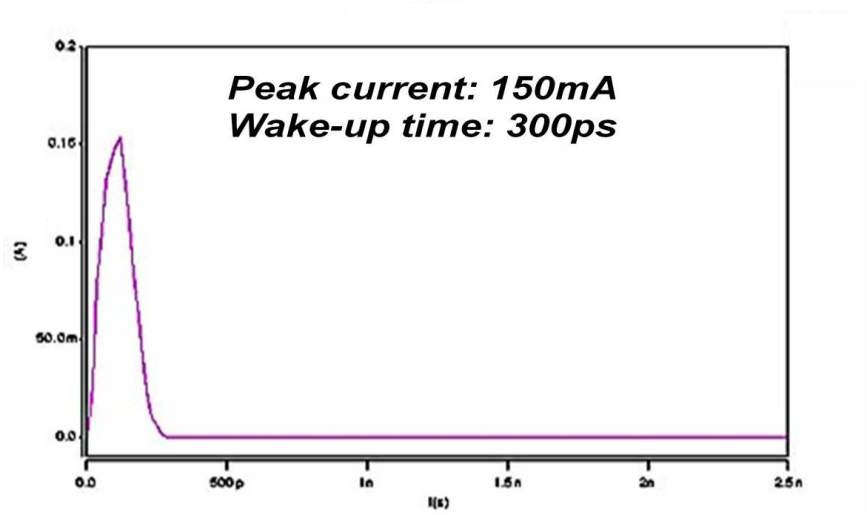


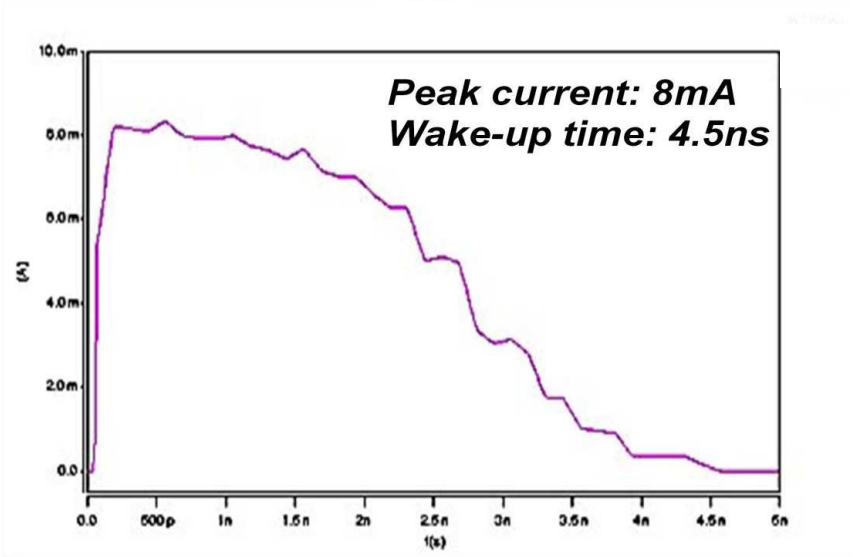
Figure 1.7: Example of coarse grain power gating (based on [3])

for gate propagation delay, Eqn. 1.7, by substituting V_{dd} with $(V_{dd} - V_x)$, where V_x is the voltage drop across the sleep transistor, the propagation delay of the gates in the power domain is increased resulting in a performance degradation. As such, this can be a particularly important design consideration in high performance systems [3, 41]. The effective resistance of the power gating transistors is directly proportional to their combined width and so many power gating transistors in parallel can help to limit IR drop [3]. However, when the power is switched back on, a large effective sleep transistor width can cause a large rush of current during the recharging of the virtual rail and internal capacitances, Fig. 1.8(a). This rush of current creates inductively induced voltage fluctuations in the power distribution networks and is known as *ground bounce* [42]. This can be problematic in neighbouring non-power gated logic as a reduction in the effective supply can cause performance reduction causing functional problems or can even cause register states to become corrupted [3, 43]. Potential solutions to reducing the rush current and ground bounce have been proposed by first switching a small power gating transistor giving a weak trickle current followed by the main power gating transistors [44] or staggering the sleep control signal to each power gating transistor [42, 45]. The effect this latter solution has on in-rush current can be seen in Fig. 1.8(b).

At the interface of power gated logic and non-power gated logic, isolation must be included to clamp the signals from the power gated region to the always-on region when the power is shut down. This is shown as *ISOL* in Fig. 1.7. This is required because the signals float when the power is disconnected and can cause short circuit (*crowbar*) currents in any always-on logic inducing high power consumption or functional problems [3]. Signals that flow in the other direction do not need to be isolated because the logic signals are constantly driven. Isolation is achieved by placing special isolation gates at the outputs of the power gated block to guard the entire fan-out of the logic signal.



(a) Simultaneous switch on



(b) Staggered switch on

Figure 1.8: In-rush current using simultaneous power gate switch on and staggered switch on [45]

There are two main types of isolation gates that are used and are shown in Fig. 1.9. On the left of Fig. 1.9 is an *AND* style 'clamp low' gate which passes the input signal to its output when *nIsolate* is high but clamps the output to a logic 0 when *nIsolate* is low. Alternatively, on the right of Fig. 1.9 is an *OR* style 'clamp high' gate which clamps the output to a logic 1 when *Isolate* is high. The *AND* and *OR* style gates ensure the correct polarity of the power gated circuit's outputs can be maintained depending on whether the output signals are active high or active low respectively. The assertion of the control signals to the power switches and isolation gates requires careful timing to ensure correct functionality of the power gating technique. This control typically comes from a controller state machine [3], labelled as *Power Gating Controller* in Fig. 1.7. Fig. 1.10 shows a typical control sequence issued by a power gating controller when

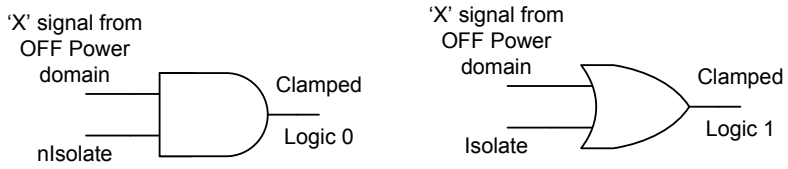


Figure 1.9: Typical clamp low and clamp high isolation gates

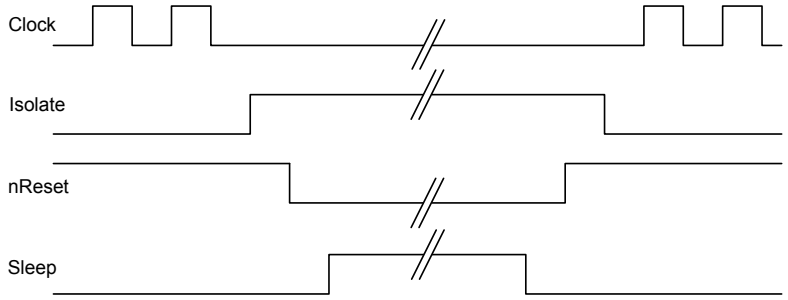


Figure 1.10: Control of signals during power down and power up

entering and exiting the sleep mode of operation [3]. Firstly the clock to the power gated block is stopped. This is followed by the assertion of the isolation signal and ensures the always-on blocks in the digital circuit are protected. This is followed by asserting the reset signal and is done so the registers are brought back into a known state when the power is later restored. Once the power gated block is isolated, finally the power is switched off. This sequence is reversed when the power is restored.

1.4.1.1 Physical Implementation

This section describes a typical physical design flow for power gating and serves as background for all chapters in this thesis but is particularly relevant to Chapter 5.

The implementation of an Application Specific Integrated Circuit (ASIC) with power gating is well supported by industry standard EDA tools by various vendors such as Cadence®, Mentor Graphics® and Synopsys®. Throughout this thesis the Synopsys tool suite will be used. A typical physical design flow to implement power gating is shown in Fig. 1.11. The flow begins with the register transfer level (RTL) of the circuit which is a high level representation of the functionality of the digital circuit often written in a hardware description language (HDL) such as Verilog or VHDL; in this thesis Verilog is assumed. The RTL is simultaneously written in conjunction with a power intent file. The purpose of the power intent file is to define the necessary power domain, power gating transistor and isolation strategy requirements of the power gating technique used in the design. The power intent also defines the power supplies to all parts of the design, for example a power gated domain will be assigned the VV_{dd} and V_{ss} whereas an always-on domain will be assigned V_{dd} and V_{ss}. Two main power intent standards currently exist: Common Power Format (CPF) and Unified Power Format

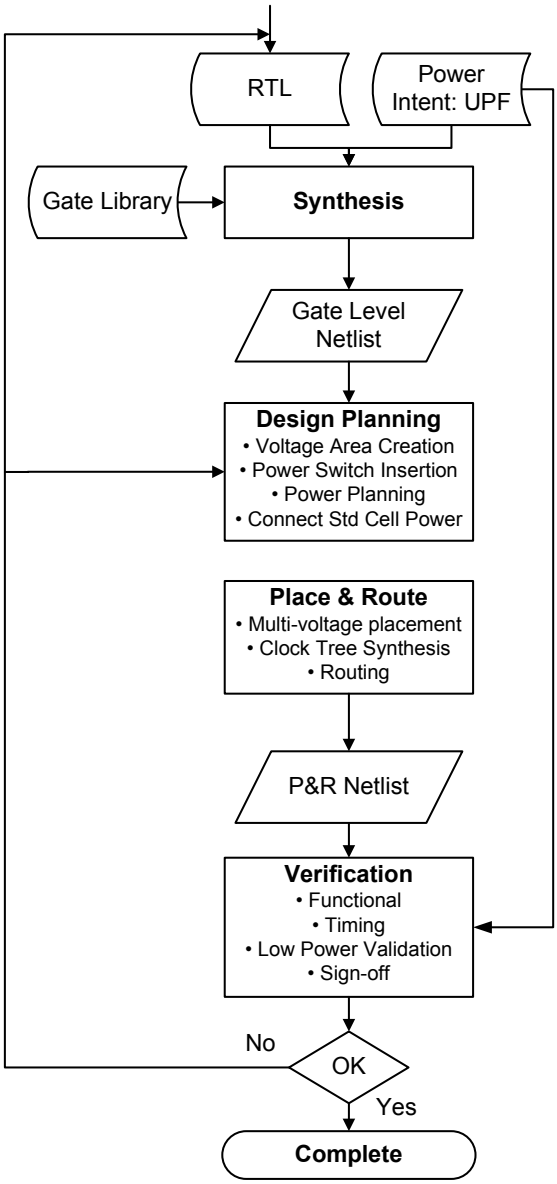


Figure 1.11: Physical design flow for power gating (based on [3])

(UPF). The UPF standard has achieved greater success in comparison to CPF and has recently become an IEEE standard, 1801 [46]. For this reason, the rest of this thesis will assume the implementation of power gating using the IEEE 1801 (UPF) standard only. An example of how power domains, supply voltages and power switches are defined in UPF is given below:

```
create_power_domain PD1 -elements {Top/switched_module}

set_domain_supply_net PD1 \
-primary_power_net VVDD -primary_ground_net VSS
```

```
create_power_switch p_switch -domain PD1 \
-input_supply_port {VDD VDD} -output_supply_port {VVDD VVDD} \
-control_port {sleep Sleep} -on_state {on_state VDD {!sleep}}
```

In this brief example a power domain *PD1* is defined. A power domain can only be created using Verilog modules and in this example *PD1* includes the instantiated Verilog module *switched_module*. The power domain is defined with a virtual supply voltage *VVDD* and ground *VSS*. To derive the virtual supply voltage a power switch is defined called *p_switch* which takes the normal *VDD* and outputs a switched *VVDD*. The switch is controlled by the signal *Sleep* and is active when the control signal is low. As described in Section 1.4.1, it is necessary to include isolation between power domains and an example is shown below where the outputs of the power domain *PD1* are set to be clamped to logic 0 with the always-on V_{dd} and V_{ss}, which is controlled by the active high signal *Isolate*:

```
set_isolation PD1_isolation -domain PD1 \
-isolation_power_net VDD -isolation_ground_net VSS -clamp_value 0

set_isolation_control PD1_isolation -domain PD1 \
-isolation_signal Isolate -isolation_sense high -location parent
```

The synthesis stage (Fig. 1.11) combines the RTL with the UPF to map the design to the desired gate library and outputs a gate level representation of the design. The basic layout of the ASIC is subsequently established in the Design Planning stage. The power domains defined in the UPF are mapped to physical instantiations known as *Voltage Areas* in the layout [3, 47]. These voltage areas ensure that standard cells belonging in each power domain are grouped together in the physical layout so that the correct power supplies can be routed to them without shorts being created between the switched and unswitched supply rails. The location of the voltage area has some impact on the eventual quality of results as it creates a placement bound for a subset of the standard cells in the design [3]. For this reason its location should be mindful of the relationship of the standard cells in the voltage area with other standard cells outside the voltage area. The primary advantage of using a voltage area is the ability to use traditional standard cells and placement techniques as all cells contained within one voltage area share the same power and ground connections [3, 47]. For example, within the voltage area the standard cells in the placement rows all share a single VV_{dd} and V_{ss} connection. Outside of the voltage area, placement rows use V_{dd} and V_{ss} instead. Further details of how and why a voltage area is used in the physical layout are given in Chapter 5. The voltage areas that use power gating are then populated with a series of distributed power gating transistors. An example of how a power gating transistor links the V_{dd} and VV_{dd} supplies in the physical layout is shown in Fig. 1.12. The V_{dd} supply is connected from a higher metal layer to the sleep transistor using Vias. The sleep transistor internally contains

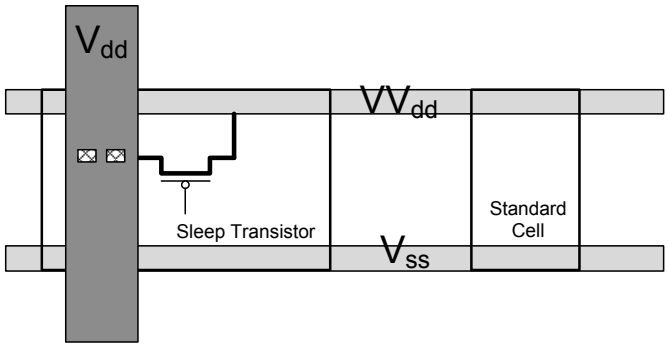


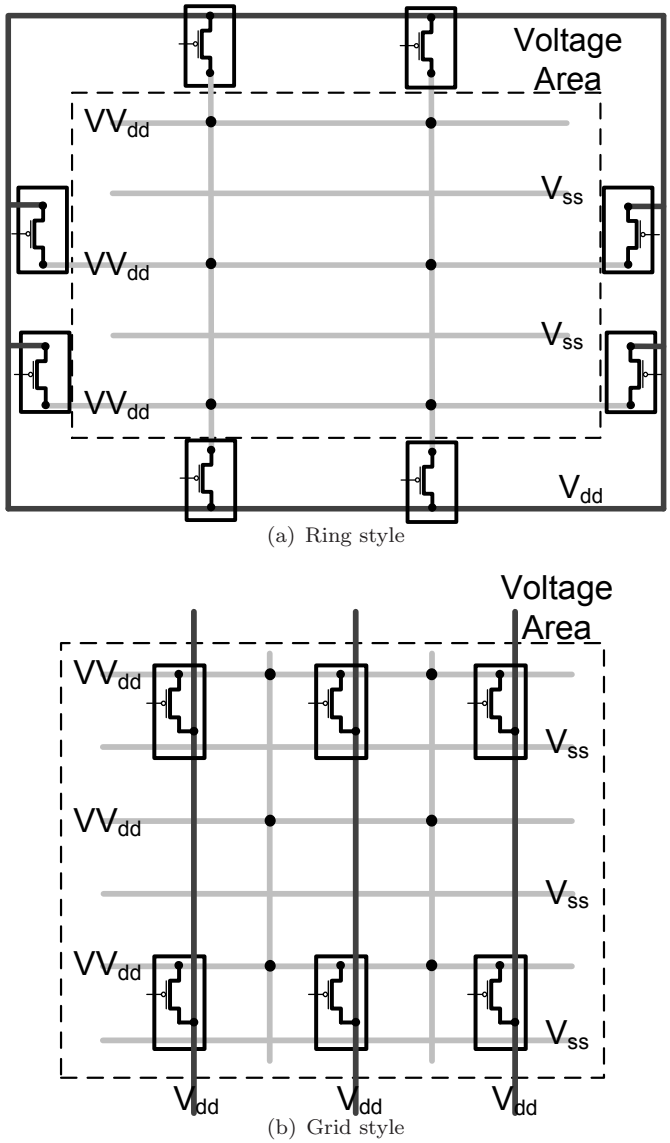
Figure 1.12: Example of sleep transistor connection in physical layout (based on [3])

the PMOS transistor(s) which links between the always-on V_{dd} and the standard cell row's VV_{dd} supply rail. Two main methods of power gate insertion exist, namely ring and grid style [40]. With the ring style placement, Fig. 1.13(a), the power gates are evenly spaced around the edge of the voltage area. In the grid style placement, Fig. 1.13(b), the power gates are staggered throughout the voltage area along the x and y directions in defined increments. A ring style placement results in an increased IR drop in the centre of the voltage area due to limited drive of the power gates and so a grid style helps to reduce this [40]. The need for multiple power rails within the power network complicates the 'power planning' substep in Fig. 1.11, but most EDA tools provide some method of automated power network synthesis which can aid with this.

The placement, clock tree synthesis and routing stages remain relatively unchanged to the user as the EDA tool is aware of the defined voltage areas during all three. Placement optimises the location of all standard cells taking into consideration a cell's assignment to a voltage area, clock tree synthesis creates a balanced clock tree such that all registers are clocked with minimum skew and routing completes all signal routing. The verification stage also remains largely unchanged but the UPF is recalled to match the original power intent with the physical realisation of the design. It is, however, necessary to perform some form of low power validation where the sequencing of the power gating and isolation strategy is checked. This is used to check critical signals are not lost due to powered down logic, power switches and isolation are driven correctly and X values do not propagate out of shut down power domains. Any issues with functionality or the power gating technique noticed in the verification stage must be fixed which may require re-architecting the design or modifying the layout.

1.4.2 Minimum Energy Computation

The subthreshold technique is well suited to energy constrained applications where performance is not a primary concern [37]. Traditionally the supply voltage used in digital



(a) Ring style

(b) Grid style

Figure 1.13: Sleep transistor insertion methods for a voltage area [40]

circuits maintains a safe margin above the threshold voltage of the transistors to guarantee robustness and performance [1]. However, as shown in Eqn. 1.6 dynamic power shares a quadratic relationship with supply voltage and from Eqn. 1.9 and Eqn. 1.1 it can be seen that subthreshold leakage current and total leakage power are reduced with lowering of supply voltage. Therefore, aggressive voltage scaling presents an attractive way to reduce power of a digital circuit. The limit of voltage scaling was theorised by Meindl et al. [48] and predicts an ideal MOSFET can be still fully functional down to a supply voltage of 36mV. At a supply voltage this low, the transistors are operated at a voltage below the threshold voltage. As the gate to source voltage (V_{gs}) drops below the threshold voltage (V_{th}) the drain to source current does not immediately drop to zero, as is implied by typical models for a MOSFET [2], but instead decreases exponentially ($I_{ds}\ \alpha\ exp(V_{gs} - V_{th})$) [36]. The on current ($I_{on}$) produced at these ultralow voltages

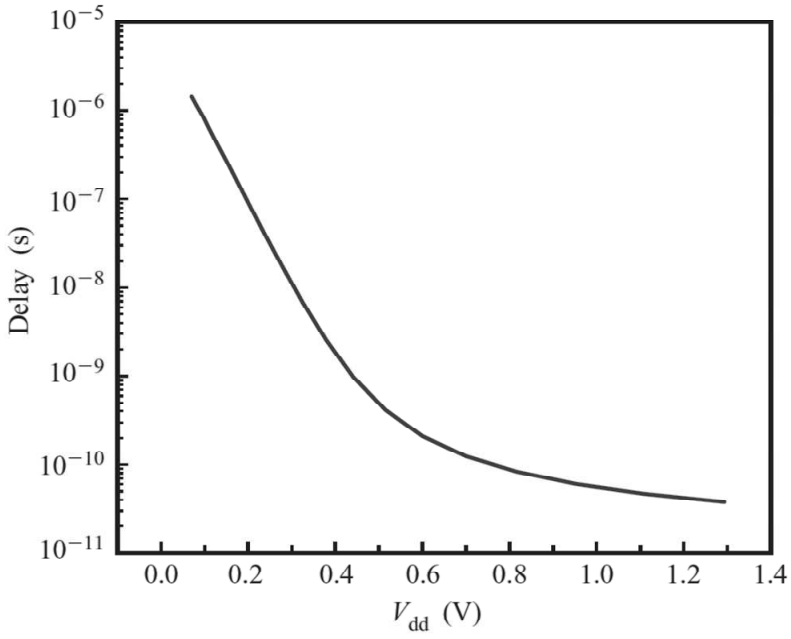


Figure 1.14: Delay of an inverter against V_{dd} (130nm Technology) [37]

therefore changes from the strong inversion current experienced at super threshold voltages and takes the form of a weak inversion sub-V_{th} current [37]. At a supply voltage below the threshold voltage of the transistors then, a non-zero gate voltage can still produce a drain current that is larger than when the gate voltage is zero [36], and using this current, nodal capacitances can be charged and discharged allowing a circuit to continue operation at very low voltages.

In practice, simulation on a 65nm process shows functionality of CMOS logic gates to approximately 100mV [37], which is higher than that predicted by Meindl but nevertheless shows functionality of CMOS to very low voltages. However, as the supply voltage is lowered the propagation delay of the gates within the circuit increases, Eqn. 1.7. As an example, the impact of supply voltage scaling on the delay of an inverter is shown in Fig. 1.14. The sharp rise seen in propagation delay of the logic gates with reduction in supply voltage has a consequential effect on the energy consumption per clock cycle when using ultralow voltages. Although reduction in the supply voltage results in lowering of both dynamic and leakage power, the increase in circuit delay and the leakage energy's dependence on the clock period (Eqn. 1.10), results in leakage energy per clock period exceeding dynamic energy. This can be seen in Fig. 1.15, where the power and energy per operation of a chain of 50 inverters in a 130nm technology is obtained over a range of V_{dd} [37]. The power against supply voltage graph in Fig. 1.15 shows that average total power decreases monotonically with reduction in supply voltage, however the energy against supply voltage graph shows an inflection in the energy per operation due to the rise in leakage energy consumption as a result of increased circuit delay. This inflection point is often referred to as the minimum energy point and corresponds to an

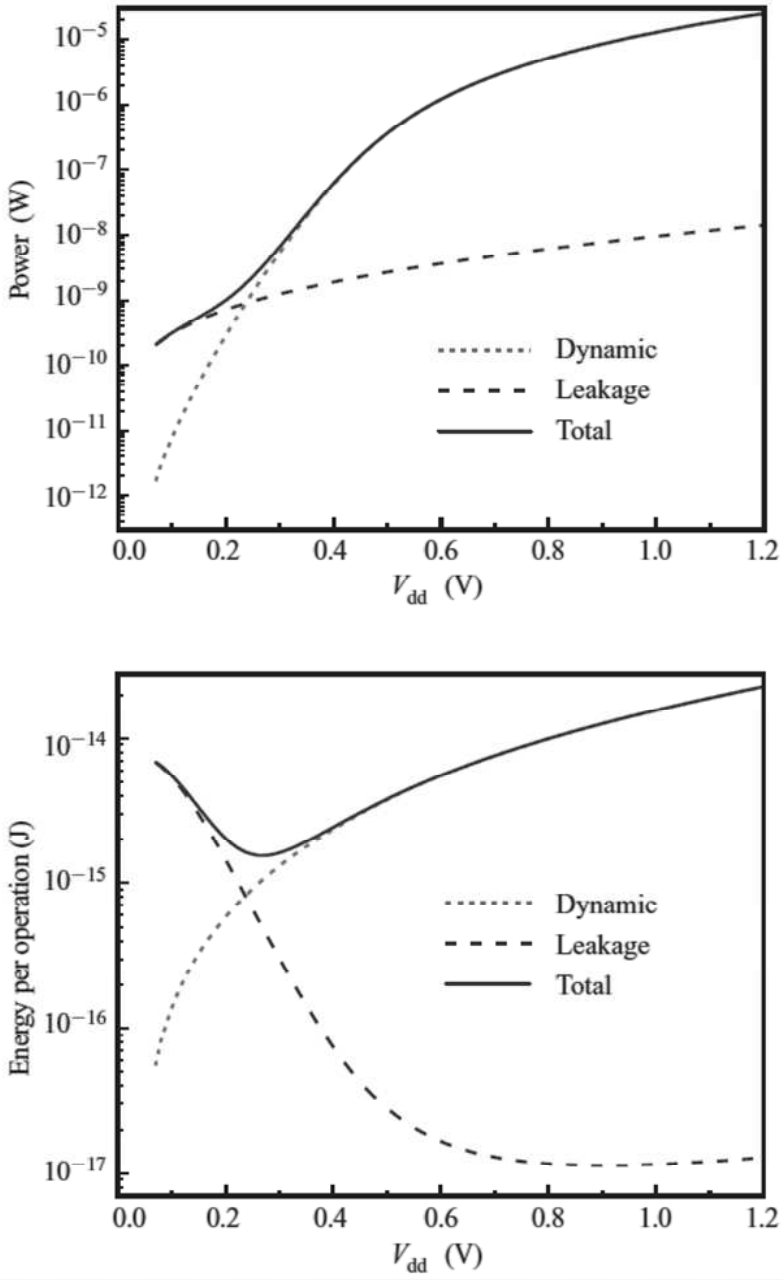


Figure 1.15: Simulation of a 50 stage inverter chain (130nm process): Top - Power as a function of V_{dd}, Bottom - Energy per operation as a function of V_{dd} [37]

equality between the energy consumed to dynamic power and leakage power in the clock period [36, 37, 49]. Increasing the supply voltage from this point would result in dynamic energy dominating per operation and reducing the supply voltage would result in leakage energy dominating. As is found in this inverter chain example, and many other circuit designs [50–53], this minimum energy point occurs below the threshold voltage of the transistors, and is why this technique is commonly referred to as subthreshold operation. Provided performance is not a key design goal, subthreshold operation provides a compelling method to maximise the energy efficiency of a digital circuit.

1.5 Applications

Historically performance has been the key criteria in processors due to the demand for feature rich user experiences in handheld mobile devices such as smart phones and tablet computers whilst power and energy efficiency has been left as a desirable but unessential goal [54–56]. However, there are some current and emerging applications where performance is not the ultimate goal and instead power and energy are the primary constraint.

Wireless sensor networks (WSN) is one such application. WSNs consist of small 'sensor nodes' with sensing, computation and communication capabilities and are used for applications like habitat monitoring [57], environment monitoring [58] and health monitoring [59]. Hundreds if not thousands of wireless sensor nodes are placed out in the field and are left to operate without maintenance to collect, process and transmit data for months to years. The type of sensing these sensor nodes do is not demanding on the processor and so it is common to see processors used with performances in the range of kHz-MHz. However, to maximise device lifetime average power consumption in the order of 10s-100s of μWs is desired. The Zebranet application for example uses a Texas Instruments MSP430 [60] and utilises its 32kHz mode of operation most of the time for device control and consumes approximately 300μW [61]. The Free2Move device is another example application that uses a PIC16F87 which toggles between 32kHz and 1MHz depending on the current operation consuming between 18μW and 152μW [62]. Whilst general purpose microprocessors are used in WSNs, a number of ASIC processors have also been developed and operate at low performance levels. For example, the event processor [63] operates at 100kHz to limit power consumption to 100μW and the ASIC microprocessor proposed by Warneke et al. is operated between 10kHz-100kHz [64].

Bio-medical applications are another area where high performance is unnecessary due to the rate at which data needs to be processed, but energy efficiency is key because of the desire for untethered operation. Heart and brain signals are in the order of Hz and are sampled and processed continuously using sensors to monitor a patient's health [65, 66]. Using processors with MHz performance would be wasteful of power due to unnecessary switching and would unnecessarily limit the device lifetime of the portable sensors [67]. An ASIC for wireless monitoring of an Electrocardiography (ECG) signal that operates at 32kHz has been developed [65]. However, because the sensor is battery operated they require power consumption in the order of μWs to maintain a reasonable device lifetime. A similar power requirement is imposed on an Electroencephalography (EEG) ASIC which uses a 32kHz digital circuit to control their device [66]. The 'Internet of Things' is an emerging application that is regarded to be the next big revolution in science and technology [68]. The main vision of the internet of things is being able to monitor, sense and track items and phenomena in the environment as a means to improve quality of life [69]. The internet of things is considered to be very similar to

wireless sensor networks requiring similar low processor performance requirements but one of the key problems is constrained resources demanding low power [70].

From this brief overview it is apparent that existing and emerging applications have low to moderate frequency (10-100s kHz) of operation requirements where power and energy is constrained (10-100s μW). To maintain a sufficient device lifetime, power consumption must be kept to a minimum while the processor is doing useful work and reducing leakage is key to this. Therefore, continuing advances in leakage power minimisation during the active mode is essential to achieving this.

1.6 Thesis Organisation

Chapter 2 - Literature Review

This chapter presents a coherent overview of widely used and recently reported techniques for reducing leakage power within an integrated circuit. The chapter also outlines the objectives of this thesis.

Chapter 3 - Active Mode Sub-Clock Power Gating

This chapter presents a power gating technique, called sub-clock power gating (SCPG), that can be used during the active mode to reduce the leakage power dissipation of an embedded processor. The motivation for the proposed technique arises from the increased combinational logic idle time that exists in the clock period from the use of low clock frequencies at a fixed supply voltage. The technique power gates combinational logic within the clock period and allows a digital circuit to operate with lower average power at a given performance point or a higher performance for a given average power. The technique is fully compatible with commercial EDA tools and power gating design flows and the steps required to augment a design with sub-clock power gating are given. Simulation results of the proposed technique from post layout netlists of a 16-bit parallel multiplier, an ARM Cortex-M0 and a recently proposed processor for wireless sensor networks are presented and a comparative analysis with the subthreshold technique is also given.

Chapter 4 - Symmetric Virtual Rail Clamping for Sub-clock Power Gating

This chapter presents a new power gating technique with lower wake-up energy cost than conventional shut down power gating and investigates its utility in the sub-clock power gating technique proposed in Chapter 3. The proposed technique reduces the virtual supply by two V_{th} rather than shutting down completely as is the case in conventional power gating and is achieved with a pair of NMOS and PMOS transistors at the head and foot of the power gated logic for symmetric virtual rail clamping of the power and ground supplies. The technique is combined with sub-clock power gating and implemented on an ARM Cortex-M0 using commercial EDA tools for fabrication in a 65nm technology. Experimental results based on the fabricated silicon show that better energy efficiency is achievable with symmetric virtual rail clamping enabling a greater range of frequencies to be used for the sub-clock power gating technique proposed in Chapter 3.

Chapter 5 - dRail: A Physical Layout Technique for Power Gating

This chapter provides deeper understanding of why a voltage area is used in the physical layout of power gating and investigates how removing the placement constraint enforced by a voltage area affects energy efficiency. To enable this investigation a new physical layout technique called dRail is proposed which allows both power and non-power gated cells to be placed adjacently. The proposed technique is achieved with three changes to the layout: modified standard cells, dual power supply rail routing and custom power hook-up per standard cell. dRail is fully integrated into a conventional physical design flow with commercial EDA tools. The use of an unconstrained placement is compared against voltage area layout in the sub-clock power gated Cortex-M0 used in Chapter 4 and a Cortex-A5. It is shown from experimental results that better energy efficiency is attainable through reduction of standard cell area and signal routing length.

Chapter 6 - Conclusions and Future Work

This chapter summarises the contributions discussed in this thesis and places them in context with state-of-the-art research. A number of areas for future work that would improve upon and extend the techniques proposed in this thesis are also highlighted.

1.7 Contributions

The contributions of the research work presented in this thesis have been published as follows:

Journal Publications

1. **Mistry, J. N.**, Myers, J., Al-Hashimi, B. M., Flynn, D., Biggs, J., *Active Mode Sub-Clock Power-Gating*, IEEE Transactions on V.L.S.I. (under review)

Conference Publications

2. **Mistry, J. N.**, Al-Hashimi, B. M., Flynn, D., Hill, S., *Sub-Clock Power-Gating Technique for Leakage Power Reduction During Active Mode*, Design, Automation and Test in Europe (D.A.T.E.), 14^{th} to 18^{th} March, 2011, Grenoble, France

3. **Mistry, J. N.**, Myers, J., *An ARM Cortex-M0 for Energy Harvesting Systems: A Novel Application of UPF with Synopsys' Galaxy Platform*, Synopsys Users Group (SNUG), 26^{th} to 28^{th} March, 2012, Silicon Valley, CA.[1]
Winner of Best Technical Paper Award

4. **Mistry, J. N.**, Biggs, J., Myers, J., Al-Hashimi, B. M., Flynn, D., *dRail: A Novel Physical Layout Methodology for Power Gated Circuits*, Power and Timing Modeling, Optimization and Simulation (PATMOS), 4^{th} to 6^{th} September, 2012, Newcastle upon Tyne, UK

[1] http://www.synopsys.com/community/snug/pages/ProceedingsAwards.aspx?loc=Silicon%20Valley&locy=2012

Chapter 2

Literature Survey

Minimising power consumption through the use of low power design techniques has been an active research area for nearly two decades [10, 71, 72]. While dynamic power has dominated the power consumption in the past, as feature sizes have shrunk for increased device integration, lower cost and higher performance, leakage power has risen and poses a major obstacle for energy efficiency in digital circuits [32]. The purpose of this chapter is to provide an account of state-of-the-art techniques for reducing leakage power in digital circuits and identify opportunities for research. The increasing dominance of leakage power has led to a number of different approaches to minimising it ranging from the device level to the architectural level which are either fixed or controllable through system or circuit processes. In general, the majority of techniques exploit opportunities for reducing leakage power reduction within two areas: design time and runtime. The runtime techniques can be further split into two categories, namely standby mode and active mode. These different opportunities for leakage power minimisation are shown in Fig. 2.1.

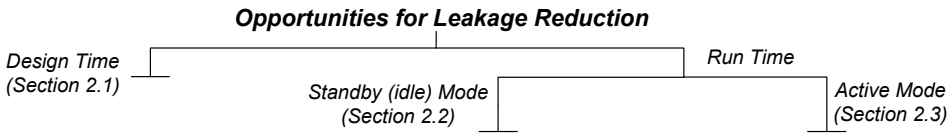


Figure 2.1: Opportunities for leakage power reduction

Section 2.1 of this chapter focusses on techniques that are used during design time to minimise leakage at the device level. Section 2.2 discusses techniques that are used during runtime but focus on leakage reduction during the standby mode. Section 2.3 describes an emerging area of leakage power control which employs techniques to capitalise on leakage reduction at runtime but during the active mode. Physical layout also plays an important role for the implementation of many of the reported techniques for reducing leakage power dissipation and is discussed in Section 2.4. The aims and objectives of this thesis are outlined in Section 2.5 and concluding remarks are given in Section 2.6.

2.1 Design Time: Transistor and Gate Level Techniques

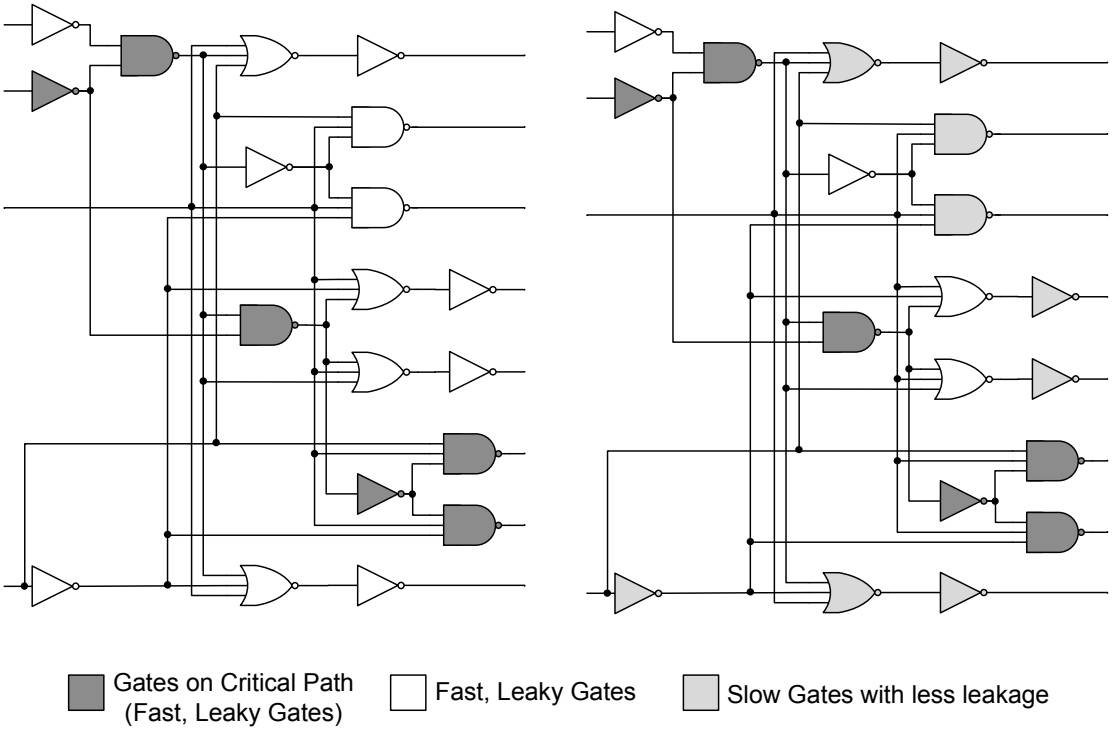


Figure 2.2: Path balancing during design time (based on [73])

One of the primary reasons for increasing leakage current in deep sub-micron technologies is the need to lower threshold voltages in transistors to maintain performance at the newly scaled supply voltages [6, 29, 32]. Although performance is increased from the use of lower threshold voltage transistors, the overall performance of a digital circuit is limited by its critical path [4]. Therefore every other path that is not the critical path may exhibit positive timing slack - defined as the difference between the time the path must complete evaluating and the time it actually finishes evaluating [2]. Design time techniques exploit this positive delay slack in non-critical paths by changing attributes of the gates or transistors to trade performance for leakage reduction. These techniques are static and so, once the decision to incorporate them into the design is made, there is no way to dynamically change them while the circuit is operating. While many algorithms exist for optimising the use of delay slack to minimise leakage using one or a combination of the techniques discussed in this section [22, 73–78], the procedure is most easily summarised as follows. The digital circuit is initially synthesised using fast, high leakage gates until the required timing constraint is met, left of Fig. 2.2. Gates that are located on non-critical paths are then identified and swapped with gates that are slower and dissipate less leakage power whilst simultaneously ensuring the length of the path does not exceed the critical path. This is repeated either for a maximum number of iterations or until the number of slower and less leaky gates is maximised, right of Fig. 2.2.

Multi-threshold voltage logic is perhaps the most common method of reducing leakage in gates that are off critical paths and many examples of multi-threshold leakage optimisation exist [73–75, 77, 79, 80]. As can be seen from Chapter 1, Eqn. 1.9, subthreshold leakage depends exponentially on threshold voltage and so a higher threshold voltage results in lower subthreshold leakage current. However, as can be seen from Chapter 1, Eqn. 1.7, a higher threshold voltage also results in an increase in propagation delay. For example, it is shown by T. Luo et al. that a low-V_{th} transistor's leakage current in a 65nm gate library can be 17.3x greater than that of a high-V_{th} transistor's, but a high-V_{th} transistor is 30% slower [77]. The modulation of the threshold voltage is done by varying the transistor's doping profile [2, 32], therefore during the manufacturing process two masks are required, one for low threshold voltage and one for high threshold voltages increasing cost. Nevertheless, multi-threshold logic is very popular and L. Wei et al. show on the ISCAS benchmark circuits that savings up to 80% are achievable in standby leakage power while active power can be reduced by 50% and 20% for low and high switching activities respectively [73].

Another method of reducing leakage power in logic gates is by modulating the channel length [74, 78, 81]. This method has been proposed as an alternate to dual threshold voltage to reduce the cost associated with manufacturing. Whereas two masks are required for the implementation of different doping profiles in a dual threshold voltage process, channel length variation is easier to manufacture [82]. An increase in channel length increases the threshold voltage of the device but reduces the drive current and increases the input capacitance. This consequently reduces performance and marginally increases dynamic power dissipation respectively [78]. It is reported that although drive strength is reduced by 10% for a 10% increase in channel length, leakage is reduced by approximately 3x per device [11]. In a study done by Gupta et al. [82] where multiple channel lengths were used on the ISCAS benchmark circuits, they were able to show up to 33% leakage reduction with a maximum of 3% increase in dynamic power using a 130nm process. In modern gate libraries channel length modulation is employed simultaneously with multiple threshold voltages and it is shown that an inverter with both high threshold voltage and extended channel length exhibits 86% leakage power reduction compared to 72% with just a high threshold voltage and 62% with just a longer channel length in a 40nm process [74].

Modulation of the gate oxide thickness has also been proposed as a method to alter the threshold voltage of the transistors to achieve lower leakage current [78, 83, 84]. A thicker oxide relates to a higher threshold voltage and also has the added advantage of a reduced gate capacitance which lowers dynamic power [78]. A greater oxide thickness is particularly useful in sub-90nm process technologies [84, 85] where gate leakage is as dominant as subthreshold leakage due to the very thin oxide thicknesses [32]. Multiple oxide thicknesses are readily available as part of commercial gate libraries and are normally combined with higher threshold voltages, and has recently been demonstrated in

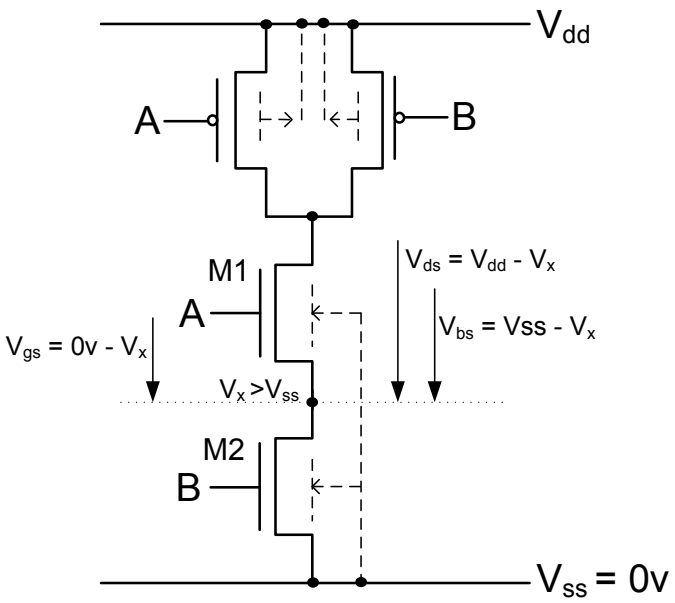


Figure 2.3: Stack Effect (based on [88])

a custom microcontroller implementation [86] and SRAM which shows 50% reduction in standby leakage in a 45nm process [87] .

Aside from modifications that can be made to the process variables (channel doping, channel length and oxide thickness), a simpler solution to reducing leakage power has been proposed by using forced transistor stacking [88, 89]. The technique capitalises on the *stack effect* that occurs when series MOSFETs are simultaneous switched off. This can be demonstrated with a two input NAND gate as shown in Fig. 2.3. Consider that both A and B are logic 0 and so M1 and M2 are both off and the output is fully charged. The intermediate voltage V_x between M1 and M2 has a small positive value due to the capacitance associated with the node and the subthreshold leakage currents that flow through the off transistors [2, 90]. Assuming the body of the transistors are grounded, this means the gate-to-source voltage of M1 becomes negative, the body-to-source voltage becomes negative and the drain-to-source potential of M1 is lowered. As can be seen from Eqn. 1.9, the reduction in V_{ds} and V_{gs} both reduce subthreshold leakage current, and as explained in Chapter 1, Section 1.1.2, the body effect causes a negative V_{bs} to increase the threshold voltage resulting in a further reduction in subthreshold leakage current. To capitalise on this effect, it has been proposed to swap a non-stacked transistor of width W with two stacked transistors of width $\frac{1}{2}$W in a logic gate to reduce leakage [89, 90]. By maintaining an equivalent iso-input load there are no adverse affects to the gate fan-in but the inclusion of the stack reduces the gate's performance [90]. Forced stacking has recently been used by Hanson et al. in a custom processor where they report 2x leakage reduction from using logic gates with forced transistor stacks [91].

2.2 Runtime: Standby Mode Leakage Techniques

The techniques discussed in the previous section are decided upon at design time and once implemented remain static within the digital circuit. Runtime techniques on the other hand allow a particular technique to be incorporated in the digital circuit at the design stage but permits activation and deactivation of it dynamically during runtime. The most common use for these runtime techniques is leakage control during the standby mode, which refers to the periods of idle time that occur between execution due to varying workload on a processor [92]. The techniques described in this section exploit these periods of idleness to place them into the low leakage mode to minimise power dissipation. These low leakage modes are generally managed by a power controller and the decision to transition to the low leakage state is often given at the system level. For example, in commercial operating systems such as Microsoft Windows the low leakage modes are controlled through the Advanced Configuration and Power Interface (ACPI) standard [92]. This standard defines four processor states ranging from C0-C3 each having increasing power savings. For example, C0 corresponds to the active state whereas C3 is the sleep state when the processor may be power gated (Chapter 1, Section 1.4.1) and can be controlled by the 'Sleep' function on a laptop. Dynamic power management (DPM) presents a much more fine grained level of control and enables power management to be embedded, for example, within the Linux kernel and triggers the low power states more frequently [93]. DPM makes the decision to employ the low leakage states dependent on a governing algorithm. These algorithms may be time-based where the device is put to sleep after a certain amount of idle time, predictive where the upcoming duration of idle period can be predicted ahead of time or stochastic where the idle period arrival and power state changes can be predicted using statistics [93]. More recent work has shown the application of online learning which permits better adaptation to a processor's workload to achieve better scheduling of the low leakage states [93].

2.2.1 Power Gating

Power gating is considered to be the most effective and practical technique to reduce the leakage power of idle circuitry and the fundamentals of the technique were introduced in Chapter 1, Section 1.4.1. Applications of power gating have an extremely diverse range, from low performance applications such as wireless sensor nodes [91] to high performance processors [41, 43]. The popularity and practicality of power gating has prompted a number of variations of the original power gating technique and this section covers a selection of the key proposed techniques.

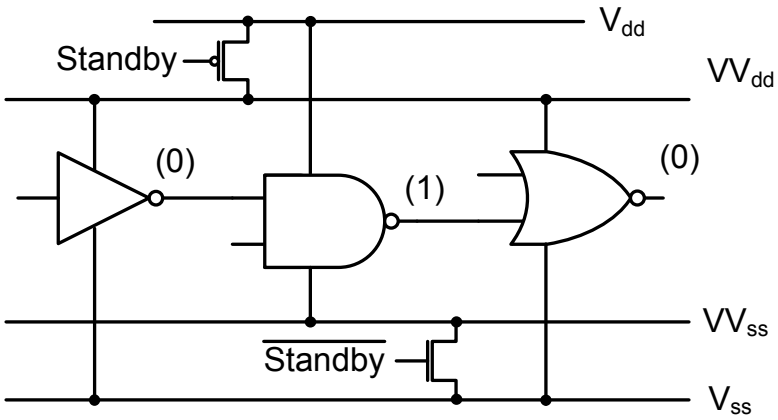


Figure 2.4: Example of zig-zag power gating [95]

2.2.1.1 Header Vs Footer

In Chapter 1, Section 1.4.1 power gating was described with PMOS header transistors to switch the V_{dd} power supply, but the use of an NMOS transistor to switch the V_{ss} power supply is equally practical and has been demonstrated in many power gating implementations [41, 91, 94]. Furthermore, circuits have also used both PMOS and NMOS sleep transistors but two power switches results in increased area and increased IR voltage drop to the power gated logic degrading performance which can be unacceptable in some designs [3]. A study has been carried out to show that NMOS transistors can provide a higher drive, hence lower IR drop, than a PMOS transistor of an equivalent size. Additionally, when sized for equivalent drive strength, a PMOS transistor exhibits higher leakage than an NMOS transistor, for example 2.67x more leakage in an 90nm library [3]. The disadvantage of using an NMOS sleep transistor is that a switched ground supply becomes more sensitive to ground noise [40]. It additionally complicates output clamping to logic 0 due to the loss of ground and level shifting because a shared ground is needed between power domains [3]. There is currently no consensus on whether a PMOS or NMOS transistor should be used for power gating and research shows that both are still equally used and may often be governed by availability in the technology library.

Zig-Zag power gating extends the choice of sleep transistor by using PMOS transistors on part of the design and NMOS transistor on another part of the design [95, 96]. The method for using zig-zag is shown in Fig. 2.4. The purpose behind this methodology is that when a virtual V_{dd} is switched, all nodes collapse to ground and when a virtual V_{ss} is switched all nodes charge to V_{dd}. By applying a known input vector to the power gated logic before power down and then selectively choosing gates that are at a logic 1 to be switched by an NMOS transistor and vice versa for gates that are switched by a PMOS transistor, the gate outputs would already be at their correct value when power is restored. This reduces the amount of charging/discharging that occurs in the circuit when coming out of a power gated state, saving both energy and time. Simulation of a

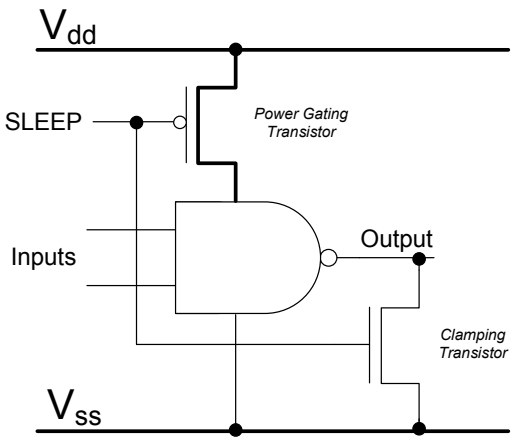


Figure 2.5: Example of fine grain power gating with an *NAND* gate [40]

c7552 ISCAS benchmark circuit, which consists of 3512 gates, shows that in comparison to simply using an NMOS sleep transistor, zig-zag power gating can reduce wake-up delay by 81% [95]. Zig-zag power gating has also been used in the peripheral word-line driver circuits of caches to reduce their idle leakage power whilst maintaining reasonable wake-up delay for cache accesses and it is shown that leakage power can be reduced by up to 100x using a 65nm process [97].

The definition of power gating has so far focussed on a *coarse grain* approach where a sleep transistor is inserted between the power rail and a group of logic gates. However, there is an alternative approach referred to as *fine-grain* power gating [40]. In fine grain power gating the power switch is included as part of each logic gate as demonstrated on a *NAND* gate in Fig. 2.5. The *SLEEP* signal is used to control the power to the *NAND* gate through the PMOS power gating transistor. The gate is active when *SLEEP* is logic 0 and is powered off when *SLEEP* is logic 1. Just as is necessary in coarse-grain power gating, isolation is placed on the output. In the *NAND* gate example in Fig. 2.5, this is achieved with a clamping NMOS transistor on the output node to ensure the output signal of the gate is held at a valid logic 0 level when the logic gate is powered down. If an NMOS sleep transistor was used then a PMOS clamping transistor would be used [44]. The fine grained power gating approach offers lower complexity during implementation as the power gating is confined within each standard cell and so a standard physical design flow can be used with very little additional input such as an Unified Power Format file [40], see Chapter 1, Section 1.4.1.1 for more details. The disadvantage of fine grain power gating, however, is the large increase in standard cell size due to the addition of a power gating transistor and clamping transistor in each cell. Area is reported to increase by between 2x-4x the size of the original cell when using the fine grain approach and this high increase in area has proven to be too high for the reduction in design effort [3]. Additionally, significant buffering is required for the sleep signal to every logic gate. The fine-grain power gated cells also become more sensitive to PVT variation because the sleep transistor is subject to PVT variation. This results

in added IR drop variation from cell to cell and hence varying performance degradation across the design [40]. Coarse grain consequently remains the preferred option for the utilisation of power gating [3].

2.2.1.2 Power Gating Alternatives

Power gating traditionally uses a high-V_{th} transistor to power gate the logic [39]. However, if the supply voltage is to be scaled down to a level lower than the threshold voltage of a high-V_{th} transistor then the transistor would never be fully switched on resulting in an increased voltage drop across the transistor and degraded circuit performance during the active mode. Alternatively a low-V_{th} transistor could be used as the power gating transistor but this comes at a cost. The subthreshold leakage of the transistor is higher than that of a high-V_{th} transistor leading to standby leakage in the order of mA for a million gate VLSI [98]. Super cut-off CMOS (SCCMOS) proposes to overcome this problem by overdriving the gate of the power gating transistor during standby to force it 'more off'. The V_{gs} term in Eqn. 1.9 is lowered resulting in a reduced sub-threshold leakage current through the power gating transistor. This subsequently results in a lower standby leakage current of the power gated block. It is suggested that an overdrive of ΔV equal to the difference between the threshold voltages of a high-V_{th} and low-V_{th} transistor can sustain the same sub-threshold leakage current as using a high-V_{th} transistor. Results show that standby leakage is equivalent to using a high-V_{th} transistor with no impact on active performance [98]. A recent study compares SCCMOS with traditional power gating to observe its effectiveness at normal supply voltages [99]. It is shown that due to the inclusion of an overdrive voltage generator, SCCMOS can improve the energy savings but is best employed when the power gated block is large and standby times are long. In comparison with regular high-V_{th} power gating, SCCMOS can show approximately 33% improvement in standby leakage when implemented on a block size of 6000 gates because fewer power gates are needed to meet a required IR drop [99]. In modern gate libraries the use of a high gate voltage exacerbates the gate leakage current of the transistors and gate-induced-drain leakage; leakage current caused by a large negative bias on the gate. Consequently, it has recently been observed that there is a growing necessity for finer control of the gate voltage to find an optimal point where gate leakage current, gate-induced-drain leakage and sub-threshold leakage are minimised when using SCCMOS in current gate libraries [100].

Since the power gated block in coarse grain power gating is completely disconnected from the power supply, the state of any registers is lost, which can be problematic when the circuit is woken and required to continue execution. To avoid loss of state in the circuit three main solutions have been proposed referred to as 'state retention power gating'. The first solution uses software to copy the state of the registers into the main memory during power down and then restores the state upon power up [101]. This process is

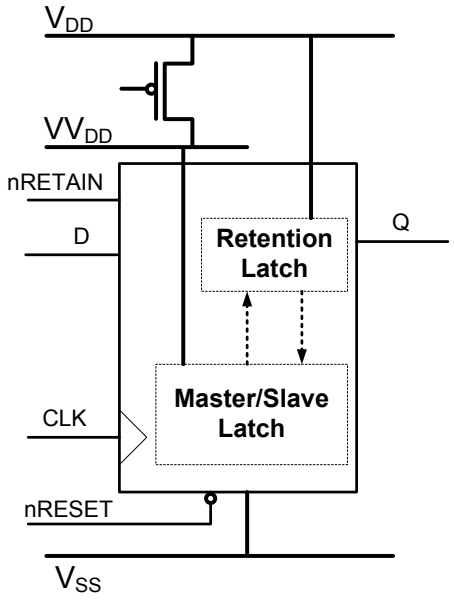


Figure 2.6: Example of retention register (based on [3])

equally useful if the processor contains cache that needs to be maintained when powered down. An alternative method is to use the in-built scan chains to shift the state of the circuit into memory. Since many digital circuits incorporate scan registers for testing, this method reuses the existing hardware and so avoids additional area but at the cost of increased design effort [3]. The final method for saving the state of a power gated circuit is by using retention registers in place of normal registers [102]. Retention registers use a low leakage 'balloon' latch constructed from high-V_{th} transistors and are used to copy the stored value of a register before power down and remains always-on whilst the main register is shut off. An example state retention register is shown in Fig. 2.6. Before the power is shut down, the active low *nRETAIN* signal is asserted to copy the data from the main register into the balloon retention latch. When the power is restored the signal is deasserted and the state is copied back. Retention registers reduce the timing impact of saving state during power down when compared with copying out to memory to single cycle, but increase the register's area by between 20%-50% [3].

Due to the large area overhead of using state retention registers a number of alternate solutions have been proposed for maintaining the state of a power gated circuit when powered down. The technique of virtual rail clamping is one such example [104]. The internal power gated logic connects to switched VV_{dd} and VV_{ss} rails. However, the VV_{dd}/VV_{ss} and the V_{dd}/V_{ss} supply rails are linked together through PMOS/NMOS power gating transistors, as normal, but also forward biased DP/DN diodes. The resulting behaviour of the circuit is as follows [104]. When the *Sleep* signal is logic 0 the circuit is in normal operation and the virtual supply rails are fully charged. However, when the *Sleep* signal is asserted the VV_{dd}/VV_{ss} supply rails drop/rise but are clamped by the built-in potential of the diodes DP/DN. Since the supply is not completely disconnected this method allows the data stored in the latch to be retained during sleep mode.

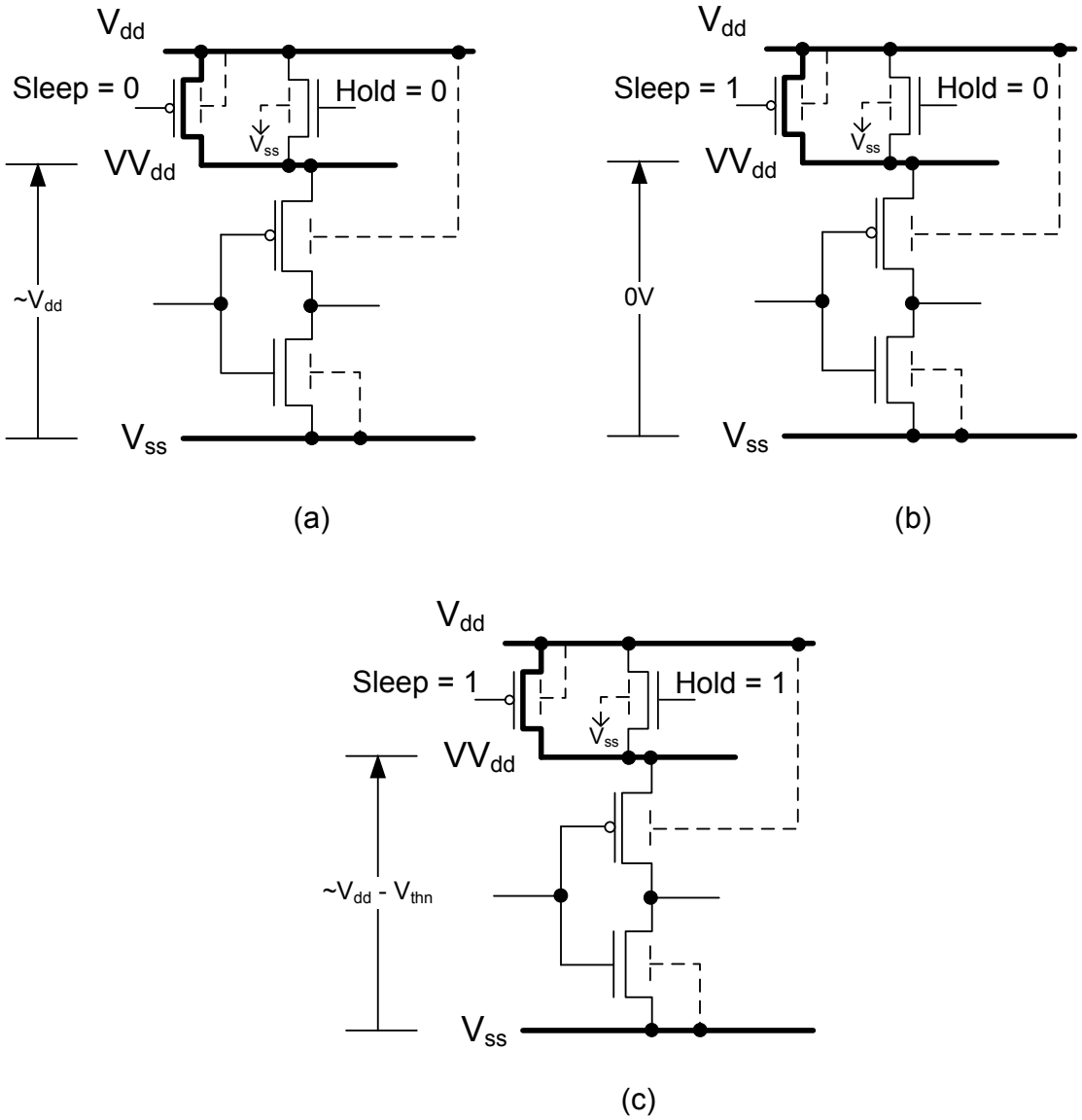


Figure 2.7: Virtual Rail Clamping using a MOSFET [103] (a) RUN/IDLE mode for normal operation, (b) COLD mode for sleep with full shut down and (c) PARK mode for sleep with state retention

Experimental results of the proposed technique showed up to 98% reduction in standby current whilst fully retaining data in a multiply-accumulate macro block fabricated in a 0.25μm process at a supply voltage of 1.2V. The concept of virtual rail clamping [104] has more recently been extended to swap the clamping diode with a single MOSFET transistor due to the reduced supply voltage of modern process technologies [103]. This is shown in Fig. 2.7. The advantage of this is that the circuit enables a sleep mode of operation in addition to the active and clamped modes found in virtual rail clamping. The authors in [103] call the first mode RUN/IDLE and is the normal mode of operation, Fig. 2.7(a). In this mode the power gate is on and the clamping MOSFET is off resulting in a potential difference close to V_{dd} between VV_{dd} and V_{ss}. The second mode of operation is called COLD, Fig. 2.7(b), and is when the power gate and MOSFET

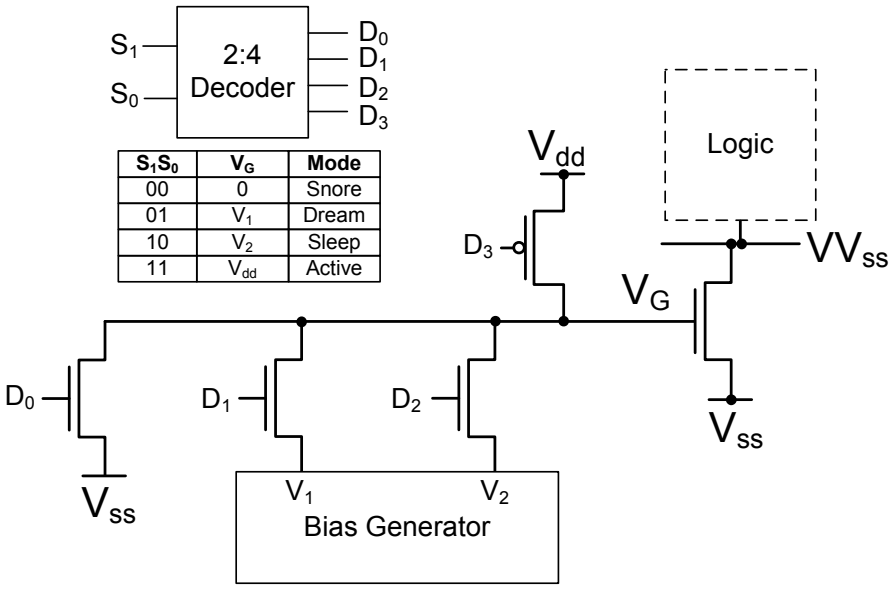


Figure 2.8: Multiple sleep mode power gating using a bias generator [105]

are both disabled resulting in a potential across the logic of 0V. The final mode, Fig. 2.7(c) is the data retention mode called PARK where the power gating transistor is disabled but the parallel NMOS is enabled resulting in a potential equivalent to $V_{dd} - V_{thn}$, where V_{thn} is the threshold voltage of the NMOS transistor. Experimental results from a 130nm test chip shows that the COLD mode of operation, i.e. traditional shut down power gating achieved up to 43x reduction in leakage power whilst the PARK mode of operation achieved up to 2.68x reduction in leakage power on their device under test with full data retention.

Rather than clamping the virtual supply rail for state retention a technique has been proposed that uses a bias generator to drive the gate of the sleep transistor with a varying gate voltage. It is shown that a gate voltage can be chosen such that the power gating transistor is not fully off resulting in the virtual rail being only partially collapsed [105]. The technique is shown in Fig. 2.8. The authors use an NMOS footer transistor and have four selectable gate voltages: V_{dd}, V_1, V_2 and 0V corresponding to active, sleep, dream and snore states where the VV_{ss} rail is progressively more charged for greater leakage savings. The voltages V1 and V2 are configured using bias generators and can be tweaked according to the application requirements. The current mode of operation is selected using a two bit select signal. It is shown that due to the virtual rail being less charged in the sleep and dream states, wake-up time and energy is reduced. Furthermore, the sleep and dream states can be configured such that the voltage maintains the state of the power gated registers. Experiments are carried out on a 64-bit Alpha processor and results show 17% greater leakage saving using intermediate strength power gating compared to traditional power gating as the shorter wake-up delays of the different levels of power gating permits entering the low leakage states more frequently. This technique has recently been improved upon by swapping the bias generators with variable sized

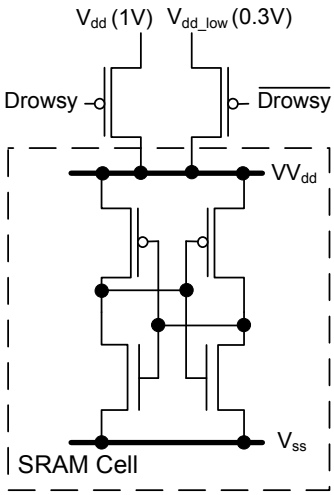


Figure 2.9: Drowsy power gating of cache line [108]

power gating transistors and is shown to be more tolerant of process variation whilst also enabling greater than 2 intermediate modes of power gating [106].

A much simpler method of achieving state retention has been demonstrated for reducing the leakage power in caches whilst still retaining their state. The concept is shown in Fig. 2.9. The method, known as drowsy power gating, switches between two independent power supplies depending on the current state of operation [107–109]. When a cache line is not being accessed the *Drowsy* signal is set and the supply voltage of the SRAM cells are switched to 0.3V to reduce the cache line's leakage power by 77-91%. When the cache line needs to be accessed the *Drowsy* signal is deasserted and the supply is restored to nominal voltage [108]. Although the technique has been applied to caches the technique of switching between two supplies is equally applicable to logic circuitry and a similar approach is presented in [94]. In the proposed technique, the logic supply is completely removed from the combinational logic during standby and the state holding elements switch to the second lower supply voltage to retain the state.

2.2.2 Natural Transistor Stacks

As was described in Section 2.1 stacks of off transistors exhibit lower leakage current due to the stack effect. Rather than statically using gates with forced stacks as is done during design time [90], this phenomenon has been taken advantage of in the idle mode by using input vectors to maximise the number of off transistors in natural gate stacks. For example, forcing a 0000 input into a four input NAND gate would force all four NMOS transistors in the existing stack off and has been shown to reduce the leakage current by two orders of magnitude in a 0.18μm technology [32]. To capitalise on this observation, a number of works have proposed algorithms for estimating the leakage current and selecting appropriate input vectors to reduce the standby leakage power of a digital circuit [88, 110–112].

To apply the input vectors during standby, deep CMOS VLSI circuits do not benefit well from simply applying the vector on primary inputs as gates down the logic path will be uncontrolled. Instead a number of techniques have been proposed to provide better control of intermediate nodes. Firstly it has been proposed that multiplexers could be added into the circuit and their outputs controlled by a *SLEEP* signal choosing between the normal input or a fixed logic one or zero. However, since one of the inputs of the multiplexer is fixed, it is found to be easier and less costly to modify a gate in a similar way to fine grain power gating, Section 2.2.1.1 [110, 113]. If the gate's output is to be forced to logic 1 then an additional NMOS transistor and a PMOS clamp transistor is added, and vice versa for logic 0. It is evident though that swapping every gate would be much too costly in area and power and instead efficient selection of control points has been proposed [110]. Alternatively to reduce the re-design cost associated with inserting control points, it has been proposed to use scan chains to force the minimum leakage input vectors into the circuit [110, 111]. More recently, gate replacement or multi-threshold voltage gate assignment has been considered simultaneously with input vector control as it is found that some gates remain in a high leakage state even with a low leakage input vector [112]. In the case of gate replacement, for example, a NAND2 gate would be replaced with a NAND3 gate where the third input is driven by the same sleep signal that activates the input vector to force it into a lower leakage state.

2.2.3 Body Biasing

Traditionally the body of a transistor is connected to the same terminal as its source, i.e. the N-Well in a PMOS is connected to V_{dd} and the P-Substrate of the NMOS is connected to ground [2]. However, body biasing allows the voltage of the N-Well or the P-Substrate to be raised or lowered with respect to the source to allow control over the V_{bs} potential which directly influences the threshold voltage through the body effect, Chapter 1, Section 1.1.2. This fourth, body terminal on a transistor is often referred to as a back gate and therefore, this technique is sometimes also called back gate biasing. Applying a negative, reverse body (back) bias (RBB) raises the threshold voltage whereas a positive, forward body (back) bias (FBB) decreases the threshold voltage. Raising the threshold voltage in Eqn. 1.9 reduces sub-threshold leakage currents, but this also increases propagation delay, Eqn. 1.7. RBB has therefore been proposed as a method to reduce leakage current during idle modes of operation so as not to affect active performance [114, 115]. The method of RBB can be explained with the diagram in Fig. 2.10. During the active mode, the body of the PMOS/NMOS transistors are connected to V_{dd}/V_{ss} and during the standby mode they are raised/lowered to $V_{dd}+\Delta V/V_{ss}-\Delta V$. The ΔV required for the biasing of the transistor bodies is often provided by a bias generator circuit or charge pump circuit and the approach shown in Fig. 2.10 requires a triple well process [32]. Alternatively, examples have been shown where the NMOS body

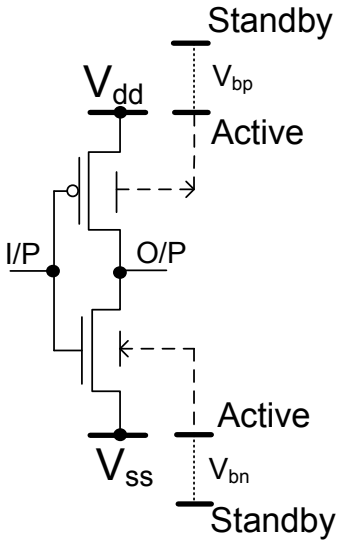


Figure 2.10: Reverse body biasing [32]

is fixed to V_{ss} and the source of the NMOS transistor is raised instead enabling the use of a standard shared substrate, independent N-well process [115].

It has recently been emphasised that in modern process technologies where the source, drain and substrate regions of a transistor are heavily doped to reduce short channel effects, a high electric field across the source to bulk and drain to bulk junction depletion region can cause a significant increase in band-to-band tunneling leakage current [114, 116]. Therefore, as reverse body bias is increased and threshold voltage increases, the leakage current does not continue to monotonically decrease and instead shows an inflection where BTBT leakage equals subthreshold leakage. To combat this, Jeon et al. propose a circuit that can monitor both the subthreshold leakage and BTBT leakage to maintain the optimal bias voltage to minimise standby leakage [116]. RBB, however, is not the only way to minimise leakage when using body biasing and alternatively, it has been proposed that high-V_{th} transistors could be used throughout the design making it inherently low leakage and FBB should be applied to decrease the threshold voltage during active mode to increase performance [117–119]. FBB has also been used with a scaled voltage. With a scaled voltage both dynamic power and leakage power improve during the active and standby modes and by using FBB the performance of the transistors can be tuned up during the active mode to achieve the required performance [117, 119].

2.3 Runtime: Active Mode Leakage Techniques

Unlike standby mode runtime leakage reduction techniques (Section 2.2) that rely on extended periods of no execution to implement a low leakage state, active mode runtime leakage power reduction techniques are used when the digital circuit is still doing useful

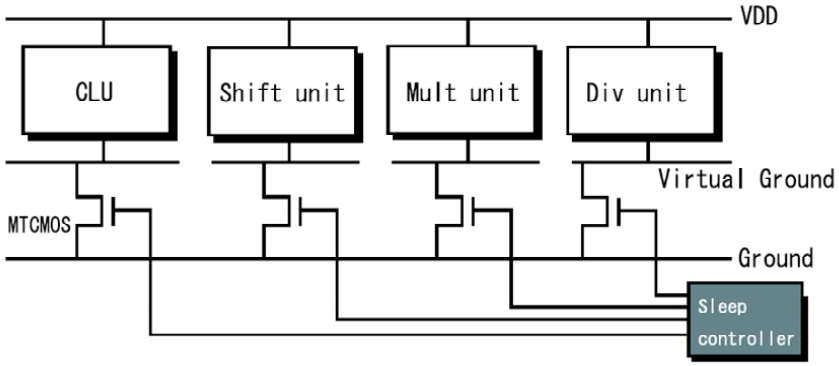


Figure 2.11: Power gating of individual executional units [121]

work. These techniques selectively employ lower leakage states when the current workload, task execution or state permits it. In some cases the decision to use the technique comes from a controller as is found in standby leakage minimisation techniques but in others the decision is made 'on-the-fly' from internal circuitry.

2.3.1 Power Gating

By implementing power gating at the CPU core level, the power gating technique is limited to times when the entire core is idle. At the level of functional units however, the opportunities for power gating can be greater allowing a block such as an ALU or multiplier to be power gated whilst the rest of the core continues to operate [120]. For example, an analysis done by Hu et al. shows that, based on a technique that detects idle periods, a floating point unit could be placed in a sleep mode for up to 28% of the execution time in floating point benchmarks and using a method based on branch mispredictions the opportunities could rise further [120]. This opportunity to power gate executional units has been exploited in a prototype MIPS R3000 processor [121]. The authors split the execution unit into an individual multiplier, divider, shifter and ALU and use the fetched instruction to determine which of the execution units is needed, Fig. 2.11. Across four different benchmarks it is shown that 47% of the total leakage power can be saved [121]. The idea of power gating executional units has been extended further to power gating parts of a functional unit, such as part of a multiplier or adder, depending on the data width [122–124]. For example, Sjalander et al. configure the precision of a 16bx16b multiplier between half and full precision using power gating. When operating at half precision, it is reported from post layout simulation that power gating half the multiplier allows 53% power saving [122]. Usami et al. on the other hand fabricated a 32bx32b multiplier where the top half can be power gated for a 16bx16b multiplication and, using input data from a MIPS R3000 CPU JPEG decoding program at a clock frequency of 100MHz, report up to 39% power saving [125].

The clock in a sequential digital circuit ensures execution happens in lock-step and many digital circuits can ultimately be thought of as finite state machines where they simply

move through a number of pre-defined states. This observation has been capitalised on by partitioning the datapath of a processor into power gateable sub-sections according to the governing finite state machines of the datapath [126]. Using the current state of the finite state machine it is possible to determine which sub-section must remain on whilst the other sub-sections can be powered down [127].

Clock gating is now a well supported dynamic power reduction technique in digital circuits [17] and was described in Chapter 1, Section 1.2. The technique stops the clocks to registers that maintain the same state over multiple clock cycles to avoid unnecessary dynamic power dissipation [13]. This consequently means the combinational gates that form the inputs or outputs of the registers can theoretically be considered redundant since, in the case of the inputs, any change is ignored and at the outputs, no change needs to be propagated. This observation has been exploited to propose the use of the clock gating enable signal to control the power gates to an integer execution core [118]. The integer execution core is fabricated alone in a 130nm technology library to study how activity profiles affect the power savings achievable. The technique targets a high performance operation at a clock frequency of 4.05GHz. It is shown that at their desired performance target at an average activity factor of 0.05, i.e. the execution core is active for 400 clock cycles and idle for 7600 clock cycles, it is possible to achieve a total power saving of 15% when compared to using clock gating alone.

The idea of using clock gating signals for power gating has recently been extended to a much finer granularity to enable power gating of the fan-in or fan-out of registers throughout a digital circuit [128–132]. The concept behind this technique is the ability to identify the logic gates that form the 'cones' of the clock gated registers, such that they can be grouped into power domains, Fig. 2.12. In the example given in Fig. 2.12, Fig. 2.12(a) shows a normal clock gated circuit whilst Fig. 2.12(b) shows the circuit with power gating implemented. To create this power gated version, the following is done [128]. A clock gated register (F_i) is found and the fan-in of that register is traced back towards a primary input or register. Each combinational gate that is encountered is labelled with the clock gating signal index (i) for the corresponding register. This is repeated for each gated register. The fan-in of every register that is not clock gated is labelled with an index $i = 0$. Every gate that has a nonzero index i forms a group G_i that can be power gated with a single clock enable signal. Some gates will have more than one clock-gating signal and so an n input NAND (in this example, or OR if using PMOS header transistors) is used to control the power to their domain. All gates labelled with an index $i = 0$, and gates that form the control of clock enables are kept always-on. The technique is designed for use in high performance systems where the power gating would be used in a manner akin to clock gating [128]. The short periods of time that the logic is power gated for therefore requires careful selection of which domains will result in net power savings as the energy required to enter and exit the power gated state may exceed the energy saved from the time spent in it. Additionally, the power gating should not

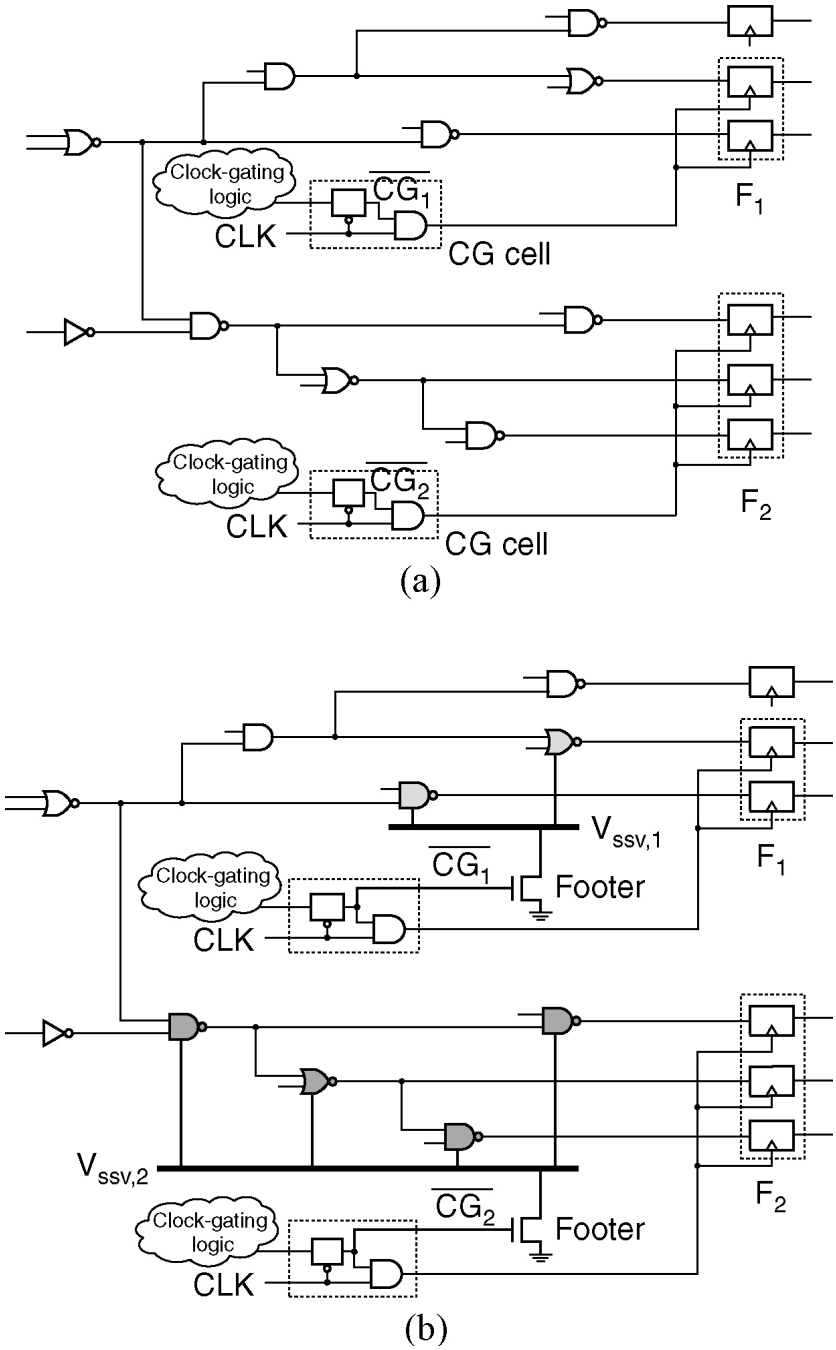


Figure 2.12: Grouping of logic gates into power domains controlled by clock enable signals [128]

affect performance but the inputs to the clock gated registers must be isolated from the power gated logic introducing an additional gate delay on the paths. Consequently, a number of works have developed algorithms and grouping methodologies to determine ways to power gate the logic [130, 133, 134]. For example, in an analysis of a 32-bit DSP microprocessor by Usami et al., they find that domains controlled by more than 3 clock gate enables would not result in power savings and end up with 34 power gateable domains in total, resulting in an 83% reduction in leakage power [130]. Seomun et al.

conversely assert three constraints to group logic together to enable implementation on any circuit. The first constraint named 'functional' is based on the logic cones and determines which gates can be power gated with which clock gate enable. The second is 'timing' and observes that the virtual rail should discharge and the logic should evaluate within half a clock period. Lastly a 'current' constraint is imposed to ensure both a large wake-up delay penalty is avoided and voltage drop across the transistor is kept within a certain percentage. For the latter two constraints, if the constraint is not met, the first logic gate of a critical path is dropped until the constraint is met. Experiments are carried out on ISCAS benchmark circuits and it is shown that their method enables 16% reduction in active leakage on average [133].

2.3.2 Adaptive Body Biasing

As described in Section 2.2.3 body biasing has been utilised as a method to reduce the leakage power of a digital circuit during the standby mode through control of the body to source potential (V_{bs}). A number of techniques, however, have exploited body biasing to adaptively vary the threshold voltage during the active mode to minimise leakage whilst continuing execution. Due to the way V_{th} variation affects performance, Eqn. 1.7, the most common use of V_{th} scaling during the active mode is for adapting to the current workload [135, 136]. For this reason adaptive body biasing can be considered very similar to dynamic voltage and frequency scaling, Chapter 1, Section 1.2, where the former is more effective for leakage power reduction and the latter is more effective for dynamic power reduction. Fig. 2.13 shows a potential implementation that uses a feedback loop to adaptively change the body bias between a set of programmable levels according to the desired performance and is called Dynamic V_{th} Scaling (DVTS) [135]. The desired frequency is determined by the operating system and is supplied to the DVTS hardware. The voltage controlled oscillator (VCO) is operated from the two body bias voltages and its frequency is matched against the desired frequency. Any error is then fed to the feedback algorithm which then sets the charge pumps accordingly. Dynamic V_{th} scaling reduces leakage power with clock frequency, while dynamic voltage scaling reduces dynamic power with clock frequency, however studies have shown that the minimum energy point for a given clock frequency is determined by both an optimum supply voltage and threshold voltage [31, 137]. As leakage power becomes a more dominant source of power dissipation in newer process technologies adaptive body bias has been coupled with dynamic voltage scaling to achieve simultaneous dynamic and leakage power reduction. A number of works have consequently proposed various algorithms and models for implementing combined voltage scaling and body biasing to maximise energy efficiency for a given performance requirement [138–140].

An alternative implementation for an adaptive body biasing scheme has been proposed that utilises two finite modes of operation rather than a continuous range and is known

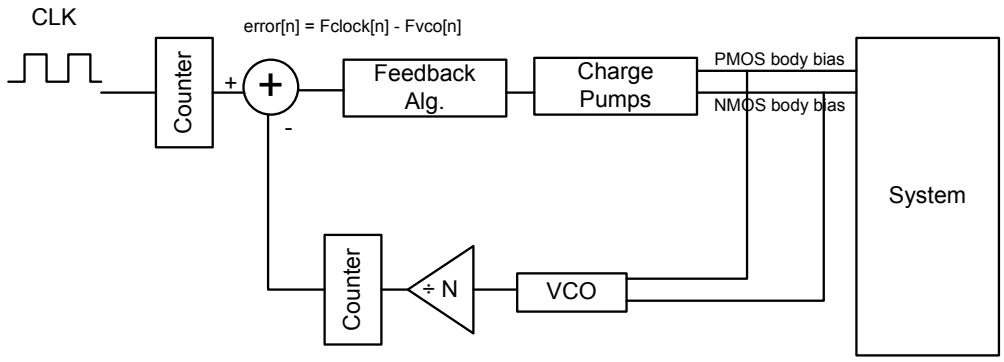


Figure 2.13: Adaptive body biasing scheme using feedback [135]

as V_{th} hopping [136]. The power control block receives a control signal typically derived in software and generates the signal lines to control the V_{th}s in the processor and corresponding clock frequency. When a *VTHlow_Enable* signal is selected the processor transistors have a low threshold voltage for performance driven tasks whereas when a *VTHhigh_Enable* signal is selected the transistors exhibit a higher threshold voltage for leakage power saving in low performance tasks. This approach has been demonstrated in a processor in which high threshold voltage transistors were used to inherently have a low leakage component of power dissipation during the active mode [136]. A zero bias is applied when *VTHhigh_Enable* is asserted and if a higher performance is required the *VTHlow_Enable* is asserted to forward bias the body of the transistors, lower the threshold voltages and increase performance. Experimental results show that at a V_{dd} of 0.9V the high threshold voltage logic with zero bias voltage allows power savings of 91% compared to when using 0.7V of forward bias. V_{th} hopping has since been proposed for use in capitalising on functional unit idle time like functional unit power gating (Section 2.3.1), rather than for matching required performance [141, 142]. Instead of two voltage hops, one voltage enables high performance and two voltages are used for the low leakage, high threshold voltage hop. A small hop, with minimal increase in threshold voltage and reduction in leakage, is used for short idle periods to reduce the energy overhead of moving between the low leakage and active states. A larger hop, with a greater increase in threshold voltage and leakage saving is used for longer periods of idle time such as traditional standby time (Section 2.2). The authors achieve an increase in leakage savings up to 19.2% when compared to just using V_{th} hopping for standby times [141]. V_{th} hopping has also recently been considered for use in a similar fashion to clock gating [143]. It is proposed that to enable body biasing to be applied and withdrawn within one clock cycle, it should only be used on selected transistors within a subset of carefully characterised logic gates. This primarily ensures that the process of employing body biasing during the active mode does not consume more energy due to the short time periods available for the hopping technique.

2.3.3 Subthreshold

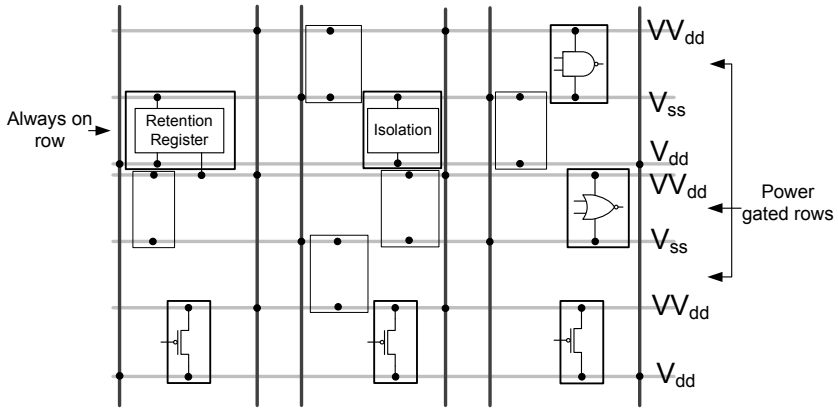
The subthreshold technique introduced in Section 1.4.2 is a compelling strategy for energy-constrained applications with low performance requirements, due to the simultaneous reduction in dynamic power and leakage power achievable with aggressive supply scaling. Consequently there have been a number of circuits designed using the subthreshold operation technique. For example, Wang et al. present a subthreshold FFT processor for wireless sensor nodes using a 0.18μm process library [50]. The processor operates at 350mV, with a clock frequency of 10kHz. It dissipates 155nJ for a 16-b 1024-pt FFT, which is reported to be 350 times more energy efficient than a low power microprocessor and 8 times more energy efficient than an ASIC. Zhai et al. on the other hand developed a full general purpose processor, named 'Subliminal', for wireless sensor network applications [53]. The processor achieves a maximum energy efficiency of 2.6pJ/instruction at 360mV, operating at a frequency of 833kHz. Jocke et al. developed a mixed signal subthreshold SoC for use in ECG applications [51]. Minimum energy operation is located at 280mV with an energy consumption of 1.51pJ/instruction at an operating frequency of 475kHz.

The main challenge with subthreshold operation is the constant battle of I_{on} current to I_{off} current within the logic gates to maintain correct functionality. Analysis on a 65nm technology node shows at super-threshold voltages the ratio can be as high as 7000x due to strong inversion, whereas due to the reliance of weak inversion currents in subthreshold operation, this ratio can be dramatically reduced at ultralow voltages to as little as 160x [37]. The impact of this is a weak I_{on} which can result in the output not swinging from rail to rail as the load capacitor is simultaneously charged and discharged by the on/off PMOS and NMOS transistors [50]. Topological features like MOSFET stacking and parallel leakage paths in particular degrades I_{on}/I_{off} [50, 52, 144]. Analysis was done on a typical XOR gate in [50], and it is shown that under one input combination, a single on transistor is pitched against three parallel off transistors resulting in the output not reaching the required 100mV value and instead falls short at 55mV. Considering the issues of degraded I_{on}/I_{off} in subthreshold operation a number of works have proposed cell library modifications and restrictions, and custom tools to aid in the design of a subthreshold design. For example, to avoid issues from long stacks of devices, Wang et al. buffer every input and output to ensure functionality in the design and then redundant buffers were removed later in the flow to limit additional area and dynamic power cost [50]. Kwong et al. limit maximum fan-in to 3 and developed a custom 62 cell gate library to avoid large stacks. They also propose a characterization flow which uses Monte Carlo simulation to ensure their custom gate library will function at ultralow voltages [52]. Similarly, Jocke et al. and Zhai et al. both propose custom tools to characterise logic cells over all input combinations to check functionality of their gate libraries [51, 53].

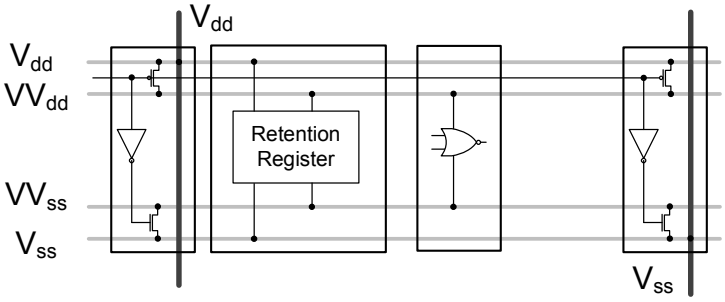
'Near threshold' is also a promising avenue for energy efficient digital circuits and shows similar energy efficiency when compared to subthreshold operation at reduced performance loss. For example, a study done on the Subliminal processor [53] shows that, when operating at near threshold voltage, the energy per operation goes up 1.4x but performance is improved by 5.1x when compared to operating at the minimum energy point ($V_{min} < V_{th}$) [145]. Due to the low performance achievable in near threshold computing, current research has proposed using the design technique for multi and parallel processing to regain the performance degradation. For example, architectural level parallelism is exploited in a JPEG co-processor to increase throughput by implementing four parallel engines instead of one [146]. This enables the authors to achieve the performance requirements for VGA video at a supply voltage near threshold. Fick et al. propose to use four ARM Cortex-M3 cores that each share a single instruction and data cache [147]. This ensures the cache is more active, reducing its idle time and hence improving the proportion of leakage to dynamic power dissipation in the cache. They combine 16 of these 'clusters' together to realise a 64 core system which achieves a performance throughput equivalent to half that of an ARM Cortex-A9 processor whilst operating at near threshold operation.

2.4 Physical Layout

There has been considerable research effort into the physical layout of ASICs but the majority of current research focusses on generic techniques for improving power, performance and area in normal ASIC layouts [148–150]. In the implementation of leakage power minimisation techniques, however, there has been little research in effective and efficient layout strategies. Power gating has prompted the majority of different physical layout methodologies, and has been driven by the need to route multiple supply rails within the design. For example, power gated cells require connection to the switched supply whereas always-on and isolation cells require connection to the unswitched supplies, Chapter 1, Section 1.4.1. This problem is further exacerbated if retention registers are required in the design and requires routing of up to four supplies to each of the registers if both the V_{dd} and V_{ss} are switched, Fig. 2.6. The concept of using a voltage area is currently the most common implementation methodology for power gated designs [3, 47, 94], and is well supported in commercially available multi-voltage EDA tools such as Synopsys IC Compiler and has already been introduced in Chapter 1, Section 1.4.1.1. The main advantage of a voltage area is its ability to use conventional standard cells and placement techniques with little additional design effort [3, 47]. This is because the physical layout in the voltage area can locally be treated as a normal generic layout, where the traditional V_{dd} power rail routing of the standard cells can simply be replaced with a VV_{dd} rail in the case of a switched power [47]. This means that all cells locally receive the switched supply rail and outside the voltage area V_{dd} can be routed without



(a) Power gated rows [151]



(b) Modified standard cells with supplies and sleep signal embedded [39]

Figure 2.14: Physical layout techniques for power gating

fear of the two supplies becoming shorted. However, as this constrains the placement of a subset of standard cells, a number of works have also proposed the use of using power gated rows to reduce this placement constraint [95, 133, 151]. This method scatters a subset of rows throughout the design where standard cells with a switched power supply may be placed. Fig. 2.14(a) shows one method of achieving power gated rows. The use of power gated rows enables the EDA placement tool such as Synopsys IC Compiler to assign a switched standard cell to the most optimal power gated row to reduce impact on area and routing from its constrained placement. Some power gating methodologies have also proposed the use of modified standard cells which enable the routing of multiple supplies through each standard cell [39, 152]. One such example is shown in Fig. 2.14(b) where both the V_{dd} and V_{ss} are switched. Each standard cell has all four power supplies (V_{dd}, V_{ss}, VV_{dd} and VV_{ss}) routed through the cell as well as the sleep signal for the power gating transistors [39]. There is considerable design and area overhead associated with this approach but the technique enables the placement tool to locate the cells in the layout wherever it sees fit thereby removing all placement restriction.

2.5 Objectives

There is little doubt from this literature review that leakage power minimisation continues to attract research effort due to the scaling of process technology and broad range of emerging applications. Leakage power minimisation during the active mode is a growing area of research due to the significance of leakage not only during standby mode but also during execution and recent research demonstrates a number of techniques that target leakage reduction opportunities during this time. The techniques discussed in this chapter can ultimately be grouped according to the types of system they target: high performance and low performance. The majority of techniques discussed in this chapter focus on leakage minimisation in high performance systems. For example, adaptive body bias can be used to trade performance for leakage power reduction but has been utilised for tuning the frequency in high performance processors [139]. Similarly many of the power gating techniques have been focussed on cutting leakage both in standby and active mode in high performance processors [118, 122, 128]. The primary focus of this thesis, however, is on leakage power minimisation techniques to improve energy efficiency in low performance (10-100s kHz), embedded processor applications and the key technique discussed in this chapter that is currently used to achieve this is subthreshold operation. Examples include the Subliminal processor by Zhai et al. [53], the FFT processor by Wang et al. [50] and the ECG processor by Jocke et al. [51] which all have pJ to nJ energy consumption achievable through sub-1MHz performance.

The subthreshold technique is a compelling strategy for low performance energy constrained systems particularly because of its ability to operate a circuit near or at its minimum energy point. However, there are a number of challenges with designing a circuit for operation at ultralow voltages. Subthreshold operation is predominately of academic interest and it is widely regarded that process-related variability is one of the critical barriers that needs to be overcome for subthreshold logic to be used in industry [36, 37, 144]. The main reason for this is the constant battle of I_{on} current to I_{off} current within the logic gates to maintain correct functionality. It has already been mentioned in Section 2.3.3 that analysis on a 65nm process shows the I_{on} to I_{off} ratio is lowered from 7000x at super threshold operation to 160x when operating in the subthreshold region, but this can be further exacerbated with process variation. It is shown that the sensitivity of I_{on} to key parameters such as V_{th} and V_{dd} increases from 1.17-1.2x at super-threshold operation to between 10-18x at subthreshold operation in a 65nm technology [37]. Therefore, global or local variation in the threshold voltage due to random dopant fluctuations (RDF) and channel length fluctuation, for example, can cause the minimum energy point to shift and may have a problematic effect on functionality from PMOS and NMOS device mismatch [37, 52, 144]. Statistical models, for example, predict that the minimum energy point can vary by as much as 78mV in a 130nm process due to process variability [37], and Zhai et al. show that each fabricated chip must be individually characterised to locate the minimum energy point and corresponding clock frequency

[53]. Furthermore, simply factoring a margin for performance variation can necessitate 1/10 degradation in performance and energy efficiency [145]. These variations can also have a negative impact on noise margins. For example, incorrect operation of inverters in a register from reduced logic swing can decrease the hold static noise margin of the latches resulting in loss of state, resulting in functional problems [52]. As discussed in Section 2.3.3 this requires careful consideration and additional design effort during the implementation of a subthreshold circuit which can be costly in both time and money. Modified gate libraries are not uncommon [50, 52] and custom tools for characterisation of gate library cells and timing analysis are also required [51–53]. It is shown that as technology scaling continues to 22nm and below the increasing off current (leakage) will have a detrimental impact on on/off current ratio and when coupled with variability, it will become the dominant limiter for low V_{dd} operation and circuit implementation [144]. For this reason there is a need for techniques that enable reduction in leakage power and improvement in energy efficiency for low performance processors during the active mode, that can be easily implemented using standard EDA tools and gate libraries.

The objectives of the research reported in this thesis are as follows:

1. Develop effective leakage minimisation techniques taking advantage of the low frequency of operation found in low performance embedded processors for energy constrained applications
2. Incorporate the techniques into industry standard EDA design flows and validate the techniques through synthesis and place and route using state of the art EDA tools and power accurate simulators such as HSpice, running a range of case studies
3. Incorporate developed techniques in a fabricated test chip and validate experimentally
4. Investigate and develop physical layout techniques for seamless integration of proposed leakage power minimisation techniques

The main focus of this thesis is to develop leakage power minimisation techniques for low performance energy constrained applications that can be implemented with a standard design flow using commercially available EDA tools. Furthermore, commercially available standard cell libraries should also be fully compatible with the proposed techniques to reduce implementation overhead such as is found in subthreshold design. These two aspects are vitally important to minimise design effort which helps to reduce time to market, risk and cost associated with development and implementation. To validate the effectiveness of the proposed technique, power simulations will form an important part of the evaluation across a set of test cases representative of the target low performance applications. Furthermore fabrication of a test chip will help to validate the effectiveness of the proposed techniques experimentally. As has been mentioned in Section 2.4,

physical layout is an important step in the implementation of leakage power minimisation techniques but has received little attention. Consequently, to improve the energy efficiency of the proposed solutions, the physical layout of their implementation will be investigated with a view to optimization.

2.6 Concluding Remarks

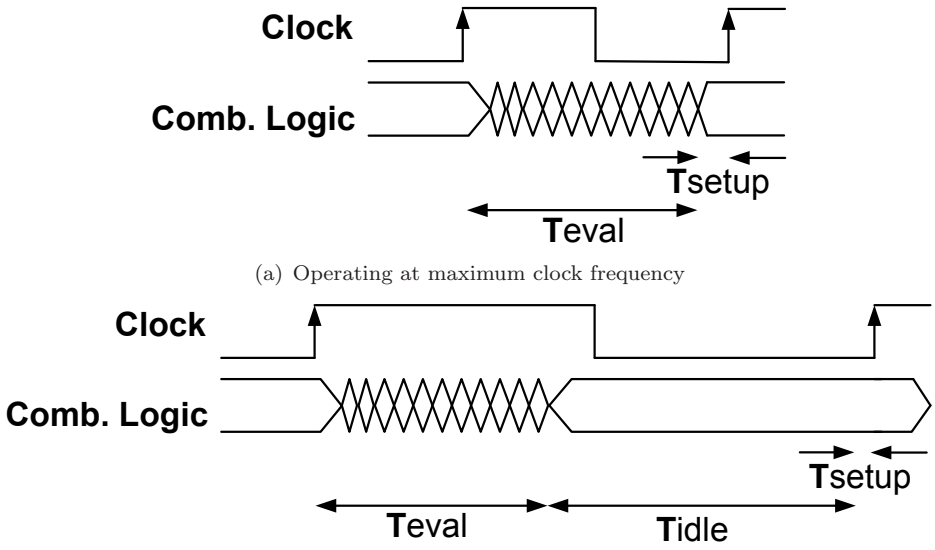
This chapter has presented an overview of state-of-the-art research in leakage power minimisation for integrated circuits ranging from techniques that can be applied and fixed at design time to techniques that are enabled and disabled at runtime dynamically. The vast diversity of techniques presented in this chapter shows that leakage power is a major problem in integrated circuits and is a mature and continuing area of research with more focus on reducing leakage during the active mode. The majority of techniques that have been discussed, however, are most applicable to high performance systems capitalising on things like standby periods, reduced performance requirements and idle functional units in 100MHz+ processors. For integrated circuits that operate at low to moderate performance points and demand high energy efficiency, the key technique currently used for reducing leakage power is operation at subthreshold voltages. This chapter has however highlighted that two critical barriers for the subthreshold technique are process related variability, which affects circuit reliability, and the complex design flow required for implementation of the technique which increases the RTL to silicon time. These two challenges are the main reason why there has been a slow uptake of the subthreshold technique in industry and why its use is limited to a small number of applications. This chapter has also shown that while physical layout is an important area of research, most research has focussed on generic techniques and little consideration is often put into the physical layout of leakage minimisation techniques. All the above drive the motivation for the research in this thesis which targets leakage power minimisation techniques and their physical layout for low performance, energy constrained processors that can be seamlessly integrated into standard design flows using commercial EDA tools.

Chapter 3

Active Mode Sub-Clock Power Gating

The significance of leakage power dissipation on energy efficiency in a digital circuit was considered in Chapter 1 and is a growing concern in nanometer technologies due to lowering of the threshold voltage, reduction of the gate oxide thickness and increased doping concentrations around the source and drain regions [4, 6, 32]. At sub-65nm technology nodes, leakage power is considered to be as dominant as dynamic power [32] and therefore poses a large source of power dissipation in current and future integrated circuits. It has been shown in Chapter 2, Section 2.3 that a number of techniques exist to reduce the leakage power during the active mode. Subthreshold operation is currently the key technique for low performance, energy constrained applications but suffers a number of challenges related to operation and implementation motivating the need for new active mode leakage minimisation techniques for low performance processors.

This chapter discusses the motivation, technique, design flow and simulation results to reduce leakage power in low performance embedded processors through the proposal of a novel active mode power gating technique that is fully compatible with a standard power gating design flow using commercial EDA tools. The negative impact of leakage power on the energy efficiency of low performance processors is discussed in Section 3.1. Section 3.2 shows how power gating can be used to exploit combinational logic idle time within the clock period to reduce power and improve energy efficiency in low performance processors. Simulation results of the proposed technique are given in Section 3.3 and a qualitative comparison of the proposed technique with the subthreshold technique is given in Section 3.4. Finally, Section 3.5 concludes the chapter.



(a) Operating at maximum clock frequency

(b) Operating at less than maximum clock frequency

Figure 3.1: Idle time within the clock period from reduced clock frequency

3.1 Motivation

In many integrated circuits the clock frequency of the system is governed by the critical path of the circuit and is set such that the clock period is as close to the critical path length as possible to maximise throughput [4]. In these situations the clock period (T_{clk}) is close to the combined evaluation time of the combinational logic (T_{eval}) and setup time (T_{setup}) as shown in Fig. 3.1(a). However, in current and emerging applications such as wireless sensor nodes and bio-medical processors speed performance is either not critical or unnecessary and energy efficiency and power are important design goals due to the limited energy available from the untethered nature of the applications. In these types of applications it is common to see operating clock frequencies ranging from 1kHz to 1MHz as the demand on the processor is low, enabling power to be saved from unnecessary dynamic switching (Chapter 1, Section 1.5). In modern technology libraries these types of clock frequencies are easily attainable during synthesis, even with slow high threshold voltage devices, and will often result in large positive combinational timing slack [153]; the difference between when the next register state must be evaluated by and the time it is actually available [4]. This may also be true if the processor is designed for operation at higher clock frequencies but can be operated at much lower clock frequencies, such as the MSP430 which is capable of execution at 8MHz but can be operated down to 32kHz [60]. The effect of a reduced clock frequency is shown in Fig. 3.1(b) where the clock period (T_{clk}) has become longer than the combined evaluation time of the combinational logic (T_{eval}) and setup time (T_{setup}) resulting in idle time (T_{idle}) within the clock period. This can be substantiated with the information presented in Table 3.1, where the capacitive loaded gate delays for a set of typical logic gates from the Synopsys 90nm education kit and TSMC 65nm technology library are given for the lowest drive

Table 3.1: Worst case capacitive loaded gate delays

(a) Synopsys 90nm [154]

Gate	Delay (ps)
INV	38
NAND2	51
NAND4	127
NOR2	64
NOR4	124
XOR3	253
XNOR3	252

(b) TSMC 65nm [155]

Gate	Delay (ps)
INV	12.4
NAND2	20.6
NAND4	50.5
NOR2	28.3
NOR3	51.6
XOR3	252
XNOR3	251

strength gates. As an example, consider 50 XOR3 gates from the Synopsys 90nm library placed in series, each with a propagation delay of 253ps. The time taken for a logical transition to propagate down the path would be equal to 12.65ns. At an operational clock frequency of 32kHz with clock period (T_{clk}) equal to 31250ns, for example, this would result in 31237.35ns of positive slack time within the clock period.

To capitalise on this relaxed clock frequency, voltage scaling can be used which increases the propagation delay of the logic gates (Eqn. 1.7) and reduces both dynamic and leakage power dissipation. As explained in Chapter 2, Section 2.5 though, lowering the supply voltage to near or below the threshold voltage increases the battle of I_{on} to I_{off} resulting in increased sensitivity to process variation and noise [36, 37, 145]. Therefore, it is desirable to maintain a super-threshold voltage. Even at scaled voltages there can still be significant idle time within the clock period. For example, Wang et al. synthesised an ASIC implementation of a processor for speech recognition systems [153]. They show that the synthesised processor can operate up to a clock frequency of 83.3MHz at a nominal supply voltage of 3.3V whereas the necessary real time processing can be achieved at a clock frequency of 234kHz [153]. To capitalise on the low performance needs of their application, the supply voltage is reduced to 1.1V reducing the maximum clock frequency to 13MHz. This corresponds to a critical path length of 77ns but at a clock frequency of 234kHz, with clock period of 4274ns, this still equates to 4197ns of idle time within the clock period. A similar calculation can also be made for the MSP430 [60]. At a fixed supply voltage of 1.8V the MSP430 can be operated up to a clock frequency of 8MHz however, it is possible to operate the processor down to 32kHz for low performance tasks and is utilised in applications such as the Zebranet wireless sensor network application [61]. Assuming 8MHz is the maximum clock frequency attainable, this corresponds to a critical path of 125ns. At an operational clock frequency of 32kHz (31250ns), this results in 31125ns of combinational idle time within the clock period.

With leakage power as significant as dynamic power in nanometer circuits [156], this combinational logic idle time can represent a significant drain of energy during the active mode. For example, the simulation of a processor used in ECG applications shows that at the nominal operating clock frequency of 1kHz, leakage power dominates

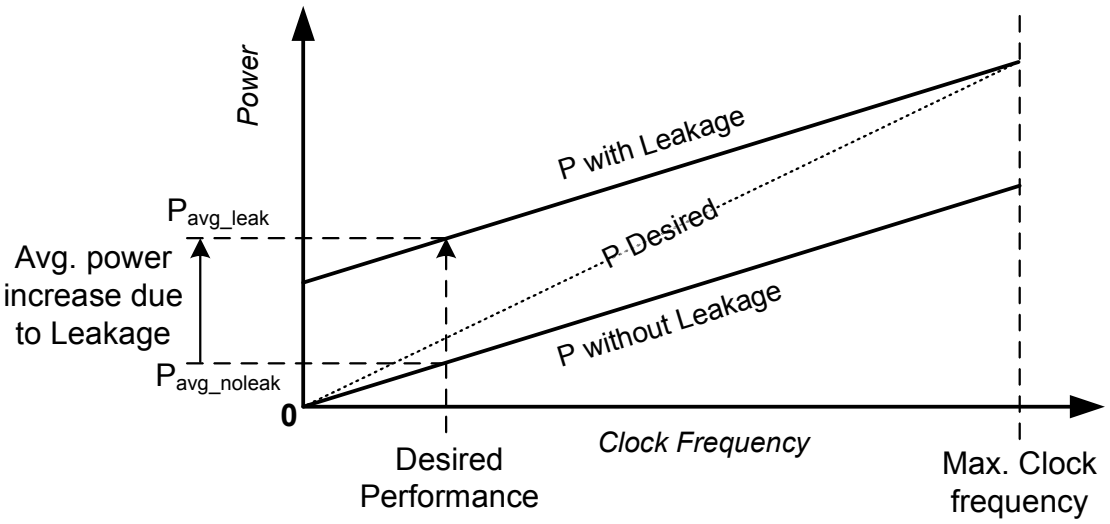


Figure 3.2: Increased power consumption at a given low performance target due to leakage power dissipation at fixed V_{dd}

the average power accounting for 60% of the active mode energy consumption [157]. Fig. 3.2 demonstrates the consequence of this problem on energy efficiency by relating power consumption to a desired clock frequency at a fixed V_{dd}. As the clock frequency is reduced from the maximum attainable clock frequency in Fig. 3.2, the average power consumption of the digital circuit reduces linearly and is governed by the equation for dynamic power, Eqn. 1.6. Leakage power, however, is not a function of the clock frequency, Eqn. 1.8, and so the power dissipation follows the *P with leakage* line as the clock frequency is lowered, crossing the Y-axis at a non-zero intersection when the clock is stopped, corresponding to the leakage power of the circuit. In an ideal integrated circuit with no leakage, the average power of the circuit would only be a function of the dynamic power and would instead follow the line given by *P without leakage*. Therefore, given a desired clock frequency the leakage power results in a higher average power than if leakage was not present in the circuit as shown in Fig. 3.2. Realistically however, leakage is something that cannot be eliminated totally whilst the combinational logic is actively switching as it is present in transistors while they are powered [6]. However, if the leakage power of the circuit could be totally eliminated during the idle time of the clock period the graph of the power consumption with leakage, *P with Leakage* in Fig. 3.2, would transform to the graph labelled as *P Desired*. This would mean the average leakage power would become a function of the clock frequency and is what is targeted within this chapter.

Power gating has most traditionally been used for cutting standby leakage power [3, 39, 41, 43] but more recently, power gating has been used more aggressively to take advantage of functional unit idle time [123, 124] and combinational logic idle time when using clock gating [129, 133]. Although these techniques can be employed over multiple clock cycles to reduce leakage power, combinational idle time within the clock period at low frequency operation is still an issue. The next section shows how power gating can

be used to capitalise on combinational logic idle time within the clock period to reduce leakage power dissipation during the active mode.

3.2 Proposed Sub-Clock Power Gating Technique

Power gating is a well known leakage power minimisation technique which cuts the power to sub-sections of an integrated circuit and due to its high practicality has been used to reduce the power of idle circuitry during standby and the active mode [41, 124]. The principles of power gating were discussed in detail in Chapter 1, Section 1.4.1. To reduce the component of leakage power during the active mode when using low clock frequencies the proposed technique capitalises on the combinational idle time to power gate it within the clock period and is referred to as sub-clock power gating. In this section the architecture of a circuit employing sub-clock power gating and the design flow to implement the technique are considered.

3.2.1 Sub-Clock Power Gating Architecture

The proposed sub-clock power gating (SCPG) technique is shown in Fig. 3.3 and has three distinct parts. Firstly, the digital design chosen for applying sub-clock power gating to is split into 2 domains: a combinational logic domain which can be power gated, marked as 'Comb. Logic', and a separate always-on sequential logic domain, marked as 'Seq. Logic', Fig. 3.3. This split is made to retain the register state and avoid the need for state retention registers which are used in traditional power gating to store state in sleep mode, as they increase area by 20-50% and introduce additional delay when changing between the sleep and active modes [3]. Secondly, a high threshold voltage PMOS header transistor is placed between the combinational logic and the power supply and is used to control the power to this power domain. The third distinguishing feature is the isolation logic between the combinational and sequential domains, shown as 'ISOL' in Fig. 3.3. This is also a common feature in traditional power gating [3], and is used to clamp the output signals to a known fixed value to ensure they do not cause crowbar currents in the always-on sequential logic when the combinational domain is powered down, Chapter 1, Section 1.4.1.

In traditional power gating schemes, the control to the power gates is usually driven by a power gating controller state machine as explained in Chapter 1, Section 1.4.1. However, a power gating controller is impractical in the proposed SCPG technique since the control needs to be issued within the clock period. The proposed technique instead uses the clock signal, as shown in Fig. 3.3. This alleviates the need for a higher clock frequency power gating controller saving area, power and implementation complexity and also minimises routing since the high-fanout clock tree of the processor can be

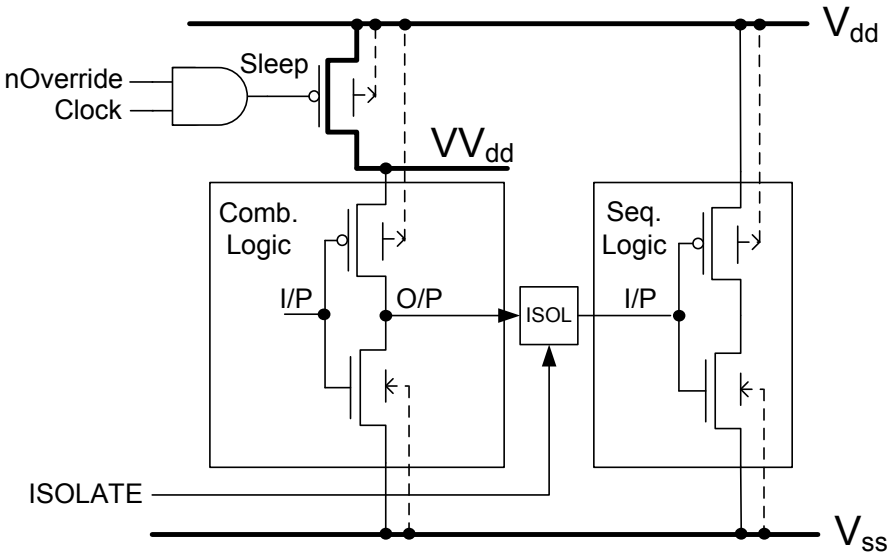


Figure 3.3: Sub-clock power gating technique

exploited for the control signal. This means, when the clock is high, the combinational logic is power gated as the virtual V_{dd} rail is disconnected from the V_{dd} supply and when the clock is low it is restored to V_{dd}. The leakage energy saved from using the proposed technique is therefore influenced by two variables: the circuit's clock frequency and the duty cycle of the clock. As the clock frequency is reduced, the idle time of the logic (T_{idle}) becomes greater as the difference between the evaluation time (T_{eval}) and the clock period (T_{clk}) increases, Fig. 3.1(b), presenting greater potential leakage saving. By limiting the clock signal to use a 50% duty cycle, it is possible to save leakage power for half the clock period ($T_{clk}/2$) but restricts the application of SCPG to when $T_{eval} < T_{clk}/2$ to allow enough time for the power to be restored and evaluation of the combinational logic. Changing the duty cycle, on the other hand, allows the application of SCPG even when $T_{clk}/2 < T_{eval} < T_{clk}$ by decreasing the duty cycle. Furthermore, for very low clock frequencies when $T_{eval} \ll T_{clk}$, changing the duty cycle to extend the high phase of the clock enables capitalisation of all the combinational logic's idle time to provide maximum leakage power saving, as will be demonstrated (Section 3.3). Notice in Fig. 3.3 that the clock signal is *ANDed* with an active low *nOverride* signal. This signal allows the sub-clock power gating to be disabled to provide the circuitry with a mode of operation with normal timing.

In traditional power gating the isolation of the combinational domain, 'ISOL' in Fig. 3.3, would also be controlled by a power gating controller [3]. In the proposed SCPG technique though, the circuit of Fig. 3.4 is used to drive the *ISOLATE* signal in Fig. 3.3 and is because of the need for control within the clock period. The proposed control is adaptive to the behaviour of the VV_{dd} supply rail and has two primary inputs: the clock signal and the value of the combinational logic VV_{dd} which is derived from a TIEHI logic gate[1] placed in the combinational domain. When the clock is logic 1, ISOLATE

[1] A logic gate that outputs a logic 1 by connecting to the V_{dd} supply rail with ESD protection

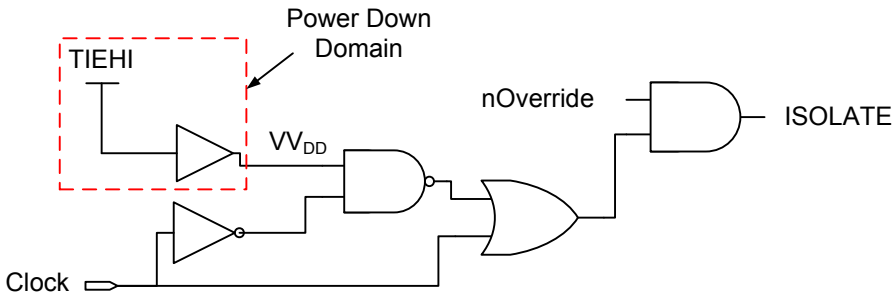


Figure 3.4: Isolation control circuit

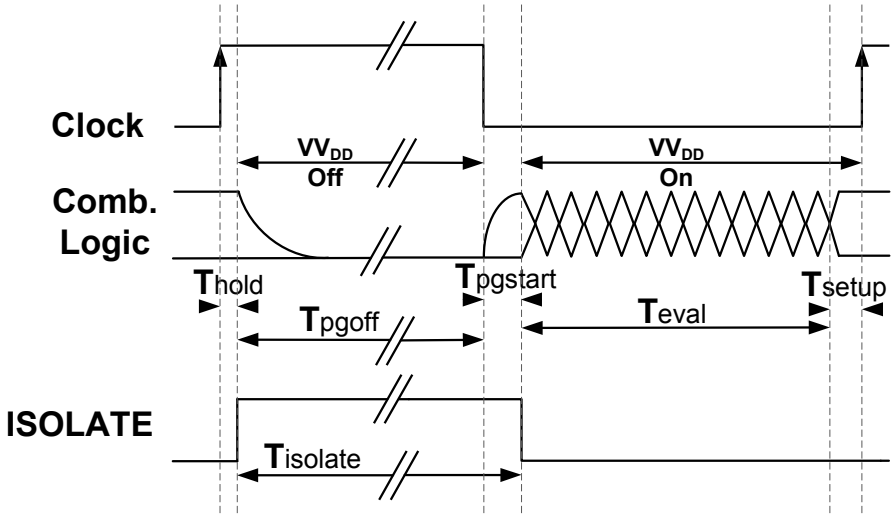


Figure 3.5: Sub-clock power gating timing

is driven to a logic 1, thereby isolating the combinational outputs. When the clock is logic 0, ISOLATE is held at logic 1 while the VV_{dd} input remains at logic 0 (discharged). This ensures the combinational outputs remain isolated until the supply rail is charged to an equivalent logic 1, eliminating short-circuit crowbar currents during wake-up. The output is *ANDed* with the *nOverride* signal used to disable the sub-clock power gating and ensures that if the proposed technique is disabled, additional switching energy is not consumed from the isolation gates.

To help understand what is happening with the combinational logic when using the proposed technique, the overall timing diagram of one clock period in sub-clock power gating is shown in Fig. 3.5. After the next logic state is clocked into the rising-edge triggered registers, the VV_{dd} rail to the combinational logic is disconnected from the V_{dd} supply but the capacitive nature of the virtual supply rail [3] means the time taken for the virtual rail to discharge ensures register hold times(T_{hold}), which are on the order of ps in modern technology libraries [155], will be met. At this point, the output isolation is also enforced. The virtual supply rail is held off for the remainder of the high phase of the clock (T_{pgoff}) minimising leakage power dissipation, and the outputs of the combinational domain remain isolated ($T_{isolate}$). Note that by changing the duty cycle of the clock it is possible to extend this off period (high phase of clock), maximising

the leakage power savings. The virtual supply rail is restored at the negative edge of the clock but the output isolation is held until the virtual supply rails are fully restored ($T_{pgstart}$). The remainder of the clock period is used for the evaluation of the next state (T_{eval}) and ensuring setup time (T_{setup}) is met before the process repeats in the next clock period.

3.2.2 Design Flow

The design flow to augment a digital design with the proposed SCPG technique is shown in Fig. 3.6; three additional steps are added to a traditional power gating design flow (Chapter 1, Section 1.4.1.1) and are indicated. A brief summary of each of these steps is given and further details will be given when discussing the fabrication of a sub-clock power gated case study in Chapter 4, Section 4.3. The design flow begins with the original RTL of the circuit that is to be mapped to a sub-clock power gating architecture. In order to achieve the power domain split shown in Fig. 3.3, the RTL must be written with separate Verilog modules for the combinational and sequential logic so that a UPF file can be used (Chapter 1, Section 1.4.1.1). This is a constraint of the UPF standard [46] and is the primary reason the first two process steps of the design flow are required. If the original RTL is easily split into combinational and sequential logic Verilog modules then the first two steps shown in Fig. 3.6 can be skipped and the split can be performed manually as will be shown in Section 3.3.1. If however the HDL description consists of intertwined combinational and sequential logic, the first step is used to synthesise the design to a generic gate library available through the EDA tool vendor, which in the case of Synopsys EDA tools is the GTECH library [17]. The output of this step is a flat gate level netlist of the circuit and enables a Perl script to be used in the second step to identify sequential and combinational logic gates and separate them into two individual Verilog modules. The output of the second step is then the same GTECH gate level netlist from the first step, but with the combinational logic in one module and the sequential in another. The third and final additional step merges this gate level netlist with the isolation circuit shown in Fig. 3.4 in a top level wrapper along with definitions for the control signals used for the power gates. The complete, split netlist can then be combined with the power intent UPF of the sub-clock power gating, which defines the power domains, power switches and isolation to match with the architecture shown in Fig. 3.3. The rest of the design flow remains the same as a traditional power gating design flow.

3.3 Simulation Results

To validate the sub-clock power gating technique, three case studies were used: a 16-bit parallel binary multiplier, an ARM Cortex-M0 microprocessor and an ASIC wireless

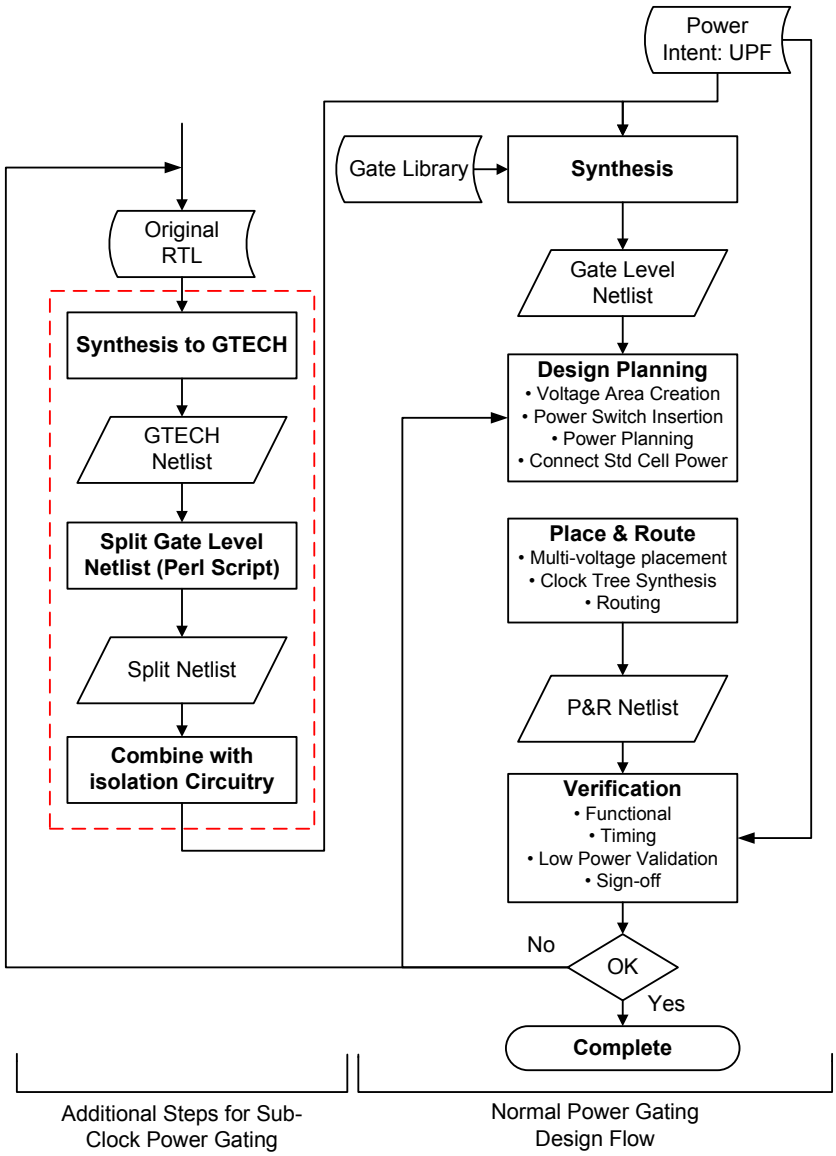


Figure 3.6: Design flow of the sub-clock power gating technique

sensor node processor, the Event Processor [63]. The flow of how all the results presented in this section were obtained is shown in Fig. 3.7. A brief summary of the steps in Fig. 3.7 is given but further details of how the HSpice simulation of the circuits was conducted can be found in Appendix B.3. Firstly, the designs were all implemented using the implementation flow described in Fig. 3.6, using a nominal 1.2V 90nm technology library[2] and the Synopsys EDA tool suite. A full transistor level netlist including parasitic resistors and capacitors (RC) of all signals and power grids was then extracted from the place and routed design using the Synopsys Star-RC tool to ensure the simulation provided an accurate representation of timing and power. Simulation vectors were captured from the Verilog simulation of the gate level netlist which were then ported into a digital vector file used in HSpice for the transistor level netlist simulation. The

[2]Synopsys 90nm Education Kit available from Synopsys

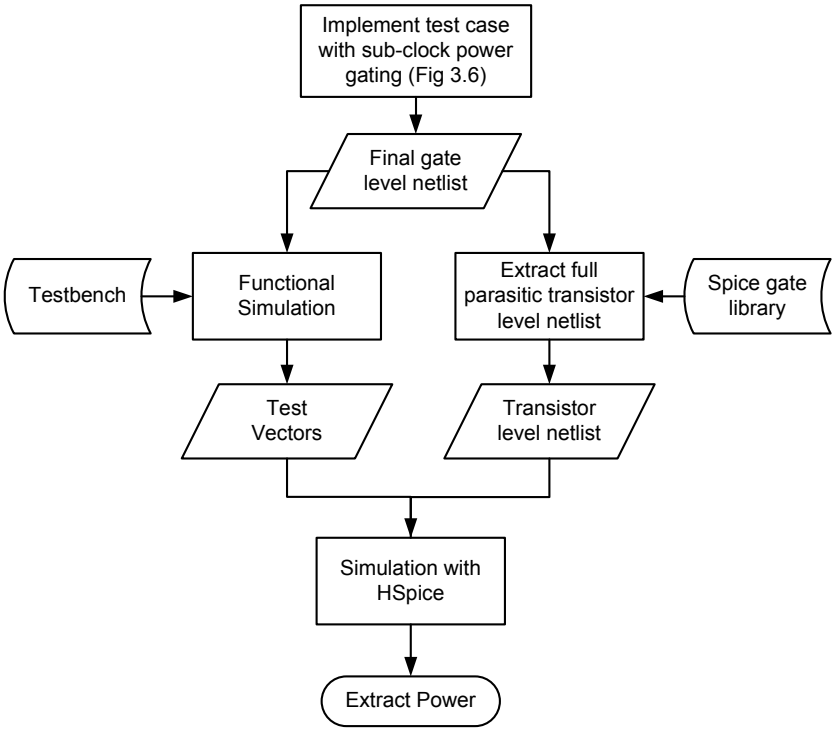


Figure 3.7: Experimental flow for generation of sub-clock power gating power results

post layout simulation was then carried out using Synopsys HSpice at a scaled voltage of 0.6V. This voltage was chosen because it remained adequately above the 0.4V threshold voltages of the transistors alleviating problems from near threshold operation whilst being half the nominal 1.2V voltage giving large dynamic and leakage power saving and is representative of the scaled voltage used in low performance applications [60, 64, 153]. Finally, the power and energy values were extracted from the simulation results and recorded, Tables 3.2, 3.3 and 3.4.

An integral part of implementing power gating is the choice of sleep transistors. In all the test cases, PMOS transistors are used and as discussed in Chapter 1, Section 1.4.1, the inclusion of the header transistors introduces a small IR drop to the power gated logic. As reported in previous publications, the header transistor size, the number of headers and their arrangement directly affects the IR drop across the power domain [3, 40]. With a lower IR drop the impact in performance is reduced and the time taken to reach an active state from power down is also reduced as a higher current can be facilitated through the power gating transistors. As shown in Chapter 1, Section 1.4.1 though, including many header transistors can have a negative impact on in-rush current causing ground bounce [3, 42]. Constraints of up to 5% IR drop are common in many power gating designs. Trying to achieve very low IR drop is important in high performance systems [45] but can result in unnecessarily large effective power gating widths, resulting in increased sleep mode leakage current and area overhead [3]. As such, iterative simulation was used to find the widths required for a 5% IR drop in

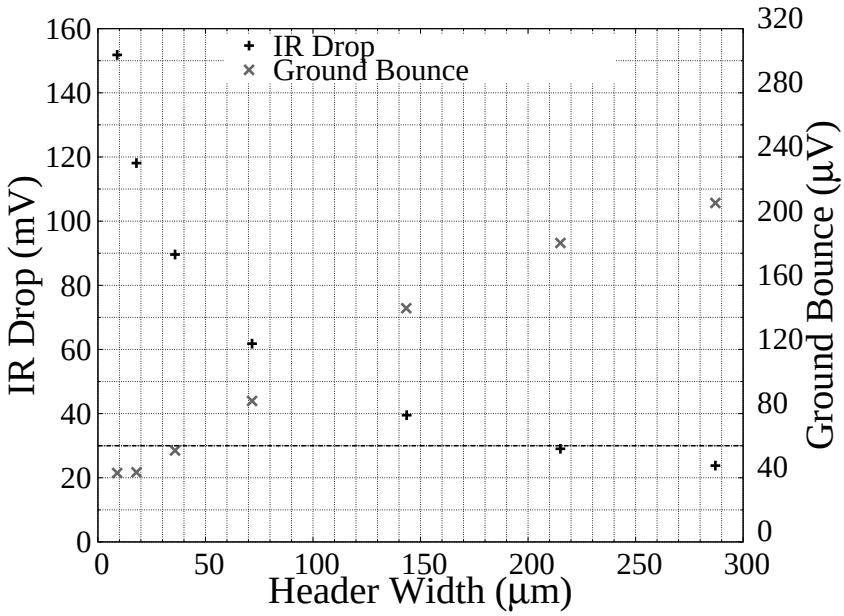


Figure 3.8: Effective PMOS power gating transistor width against IR drop and ground bounce in the 16-bit parallel multiplier

the test cases. Fig. 3.8, shows the effect of the header width on the IR drop in the 16-bit parallel multiplier test case. As expected, the smallest header width exhibits the largest IR drop equivalent to 25% of the supply voltage. With increasing effective header width the IR drop is lowered but improvements begin to diminish as size increases. The header width that achieves a 30mV IR drop was found to be 215μm. This width was also sufficient in the Event Processor test case used in this section, however, the Cortex-M0 test case is larger and an equivalent header transistor width of 286μm was required instead.

While IR drop was set as the main constraint, ground bounce due to in-rush current is an important effect to monitor as a large ground bounce can impair the reliability of the always-on registers in the sub-clock power gating design [42] as explained in Chapter 1, Section 1.4.1. Fig. 3.8 also shows header width against ground bounce. As expected the smallest header width achieves the lowest ground bounce but it is found that over all tested widths the ground bounce remains less than 0.04% of the supply voltage which can be attributed to the small size of the power gated domain. It can be concluded from these results that ground bounce is not an issue in this test case. A similar analysis on the Cortex-M0 and the Event Processor test cases used in this section shows ground bounce magnitudes of 560μV ($< 0.1\%$) and 180μV ($< 0.03\%$) respectively. The larger magnitude in the Cortex-M0 can be attributed to the larger power gated domain but these results further verify that ground bounce is not an issue for the state holding elements. In Fig. 3.8, it can be seen that increasing the header width to greater than 215μm can result in reduced IR drop with negligible impact on ground bounce, however, an increased header size also increases the sleep mode leakage current of the power gated block because subthreshold leakage is a function of transistor width, Eqn. 1.9. Fig. 3.9

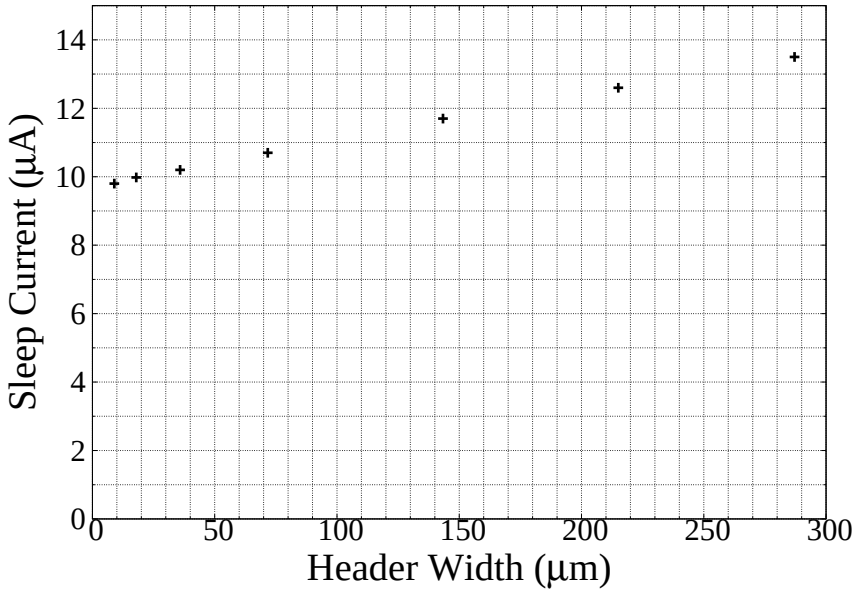


Figure 3.9: Effective PMOS power gating transistor width against sleep current in the 16-bit parallel multiplier

shows the effect increasing header width has on the sleep current and as expected leakage current linearly increases. This analysis shows that, while a header width of 287μm can improve the IR drop, sleep mode leakage current will probably suffer resulting in a higher overall energy consumption. Therefore, to maximise leakage savings during the sleep mode, the header is sized for 5% IR drop reduction only.

3.3.1 Case Study 1: 16-bit Multiplier

This circuit was chosen because of its large concentration of combinational logic to highlight gains in datapath and control blocks. The multiplier is a 16x16 unsigned binary multiplier that uses sum of partial products to compute the multiplication result. As the complexity of a multiplication increases quadratically, an example of how a sum of partial products multiplier operates is given in Fig. 3.10 using a 4x4 multiplication. In this example, A is the multiplier and B is the multiplicand. Firstly, the multiplier is bit-wise *ANDed* with each bit of the multiplicand and is lined up with the corresponding multiplicand bit in the table, as is shown, giving the partial products A_xB_y. Each column is then summed starting with the lowest significant bit to the highest significant bit to calculate the value of each bit in the product P; any carry bits are propagated to the following column. As can be seen, a 4x4 multiplication results in an 8-bit product, hence in the 16-bit multiplier test case used, a 32-bit result is obtained. The hardware implementation of the sum of products multiplication algorithm is shown to the bottom of Fig. 3.10. Since a number of partial product additions need to be computed, the hardware largely consists of full adders, labelled as *FA* in Fig. 3.10.

				A_3	A_2	A_1	A_0	Multiplier
				B_3	B_2	B_1	B_0	Multiplicand
				A_3B_0	A_2B_0	A_1B_0	A_0B_0	
			A_3B_1	A_2B_1	A_1B_1	A_0B_1		Partial Products
		A_3B_2	A_2B_2	A_1B_2	A_0B_2			
	A_3B_3	A_2B_3	A_1B_3	A_0B_3				
P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0	Product = Sum of Partial Products

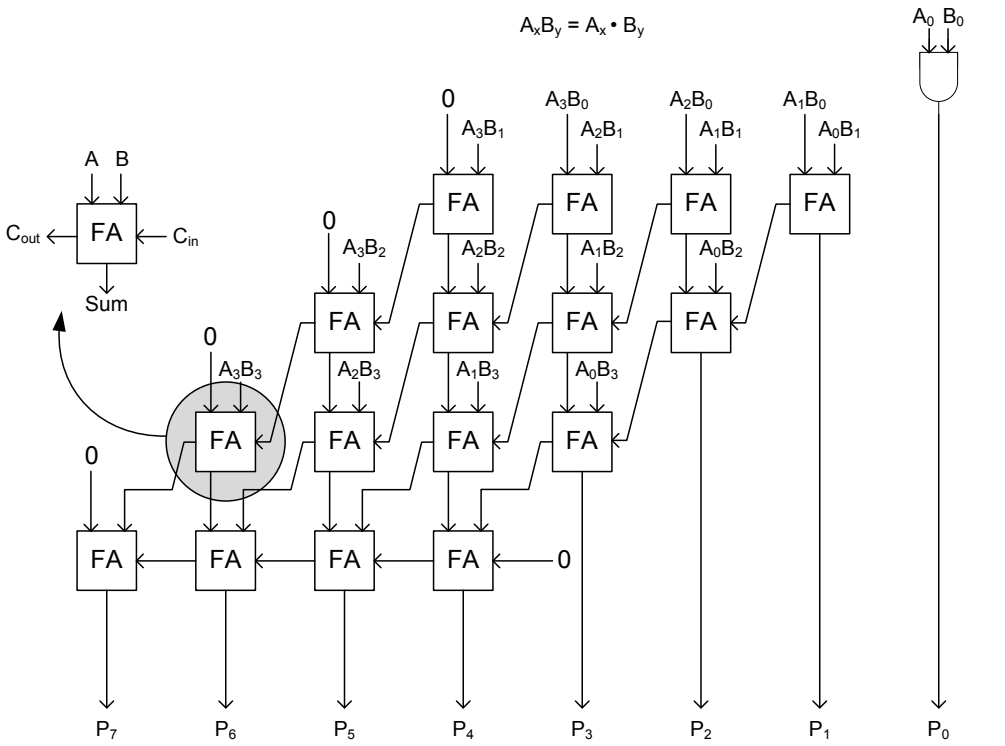


Figure 3.10: Example of a 4x4 sum of partial products parallel multiplier

To understand how a digital circuit can be mapped to the sub-clock power gating technique an explanation using the 16-bit multiplier test case is given and is explained diagrammatically in Fig. 3.11. The multiplier uses two 16-bit register inputs, A and B, which are fed to the partial product multiplication hardware. The result is then output from this hardware to the 32-bit product register P. All the combinational elements of the digital circuit, which in this case is the multiplication hardware, is separated into the 'Comb. Logic' block so that it can be power gated using sub-clock power gating. The registers on the other hand must remain always-on and they are grouped together into the 'Seq. Logic' block. As was explained in Section 3.2, the first two steps of the design flow, Fig. 3.6, require the circuit to be split into separate combinational and sequential Verilog modules. In the case of the 16-bit multiplier test case this is easily accomplished manually as the design is small and allows the registers to be placed in one Verilog module, whilst the combinational part of the RTL can be split into another module, as shown in Fig. 3.12. This means the first two steps of the design flow can be skipped. As previously mentioned the purpose of this Verilog rewrite is to enable the use of the UPF

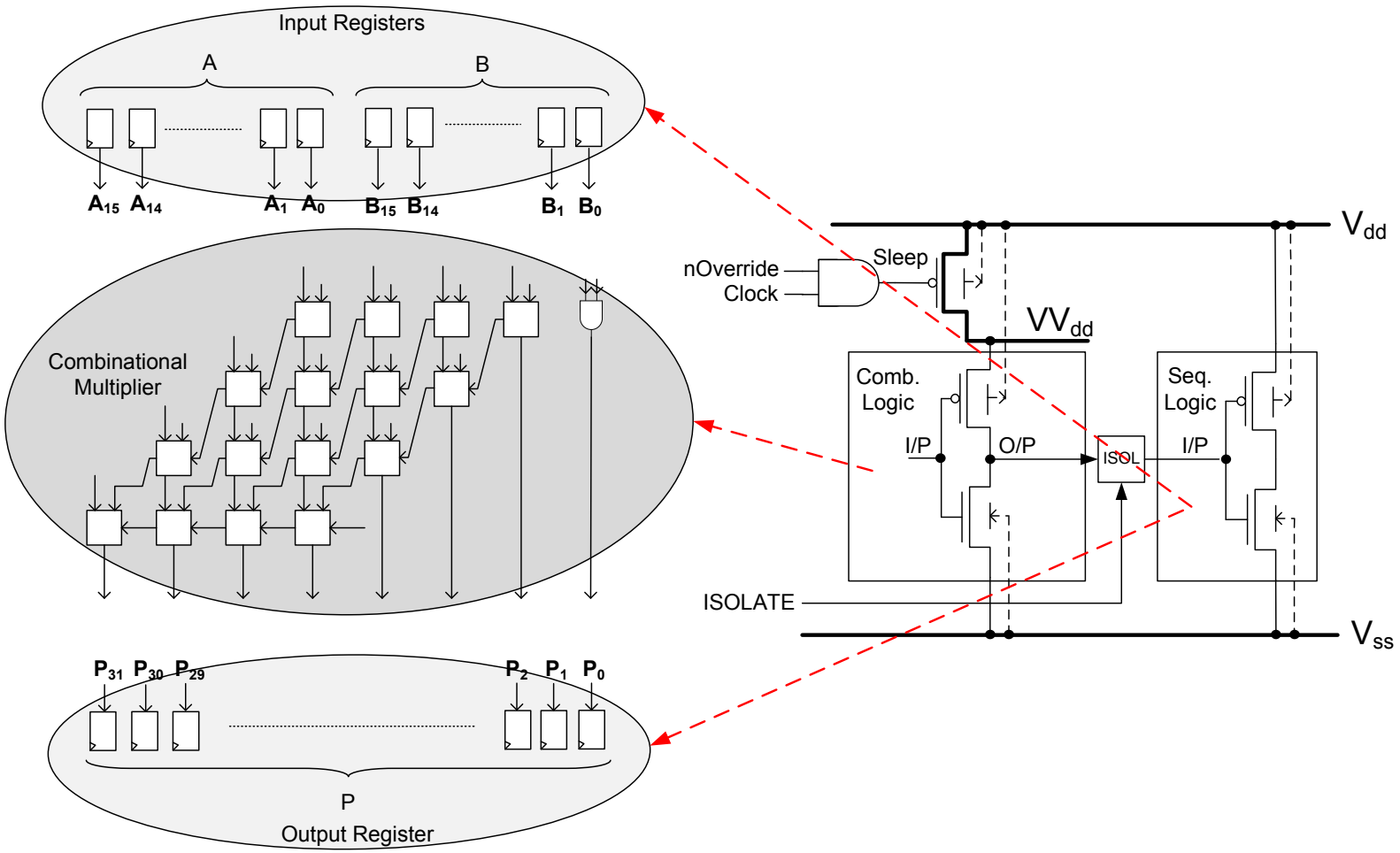


Figure 3.11: Example of how the multiplier circuit is mapped into the SCPG technique

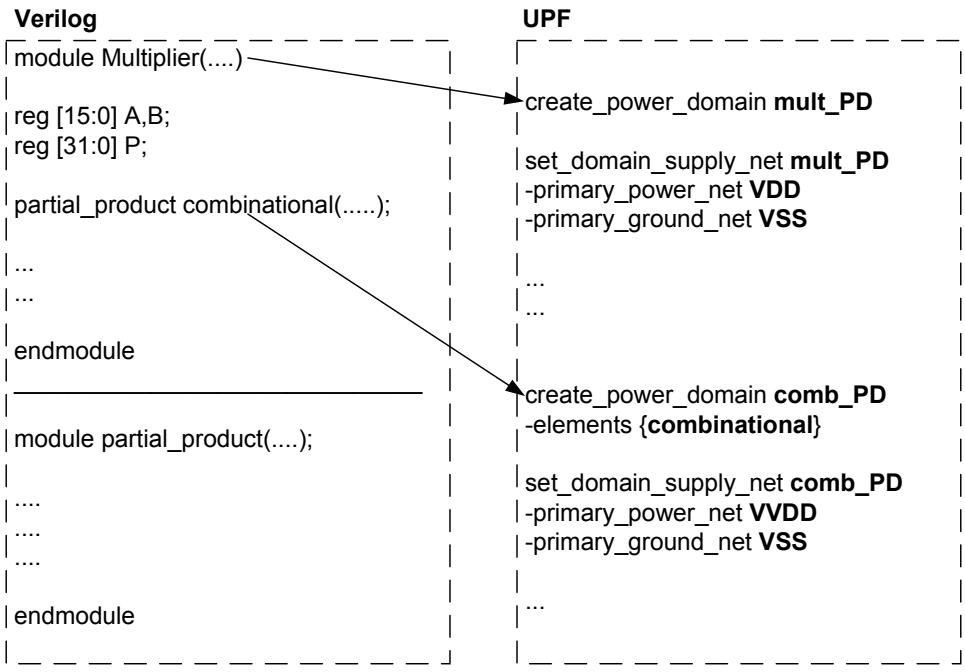


Figure 3.12: Example of how modules in Verilog are mapped into the power domain definitions in the UPF

power intent standard and Fig. 3.12 shows how the two Verilog modules are assigned to different power domains in the UPF file. A power domain *mult_PD* is created for the top level module with power and ground supplies V_{dd} and V_{ss} and contains all the registers of the circuit, whilst a separate power domain is created containing only the *partial_product* instantiation 'combinational' with a switchable VV_{dd} power supply.

During the implementation of the multiplier test case, the Synopsys EDA tool suite was set to optimise the hardware for minimum area resulting in a number of full adders in the partial product hardware being optimised. This resulted in the circuit being dominated by full adders (total 223) but also NOR gates (total 214) in the final implementation. Altogether there are 556 combinational logic gates and 64 registers and the combinational gates account for 85% of the circuit's leakage power. The inclusion of the SCPG technique results in a 3.9% increase in total area and can be accounted to the power gates, isolation gates, control circuitry, and the addition of buffers to compensate for the splitting of the combinational and sequential logic into separate voltage areas in the physical layout. The effect of voltage area splitting will be discussed further in Chapter 5.

Table 3.2 gives the average power dissipation and energy per operation values for a range of clock frequencies with no power gating, SCPG using a 50% clock duty cycle (Proposed SCPG-50%), and using greater than 50% duty cycle (Proposed SCPG-Max) to maximise leakage power reduction. In the SCPG-Max simulations the clock duty cycle is an ideal optimisation of clock duty-cycle. No timing constraints were set on the synthesis and through simulation it is found that the combined critical path and wake-up time for

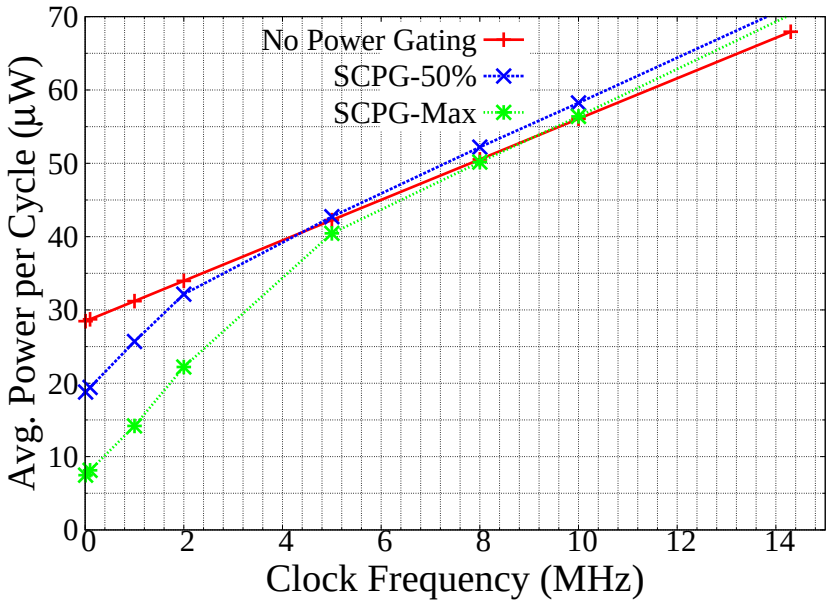
Table 3.2: Power and energy per operation of sub-clock power gated multiplier, V_{dd}=0.6V

Clock (MHz)	No Power Gating		Proposed SCPG-50%			Proposed SCPG-Max		
	Power (uW)	Energy (pJ)	Power (uW)	Energy (pJ)	Saving (%)	Power (uW)	Energy (pJ)	Saving (%)
0.01	28.47	2847	18.81	1881	33.9	7.47	747	73.8
0.1	28.72	287	19.44	194	32.3	8.13	81.33	71.7
1	31.20	31.20	25.70	25.7	17.6	14.18	14.18	54.6
2	33.96	16.98	32.15	16.1	5.3	22.22	11.11	34.6
5	42.25	8.45	42.72	8.54	-1.1	40.45	8.09	4.2
8	50.53	6.32	52.20	6.53	-3.3	50.17	6.27	0.7
10	56.07	5.61	58.23	5.82	-3.8	56.40	5.64	-0.5
14.3	67.94	4.75	71.48	5	-5.2	70.28	4.91	-3.4

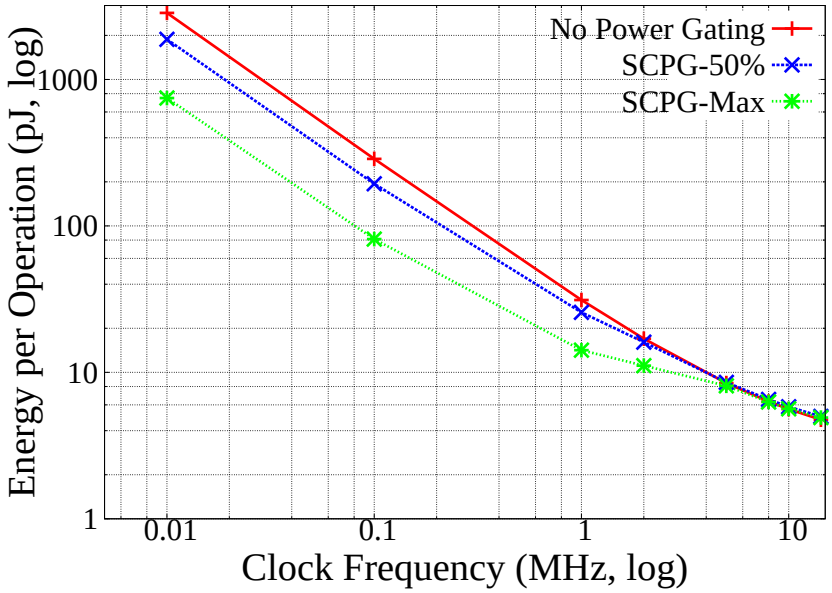
the multiplier is 20ns. While an optimised duty cycle is used in the simulations in this chapter, it is possible to implement the duty cycle modulation in hardware and will be described in Chapter 4. It can be seen in Table 3.2 that the proposed SCPG technique reduces the average power when compared to no power gating, and greater savings are achievable at lower frequencies because of increased idle time of combinational logic. Furthermore, modulating the high phase of the clock maximises the savings achievable. For example, at 10kHz, the power saving rises from 33.9% to 73.8%.

Fig. 3.13(a) depicts the trends in average power dissipation with clock frequency. As can be seen the average power dissipation of the 3 setups converge with increase in clock frequency because, as the idle time of logic decreases, the power dissipated from recharging the virtual supply rail from switching between the sleep mode and active mode begins to dominate over leakage power savings. Thus, the point is reached when the power dissipated switching modes equals the leakage power saved and beyond that frequency an SCPG design would not save any power. For the multiplier, it was found that the SCPG-Max and No Power Gating setups converge at approximately 8MHz. In the sub-clock power gating technique though, the *nOverride* signal provides a method by which the power gating can be disabled (Fig. 3.3) and so, if the circuit was required to be operated at clock frequencies above and below 8MHz, the circuit could be switched to no power gating at frequencies above 8MHz. It should be noted that the addition of the isolation cells between the combinational and sequential logic introduces additional leakage and dynamic power overhead which are absorbed at low frequencies by the leakage power that is saved. However, at frequencies over 8MHz where SCPG would not save any power the power gating technique can be disabled but the isolation cells would still dissipate power in a circuit employing sub-clock power gating. In the multiplier test case it is found that the 'override' mode of operation dissipated 5% additional power compared to the original circuit with no power gating circuitry.

Fig. 3.13(b) shows the energy per operation against clock frequency for the three designs. As expected, energy per operation decreases with increasing clock frequency as the idle time within the clock period shortens resulting in less energy being wasted as leakage.



(a) Power



(b) Energy

Figure 3.13: 16-bit parallel multiplier, V_{dd}=0.6V

Note though, that at any clock frequency below 8MHz, the SCPG design is more energy efficient due to savings of combinational leakage power. To demonstrate the benefit of sub-clock power gating consider a clock frequency of 100kHz. Without sub-clock power gating the circuit dissipates an average of 28.72μW consuming 287pJ per operation (Table 3.2). With sub-clock power gating on the other hand the circuit dissipates 19.44μW consuming 194pJ per operation or if the high phase of the clock is modulated the circuit dissipates 8.13μW consuming 81.3pJ per operation representing an energy reduction of 3.5x. Up to now, the focus has been on reducing power at a given performance point, however there is an alternate utility for the proposed technique. There is increasing interest recently in utilising energy harvesting, a method of scavenging energy from the

environment, to power a device indefinitely [158]. In energy harvesting, the available energy can be thought of as infinite while the rate at which energy is scavenged limits the power available. Therefore, the idea of energy neutral operation has been proposed which enables a system to operate perpetually from an energy harvesting source provided the energy being delivered by a harvester is greater than or equal to the energy consumed [159]. This leads to the enforcement of a power budget. Vullers et al. report that a typical energy harvester power budget can be between 10 to 100s of μW [158] and so for demonstration purposes, consider a power budget of 30μW for the 16-bit parallel multiplier. Even with voltage scaling to 0.6V, the multiplier with no SCPG would need to operate at 100kHz and would consume 287pJ per operation (Table 3.2). With SCPG, the given power budget can be met at an operating frequency between 1-2MHz consuming less than 25.7pJ per operation (Table 3.2) and with a modulated high phase of clock, SCPG can achieve an operating frequency between 2-5MHz consuming less than 11.11pJ per operation (Table 3.2). This analysis shows that SCPG enables over 20x increase in clock frequency with over 26x improvement in energy efficiency within the same given power budget.

3.3.2 Case Study 2: ARM Cortex-M0

The second case study used is the ARM Cortex-M0 microprocessor and was chosen because of its low power design and relevance to low performance, energy constrained applications, which serves as a good example to demonstrate the gains of SCPG in a real world application setting. The ARM Cortex-M0 is provided for use in this thesis by the industrial project partner and a block diagram of the microprocessor can be seen in Fig. 3.14. The RTL allows the various elements of the microprocessor to be enabled or disabled during implementation depending on the constraints on power, performance and area. Therefore, the blocks in the microprocessor that were implemented in the Cortex-M0 test case are shaded in Fig. 3.14. Notice that the debug unit was omitted (Fig. 3.14) primarily to keep both area and power down in the test case. The Cortex-M0 processor core consists of a 3 stage pipeline and a 32-bit RISC architecture. The microprocessor does not have any cache and uses the Von Neumann architecture and so shares its memory for both instruction and data. The instruction set used in the microprocessor is the compact Thumb2^{TM} instruction set which is specifically designed to reduce code footprint and aid in reducing power. Not shown in Fig. 3.14 is the ability to choose between either a 32 cycle or single cycle multiplier hardware implementation in the Cortex-M0 Processor Core. The former offers lower power and area whereas the latter can be chosen for high performance. In the implementation used, a 32 cycle multiplier was chosen as the applications targeted by this work do not demand high performance. Further details of the Cortex-M0 microprocessor can be found in Appendix A.1.

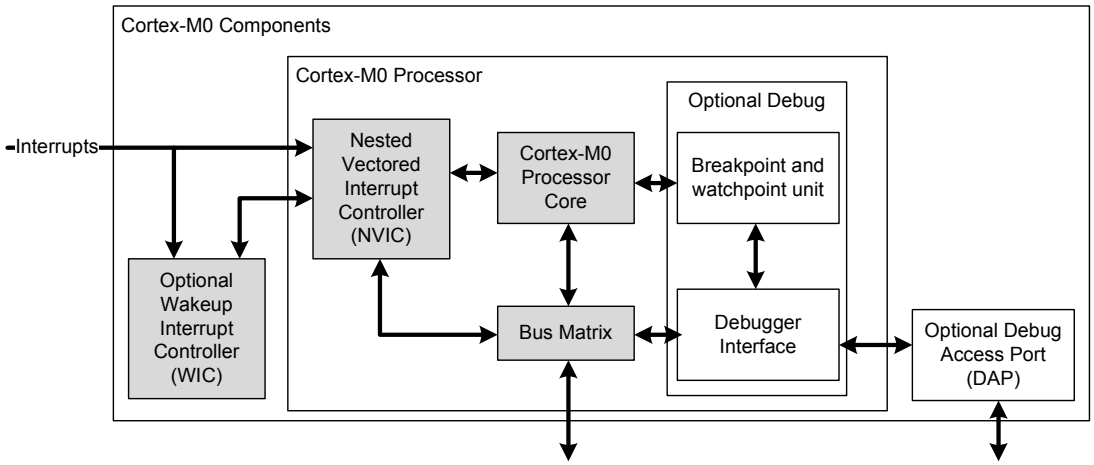


Figure 3.14: Complete ARM Cortex-M0 block diagram [160], implemented blocks highlighted

As the microprocessor provided by ARM is a synthesisable RTL core, it allowed the SCPG technique to be tested using the design flow presented in Section 3.2 (Fig. 3.6). Unlike the 16-bit multiplier, where the circuit is clearly defined into input registers, combinational hardware and output registers, the Cortex-M0 is not easily split into the SCPG architecture. This is because each of the Verilog modules that make up the Cortex-M0 contains intertwined combinational and sequential elements and is representative of most RTL circuit designs. Each of the four blocks of the Cortex-M0 microprocessor that were synthesised - WIC, NVIC, Core and Bus Matrix - contain a combination of both sequential and combinational logic. To demonstrate what happens when a circuit such as the Cortex-M0 is split using the proposed design flow, Fig. 3.6, an example is given using the 'Cortex-M0 Processor Core' sub-block and is shown in Fig. 3.15. Different shading patterns are used in Fig. 3.15 to help differentiate between which parts are mapped into the 'Comb. Logic' and 'Seq. Logic' power domains. Blocks such as the ALU, the multiplication unit and shift unit are purely combinational and therefore become part of the combinational domain. The register bank is purely sequential logic and maps directly into the sequential domain. However, units such as the 'core control' and 'instruction prefetch unit' are split such that their registers are mapped into the sequential domain and the combinational circuitry is mapped into the combinational domain. From the physical layout, it was found that the design consisted of 5795 combinational logic cells and 783 registers and combinational logic accounts for 69% of the circuit's leakage power. The inclusion of the SCPG technique in the physical implementation increased area by 6.6% due to the addition of power gates, isolation cells and power gating circuitry.

To obtain the power characteristics of the microprocessor at a specific performance point, the Dhrystone benchmark [161] was used. The benchmark is a synthetic benchmark first developed in 1984 and is representative of integer operations and memory accesses in modern processors. The benchmark is written in C and can be compiled using the ARM

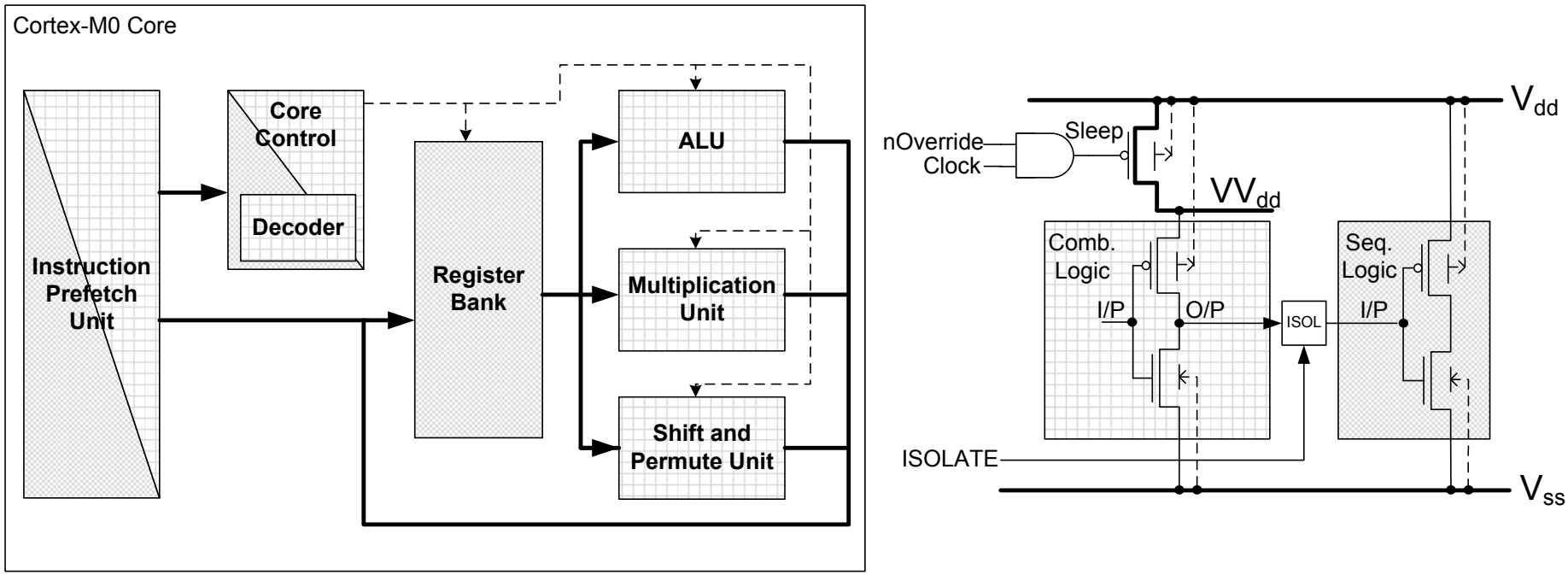


Figure 3.15: Example of how the Cortex-M0 Core block is mapped into the proposed SCPG technique

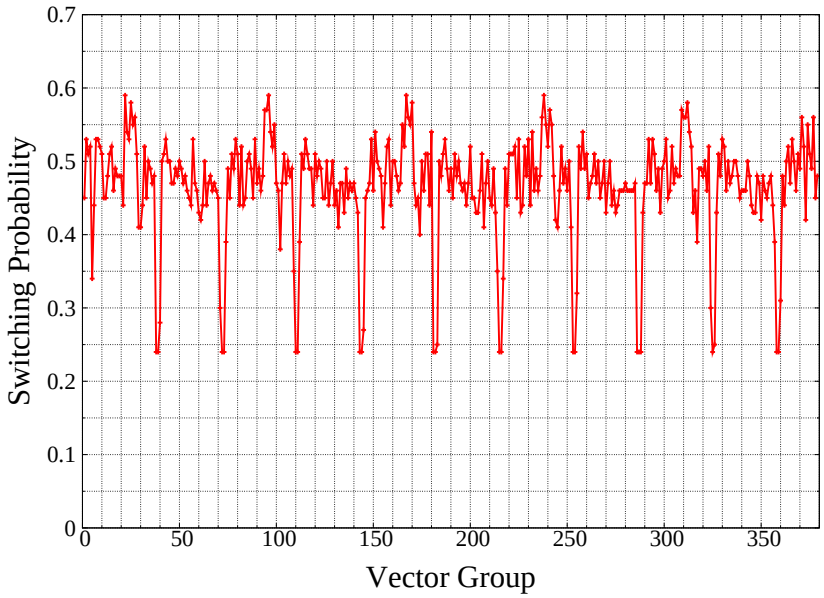


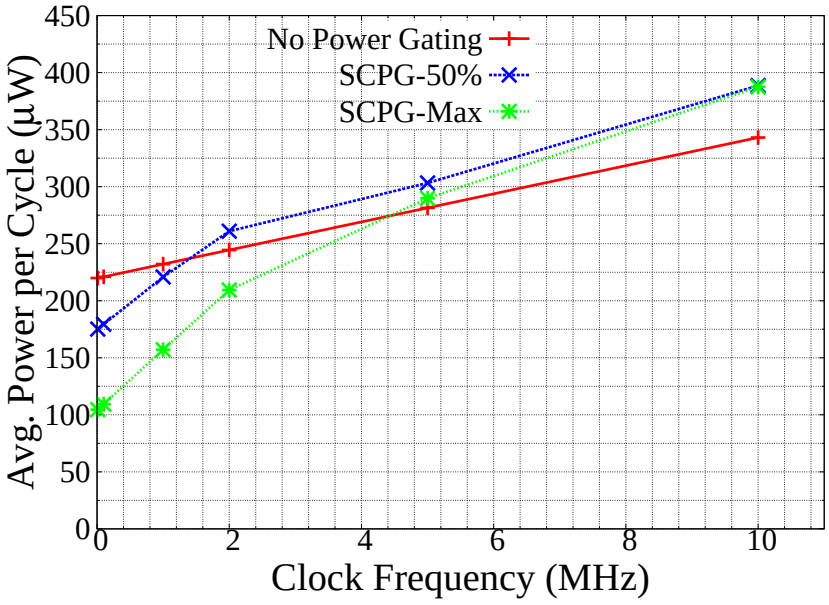
Figure 3.16: Switching probability of the Cortex-M0 for each set of 10 vectors from Dhrystone benchmark

C compiler for execution in the ARM Cortex-M0 and the total instruction code size of the benchmark is 1.2kB. Further details of the Dhrystone benchmark can be found in Appendix B.1. The experimental flow shown in Fig. 3.7 was followed to obtain the power and energy results, however, to keep HSpice simulation time reasonable (less than 24 hours), the steps presented next were followed in obtaining the simulation vectors. The Cortex-M0 netlist was simulated with the Dhrystone benchmark in Mentor Modelsim and a value change dump (VCD) file was created from the switching activity of the circuit. The complete benchmark (3700 vectors) was divided into groups of 10 vectors and each group's average switching activity was obtained with Synopsys Primetime-PX using the VCD file obtained, Fig. 3.16. Three groups of test vectors representing maximum, minimum and average switching activity were then extracted from these 370 groups. The test vectors representing these three test cases were then simulated in HSpice and the power numbers obtained were used to estimate the average power dissipation from a complete Dhrystone benchmark by process of a weighted average. To validate the accuracy of the weighted average estimation a simulation of the whole Dhrystone benchmark was performed at a clock frequency of 1MHz and compared with the estimated power value. The difference was found to be less than 2%.

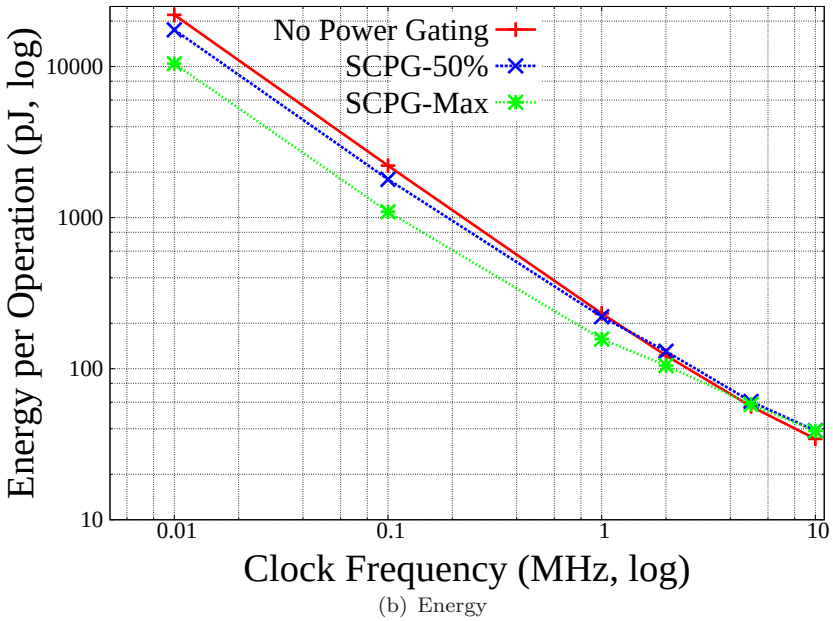
The simulation results for the Cortex-M0 design are reported in the same format as those for the 16-bit parallel multiplier in Section 3.3.1, and are given in Table 3.3. For the SCPG-Max case, it is found through simulation that the wake-up time and evaluation of the next state is 40ns. A similar trend to the multiplier test case is observed with sub-clock power gating achieving power savings over a range of frequencies, Fig. 3.17(a). However, it can be seen that lower savings are achieved compared to the multiplier at a given clock frequency (20.3% vs 33.9% at 10 kHz, Table 3.2). This difference can be

Table 3.3: Power and energy per operation of sub-clock power gated Cortex-M0, V_{dd}=0.6V

Clock (MHz)	No Power Gating		Proposed SCPG-50%			Proposed SCPG-Max		
	Power (uW)	Energy (pJ)	Power (uW)	Energy (pJ)	Saving (%)	Power (uW)	Energy (pJ)	Saving (%)
0.01	219.9	21990	175.19	17519	20.3	104.56	10456	52.4
0.1	221.0	2210	179.37	1793.7	18.8	109.31	1093	50.5
1	232.1	232.1	220.87	220.87	4.84	157.08	157	32.3
2	244.5	122.3	260.87	130.48	-6.73	209.43	105	14.3
5	281.4	56.3	303.21	60.64	-7.75	289.79	57.96	-2.98
10	343.1	34.3	388.63	38.86	-13.27	387.52	38.75	-12.9



(a) Power



(b) Energy

Figure 3.17: Cortex-M0, V_{dd}=0.6V

explained by the contrast in size between the two designs. Firstly, the combinational gates account for only 69% of the circuit's leakage power in the Cortex-M0 whereas the combinational logic accounted for 82% in the multiplier. Secondly, more isolation cells (874 vs 32 in the multiplier) are required which increases power by an average of 7% as opposed to just 5% in the multiplier. Thirdly, the increased size of the combinational domain in the Cortex-M0 (5795 gates vs 556 gates in the multiplier) increases the energy required to charge the virtual supply rail. As the power domain is restored back to an active state the combinational gates also experience a series of additional glitches resulting in an increased wake-up energy cost and is more significant in a larger design [162]. These two effects therefore increase the wake-up energy overhead of the power gating technique in a larger design. Analysis with HSpice found the recharge cost to be 20x higher compared to the multiplier circuit. The combination of these effects means a lower convergence point as can be seen in Fig. 3.17(a). Note that no power saving is achieved at 5MHz and above, whereas the multiplier design showed power saving up to 8MHz (Fig. 3.13(a)). At frequencies above 5MHz the Cortex-M0 could be switched over to no power gating with the *nOverride* signal. It was found that at the simulated clock frequencies where the override mode would be applicable, the active mode power was 7% greater than a circuit with no power gating circuitry due to addition of isolation cells.

Despite the increased power overhead associated with moving between the sleep and active mode in shut down power gating, the leakage power reduction achievable with sub-clock power gating enables better energy efficiency at a given performance point, Fig. 3.17(b). The benefit of the technique can be demonstrated with an example. Consider an operating frequency of 100kHz (Table 3.3), with no power gating the circuit would dissipate 221μW consuming 2210pJ per operation. With sub-clock power gating though, the circuit would dissipate 179.37μW with a 50% duty cycle and 109.31μW with the high phase of the clock modulated, consuming 1793pJ and 1093pJ per operation respectively. This represents an improvement in energy efficiency of 2x enabled by the application of sub-clock power gating. The benefit of the proposed technique can also be demonstrated with a power budget as was shown with the 16-bit multiplier. For example, considering a power budget of 220μW, a Cortex-M0 design without SCPG would need to operate at approximately 100kHz consuming 2210pJ per operation (Table 3.3). On the other hand, an SCPG design using 50% clock duty cycle operate at 1MHz consuming 220.87pJ per operation, while operating at maximum duty cycle consumes 105pJ per operation at an operating frequency of 2MHz (Table 3.3). This represents over 21x improvement in energy efficiency achieved by operating at 20x higher clock frequency.

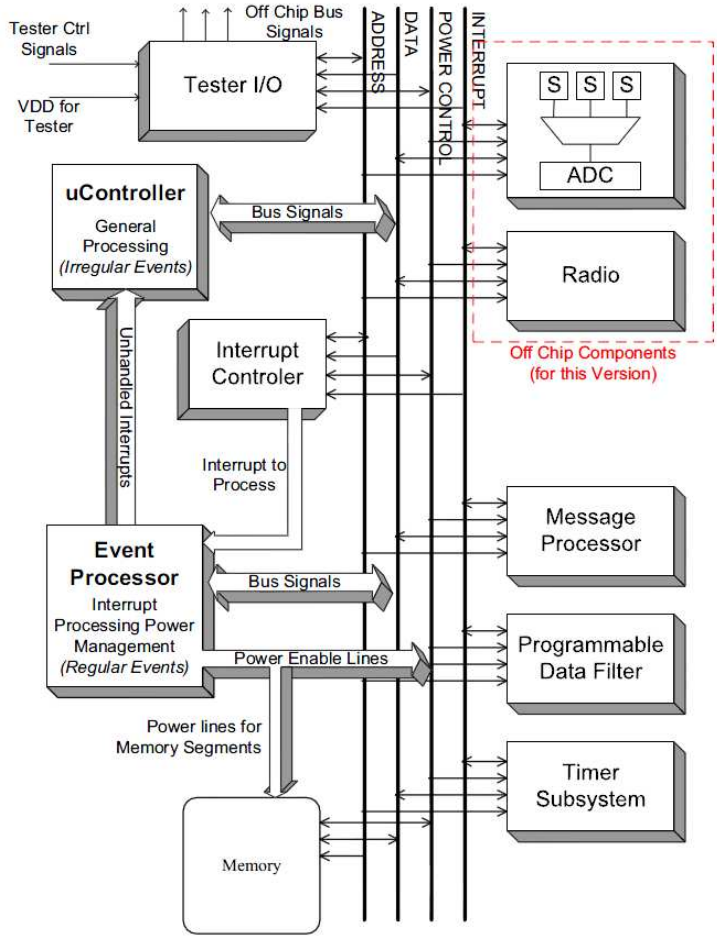


Figure 3.18: Architecture of the Event Processor [63]

3.3.3 Case Study 3: Event Processor

The Event Processor [63] was chosen as a third test case because of its ultra low power design and to demonstrate the SCPG technique in an ASIC processor designed specifically for Wireless Sensor Networks. A block diagram of the entire system is shown in Fig. 3.18 and is a stripped down architecture that utilises hardware accelerators to perform each of the most common tasks done on a wireless sensor node: data acquisition, data processing (filtering), message processing and message transmission. A typical sensing task would involve reading data from the off-chip ADC, filtering the data in the data filter and preparing the data in the message processor before sending it to the off-chip radio for transmission [63]. To exploit the regularity of operation often found in a wireless sensor node, a timer subsystem is included and triggers periodic interrupts. Between interrupts the hardware accelerators are powered down to save power and are only powered up when they are needed. A general purpose microcontroller is included in the system but remains off as long as the current task can be handled by the hardware accelerators. The control of the whole system comes from a central event processing state machine which remains always-on and is used to handle interrupts, enable/disable

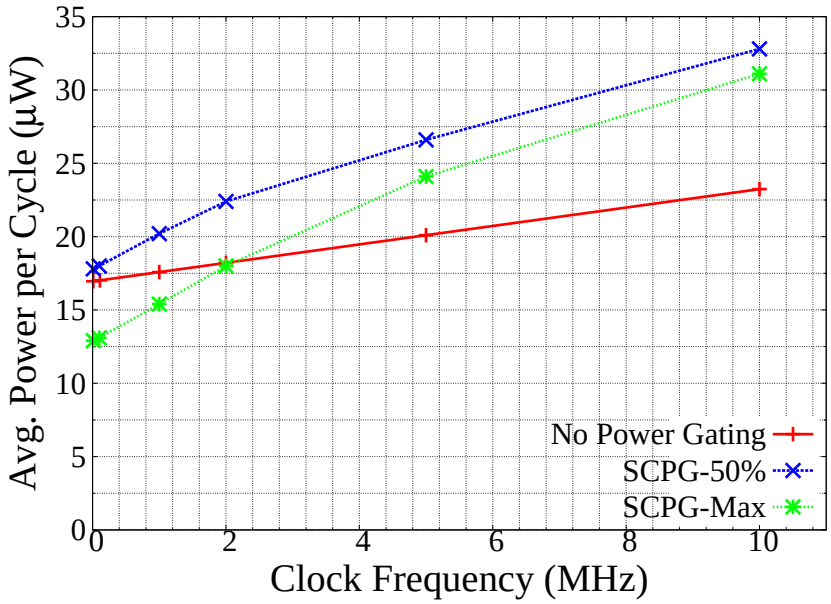
Table 3.4: Power and energy per operation of sub-clock power gated Event Processor, V_{dd}=0.6V

Clock (MHz)	No Power Gating		Proposed SCPG-50%			Proposed SCPG-Max		
	Power (uW)	Energy (pJ)	Power (uW)	Energy (pJ)	Saving (%)	Power (uW)	Energy (pJ)	Saving (%)
0.01	16.96	1696	17.8	1780	-5.0	12.9	1290	23.9
0.1	17.02	170.2	18.0	180	-5.9	13.1	131	22.8
1	17.58	17.58	20.2	20.2	-15.1	15.4	15.4	12.2
2	18.21	9.11	22.4	11.2	-23.2	18.0	9.0	1.32
5	20.1	4.02	26.6	5.32	-32.2	24.1	4.82	-20.05
10	23.24	2.32	32.8	3.28	-41.3	31.1	3.11	-33.86

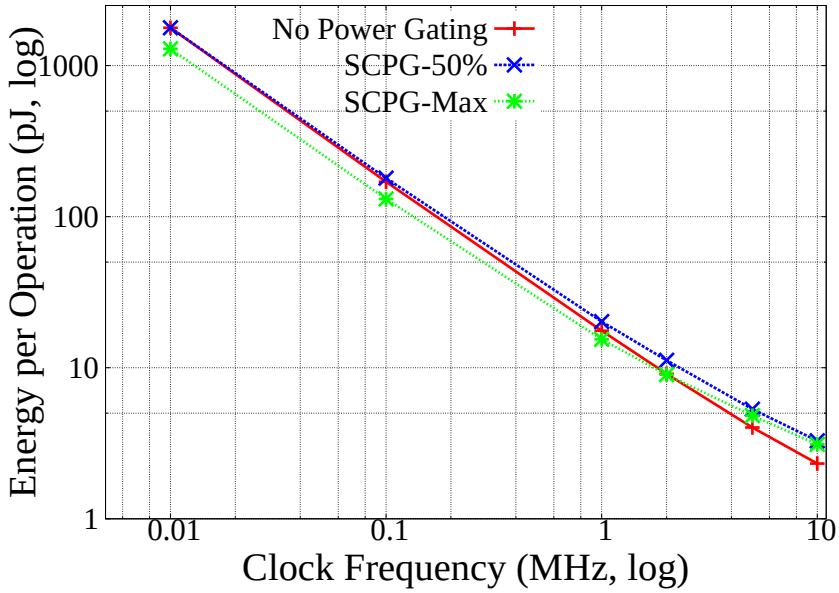
the power to each of the hardware accelerator blocks and move data about the system. The entire system is designed to be operated at a low clock frequency of 100kHz and is done so to maintain a low average power [63]. Due to the low frequency operation and the always-on nature of the central event processor state machine, it is a prime candidate for the application of the sub-clock power gating technique.

Using the information available from [63], the Event processor has been prototyped and synthesised for use as a test case. The sub-clock power gating technique was then embedded using the design flow shown in Fig. 3.6. The event processor consists of 441 combinational logic cells and 82 registers and combinational logic accounts for 64% of the total leakage. The additional power gating circuitry increases total area by 21% and is a greater cost than in the multiplier and Cortex-M0 test case due to the need for a larger number of isolation gates (111 Vs 32 in the multiplier) and the relatively small size of the design. The processor is simulated during instruction execution for a typical sensing task comprising of data acquisition, processing and transmission and power numbers are recorded using the experimental flow from Fig. 3.7. Table 3.4 gives the power and energy values for the Event Processor in active mode with no power gating, with sub-clock power gating and with sub-clock power gating using a modulated duty-cycle. The results are presented in the same format as with the multiplier and the Cortex-M0 results.

It can be observed from the results in Table 3.4 and trends depicted in Fig. 3.19(a) that using a 50% duty cycle does not result in power savings over the same circuit with no power gating, and if power saving is to be achieved then a modulated clock must be used. This is due to three main observations. Firstly, the combinational logic accounts for only 64% of the total leakage in the original circuit. Secondly, the circuit requires a large number of isolation cells (111 in total) accounting for a 21% increase in the total cell count in a sub-clock power gating design, contributing 34% additional leakage and dynamic power overhead. Thirdly, the wake-up energy cost of moving from the sleep mode to active mode also counts against potential energy savings. Nevertheless, if the high phase of the clock is extended power and energy savings are achieved up to 2MHz. The nominal operating frequency of the Event Processor is 100kHz, and it is



(a) Power



(b) Energy

Figure 3.19: Event processor state machine, V_{dd}=0.6V

observed that at this clock frequency the sub-clock power gating technique utilising a maximised duty cycle enables 1.3x improvement in energy efficiency. Alternatively, if the power consumption of the circuit with no power gating at 100kHz is considered to be the power budget of the state machine then it is possible to increase the clock frequency of the Event Processor to over 1MHz, achieving a 10x improvement in performance and 11.1x improvement in energy efficiency.

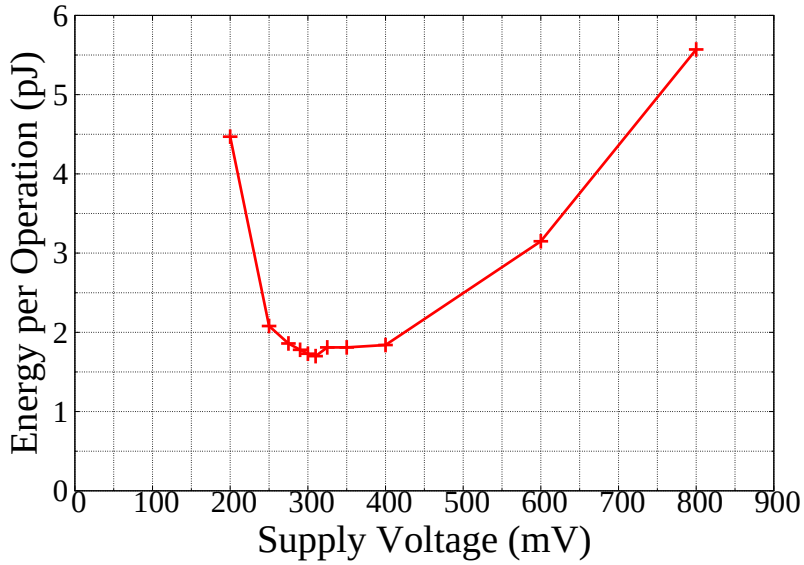


Figure 3.20: Supply voltage Vs energy per operation, 16-bit parallel multiplier

3.4 Comparative Analysis with Subthreshold

Subthreshold design technique enables realization of minimum energy computation by aggressively lowering the supply voltage until a minimum energy point is found where the energy consumed to dynamic switching per clock cycle is equivalent to the leakage energy consumed per clock cycle [36, 37]. The principles of the technique were given in Chapter 1, Section 1.4.2. The work proposed in this chapter can be considered as an alternate, orthogonal approach to the subthreshold technique to achieve low performance operation whilst maximising energy efficiency. This section investigates the subthreshold operation of two of the test circuits, namely the 16-bit parallel multiplier and the ARM Cortex-M0, with the aim of establishing the performance of sub-clock power gating relative to the subthreshold technique. The two test circuits were implemented using the same 90nm technology library used in Section 3.3 however, to be representative of the constraints typically enforced on gate libraries used when designing subthreshold circuits, gates with transistor stacks greater than 3 were banned from synthesis [50, 52]. The experimental flow in Fig. 3.7 was followed to obtain the power results and the HSpice simulation stage was conducted for a range of supply voltages. For each supply voltage the circuit was first simulated once at a low clock frequency to obtain the critical path length of the circuit and then again at the corresponding maximum operating frequency. The netlists were full parasitic netlists but the transistor models were kept ideal and effects of process variation such as threshold voltage, supply voltage and channel length variation, which can affect both the reliability and performance of a subthreshold circuit [36, 37, 53] were omitted resulting in ideal subthreshold results. However, due to the up to 24 hour simulation time required with HSpice this method kept turnaround time reasonable.

Fig. 3.20 shows the energy per operation against supply voltage of the 16-bit multiplier when using subthreshold design. In Table 3.5, the sub-clock power gating technique using

Table 3.5: Comparison of sub-clock power gating relative to subthreshold operation performance points, 16bit Multiplier

Comparison Point	Subthreshold			Proposed SCPG-Max		
	Freq. (MHz)	Power (uW)	Energy (pJ)	Freq. (MHz)	Power (uW)	Energy (pJ)
Freq @ Min. Vdd	1.7	7.6	4.47	2	22.22	11.11
Freq @ Min. Energy	10	17	1.7	10	56.4	5.64
Power @ Min. Energy	10	17	1.7	1-2	14.18	14.18

a maximised duty cycle has been compared against the subthreshold technique at three different power/performance points. The first compares SCPG against subthreshold operation at the same clock frequency achievable at the minimum operational V_{dd} of the subthreshold circuit. The minimum voltage the subthreshold circuit was still functional was at 200mV where the maximum attainable frequency was 1.7MHz consuming 4.47pJ per operation. In a sub-clock based design at a clock frequency of 2MHz, the circuit consumes 2.5x more energy equal to 11.11pJ per operation, Table 3.5. The second comparison made is at the same clock frequency attainable at the minimum energy point of the subthreshold circuit. It was found from HSpice simulation that the minimum energy point in the subthreshold circuit of 1.7pJ per operation is obtained at a supply voltage of 310mV corresponding to an operating frequency of approximately 10MHz. At 10MHz the sub-clock power gated 16-bit multiplier consumes 5.64pJ per operation, representing a 3.3x increase in energy. The final comparison made assumes the power consumption of 17μW at the minimum energy point as the power budget of the circuit. For a power budget of 17μW using the SCPG multiplier this dictates operation between 1-2MHz consuming 14.18pJ per operation; a 5x reduction in performance and a 6.5x increase in energy.

Fig. 3.21 shows the energy per operation at different supply voltages for the Cortex-M0 using subthreshold operation. The same trend is observed here as in the case of the 16-bit multiplier with an inflection at the minimum energy point. Note, however, that the increased density of logic in this circuit pushes the minimum energy point towards a higher supply voltage. This is because the leakage energy of the increased number of gates dominates at a higher clock frequency and is a common observation in larger subthreshold circuits due to more circuitry [37]. The same three power/performance comparisons made with the subthreshold and SCPG multiplier circuits have been made with the Cortex-M0 and are shown in Table 3.6. The minimum operating voltage of the subthreshold circuit was located at 200mV at a clock frequency of 1.4MHz consuming 31.15pJ per operation. This is a 3.3x lower energy consumption than using a sub-clock based Cortex-M0 at 2MHz as shown in Table 3.6. For the second comparison, simulation locates the minimum energy point of the subthreshold circuit at a supply voltage of 450mV, corresponding to an operating frequency of 24MHz, consuming 12.01pJ per operation or average power consumption of 288.24μW. In this case, a direct comparison cannot be made as a sub-clock power gated Cortex-M0 cannot be operated at a clock

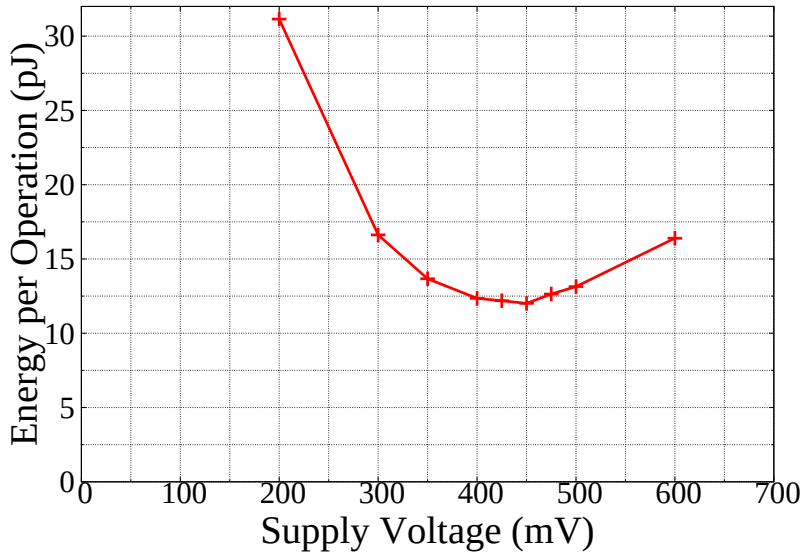


Figure 3.21: Supply voltage Vs energy per operation, Cortex-M0

Table 3.6: Comparison of sub-clock power gating relative to subthreshold operation performance points, Cortex-M0

Comparison Point	Subthreshold			Proposed SCPG-Max		
	Freq. (MHz)	Power (uW)	Energy (pJ)	Freq. (MHz)	Power (uW)	Energy (pJ)
Freq @ Min. Vdd	1.4	43.61	31.15	2	22.22	11.11
Freq @ Min. Energy	24	288.24	12.01	-	-	-
Power @ Min. Energy	24	288.24	12.01	5	289.79	57.96

frequency of 24MHz. However, the third comparison which assumes the same power budget of the subthreshold circuit's minimum energy point shows the sub-clock power gated Cortex-M0 can be operated at 5MHz consuming 57.96pJ per operation, a 5x reduction in performance and 4.8x increase in energy, as shown in Table 3.6.

This analysis from the ideal subthreshold test cases shows, as expected, that the subthreshold technique offers better energy efficiency than sub-clock power gating since it enables minimum energy computation. As mentioned in Chapter 2, Section 2.5 though, the subthreshold technique has a number of design challenges associated with it due to the ultra-low operating voltages. The circuit is more sensitive to process variations such as variations in threshold and supply voltages [36, 37] requiring careful design considerations including custom or modified gate libraries [50, 52] and custom tools for characterisation of gate library cells and timing analysis [51–53]. The sub-clock power gating technique on the other hand is utilised sufficiently above the threshold voltage maintaining greater stability with process variations and is fully compatible with a standard power gating design flow, using commercially available standard cell libraries and EDA tools with little additional design effort. In addition to the design challenges of using subthreshold regime, the circuits are also optimised for operation at ultra low supply voltages and low operating frequencies only. Sub-clock power gating conversely provides

a performance/power trade-off. The performance of the circuit can easily be changed between various clock frequencies whilst minimising leakage energy. Additionally, the *nOverride* signal enables the circuit to achieve normal timing which can be particularly useful in devices such as the MSP430 which utilises clock frequencies in the range of 32kHz to 8MHz.

3.5 Concluding Remarks

Leakage power is a major concern in current and future nanometer technologies, and its reduction is key to improving the energy efficiency of integrated circuits. In applications that demand low to moderate performance, leakage becomes a major contributor of active mode power due to idle time of combinational logic that occurs within the clock period. This chapter has addressed this problem with power gating which can be used to cut leakage power dissipation within the clock period and improve overall energy efficiency of an integrated circuit operating at low clock frequencies.

This is the first investigation into employing power gating within the clock period to minimise leakage power during the active mode and it is shown through simulation of a 16-bit multiplier, an ARM Cortex-M0 microprocessor and the Event Processor [63], using a 90nm technology library, that considerable savings are achievable with the sub-clock power gating technique. Power gating is used only on the combinational logic while the sequential logic is kept always-on during the active mode. The control to the power gates is provided by the clock signal by cutting the power when the clock is high and enabling power when clock is low. It is shown that taking control of the duty cycle and extending the high phase of the clock, leakage power saving can be maximised by capitalising on all the combinational logic idle time within the clock period. The sub-clock power gating technique is fully compatible with a standard power gating design flow using commercially available gate libraries and EDA tools.

Chapter 4

Symmetric Virtual Rail Clamping for Sub-Clock Power Gating

In Chapter 3 a new technique for power gating within the clock period was introduced to reduce active mode leakage power in digital circuits operating at low clock frequencies. The technique capitalises on the idle time of combinational logic within the clock period to power gate it, and it was shown through simulation that significant energy savings could be achieved in three test cases. In the simulation results in Chapter 3, Section 3.3, it was observed that as the clock period shortened, and hence also the combinational logic idle time shortened, the power savings with the sub-clock power gating technique also decreased and the cost of moving between the sleep and active mode began to dominate, Figures 3.13(a), 3.17(a) and 3.19(a). A point was then reached where the energy saved using sub-clock power gating equalled the energy dissipated from moving between the power gated and active mode, after which, the sub-clock power gating technique dissipated more power than it saved. This chapter investigates the effectiveness of a new power gating technique called symmetric virtual rail clamping with the intention of improving the energy efficiency and extending the applicable range of the sub-clock power gating technique. The proposed power gating technique is incorporated with sub-clock power gating in an ARM Cortex-M0 and validated experimentally with a fabricated test chip.

This chapter is organised as follows. Section 4.1 compares the transition energy costs of three different power gating techniques including the symmetric virtual rail clamping technique introduced in this chapter. Section 4.2 shows how the sub-clock power gating technique is modified to utilise symmetric virtual rail clamping. The design and implementation of a Cortex-M0 with sub-clock power gating and symmetric virtual rail clamping is outlined in Section 4.3 and experimental results are given in Section 4.4. Finally concluding remarks are presented in Section 4.5.

4.1 Wake-Up Energy Cost

In shut down power gating the energy overhead of moving between power modes is dominated by the recharging of the virtual supply rail and glitching of internal signals resulting from the re-evaluation of logic cones [3, 162]. These overheads stem from the supply rail being fully discharged and subsequent loss of valid logic gate outputs when the circuit is powered down. The recharging of the virtual rail is a discernable energy overhead, however, it has recently been reported that spurious glitches from logic re-evaluation during the transition from sleep to active mode in power gating can account for 75.71% of the wake-up energy [162]. This signal glitching is unlike glitching experienced during normal operation which can be caused by unbalanced paths [19] and is instead a side-effect from the charging of the virtual rail. This can be explained with a simple example. Consider an XOR gate, if at power-down both inputs were logic 1, then the capacitive loads associated with these inputs need to re-charge at power-up. As the virtual rail is restored, if one input resolves to logic 1 before the other, due to its size or physical placement, the XOR gate momentarily takes a logic 1 output resulting in unnecessary energy dissipation. Furthermore, this glitch can propagate down a logic cone increasing the energy cost. In traditional applications of shut down power gating the wake-up energy overhead associated with moving between power modes (E_{ovhd}) is generally negligible as the length of the idle period is typically very long [3] and therefore energy saved (E_{sav}) $\gg E_{ovhd}$. As the length of the idle period becomes shorter, the proportion of E_{ovhd} to E_{sav} increases, and therefore the magnitude of E_{ovhd} becomes an important element of the total energy savings. This was observed in the sub-clock power gating technique introduced in Chapter 3 and it was shown that as clock frequency increased, the energy overhead of the power gating technique quickly overtook the energy savings.

4.1.1 Power Gating Techniques

The wake-up energy cost is now considered for three different power gating techniques: traditional shut down power gating [3], virtual rail clamping [103] and symmetric virtual rail clamping, which is introduced in this chapter. Virtual rail clamping (VRC) was proposed as a way to maintain a voltage across the power gated logic to retain register state [103] but maintaining a voltage across the power gated logic has been proven to reduce the recharge and glitching cost associated with power gating [105]. An example of a circuit using virtual rail clamping was briefly introduced in Chapter 2, Section 2.2, but further details are given here. An inverter employing virtual rail clamping is demonstrated in Fig. 4.1(a) where a pair of NMOS and PMOS transistors are used at the head of the power gated logic to enable reduction in the supply voltage. In this circuit, the PMOS transistor, marked as *Sleep*, is a conventional power gating transistor and the NMOS transistor, marked as *Ret*, is used to clamp the virtual rail. When *Sleep*

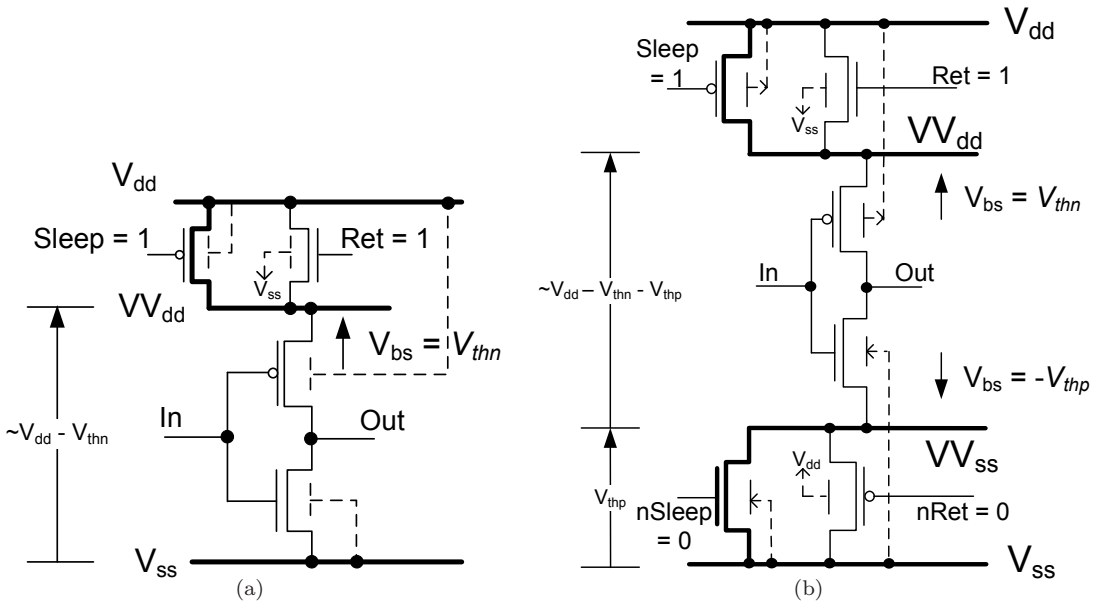


Figure 4.1: An inverter with (a) single rail clamping [103] (b) symmetric virtual rail clamping

and *Ret* are logic 1 the VV_{dd} rail is reduced to $V_{dd} - V_{thn}$ where V_{thn} is the threshold voltage of the NMOS transistor. In this mode, the inverter's PMOS transistor can additionally benefit from reverse body biasing (RBB) by connecting its body to V_{dd}, as shown. The V_{thn} potential across V_{bs} increases the threshold voltage of the PMOS transistor due to the body effect (Chapter 1, Section 1.1.2) and reduces subthreshold leakage currents [118]. The reason the virtual rail is clamped to $V_{dd} - V_{thn}$ can be explained as follows. An NMOS transistor's source is always considered to be the node with lowest potential [2] and so in this case it would be VV_{dd} rail. To start with, assume the VV_{dd} rail is fully charged to V_{dd}. When the *Sleep* and *Ret* signals are switched to logic 1 (V_{dd}) the VV_{dd} begins to discharge and the NMOS transistor has the condition $V_{gs} < V_{thn}$ and is therefore cut-off. When the VV_{dd} rail falls below $V_{dd} - V_{thn}$ the NMOS transistor is no longer cut-off, $V_{gs} > V_{thn}$ and is in the saturated mode of operation, $V_{ds} > (V_{gs} - V_{thn})$, pulling the VV_{dd} back up. When the virtual rail rises to the value $V_{dd} - V_{thn}$ the NMOS transistor is cut-off again as V_{gs} falls below V_{thn}. The VV_{dd} supply rail is thereby continuously clamped to $V_{dd} - V_{thn}$. Using MOS transistors to implement the virtual rail clamping also has the added advantage of being able to achieve shut down power gating by forcing *Sleep* to logic 1 and *Ret* to logic 0.

Virtual rail clamping enables a single threshold voltage drop reduction across the power gated logic. To maximise leakage power savings of the power gated combinational logic in sub-clock power gating, it is desirable to reduce the clamped voltage by more than a single threshold voltage, since subthreshold leakage is proportional to V_{ds}, Eqn. 1.9, and power is the product of voltage and current [2], Eqn. 1.1. Multiple NMOS transistors placed in series at the head of the power gated logic can be used to realise this, but in

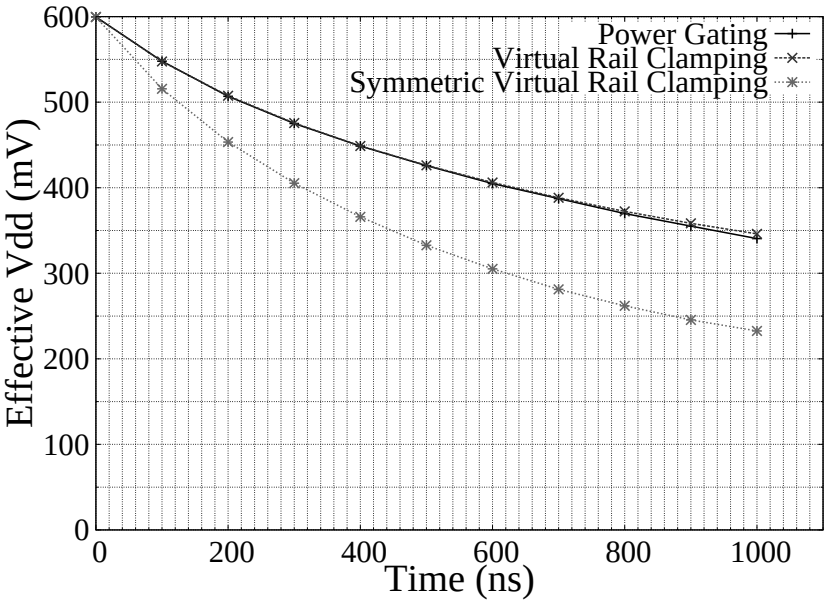


Figure 4.2: V_{dd} reduction against time for three power gating techniques

this chapter it is proposed to mirror the V_{dd} virtual rail clamping on the V_{ss} supply rail instead. This proposed symmetric virtual rail clamping (SVRC) technique is shown in Fig. 4.1(b), where there is now a pair of NMOS and PMOS transistors at the head and foot of the example inverter circuit. When *Sleep* and *Ret* are logic 1 (and thus *nSleep* and *nRet* are logic 0) the VV_{dd} is clamped to $V_{dd} - V_{thn}$ and the VV_{ss} is clamped to $V_{ss} + V_{thp}$. The result is a much more aggressive reduction in voltage across the power gated logic but also has three advantages over single rail clamping [103].

Firstly, when the sleep mode is activated, the charge that is stored in the VV_{dd} supply rail is recycled to charge up the VV_{ss} supply rail [163] and similarly when the sleep mode is exited the only charge that needs to be returned is to the VV_{dd} supply rail. This means that greater supply reduction is achievable with symmetric clamping but at an equivalent wake-up energy cost to single rail clamping. The recycling of the charge also improves the speed at which the effective V_{dd} across the logic ($VV_{dd} - VV_{ss}$) is lowered. Using a ring oscillator test circuit of 101 inverting stages the effective V_{dd} reduction of shut down power gating, virtual rail clamping and symmetric virtual rail clamping when put into the sleep mode has been measured over 1μs and is plotted in Fig. 4.2. A 101 stage ring oscillator is chosen because it provides a good representation of deep logic paths in a digital circuit with high levels of inversion. As can be seen in Fig. 4.2, virtual rail clamping and shut down power gating follow the same discharge rate with virtual rail clamping slowing down as it approaches its clamped VV_{dd} value. Conversely, symmetric virtual rail clamping achieves almost 30% greater reduction in the effective V_{dd} in the same time frame due to the simultaneous fall and rise of VV_{dd} and VV_{ss} enabled by charge recycling. Over very short idle periods, such as in sub-clock power gating, this is advantageous as the lowest leakage state can be entered faster.

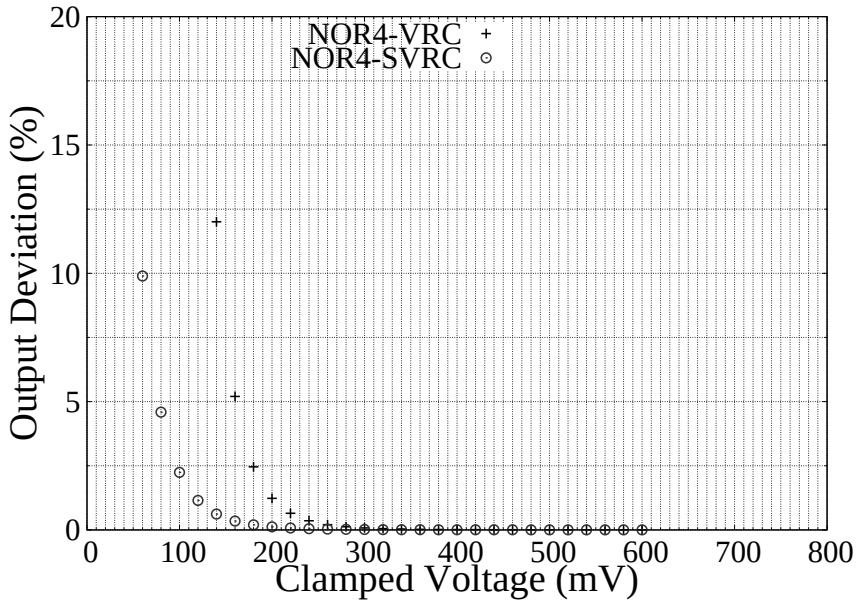


Figure 4.3: Output deviation of 4-input NOR gate from Synopsys 90nm library with virtual rail clamping (VRC) and symmetric virtual rail clamping (SVRC) with reduced supply voltage

The second advantage with symmetric clamping is that it is possible to connect not only the body of the PMOS transistor in the inverter to V_{dd} but also the body of the NMOS transistor to V_{ss}. The resulting V_{th} reverse body bias increases the threshold voltage of all transistors in the digital circuit to provide further leakage reduction.

Thirdly, the symmetrical RBB also ensures better equality of NMOS and PMOS drive strength degradation, as strong RBB of the PMOS transistors from single rail clamping (Fig. 4.1(a)) can result in significant loss of logic 1 drive. This can cause gate output logic levels to be lost, causing unwanted signal glitching at wake-up [2]. This is because, when a gate's supply voltage is lowered and threshold voltage is increased, the I_{on} current of the transistors degrades and can become comparable with the I_{off} current resulting in a battle between the on/off transistors to maintain the correct output voltage [36, 37]. It is known that NOR and NAND gates suffer the greatest effects of this because of the large number of parallel transistors in the logic gates [37]. For example a 4 input NOR gate with a logic 1 output has 4 parallel off NMOS transistors with which the PMOS transistors must compete. To observe the advantage of equal drive strength degradation in symmetric virtual rail clamping over virtual rail clamping, a NOR4 gate from the Synopsys 90nm gate library used in Chapter 3 was simulated. Fig. 4.3 shows the percentage deviation of a logic 1 output with respect to the effective clamped voltage in both cases. As can be seen, the reverse body biasing used on only the PMOS transistors in virtual rail clamping causes the NOR4 gate output to begin deviating from the expected output at supply voltages below 300mV. Conversely, the symmetric virtual rail clamping enables the NOR4 gate to hold its output for supply voltages down to 200mV. The use of MOSFET transistors in symmetric virtual rail

Table 4.1: Ring oscillator wake-up energy, leakage saving and wake-up time

	Wake-up Energy (fJ)	Leakage Saving (%)	Wake-up time (ns)
Shut Down Power Gating [3]	223.0	87.3	12
Virtual Rail Clamping [103]	76.82	75.4	6.5
Proposed Symmetric Virtual Rail Clamping	74.49	78.4	6.5

clamping means independent control of *Sleep*, *Ret*, *nSleep* and *nRet* in Fig. 4.1(b) can enable shut down power gating.

4.1.2 Power Gating Techniques Comparison

To quantify the wake-up energy cost of shut down power gating, single rail clamping and the proposed symmetric virtual rail clamping, the three approaches have been implemented on a 101 stage ring oscillator using high drive strength inverters from the Synopsys 90nm library. A 101 stage ring oscillator was chosen because it provides a representation of high levels of inversion and glitching from logic re-evaluation when returning from the sleep mode for a logic path in a processor. In line with the results presented in Fig. 4.3, a 300mV clamped voltage was chosen for virtual rail clamping achieved with a high threshold voltage NMOS clamping transistor and a 200mV clamped voltage was chosen for symmetric virtual rail clamping achieved with low threshold voltage PMOS and NMOS clamping transistors. All three circuits were simulated with 0.6V V_{dd} using HSpice to obtain the results presented. Table 4.1 shows the wake-up energy, sleep mode leakage current saving and wake-up time for the three power gating approaches. As can be seen, the proposed power gating with symmetric virtual rail clamping has the lowest wake-up energy and is 3x lower than shut down power gating. This is because the voltage maintained across the power gated logic from equal reduction in both VV_{dd} and VV_{ss} supply rails eliminates signal glitching from the logic re-evaluation present in shut down power gating. Furthermore, despite using a lower clamped voltage than virtual rail clamping, the charge recycled in symmetric virtual rail clamping results in lower wake-up energy cost. As expected, Table 4.1 shows that standby leakage saving is highest in shut down power gating because the power supply is fully disconnected achieving greater reduction in effective V_{dd}. However, the proposed symmetric virtual rail clamping has a greater standby leakage saving compared to single rail clamping which can be attributed to exploitation of reverse body biasing of both NMOS and PMOS transistors and the lower achievable effective V_{dd}. Finally, Table 4.1 shows that the proposed symmetric virtual rail clamping has a shorter wake-up time compared to shut down power gating, which permits a longer power gated period and is particularly useful over short power gated periods.

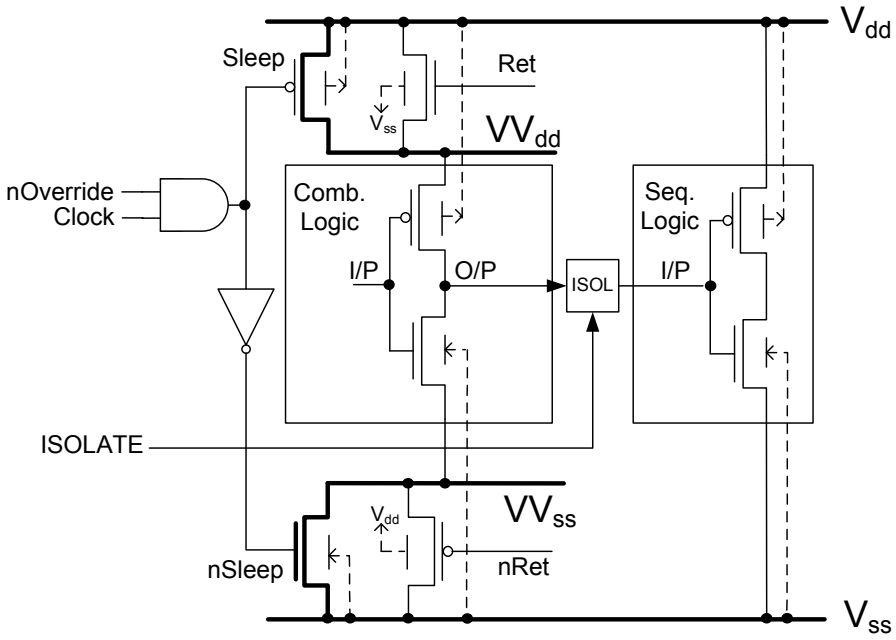


Figure 4.4: Sub-clock power gating technique with symmetric virtual rail clamping

4.2 Sub-Clock Power Gating with Symmetric Virtual Rail Clamping

The architecture of a digital circuit using sub-clock power gating was introduced in Chapter 3, Section 3.2, Fig. 3.3. There were three distinctive parts to the circuit. Firstly, the circuit was split into two blocks: a combinational block which could be power gated and a sequential block which remained always-on. Secondly, the combinational block was power gated using traditional shut down power gating and was achieved with the use of a PMOS header transistor. Lastly the interface from the combinational logic to the sequential logic went through an 'ISOL' block which was used to isolate the outputs of the combinational domain. To take advantage of the low wake-up energy cost associated with symmetric virtual rail clamping introduced in Section 4.1, the circuit must be modified as shown in Fig. 4.4. Instead of using a single PMOS header transistor as in Fig. 3.3, the combinational logic is now power gated with pairs of PMOS and NMOS transistors at the head and foot of the block. The combinational and sequential logic split of the digital circuit remains the same to avoid the need for retention registers, but also enables a much more aggressive voltage reduction to be used for the combinational logic with symmetric virtual rail clamping, whilst simultaneously ensuring the saved state remains intact during active operation. The isolation 'ISOL' between the combinational logic and sequential logic is still required to stop the output signals of the combinational domain causing short-circuit currents in the sequential logic when the combinational domain is in sleep mode.

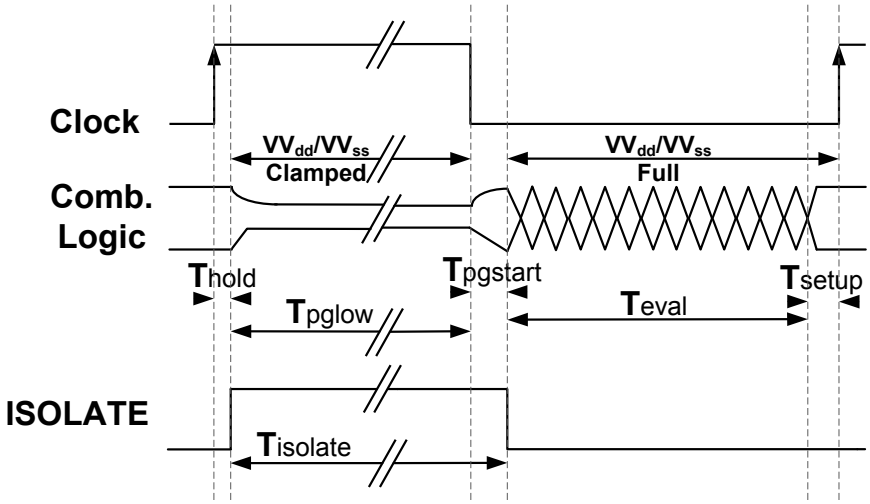


Figure 4.5: Combinational logic timing of sub-clock power gating technique with symmetric virtual rail clamping

The addition of the NMOS transistor opposite the PMOS header transistor and the pair of transistors at the foot of the logic introduces small additional control requirements. The architecture in Fig. 3.3 controlled the PMOS header power gating transistor with the clock signal and this meant the combinational logic was power gated when the clock was high and was enabled when the clock was low. To achieve the same behaviour using symmetric virtual rail clamping, the NMOS and PMOS transistors at the head of the combinational logic (Sleep & Ret) use the normal clock signal whilst the NMOS and PMOS transistors at the foot (nSleep & nRet) use the inverse of the clock signal. Therefore, when the clock is high, the combinational logic is clamped by two V_{th} using symmetric virtual rail clamping and when the clock is low it is restored to V_{dd}. The *nOverride* signal included provides a method to disable the SCPG technique. The circuit to control the isolation in the 'ISOL' block remains the same as in Fig. 3.4 introduced in Chapter 3, Section 3.2. The timing diagram of the combinational logic in the sub-clock power gating mode of operation with symmetric virtual rail clamping is largely unchanged from the timing diagram with shut down power gating, Fig. 3.5, and is shown in Fig. 4.5. After the positive edge of the clock, instead of the combinational logic voltage collapsing down to zero as in Fig. 3.5, the voltage to the combinational logic is instead clamped on both the V_{dd} and V_{ss} resulting in a supply voltage of less than V_{th}. The isolation and evaluation of the next state is unchanged when compared to Fig. 3.5 and duty cycle can still be manipulated to maximise power saving with the sub-clock power gating technique as will be shown in Section 4.4.

4.3 Implementation

The Cortex-M0 used for the simulation results of sub-clock power gating (SCPG) in Chapter 3 is a widely licensed commercial microprocessor developed by ARM for low

Table 4.2: Control signals to power gates and corresponding mode of operation

Off	**nOverride**	**Drowsy**	**Mode of operation**	**State**
1	X	X	Shut Down	Off/lost
0	0	X	Fully On	Clocked
0	1	1	SCPG with Symmetric Clamping	Retained
0	1	0	SCPG with Shut Down	Retained

performance, energy constrained applications, and therefore serves as an ideal test case for fabrication and analysis of the sub-clock power gating and symmetric virtual rail clamping techniques in silicon. This section discusses the power intent of the fabricated Cortex-M0 with sub-clock power gating, the additional steps in the RTL to GDSII design flow (Fig. 3.6) used for the implementation of sub-clock power gating previously summarised in Chapter 3, Section 3.2.2, and an overview of the entire test chip is given with details of additional circuitry necessary for validation of the proposed techniques. For the fabrication of the test chip, a 65nm Low Power (LP) TSMC process and an ARM Artisan^{TM} Library was used. The test chip was fabricated through the multi-project 'Mini@sic' scheme by Europractice. The die size available for the entire project was 2mm×2mm and therefore allowed additional experiments to be fabricated on the same test chip. This consequently had some influence on the design of the test chip and the sub-clock power gated Cortex-M0.

The Cortex-M0 was implemented such that both symmetric virtual rail clamping and shut down power gating could be employed in the sub-clock power gating technique (Chapter 3, Section 3.2) for comparison purposes. This meant that the sub-clock Cortex-M0 needed to be able to switch between either mode. The 'override' mode of operation shown in Fig. 4.4 was also made available for comparison against no power gating. Finally, it was necessary for the Cortex-M0 to support a fourth fully power gated mode to facilitate analysis of other experiments on the test chip. These four modes of operation are summarised in Table 4.2 along with the set of control signals used. The *Off* signal switches the power to the entire Cortex-M0. The *nOverride* control signal forces the microprocessor into normal operation. The choice between sub-clock power gating with shut down power gating or symmetric virtual rail clamping is controlled with the *Drowsy* signal. State retention registers are not used in the Cortex-M0 and so in the fully shut down mode of operation the state is lost, however the registers are kept always-on in the sub-clock modes of operation (Fig. 4.4) meaning state is retained. Not shown in the table is an additional 4-bit *nPG* signal which splits the power gates used for the combinational logic into four equal groups, allowing each group to be enabled or disabled for investigation of charge up time as will be shown in Section 4.4.

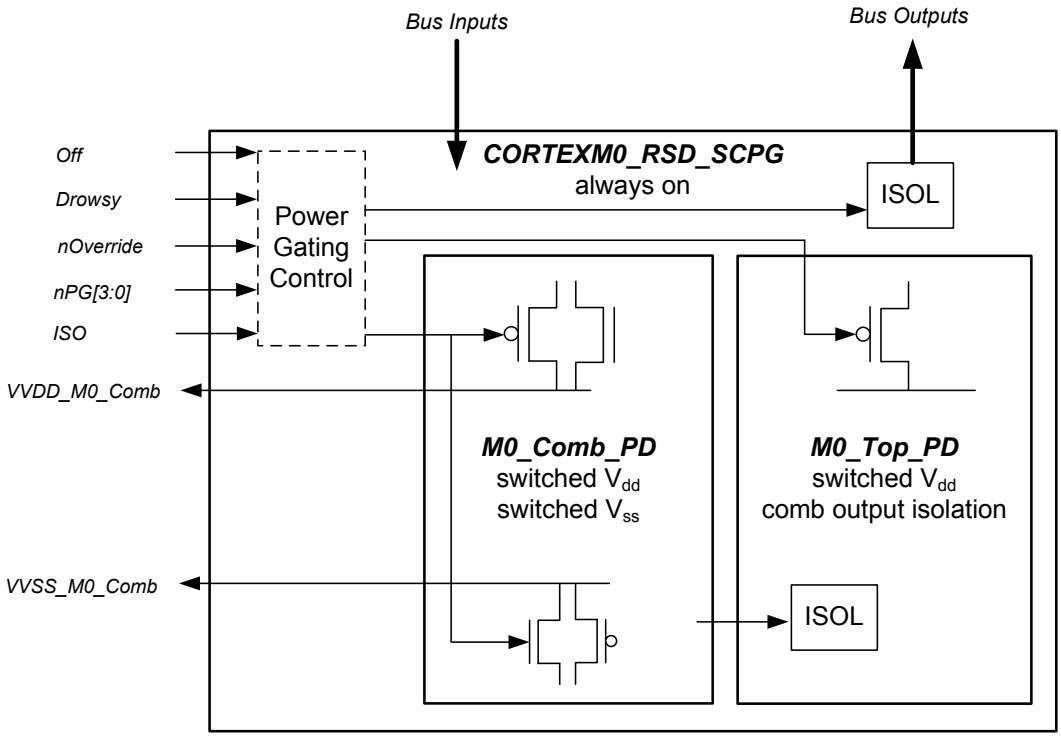


Figure 4.6: Power intent of sub-clock Cortex-M0 microprocessor

Due to the requirement for the registers to be power gated, the Cortex-M0 required three power domains as opposed to just two as shown in Fig. 4.4. The power intent (later used to write the UPF) of the Cortex-M0 is represented diagrammatically in Fig. 4.6. The three power domains were: a combinational (*M0_Comb_PD*) and sequential (*M0_Top_PD*) power domain as is standard for a sub-clock design, but also an additional always-on domain for the power gating control (*CORTEXM0_RSD_SCPG*). If the sequential logic was not switched then it would have been grouped into a single always-on power domain with the power gating control logic. The *M0_Comb_PD* power domain is where all the combinational logic of the microprocessor is located and the power is controlled through the pairs of PMOS and NMOS transistors. The *M0_Top_PD* power domain accommodates all the sequential logic of the Cortex-M0 and the output isolation from the combinational power domain and the power is controlled through a PMOS transistor. In sub-clock power gating this domain remains fully powered at all times. The *CORTEXM0_RSD_SCPG* power domain is always-on and is where the power gating control and Cortex-M0 output isolation is located for when the entire Cortex-M0 is shut down. This output isolation is controlled with an *ISO* control signal. The *Off*, *nOverride*, *Drowsy*, *nPG* and *ISO* signals are controlled externally to the Cortex-M0. The power intent also includes the ability to observe the VV_{dd} and VV_{ss} supply rails by outputting them to analogue pads made available on the test chip and will be shown in Section 4.4.

4.3.1 Silicon Design Flow

The design flow for implementing the sub-clock power gating technique was introduced and summarised in Chapter 3, Section 3.2.2. Further details of the additional steps required in the design flow and design considerations during layout are given using the ARM Cortex-M0 implementation as an example. For the fabrication of the test chip the Synopsys tool suite was used. Design Compiler is used for synthesis, IC Compiler is used for place and route and verification is completed with Star-RC, PrimeTime, VCS and HSpice. All the tools used were version E-2010.12. Design rule checking (DRC) and layout versus schematic (LVS) sign-off was completed with Calibre from Mentor Graphics.

4.3.1.1 Design Preparation for Sub-Clock Power Gating

As described in Chapter 3, Section 3.2.2, power domains can only be defined in the UPF using Verilog modules and is the primary reason the first two steps are required in a sub-clock power gating design flow, Fig. 3.6. In the first 'Synthesis to GTECH' step, Synopsys Design Compiler is used to synthesise the design to the generic GTECH gate library [17] and is done to enable the second stage of the design flow, 'Split Gate Level Netlist', to use a script to perform the correct combinational/sequential logic split. To simplify the process, the modularity of the Cortex-M0 was capitalised on to identify purely combinational modules and mark them as *dont_touch* to skip their synthesis. Please see Fig. 3.15 for a reminder of the Cortex-M0 mapping into sub-clock power gating. This meant that the final netlist from the GTECH synthesis included modules for the ALU, Multiplier, Shift and Permute Unit and Decoder which could be treated as black boxes and swapped with the original RTL. This also meant better optimisation of arithmetic terms could be achieved during the real synthesis as the target 65nm gate library consists of a richer, more complex set of logic gates, when compared to the GTECH library, improving the footprint of the design. An example of gates in the GTECH netlist is given below:

```
GTECH_FD4 zflag_reg ( .D(n106), .CP(pclk), .SD(HRESETn), .Q(\psr_apsr[2]));
GTECH_NOT U1468 ( .A(ctl_ra_addr[3]), .Z(n763) );
GTECH_AND_NOT U1467 ( .A(ctl_ra_addr[2]), .B(n763), .Z(n764) );
```

The script used to parse the netlist in the 'Split Gate Level Netlist' step, Fig. 3.6, is written in Perl because of its efficiency with text file manipulation. Synopsys Design Compiler suffixes every register in the output netlist with an *_reg* which makes identifying registers much easier; see the zflag_reg definition above for an example. The script therefore uses a rule by which any gate with an *_reg* suffix is assigned to a new sequential Verilog module and any other gate or Verilog instantiation (including the

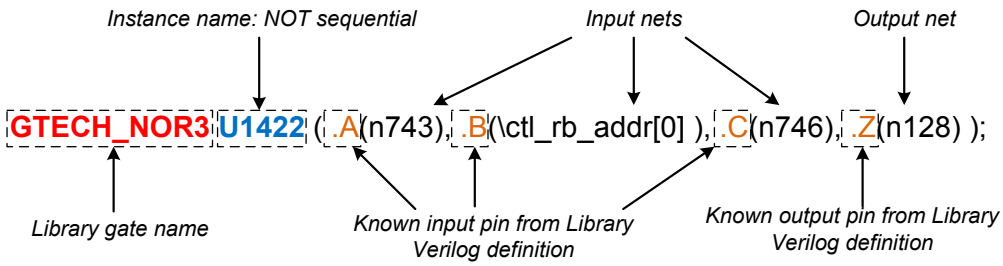


Figure 4.7: How the Perl script identifies gates and records their connections

ALU, Multiplier etc. modules) is assigned to the combinational Verilog module. This is achieved by reading the netlist line by line and comparing the instance name with a regular expression as demonstrated in Fig. 4.7. Although the instances in the netlist can be identified and split into their respective Verilog modules the signals to and from each gate must also be maintained to preserve functionality. This is done by reading the gate library's Verilog definitions and learning the input and output ports of each gate; for example, with the GTECH gate library this is read from the file *gtech_lib.v* and a NOR3 gate definition would show that the input ports are 'A', 'B' and 'C' and the output port is 'Z', Fig. 4.7. Once the names of the ports are learnt for each gate the script can record the input and output nets for each instance in the GTECH netlist into a tree structure. The tree is then traversed and any nets crossing between the combinational and sequential Verilog modules are assigned as primary inputs/outputs accordingly to maintain their connections and functionality.

Before synthesis to the target gate library can be performed, the sequential and combinational modules are combined with the isolation control circuit and the control statements for the power gating transistors in a top level Verilog module wrapper. The control to the power gates were added by using Verilog assign statements such as:

```
assign HEAD_SLEEP[0] =
    Off ? 1 : (!nOverride) ? 0 : nPG[0] ? 1 : HCLK;
```

This command is used for the control to one of the header power gates and complies with the control set out in Table 4.2. The power gate is switched off when the *Off* signal is asserted, otherwise if the active low *nOverride* signal is asserted the power gate is forced on. If neither of these control signals is asserted the power gate must first be enabled by setting the *nPG* signal to logic 0 and then the clock controls the power gate state as is required in the sub-clock power gating mode of operation. In the case of the footer power gates and the opposing NMOS and PMOS clamping transistors, the additional *Drowsy* signal is also used which allows choice between sub-clock power gating with shut down power gating or the proposed symmetric virtual rail clamping, for example:

```
assign FOOT_SLEEP[0] =
    OFF ? 0 : (!nOverride) ? 1 : Drowsy ? (nPG[0] ? 0 : ~HCLK) : 1;
```

The isolation circuit on the other hand was created in a separate Verilog module and then instantiated in the top level wrapper. As one of the primary inputs to the control circuit is the value of the combinational domain's virtual V_{dd} supply rail, Fig. 3.4, a TIEHI gate from the gate library is manually placed in the combinational Verilog module and the signal provides the control output to the top level instantiation of the isolation circuit. The isolation circuit was defined as a netlist using logic gates from the target library to explicitly control the functionality and behaviour of the circuit and stop any optimisation from the EDA tools. During synthesis the TIEHI gate along with the isolation circuit were additionally marked as *dont_touch* to ensure the EDA tool did not perform any unwanted optimisation as the TIEHI signal appears as a constant logic 1. The isolation circuit netlist used in the Cortex-M0 fabrication can be found below and is a gate for gate map of the circuit presented in Fig. 3.4:

```
module ISO_CTRL(virtual_rail, nOverride, CLK, ISOLATE);

input virtual_rail, nOverride, CLK;
output ISOLATE;

wire net1, isol1, isol2, ISOLATE;

INV_X4M_A12TR u0 ( .A(CLK), .Y(net1) );
NAND2_X1A_A12TR u1 ( .A(net1), .B(virtual_rail), .Y(isol1) );
OR2_X8M_A12TR u2 ( .A(CLK), .B(isol1), .Y(isol2) );
AND2_X11M_A12TR u3 ( .A(nOverride), .B(isol2), .Y(ISOLATE) );

endmodule
```

An integral part of the design preparation is the UPF power intent file. The power intent of Fig. 4.6 was translated into a UPF and is listed in full in Appendix C.1. The newly defined Verilog combinational and sequential modules enable the UPF to be used as you would for any other power gating design. An example of how a power domain and its corresponding supply nets are defined in the UPF was given in Fig. 3.12. Using the power gating control signals created in the top level wrapper the power gates are defined using standard UPF commands:

```
create_power_switch VVDD_M0_Comb_sw0 -domain M0_Comb_PD \
-input_supply_port {VDD VDD} -output_supply_port {VVDD_M0_Comb VVDD_M0_Comb} \
-control_port {sleep M0/HEAD_SLEEP[0]} \
-on_state {on_state VDD {!sleep}}
```

One crucial requirement in the UPF is to ensure that isolation is defined for all outputs from the combinational domain, 'ISOL' block in Fig. 4.4, but the VV_{dd} virtual rail signal from the TIEHI gate in the combinational power domain must remain un-isolated as it

serves as a primary input for the isolation control. As an example, the combinational output isolation is defined as follows:

```
set_isolation M0_Comb_isol -domain M0_Comb_PD \
-isolation_power_net VVDD_M0_Top -isolation_ground_net VSS -clamp_value 0

set_isolation_control M0_Comb_isol -domain M0_Comb_PD \
-isolation_signal M0/ISOLATE -isolation_sense high -location parent

set_isolation M0_Comb_noIso -domain M0_Comb_PD \
-elements M0/combinational/VIRTUALRAIL \
-no_isolation
```

4.3.1.2 Layout: Design Planning

As mentioned in Chapter 1, Section 1.4.1.1 the placement of voltage areas in the physical layout can have an effect on the area and routing. In a small design such as the ARM Cortex-M0 this impact can be small but is still considered in the layout. The separation of the combinational and sequential logic required in the sub-clock power gating technique demands that both types of logic are confined within their own respective voltage areas in the physical layout. This effectively creates an unusual placement constraint on the logic gates as combinational and sequential logic would conventionally be placed together to minimise physical distance and improve routing. Therefore, to minimise the placement impact, the combinational logic, *M0_Comb_PD*, is constrained to the centre of the floorplan with the sequential logic, *M0_Top_PD*, on its periphery, this is shown in Fig. 4.8. This placement choice will be explained in Chapter 5. The always-on power domain, *CORTEXM0_RSD_SCPG*, is constrained to the top of the floorplan since it mainly consists of the Cortex-M0 output isolation gates and the output pins are placed at the top of the layout due to the Cortex-M0's position in the test chip.

The power gates were placed in the floorplan in a grid pattern, as opposed to a ring pattern, for better distribution of the IR drop across the power network [40]. Only header transistors were used in the *M0_Top_PD* power domain, Fig. 4.8, as conventional shut down power gating is implemented for that voltage area. The combinational *M0_Comb_PD* power domain on the other hand required placement of both header and footer power gates but also the placement of the NMOS and PMOS clamping power gates. The clamping power gates are purely used during the sleep mode and hence do not contribute to the active mode power gating IR drop as is the case for power gating transistors [3]. A similar experiment to the one done with the NOR4 gate in Section 4.1 was done to estimate the minimum voltage that the logic in the target 65nm library can be clamped to without loss of valid logic outputs. A simulation was done using a 39 stage logic path of alternating NOR3 and NAND4 gates to represent a worst case

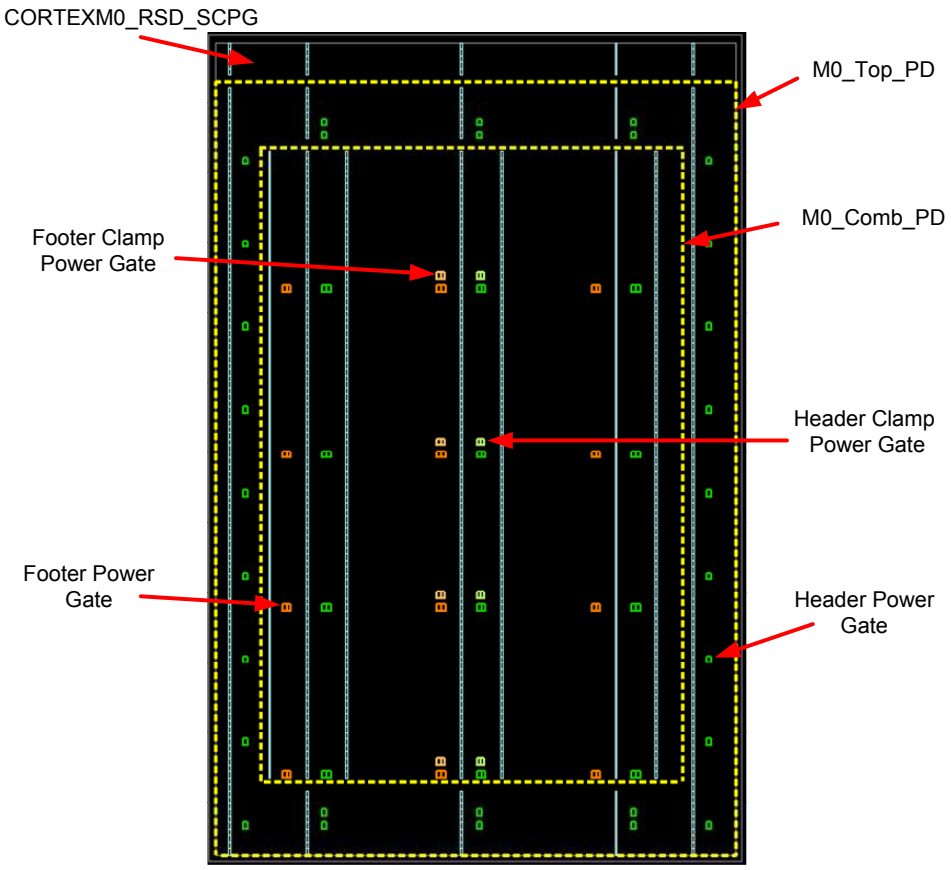


Figure 4.8: Voltage area and power gate placement in sub-clock Cortex-M0

critical path. 39 gates are used because the longest logic depth in the Cortex-M0 is found to be 40 and NAND and NOR gates are used because they experience the worst effects of supply degradation due to the large number of parallel transistors [37]. In this series circuit a degraded logic 1 from the first NOR3 gate is used to switch the NMOS transistors in the NAND4 which outputs a degraded 0 and so on. When driving a MOSFET with a degraded logic level, the transistor doesn't fully switch on resulting in an amplification of the output degradation further down the logic path [2]. The output of the last gate is monitored to determine if the logic level is correctly held. It was observed that the output collapsed at effective V_{dd} values below 160mV using symmetric virtual rail clamping, and at voltages above that, the output did not deviate more than 6%. It must be noted that while the circuit supply voltage is clamped down to below the threshold voltage it is not operated at this voltage and so if some logic gate outputs were to flip whilst clamped, then this would not be an issue as they would be rectified when the supply is returned to its nominal value. Through simulation it was found that a total of 8 regular threshold voltage transistors with width 3.6μm placed on the VV_{dd} and VV_{ss} supply rails achieved a clamped voltage of 170mV at a supply voltage of 700mV as shown in Fig. 4.9. A 5% IR drop was targeted for the power gate transistor widths which are found through simulation. A total of 24 header and footer power gates were placed in the *M0_Comb_PD* power domain. All the power gates, including

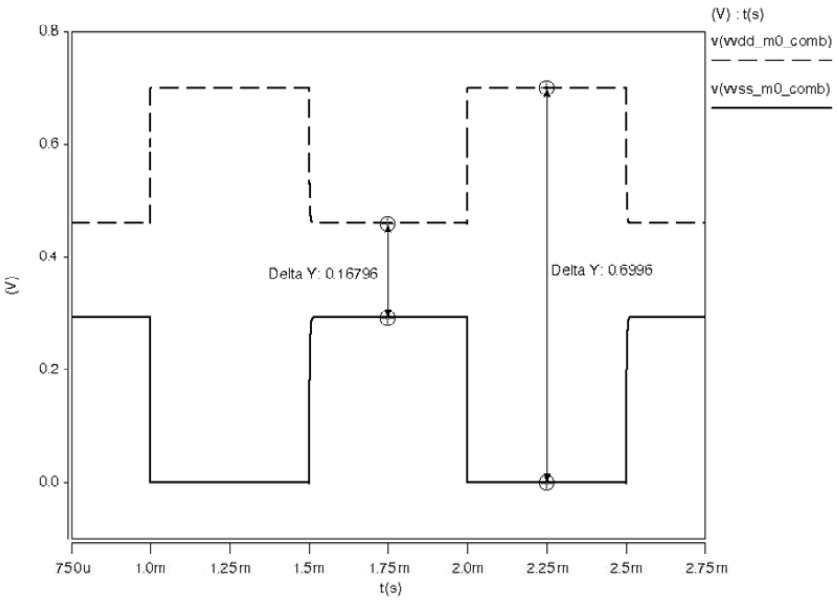


Figure 4.9: Simulated VV_{dd} and VV_{ss} in Cortex-M0 with symmetric virtual rail clamping

clamping transistors, were split into 4 groups, i.e. 6 header and footer transistors and 2 NMOS and PMOS transistors, for each nPG control signal. This meant that each of the four nPG control lines enable/disable a quarter of the power gating transistors in the combinational domain for investigation of charge up time (Section 4.4).

4.3.1.3 Verification

The verification stage that was shown in Fig. 3.6 is simplified and can instead be represented as Fig. 4.10, and is almost identical to a traditional power gating design flow [3]. The timing analysis uses accurate representation of the signal resistance and capacitance parasitics in the design (Standard Parasitic Exchange Format) to check register setup and hold timings and outputs a standard delay format file (SDF) which is used to check the functional behaviour of the circuit. Although functional verification is a transient simulation, in a sub-clock power gating circuit, functional verification only enables the static power gating behaviour of the circuit to be checked. This is because functional verification assumes ideal transitions between power modes, and so it is only possible to place the combinational logic in the power gated state and then check 'X' values do not propagate out of the combinational domain and that the header and footer power gates are correctly driven. Consequently, for correct transient verification of the sub-clock power gating mode of operation it was necessary to use HSpice as it enables observation of the virtual rail behaviour which is key in the functionality of the isolation control circuit, Fig. 3.4. This HSpice step is highlighted in Fig. 4.10 and does not typically appear in a traditional verification flow [2]. The parasitic extractor (Star-RC) is used to extract the Spice netlist of the design with full resistance and capacitance to

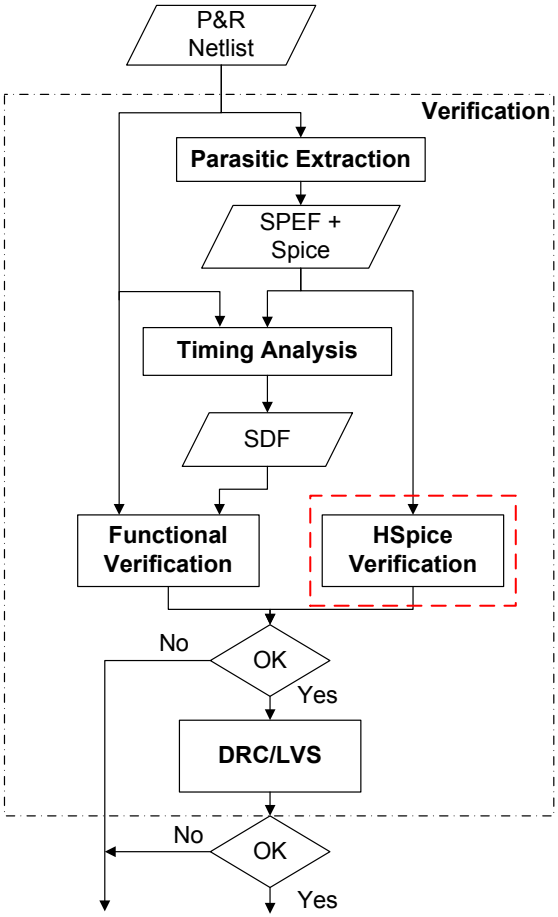


Figure 4.10: Expanded verification flow for the silicon fabrication

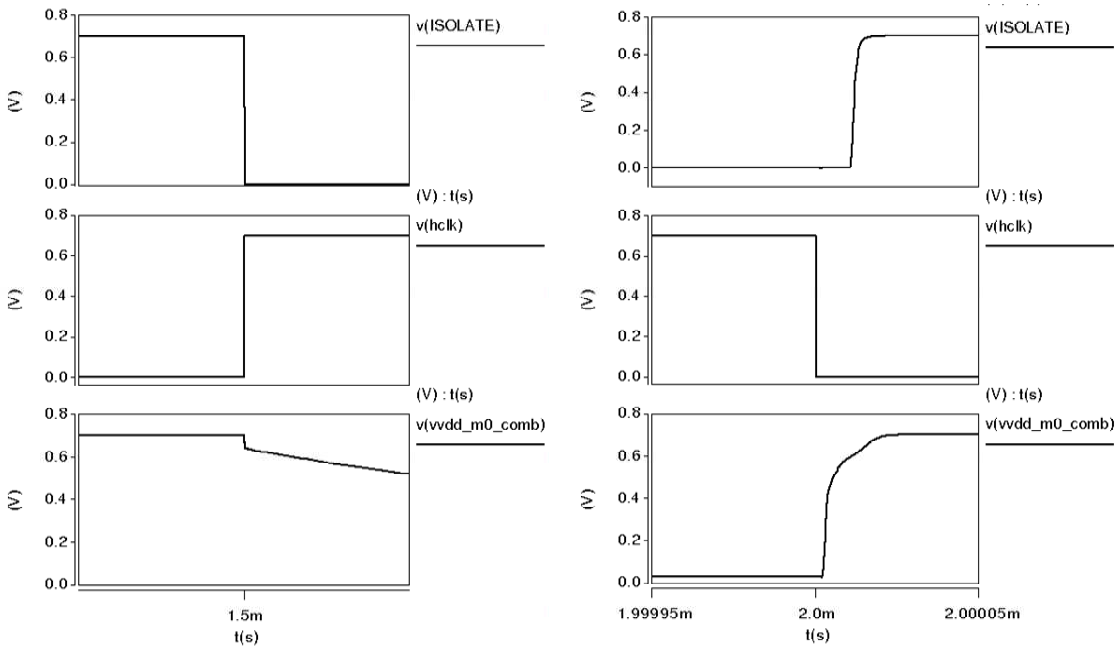


Figure 4.11: Simulated transient behaviour of isolation enable signal *ISOLATE*. Left - entering sleep, Right - exiting sleep

obtain accurate power and timing transients. Note, this is the same Spice netlist that was used for post-layout simulation in Chapter 3. With this netlist it is then possible to validate the behaviour of the virtual supply rails, as was shown in Fig. 4.9, and the functionality of the isolation control circuit. An example of the transient behaviour of the isolation enable signal is shown in Fig. 4.11. As can be seen, the active low *ISOLATE* control signal is asserted when the clock goes high (power goes off) but the isolation gates are not enabled again until the virtual rail has charged by two thirds which is the desired behaviour, Fig. 4.5.

4.3.2 Test Chip Overview

The complete test chip is shown to the right of Fig. 4.12 with the key features necessary for validation of the sub-clock power gating and symmetric virtual rail clamping techniques highlighted. Unmarked regions of the test chip accommodate other experiments which are not relevant to this thesis. The Cortex-M0, marked as CM0 in Fig. 4.12, was allocated its own power supply in the test chip to allow power measurement of just the microprocessor core once fabricated. In addition to this, two analogue pads were included to enable direct observation of the VV_{dd} and VV_{ss} supply rails in the sub-clock power gating technique. The rest of the SoC was made up of SRAM, an ASCII Debug Protocol (ADP) and a clock modulator circuit. The SRAM serves as the instruction and data space for the microprocessor. The SRAM must be operated at the nominal 1.2V and consequently, the outputs of the microprocessor are level shifted to allow the processor's voltage to be scaled down. The ADP is a simple state machine which communicates with an off chip USB protocol converter over an 8-bit bus and provides read and write access to the entire memory map of the SoC via USB. This interface provides the means to download program code to the SRAM and set and unset bits in control registers to select the mode of operation for the sub-clock Cortex-M0. As shown in Chapter 3 the high phase of the clock can be extended in the sub-clock power gating technique to maximise power savings. The clock modulator circuit provides the ability to change the duty cycle of the clock to the Cortex-M0 microprocessor on-chip. A diagrammatical representation of the circuit used to achieve this modulation is shown in Fig. 4.13. The off chip clock is fed into the modulator and is divided down to a period of $(1+n).T_{clk}$, where n can be programmed to values up to 2^{32}. The resulting output clock from the modulator is low for T_{clk} and high for $n.T_{clk}$ as shown in Fig. 4.13. Therefore with a 5MHz external clock, for example, T_{clk} is 200ns and the n value required for a 100kHz clock is obtained as follows:

$$n = \frac{1}{f \times T_{clk}} - 1 = \frac{1}{100000 \times (200 \times 10^{-9})} - 1 = 49$$

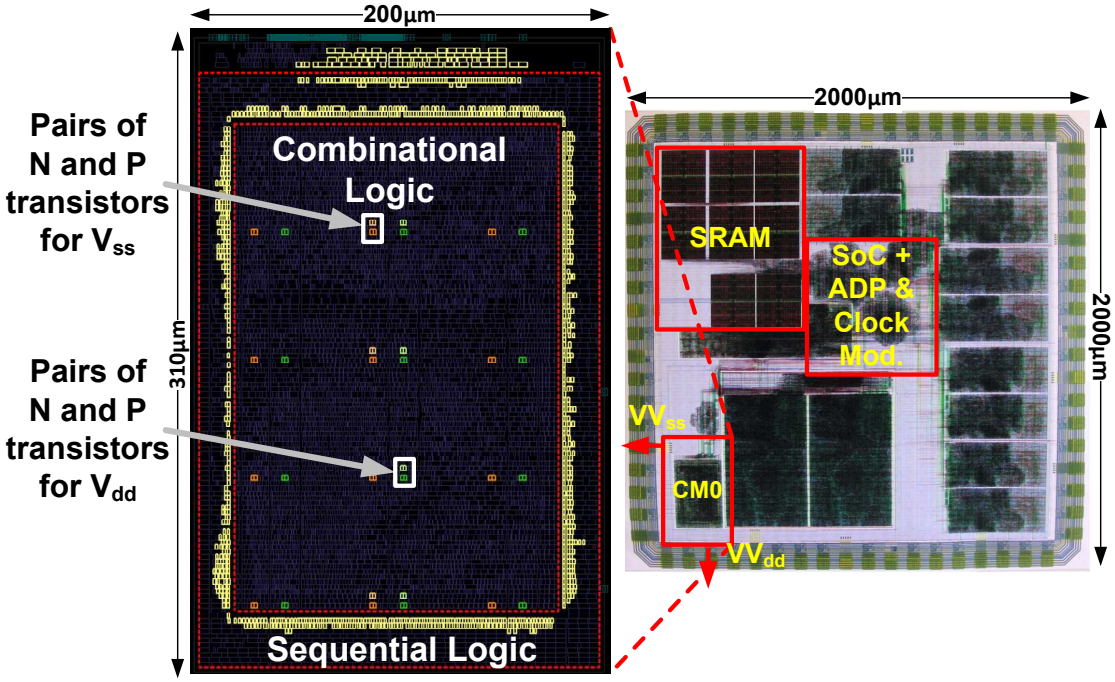


Figure 4.12: Final Layout of test chip and sub-clock ARM Cortex-M0

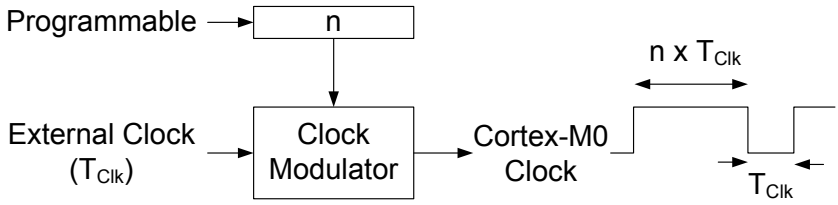


Figure 4.13: Clock modulator circuit

The left of Fig. 4.12 shows the final Cortex-M0 floorplan with sub-clock power gating. The final layout was 200×310μm and the total area is increased by 15.6% compared to a design with no power gating due to the additional circuitry for sub-clock power gating and placement constraints. The combinational logic and sequential logic are marked and correspond with the 'Comb. Logic' and 'Seq. Logic' blocks in Fig. 4.4. It is interesting to note that the placement tool has grouped the isolation cells, which correspond to the 'ISOL' block from Fig. 4.4, on the boundary of the combinational and sequential areas of the layout to minimise distance and routing. The final design was signed off at the worst case characterisation corner with a critical path length of 4.978ns.

4.4 Experimental Results

Four experiments were carried out to demonstrate the proposed sub-clock power gating technique and symmetric virtual rail clamping technique. The first compares the sub-clock power gating technique using conventional shut down power gating and symmetric virtual rail clamping proposed in this Chapter, Section 4.1. The second investigates

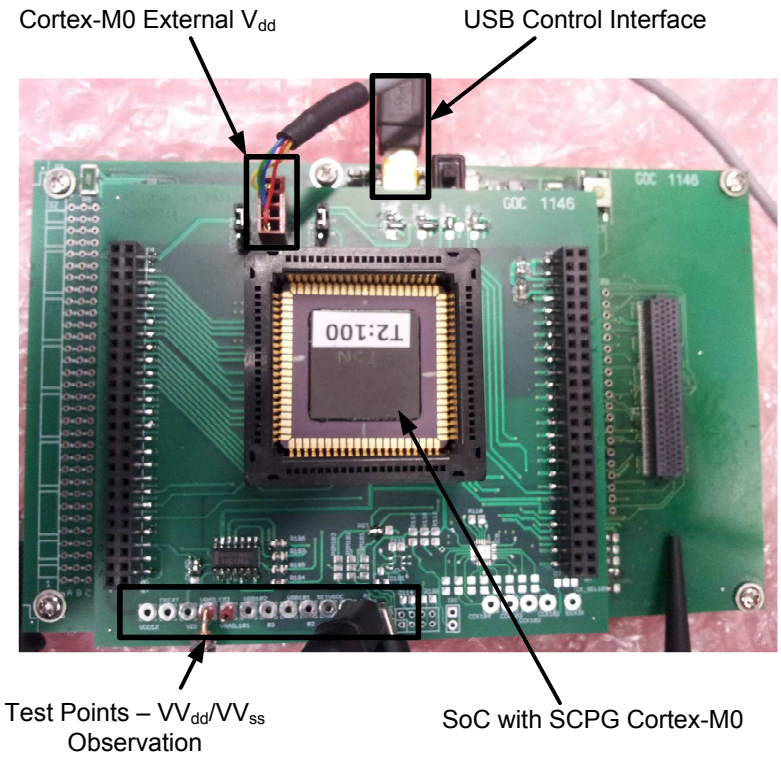


Figure 4.14: Testboard for experimental measurement

the effect of varying the clock duty cycle to show how power saving changes with the length of the high phase of the clock in sub-clock power gating. The third experiment compares the power consumption of the Cortex-M0 with and without sub-clock power gating using symmetric virtual rail clamping. The fourth measures sub-clock power gating with symmetric virtual rail clamping's impact on ground bounce.

All experiments on the fabricated chip were performed using the testboard shown in Fig. 4.14. In line with the scaled voltage typically found in processors designed for the target applications [60, 64, 153], a 0.7V external power supply is used for the Cortex-M0's independent V_{dd}, which is the limit for the level shifters used for the interface between the processor and SRAM but also remains adequately above the transistor threshold voltages. To emphasise the negative impact of leakage in scaled nanometer technologies a temperature of 90°C is used. An ammeter with 10nA resolution is connected in series with the power supply to allow current measurement of the microprocessor. The USB interface and ADP is used to download the benchmark programs and set up the test chip. The analogue pads included in the test chip are connected to the board's test points shown in Fig. 4.14, for observation of the combinational logic virtual rails. Fig. 4.15 shows an oscilloscope trace of the VV_{dd}/VV_{ss} supply rails when using the proposed SCPG technique with symmetric virtual rail clamping. An 8kHz clock is used in this trace with a 2:1 (high:low) duty cycle ratio for measurement purposes. Over the first part of the clock period (T_{clk}) the VV_{dd} and VV_{ss} rails are clamped to approximately 450mV and 270mV respectively, aggressively reducing the combinational voltage supply

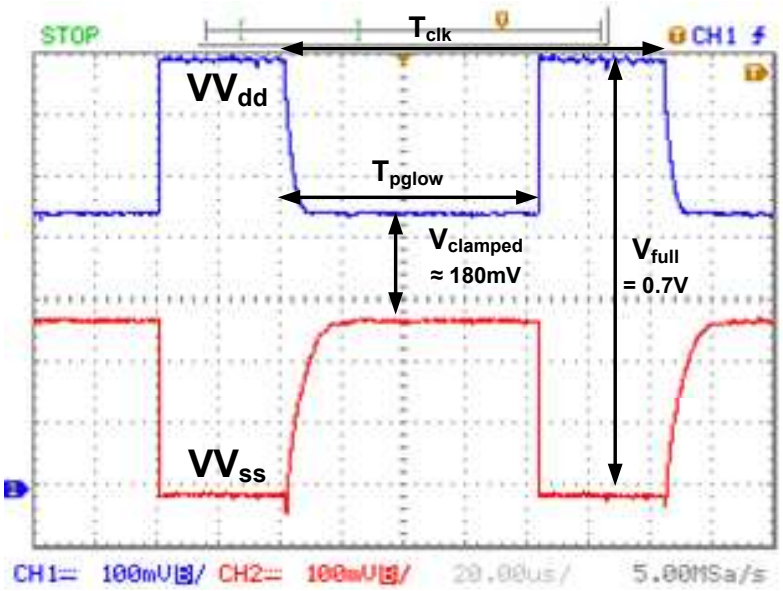


Figure 4.15: Measured VV_{dd} and VV_{ss} behaviour in sub-clock power gating using symmetric virtual rail clamping

to 180mV which closely matches with the 168mV simulated value shown in Fig. 4.9. The time taken to restore the virtual supply rails to full 0.7V supply voltage is measured to be 45ns and is kept at this value over the second part of the clock period.

4.4.1 Symmetric Virtual Rail Clamping Vs Shut Down Power Gating

Table 4.1 showed that the wake-up energy associated with shut down power gating was higher than the proposed symmetric virtual rail clamping circuit through ring oscillator simulations. The Cortex-M0 microprocessor has been implemented with both symmetric virtual rail clamping and shut down power gating to allow practical comparison of the two techniques in sub-clock power gating. Experimental measurement from the chip shows that when the Cortex-M0 is fully powered but the clocks are stopped, the leakage power dissipation is 7.51μW. On the other hand, when the combinational logic is fully shut down, power dissipation is 1.46μW, representing an 80.6% reduction in power. Alternatively, when the combinational logic supply is clamped using symmetric virtual rail clamping the power dissipation is 2.44μW, a 67.5% reduction in leakage power. This is to be expected, since shut down power gating completely disconnects the supply whereas symmetric virtual rail clamping maintains a voltage across the combinational logic, and matches with the trends shown in Table 4.1.

Fig. 4.16 shows the behaviour of the virtual rails when using sub-clock power gating with conventional shut down power gating. An 8kHz clock is used in this trace with a 2:1 (high:low) duty cycle ratio for measurement purposes. Unlike symmetric virtual rail clamping, that was shown in Fig. 4.15, the VV_{ss} rail is unclamped and the VV_{dd} rail is fully discharged in the first part of the clock period (T_{pgoff}). In the second

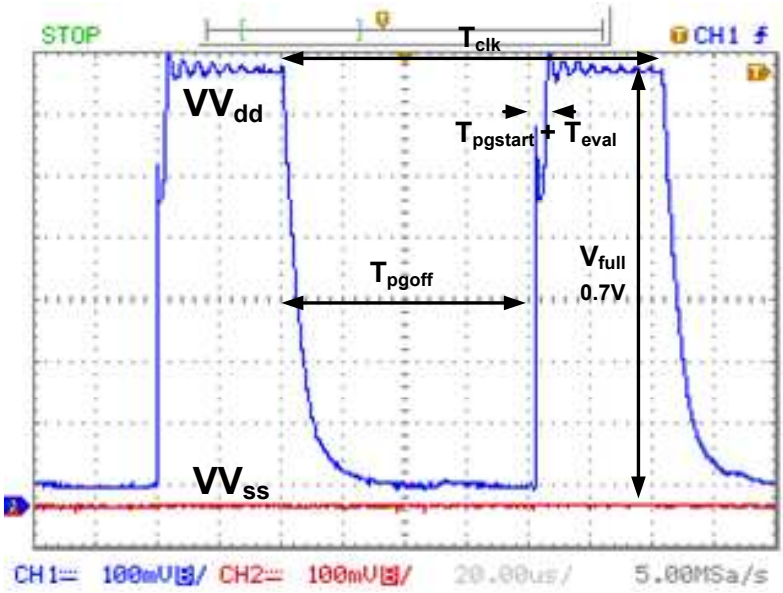


Figure 4.16: Measured VV_{dd} and VV_{ss} behaviour in sub-clock power gating using shut down power gating

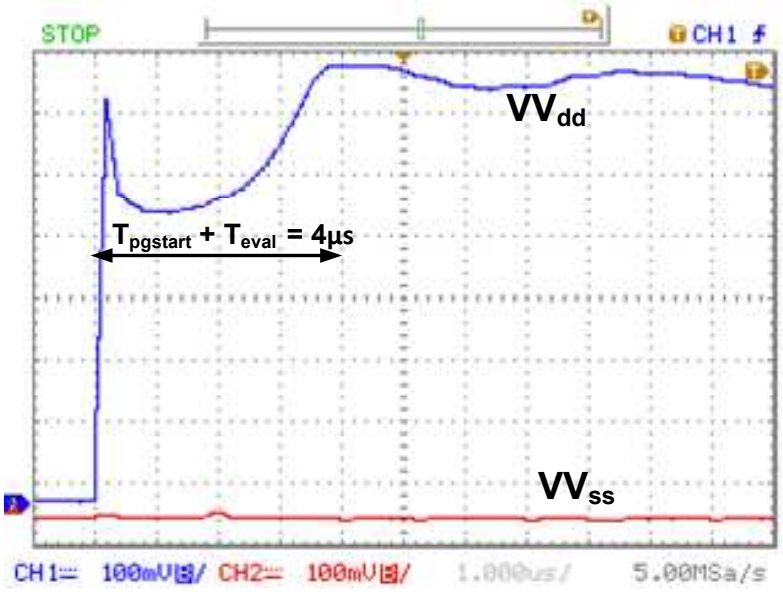


Figure 4.17: Measured VV_{dd} charge-up and evaluation time in SCPG with shut down power gating

part of the clock period the VV_{dd} rail is restored to the full 0.7V supply. Notice that the charge-up ($T_{pgstart}$) and evaluation time (T_{eval}) of the combinational logic is clearly visible in Fig. 4.16 and is labelled as '$T_{pgstart} + T_{eval}$'. Fig. 4.17 shows this time frame more clearly. The droop seen in the VV_{dd} rail can be attributed to the high volume of signal glitching that occurs as the combinational logic is brought out of shut down and re-evaluates. The glitching demands a large surge of current which cannot be delivered by the number of power gates used and results in the droop of the virtual V_{dd} rail. The droop subsequently slows the combinational re-evaluation further exacerbating the

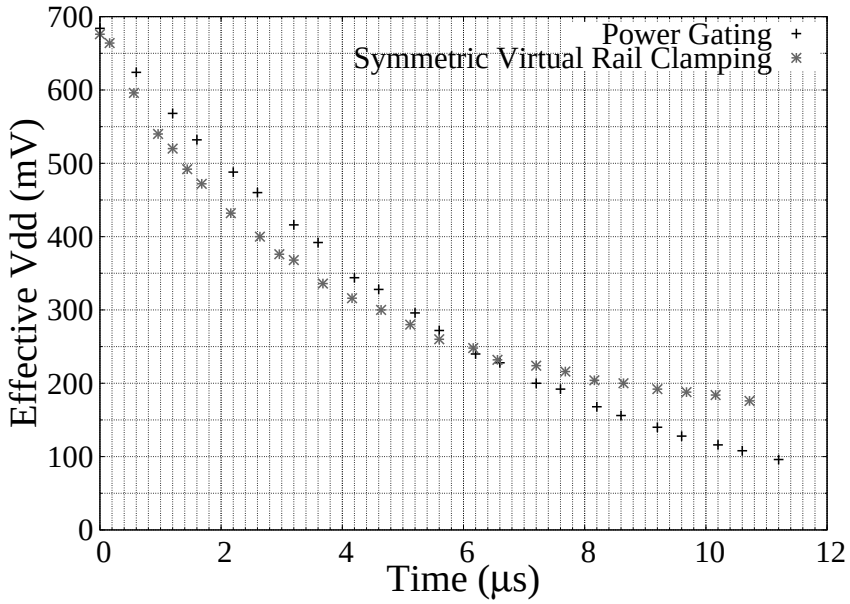


Figure 4.18: Measured effective V_{dd} reduction against time

Table 4.3: Dhrystone - Average measured power and energy in three modes of operation, V_{dd}=0.7V

Clock	**No Power Gating**		**Proposed SVRC**			**Shut Down**		
Freq. (kHz)	**Power (uW)**	**Energy (pJ)**	**Power (uW)**	**Energy (pJ)**	**Saving (%)**	**Power (uW)**	**Energy (pJ)**	**Saving (%)**
0.5	7.57	15140	2.57	5140	66.05	3.70	7405	51.09
1	7.58	7577	2.58	2584	65.89	5.79	5793	23.53
2	7.59	3793	2.62	1310	65.45	10.03	5014	-32.17
5	7.60	1519	2.75	550.5	63.76	22.06	4412	-190.4
10	7.63	762.6	2.95	294.7	61.35	42.00	4199	-450.6
20	7.63	381.6	3.40	170.1	55.42	76.51	3825	-902.3
50	7.78	155.5	4.52	90.33	41.95	33.37	667.3	-328.8
125	10.19	81.50	7.20	57.61	29.31	7.48	59.87	26.5

length of T_{eval} to 4μs as shown. This is unlike symmetric virtual rail clamping shown in Fig. 4.15 where the voltage maintained across the combinational logic helps to eliminate signal glitching during charge up, enabling supply restoration in 45ns. The consequence of this increased wake-up time when using shut down power gating is the need for a clock with a low phase of at least 4000ns to ensure there is enough time for next state evaluation. In addition to comparison of wake-up time, the discharge rate of the virtual rails in both shut down power gating and symmetric virtual rail clamping have also been measured and the results are shown in Fig. 4.18. As can be seen, symmetric virtual rail clamping achieves faster reduction in effective V_{dd} than shut down power gating and the results presented in this graph correspond with trends of the simulated results shown in Fig. 4.2. This means that for power gated periods shorter than 6μs symmetric virtual rail clamping can enable greater reduction in supply voltage in the same time frame.

To compare shut down power gating with symmetric virtual rail clamping in the sub-clock power gating technique, the Dhrystone benchmark used in Chapter 3 is loaded

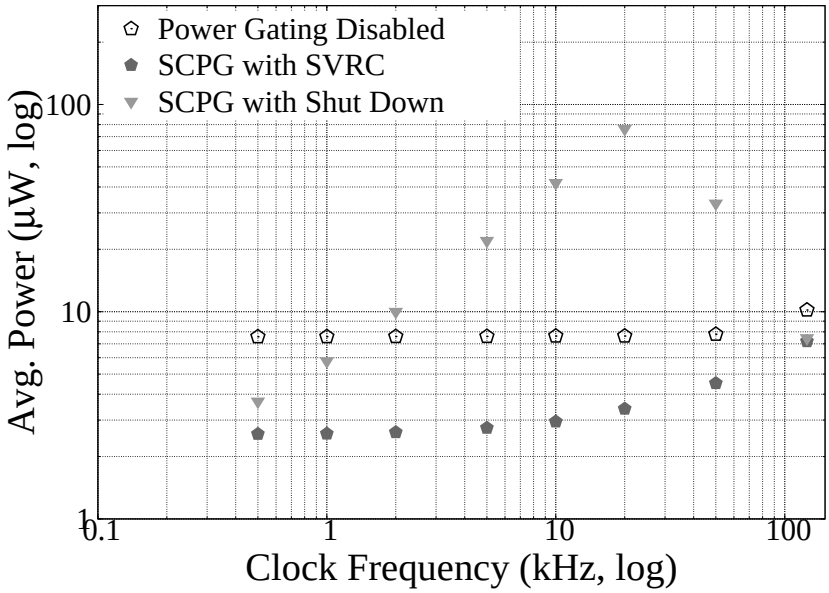


Figure 4.19: Measured Cortex-M0 power with power gating disabled, proposed SCPG with symmetric virtual rail clamping and SCPG with shut down power gating

into the SRAM and the average current during execution is recorded. To permit direct comparison of the two power gating techniques, both sub-clock power gating modes of operation used a clock duty cycle with a 4μs low period, achieved by using a 250kHz external clock signal for the clock modulator circuit, Fig. 4.13. Therefore, the applicable clock frequency range of sub-clock power gating is limited to 125kHz. The average power and energy of the Cortex-M0 in all three modes of operation are given in Table 4.3. Fig. 4.19 compares, graphically, the measured average power of sub-clock power gating using shut down power gating, sub-clock power gating using the proposed symmetric virtual rail clamping (SVRC) technique and power gating disabled. Note that at 500Hz and 1kHz, sub-clock power gating using conventional shut down power gating has lower power consumption than without power gating but is higher than the proposed symmetric virtual rail clamping. This can be attributed to the high wake-up energy cost associated with the signal glitching when restoring the virtual rail in shut down power gating. Note also, this high wake-up energy cost causes the energy overhead of shut down power gating to exceed the energy saving at frequency points above 1kHz resulting in higher power dissipation in comparison to no power gating. Conversely, symmetric virtual rail clamping maintains a lower power consumption than no power gating at all frequency points. The increasing power trend of the shut down power gating mode is reversed after 20kHz. This is because the virtual rail does not discharge fully during shut down, due to the shorter idle time within the clock period, and eliminates some signal glitching. This behaviour is very similar to using single virtual rail clamping [103] (Fig. 4.1(a)) and gives insight into how virtual rail clamping would compare with the proposed symmetric virtual rail clamping. As can be seen, despite the VV_{dd} remaining partially charged at 125kHz, the power dissipation is higher than symmetric virtual rail clamping (Table

4.3) and is a result of the asymmetric reverse body biasing of logic gates and the lack of charge recycling as discussed in Section 4.1. It should be noted that the upward and then downward trend seen with sub-clock power gating using shut down power gating shown here was not observed in the simulation results in Chapter 3. This can be explained by the difference in leakage power magnitudes between the two technology libraries used. With the 90nm technology library in Chapter 3, leakage power was much higher - approximately 200μW Vs 10μW with the 65nm library here. This resulted in the glitching energy cost being absorbed at all frequency points and subsequently a continual upward trend in power consumption. As leakage is much lower in the 65nm library used for the test chip, the glitching energy cost has a greater impact on the overall power resulting in the trend shown in Fig. 4.19. The observations seen here, through comparison of shut down power gating and symmetric virtual rail clamping proposed in Section 4.1, provide further validation for the symmetric virtual rail clamping to improve the energy efficiency of the sub-clock power gating technique and extend its applicable range. It should be noted that this comparison is made with symmetric virtual rail clamping using a 4μs low period but shorter low periods are achievable enabling greater savings as will be shown in the next sections.

4.4.2 Effect of Duty Cycle

As shown in Chapter 3, modulation of the clock duty cycle has a positive impact on the power savings attainable with the sub-clock power gating technique. This is because the combinational logic is power gated when the clock is high and extending this period of the clock exploits a greater amount of the combinational logic idle time. Fig. 4.20 shows experimental measurements of the effect of duty cycle on the power saving attainable with the proposed SCPG technique with symmetric virtual rail clamping. The clock frequency used in these measurements is 10kHz and the power values are normalised to the Cortex-M0 operating at 0.7V with no power gating. The high phase of the clock period increases from left to right in Fig. 4.20 and as can be seen, the power goes down (savings increase) as it does so. It is interesting to note that the normalised power reduces but a lower bound is slowly reached. This is because the combinational logic has some leakage in the clamped state and the registers and other always-on logic also remain active.

This analysis experimentally validates the positive influence using a maximised duty-cycle has on the power savings achievable with sub-clock power gating. To determine a suitable low period of the clock for use in sub-clock power gating, the critical path and charge up time of the virtual rails must be known. The critical path at worst case 1.08V is known to be approximately 5ns from static timing analysis, Section 4.3, however using a scaled voltage increases propagation delay of logic gates, Eqn. 1.7. To estimate the impact on delay with scaled V_{dd}, a 95 stage ring oscillator using NAND gates was placed

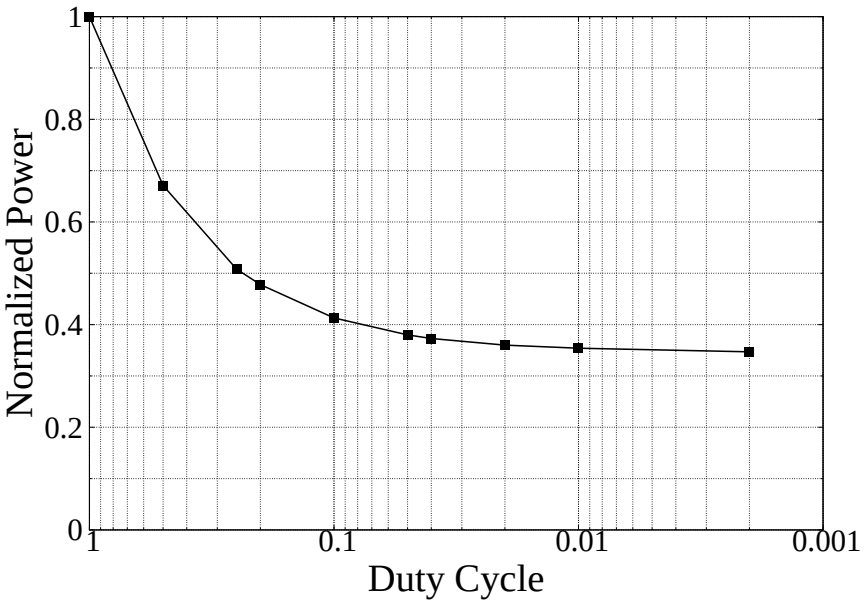


Figure 4.20: Normalised measured power of ARM Cortex-M0 microprocessor with 10kHz clock at varying duty cycle in SCPG mode, V_{dd}=0.7V

on the same supply voltage as the fabricated Cortex-M0 and a method to record its oscillating frequency was also implemented. From measuring the oscillator's frequency at 1.2V and then again at 0.7V, it is found that the propagation delay of the NAND gates increase by 13.87x. This equates to the Cortex-M0 critical path increasing from 5ns to 70ns. The charge up time of the virtual rails in symmetric virtual rail clamping is measured from the oscilloscope traces and is found to be 45ns. To ensure no timing violations occur during operation, a margin is introduced and the low period of the clock is therefore chosen to be 200ns for measurements in the following section. Using the clock modulator circuit (Fig. 4.13) a 200ns low period corresponds to an external clock frequency of 5MHz, and n can then be programmed according to the desired clock frequency.

4.4.3 Sub-Clock Power Gating with Symmetric Virtual Rail Clamping Analysis

Next, the power consumption of the Cortex-M0 with and without the proposed SCPG technique using symmetric virtual rail clamping over a range of clock frequencies is compared. The Dhrystone benchmark [161] previously used in Chapter 3 is used again. The measured results across five test chips are presented in Fig. 4.21 and show that the proposed SCPG technique achieves lower power consumption at all frequency points up to a clock frequency of just over 400kHz. At all of these frequency points, the energy saved (E_{sav}) from using the proposed SCPG technique exceeds the energy overhead (E_{oh}) of power gating resulting in the savings seen. However, as clock frequency increases, E_{sav}

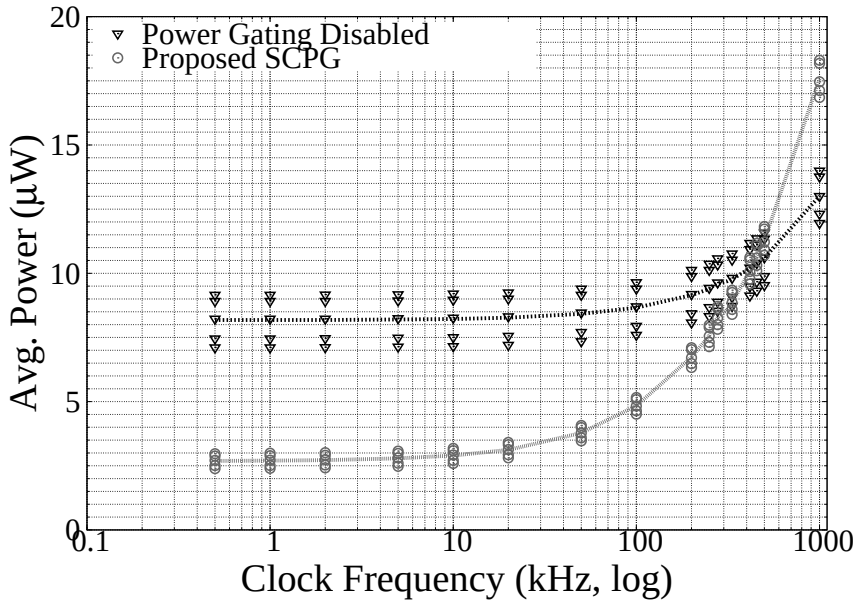


Figure 4.21: Dhrystone - Measured power of ARM Cortex-M0 at varying clock frequency, V_{dd}=0.7V

Table 4.4: Dhrystone - Average measured power and energy over five test chips with power gating disabled (No-PG) & sub-clock power gating (SCPG)

Clock Freq. (kHz)	**No Power Gating**		**Proposed SCPG**		
	Power (uW)	**Energy (pJ)**	**Power (uW)**	**Energy (pJ)**	**Saving (%)**
0.5	8.18	16351	2.69	5385	67.06
1	8.18	8179	2.70	2702	66.96
2	8.18	4091	2.72	1361	66.72
5	8.20	1639	2.79	558.0	65.97
10	8.22	822.3	2.90	290.0	64.73
20	8.27	413.5	3.12	156.0	62.27
50	8.42	168.3	3.78	75.58	55.11
100	8.66	86.61	4.84	48.42	44.10
200	9.15	45.73	6.73	33.64	26.43
250	9.39	37.55	7.57	30.30	19.31
263.2	9.52	36.18	8.02	30.48	15.73
277.8	9.60	34.56	8.28	29.81	13.73
312.5	9.69	31.00	8.57	27.41	11.57
333.3	9.79	29.37	8.89	26.66	9.22
357.1	9.90	27.73	9.24	25.88	6.67
384.6	10.04	26.09	9.65	25.09	3.86
416.6	10.19	24.46	10.11	24.27	0.79
454.5	10.38	22.83	10.65	23.43	-2.62
500	10.60	21.19	11.28	22.55	-6.42
1000	13.01	13.01	17.58	17.58	-35.13

reduces because of the shorter combinational idle time and eventually becomes comparable to E_{oh} resulting in the convergence point around 400kHz in Fig. 4.21. At clock frequencies above 400kHz, $E_{oh} > E_{sav}$ and the power consumed by the Cortex-M0 when

using SCPG exceeds that of the Cortex-M0 without power gating. This maximum applicable clock frequency of sub-clock power gating with symmetric virtual rail clamping is 400x higher than the convergence point seen with shut down power gating in Fig. 4.19 which was between 1kHz and 2kHz. In the intended applications of sub-clock power gating, if clock frequencies above and below 400kHz are required, the processor could be switched to no power gating mode by using the *nOverride* signal (Fig. 4.4) for clock frequencies above 400kHz.

The reason five test chips were used for the data shown in Fig. 4.21 is to compare results across multiple dies and, as can be seen, the measurements all follow the same trend. The spread between plotted points can be explained by die to die process variation. The average power and energy per operation across the five test chips is shown in Table 4.4. In the final column the percentage saving achieved when using the proposed technique is stated. As can be seen, the proposed technique saves up to 67% of the energy compared to without power gating and demonstrates sub-clock power gating's ability to improve energy efficiency for a circuit operating at low clock frequencies. At 455kHz, the processor would need to switch to no power gating with the *nOverride* signal to remain in the lowest energy mode of operation. The measurements were also repeated at 0.8V to investigate variation with voltage and results showed the same trend in power saving.

Wireless Sensor Node Program

To investigate the utility of sub-clock power gating in one of the target applications a second test program is used. The program is an algorithm found in a real wireless sensor node used in the 'Next Generation Energy Harvesting Electronics' project which tracks the resonant frequency (between 42Hz and 55Hz) for a vibrational energy harvester [164]. The same five chips as were used with the Dhrystone program were used for the tuning program and the results are presented in Fig. 4.22. The average power values can be seen in Table 4.5. In the real application, an analogue to digital converter (ADC) is used to measure the acceleration on an accelerometer at a rate of 2kHz and every new reading triggers an algorithmic computation on the core processor. Over a set of 1000 samples the processor is capable of calculating the current frequency of vibration which is used to set a stepper position on the energy harvester. In the fabricated Cortex-M0 an example set of 1000 samples from a 48Hz vibration source is loaded into the SRAM to emulate obtaining a new reading from the ADC. Per ADC sample, the tuning program loops around a maximum of 85 instructions, therefore at a sampling rate of 2kHz the Cortex-M0 could operate at 200kHz without missing a new sample. At 200kHz, without sub-clock power gating the processor would consume 45pJ/operation and with sub-clock power gating the processor consumes 33.20pJ/operation, representing a 1.4x improvement in energy efficiency.

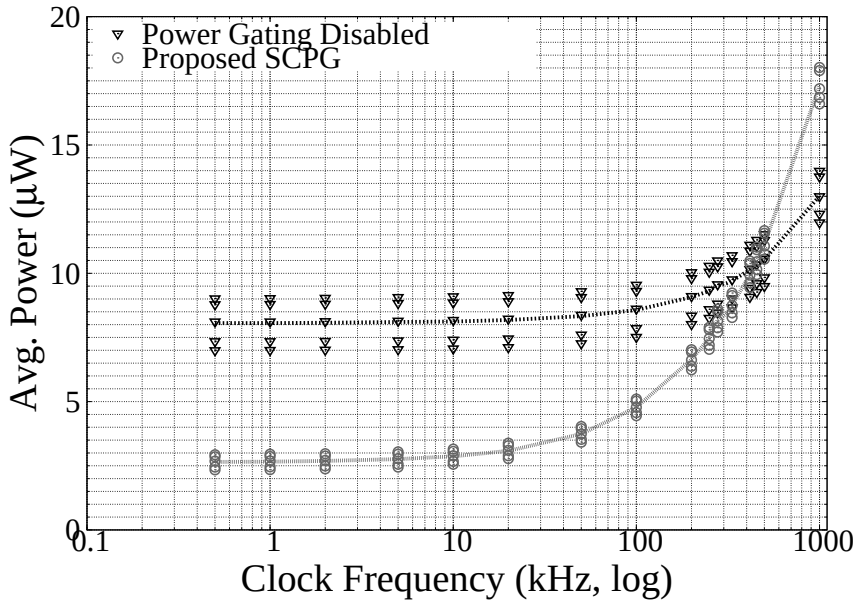


Figure 4.22: Tuning Program - Measured power of ARM Cortex-M0 at varying clock frequency, V_{dd}=0.7V

Table 4.5: Tuning Program - Average measured power and energy over five test chips with power gating disabled (No-PG) & sub-clock power gating (SCPG)

Clock Freq. (kHz)	**No Power Gating**		**Proposed SCPG**		
	Power (uW)	**Energy (pJ)**	**Power (uW)**	**Energy (pJ)**	**Saving (%)**
0.5	8.06	16117	2.65	5300	67.11
1	8.06	8061	2.66	2660	66.99
2	8.07	4035	2.69	1342	66.72
5	8.10	1619	2.76	551.0	65.97
10	8.12	812.4	2.87	286.6	64.72
20	8.18	408.8	3.08	154.2	62.27
50	8.33	166.5	3.74	74.71	55.14
100	8.57	85.74	4.78	47.84	44.20
200	9.07	45.33	6.64	33.20	26.77
250	9.31	37.26	7.48	29.91	19.72
263.2	9.45	35.90	7.91	30.07	16.23
277.8	9.53	34.30	8.17	29.41	14.26
312.5	9.62	30.78	8.45	27.05	12.14
333.3	9.72	29.18	8.77	26.31	9.83
357.1	9.84	27.56	9.12	25.53	7.34
384.6	9.98	25.94	9.52	24.74	4.60
416.6	10.13	24.32	9.97	23.94	1.59
454.5	10.32	22.71	10.50	23.10	-1.74
500	10.54	21.09	11.12	22.24	-5.44
1000	13.01	13.01	17.31	17.31	-33.06

4.4.4 Ground Bounce Analysis

A potential concern when applying power gating is the ground bounce that is induced on the always-on supply rail [3, 42]. As discussed in Chapter 1, Section 1.4.1, this ground

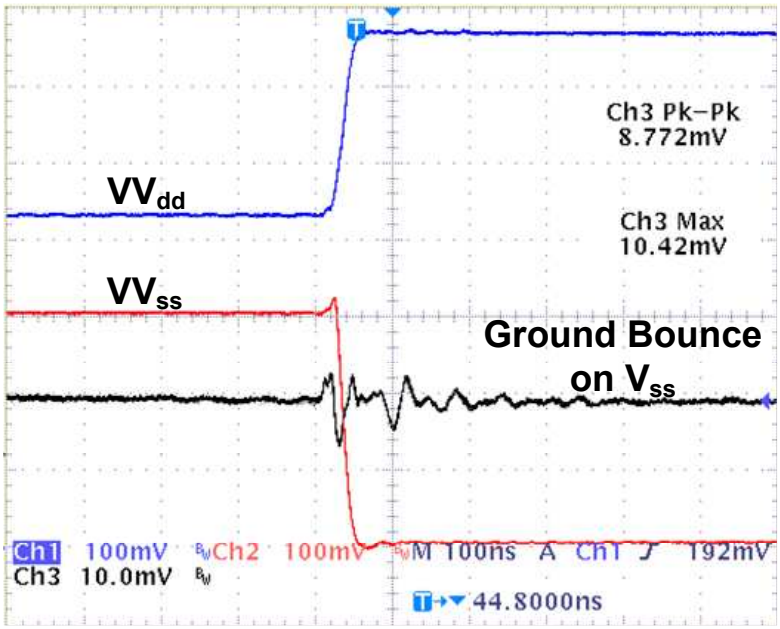


Figure 4.23: Measured ground bounce on the always-on V_{ss} supply rail

bounce occurs as a result of the rush of current that is required to restore the virtual rails, and can potentially cause signal integrity issues and corruption to registers in the always-on areas of the digital circuit. The ground bounce is measured from the chip when using sub-clock power gating with symmetric virtual rail clamping to investigate its significance. Fig. 4.23 shows the measured ground bounce on the V_{ss}. The maximum amplitude of the measured ground bounce is approximately 10mV, which for a supply of 700mV is just over 1%. At a ground fluctuation of this magnitude, the register states will not be affected by the ground bounce induced by the proposed technique. This correlates well with the low ground bounce magnitudes measured in the sub-clock power gating technique in Chapter 3 due to the small size of the power gated block. The application of the symmetric virtual rail clamping as part of the sub-clock power gating technique has, however, helped in controlling the effect of in-rush current on ground bounce. This is because less energy is required for recharging the virtual rails and because signal glitching has been eliminated as seen from comparison of the virtual supply waveforms of shut down power gating, Fig. 4.17, and symmetric virtual rail clamping, Fig. 4.15.

The proposed SCPG technique is primarily targeted for small embedded processors such as the Cortex-M0, and therefore a problem with ground bounce is not envisaged. However, in larger designs, the magnitude of the ground bounce increases [42] and may pose a more significant problem. Using fewer power gates can help to limit in-rush current as the virtual supply rail is 'trickle' charged [3] but comes at the cost of increased wake-up time. By using the additional *nPG* control signal previously mentioned in Section 4.3, the number of active power gates was varied to investigate how fewer power gates affects the charge-up time in the proposed symmetric virtual rail clamping technique. As shown in Fig. 4.24, one quarter of the power gates can restore the virtual rails in

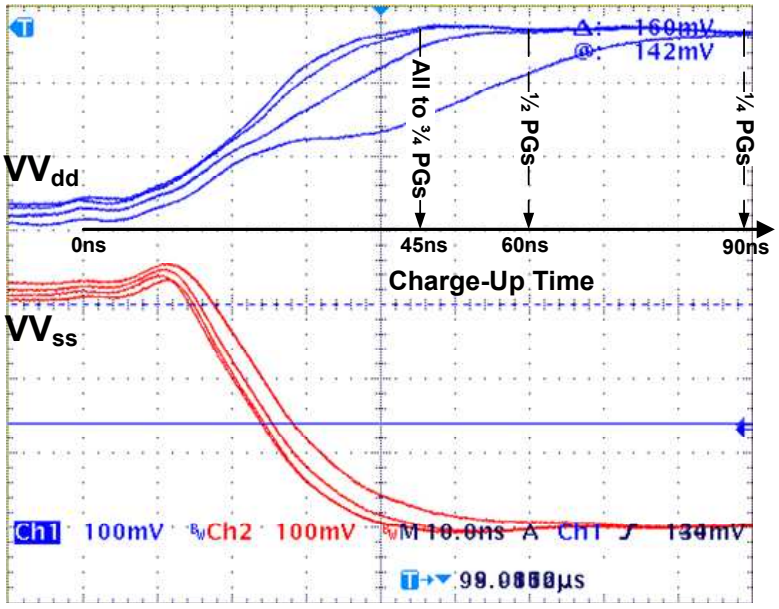


Figure 4.24: Measured charge-up time with varied number of active power gates (PGs) in proposed SCPG with symmetric virtual rail clamping

90ns, as opposed to 45ns with all 24 power gates. This shows the little impact reducing the number of power gates has on the wake-up time in symmetric virtual rail clamping enabling the application of long duty cycles in the sub-clock power gating technique.

4.5 Concluding Remarks

Wake-up energy can have a detrimental impact on the use of power gating over very short periods and is primarily contributed to by recharging of the virtual supply rail(s) and signal glitching from logic re-evaluation. This chapter has investigated the use of a new power gating technique, which can be used in place of traditional shut down power gating, to reduce power mode transition energy cost, improve the energy efficiency and extend the applicable frequency range of the sub-clock power gating technique, introduced in Chapter 3. In this chapter, both symmetric virtual rail clamping and sub-clock power gating have been validated experimentally.

The power gating technique proposed in this chapter reduces the voltage of the power gated logic by two threshold voltages, to a subthreshold voltage, rather than shutting down completely during the sleep mode as is the case in conventional power gating. To achieve this reduced voltage, a pair of NMOS and PMOS transistors are used on the V_{dd} and V_{ss} supply rails for symmetric virtual rail clamping of the power and ground supplies. The proposed power gating technique has been incorporated with sub-clock power gating and implemented on an ARM Cortex-M0 microprocessor. The ARM Cortex-M0 has been fabricated using a 65nm technology library. The additional steps required to prepare a design for implementation with sub-clock power gating previously summarised

in Chapter 3 have been described in detail and considerations that must be made during the physical layout have also been discussed. A simple circuit that enables duty cycle modulation to be achieved on chip has also been shown such that power savings achieved with sub-clock power gating can be maximised. Experimental validation of the sub-clock power gating technique proposed in Chapter 3 confirms that significant power savings are achievable when operating at low clock frequencies due to the combinational idle time that occurs within the clock period. Experimental comparison of the proposed symmetric virtual rail clamping technique and conventional shut down power gating shows the proposed technique achieves better energy efficiency at all frequency points and enables a 400x improvement in sub-clock power gating's applicable clock frequency range, allowing it to be used up to a clock frequency of 417kHz, instead of 1kHz as in the case of shut down power gating.

Chapter 5

dRail: A Physical Layout Technique for Power Gating

In Chapters 3 and 4, two new techniques called sub-clock power gating and symmetric virtual rail clamping have been demonstrated for minimising leakage power during the active mode in embedded processors for applications where performance is not the primary concern. The nature of the sub-clock power gating technique requires the combinational logic to be separated from the sequential logic into its own power domain, as was shown in Fig. 4.4. This is necessary because the combinational logic's power supply is disconnected through the use of power gating whilst the sequential logic's power supply remains always-on. In Chapter 4, Section 4.3 the physical implementation of an ARM Cortex-M0 with the sub-clock power gating technique using symmetric virtual rail clamping was described in detail for the fabrication of a silicon test chip. The separation of the combinational and sequential logic in the sub-clock power gating architecture was mapped onto the ARM Cortex-M0 and the physical layout reflected this separation by creating voltage areas. A combinational logic voltage area was used in the centre of the layout, and a sequential logic voltage area was used on its periphery, Fig. 4.8. The aim of this chapter is to gain deeper understanding of why voltage areas are required in the implementation of power gating and investigate how completely removing the placement constraint on standard cells in a power gating physical layout can improve energy efficiency. Firstly, the chapter provides a discussion of why a voltage area is needed in the physical layout of power gated circuits using modern commercial gate libraries and how this can affect energy efficiency. Secondly, to enable investigation into how an unconstrained placement affects energy efficiency of a power gated circuit a new physical layout technique called dRail is proposed. The dRail technique eliminates sharing of power and ground between abutted standard cells and breaks the traditional M1 power rails used for connecting standard cell power to make them 'derailed'. This allows both power gated and non-power gated cells to be placed adjacently in the physical layout unlike conventional voltage area layout. The dRail technique is compared

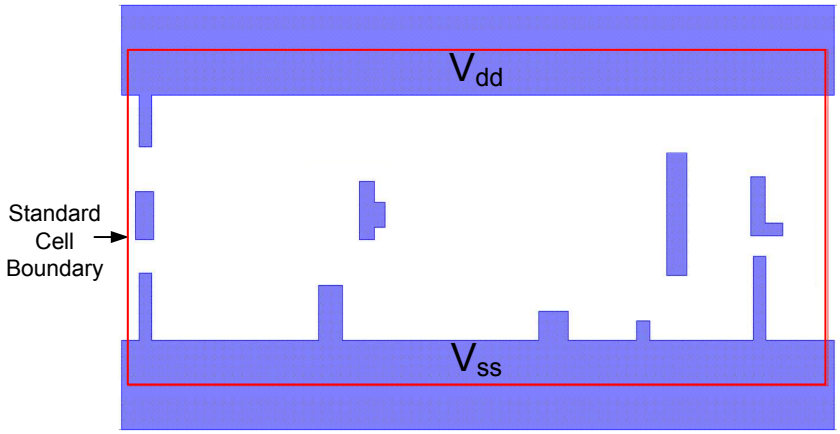


Figure 5.1: D-type flip-flop standard cell in TSMC 65nm ARM Artisan™ library [155]

with the voltage area layout approach in terms of area, signal routing length and power using a commercial gate library in two test cases.

This chapter is organised as follows. Section 5.1 provides background on the conventional voltage area physical layout approach and explains its limitations. Section 5.2 describes the proposed dRail technique, including the principles (Section 5.2.1) and how it is achieved in the physical design flow (Section 5.2.2). Two case studies are used for comparison of constrained voltage area layout and unconstrained dRail layout in Section 5.3: the Cortex-M0 with sub-clock power gating used in Chapter 4 and an ARM Cortex-A5 for comparison in a conventional power gating example. Finally, Section 5.4 concludes the chapter.

5.1 Motivation

Fig. 5.1 shows the standard cell layout of a D-type flip-flop gate taken from the TSMC 65nm ARM Artisan™ library [155] used for the test chip fabrication in Chapter 4 and is representative of standard cells in modern gate libraries. Labelled in Fig. 5.1 are the flip-flop's V_{dd} (power) and V_{ss} (ground) pins which are exposed on Metal 1 (M1); M1 will be assumed for the V_{dd} and V_{ss} connections of standard cells for the rest of this chapter. Also shown in Fig. 5.1 is the placement boundary of the D-type flip-flop and defines the space the standard cell occupies when placed in the physical layout. As can be seen the V_{dd} and V_{ss} extend horizontally beyond the boundary of the standard cell and is a common design technique used in standard cell gate libraries [2]. The reason for the extension of these connections is to simplify power routing and enable the sharing of power and ground between abutted standard cells when they are placed in the site rows of the physical layout. For example, consider a second identical D-type flip-flop placed flush to the right of the one shown in Fig. 5.1. As the V_{dd} and V_{ss} metal extend beyond the boundary of the flip-flops, both flip-flops would automatically share the V_{dd} and V_{ss}

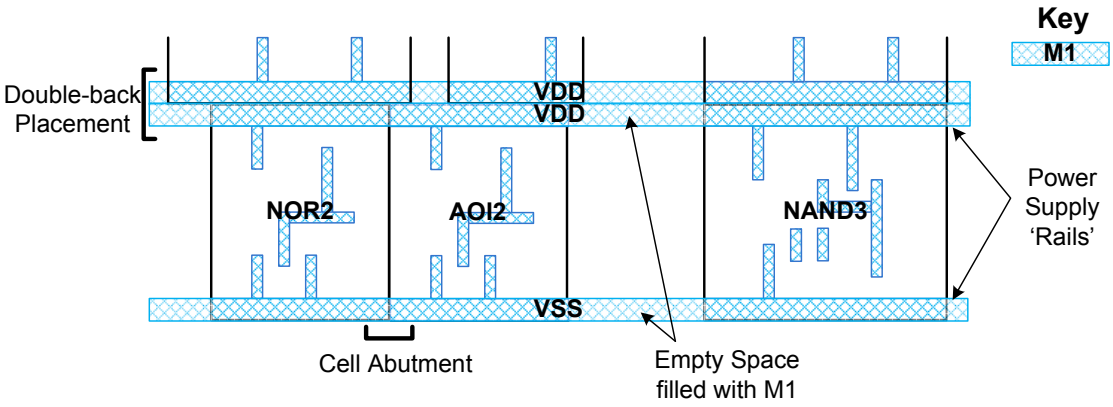


Figure 5.2: Conventional standard cell placement and power delivery with no power gating

voltages. The standard cell shown in Fig. 5.1 not only extends V_{dd} and V_{ss} horizontally for power and ground sharing with adjacent cells in the same placement row as itself, but also vertically for sharing with adjacent placement rows. This is because in modern multi-metal layer fabrication processes, the standard cell rows are placed in a way known as double-back where every other row is flipped [2], an example of which is shown in Fig. 5.2. Double-back placement eliminates the need for space to be left between site rows to stop V_{dd} and V_{ss} becoming shorted and well clearance.

In a conventional Application Specific Integrated Circuit (ASIC) physical layout with a single power and ground supply voltage, the abutment and double-back placement of standard cells is straightforward as all cells share the same supply voltage. The gates are placed in standard cell site rows within the layout by the EDA tool, such as Synopsys IC Compiler, without concern over their location as demonstrated in Fig. 5.2. As can be seen in this example, the NOR2 and AOI2 are abutted and as has already been shown, double-back placement is used with the next standard cell row. To ensure all the standard cells in the entire site row are connected to the V_{dd} and V_{ss} supplies, any empty space at the top and bottom of the site row is filled with M1 to create an uninterrupted M1 connection as shown. These continuous M1 connections that form at the top and bottom of the site rows are often referred to as standard cell power rails [4], and will be referred to as *rails* for short. The purpose of this is to ensure power reaches all standard cells in the placement row and is because power is often delivered from higher metal layers through Vias to the rails for the standard cells [2]. In a power gating design, however, this layout technique introduces a problem. For simplicity, assume a power gated circuit with a single switched power supply and one power domain. All the standard cells that are to be power gated need to be connected to the switched power supply while non-power gated cells must connect to the true supply (Chapter 1, Section 1.4.1) [3]. Due to the inherent supply sharing that occurs between abutted standard cells (Fig. 5.1), and the standard cell power supply rails created at the top and bottom of a placement row (Fig. 5.2), placing both switched and unswitched standard

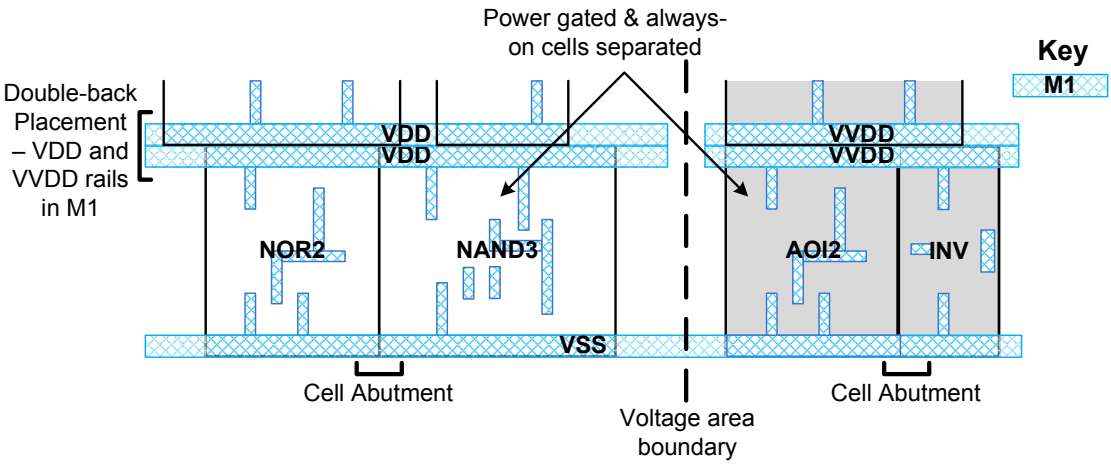


Figure 5.3: Example of standard cell separation and power delivery with power gating and voltage area

cells together would cause the two power supplies to become shorted. To overcome this problem the voltage area layout approach [3, 47, 94] has been proposed and has already been introduced in Chapter 1, Section 1.4.1.1. The primary advantage of using a voltage area is the ability to use conventional standard cells and double-back layout [3, 94]. Voltage area layout is therefore the most common method of implementing power gating in a physical design flow, is well supported by EDA tools [3, 47] and has consequently been used throughout this thesis. The physical layout of the fabricated Cortex-M0 in Chapter 4, for example, used three voltage areas to achieve the power gating needed in the sub-clock power gating technique: one for the combinational logic, one for the sequential logic and an always-on voltage area for the power gating control (Fig. 4.6).

An example of standard cell placement in a power gating physical layout using the voltage area approach is shown in Fig. 5.3. In this example a single switched VV_{dd} rail is assumed. The shaded standard cells shown in Fig. 5.3 are power gated whereas the unshaded standard cells remain always-on. To enable the use of power gating a voltage area is introduced to the right of the figure to group the power gated cells together, which separates them from the always-on cells as shown. This grouping enables always-on cells outside of the voltage area to be abutted and double-back as you would in a conventional physical layout, like Fig. 5.2, because all the standard cells connect to the same power supplies. The same can be done with the standard cells inside the voltage area. Unlike Fig. 5.2 though, a continuous V_{dd} M1 rail cannot be created across the top of the entire site row, and instead the voltage area forces a break in this rail as shown in Fig. 5.3. This break is required to stop the two power supplies from becoming shorted and means inside the voltage area the top of the standard cell row has an M1 VV_{dd} rail while outside of the voltage area the standard cells connect to an M1 V_{dd} rail. The break in the M1 power rail on both sides of voltage area boundary introduces a requirement for a small space or *guard band* to be left between both regions of the layout

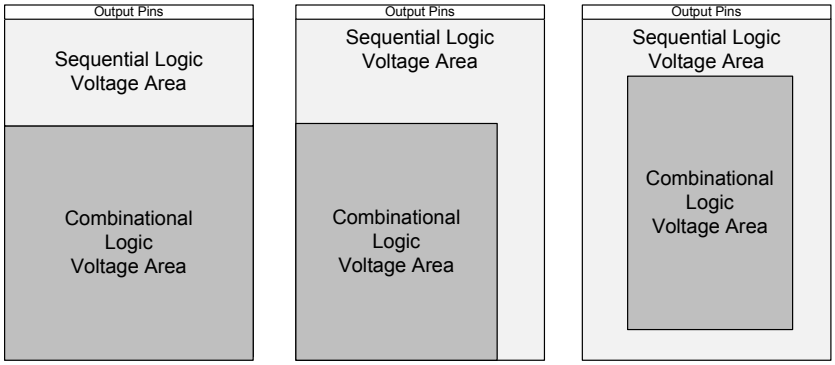


Figure 5.4: Different combinational and sequential voltage area locations for Cortex-M0 from Chapter 4

to ensure metal spacing rules are conformed to. This introduces a small area overhead as this space cannot be used for standard cell placement. As only the V_{dd} is switched in this example, the V_{ss} is common to both the always-on and the power gated standard cells and so it is possible for the V_{ss} rail to be connected between the two areas of the physical layout. A similar layout can be employed for a single switched VV_{ss} supply rail or when both V_{dd} and V_{ss} are switched for the power gated logic gates.

It was explained in Chapter 1, Section 1.4.1.1 that the location of a voltage area must be mindful of the relationship of standard cells inside and outside of the voltage area [3]. This is because the physical separation of standard cells, as shown in Fig. 5.3, can cause a greater distance to arise between logically connected cells. This increase in distance has two major consequences [2]. Firstly, it can require the addition of extra or higher drive strength gates (with higher leakage) to reduce propagation delay and transition times on signals that need to traverse a greater distance. Secondly, signal routing length can be increased between connected logic gates. For example, for the physical layout of the ARM cortex-M0 in Chapter 4, Section 4.3 a simple experiment was conducted to determine a suitable location for the combinational logic voltage area. An implementation was created with no constraint on the combinational logic and compared with implementations that constrained the placement of the combinational logic to the bottom, a corner and the centre as demonstrated in Fig. 5.4. It was found that placing the combinational logic to the bottom increased standard cell area and routing by 3.3% and 55% respectively, and placing it in the corner increased standard cell area and routing by 1% and 20.3%. Constraining the combinational logic to the centre on the other hand, increased standard cell area and signal routing by 0.7% and 5.2%, which is why the placement shown in Chapter 4, Fig. 4.8 was chosen, and demonstrates the effect voltage area placement can have on area and routing. The consequence of these increases in area and routing is a reduction in energy efficiency. This is because more and/or larger logic gates introduce additional leakage and dynamic power and greater routing length increases the output capacitive load of a logic gate also increasing dynamic power [2]. It will be shown in Sections 5.3.1 and 5.3.2 that power can be increased by up to 9% due

to increased area and routing. This discussion demonstrates how, although the use of a voltage area simplifies the implementation of power gating in the physical layout, its inclusion and location can reduce energy efficiency of a digital circuit. Previous work has proposed an alternative to voltage area layout by using power gated rows [95, 133, 151]. This technique reduces the placement constraints in a power gated design but does not completely eliminate it, still requiring careful consideration of which rows should be power gated. Instead, this chapter investigates how completely removing the placement constraint associated with the use of a voltage area can reduce the area and routing overheads associated with the physical layout of a power gated circuit and how this can lead to improvements in energy efficiency. To enable this investigation though, a method of being able to place both power gated and non-power gated cells adjacently anywhere in the physical layout is required and the next section proposes a new physical layout technique called "dRail".

5.2 Proposed dRail Technique

The sharing of power and ground between conventional standard cells (Fig. 5.1) and use of M1 rails (Fig. 5.2) in a conventional layout means the physical layout of power gating forces the separation of power gated and non-power gated standard cells into monolithic voltage areas, Fig. 5.3. The main goal of the dRail technique proposed in this section is to bring the power gated and non-power gated cells together to eliminate the separation associated with using a voltage area and reduce standard cell area and signal routing overheads to improve energy efficiency. Three challenges arise when placing both power gated and non-power gated cells together, requiring the following three changes:

1. **Modified standard cells**
   To be able to place both power gated and non-power gated cells together, the standard cell layout shown in Fig. 5.1 must be modified such that the V_{dd} and V_{ss} pins do not overlap between adjacent standard cells to stop the switched and unswitched power supplies from becoming shorted

2. **Dual power supply rail routing**
   By eliminating sharing of power supplies between adjacent cells, the continuous M1 rail conventionally used, Fig. 5.2, is lost, and the cells are 'derailed' meaning a method to route power to all cells is required. A routing channel is introduced between placement rows and switched and unswitched supplies are routed to all site rows: one over the standard cells and one between the placement rows

3. **Power Hook-up**
   Two power supplies are made available through dual power supply routing but each standard cell in the placement row must connect to the correct supply, this

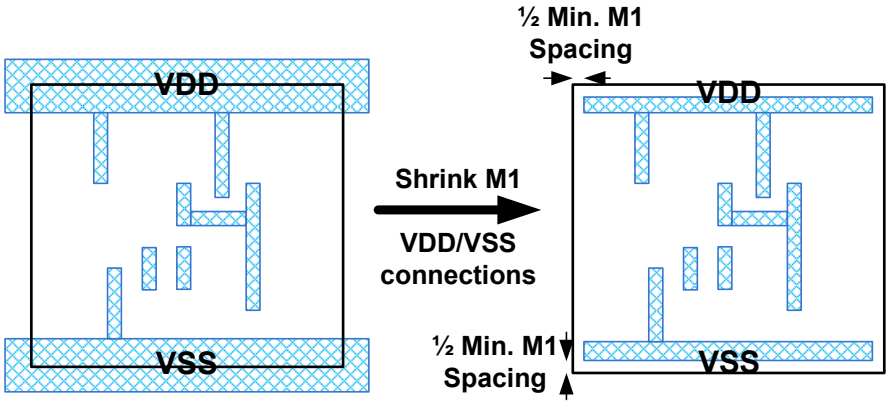


Figure 5.5: Shrinking of V_{dd} and V_{ss} pins to stop power and ground sharing

is achieved with either a Via or a small stub between the standard cells and the power rails

In this section, the method for implementing these three changes in the physical layout and how they are incorporated into a conventional power gating physical design flow is discussed. A number of considerations that must be made when using the dRail layout technique are also highlighted.

5.2.1 dRail Layout

The first challenge of the dRail layout technique is the ability to place both power gated and non-power gated cells adjacent to one another without the switched and unswitched power supplies becoming shorted. To achieve this the conventional standard cell layout shown in Fig. 5.1 must be altered such that the V_{dd} and V_{ss} pins do not overlap between adjacent standard cells. The proposed alteration is shown in Fig. 5.5. Instead of the V_{dd} and V_{ss} M1 pins extending out of the boundary of the standard cell, the new standard cells confine the pins inside the placement boundary. Both the V_{dd} and V_{ss} pins are shrunk to stop sharing on both power and ground and enables the dRail layout technique to be used for a switched V_{dd} and/or a switched V_{ss}. To ensure the alterations shown in Fig. 5.5 do not introduce M1 design rule spacing violations between abutted standard cells, the V_{dd} and V_{ss} pins must be taken inside the cell boundary by at least $\frac{1}{2}$ the M1 design rule spacing. By shrinking the power and ground pins of the standard cells in the gate library, each standard cell now has an independent V_{dd} and V_{ss} connection in the physical layout, and placing two cells next to each other no longer results in the automatic sharing of power and ground. Instead, this enables independent connections to be created between a standard cell and its respective power supply. It should be noted that the only change made to the standard cells is the cropping of the V_{dd} and V_{ss} pins, and so the logic gate is otherwise completely unchanged. Attributes such as the standard cell's internal functionality and boundary remain identical meaning the modifications

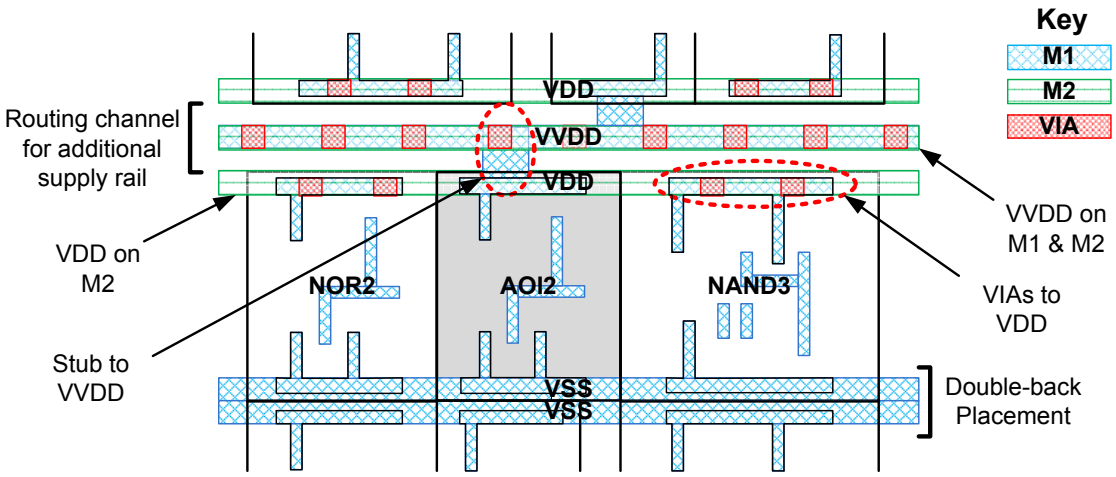


Figure 5.6: Power routing and hook-up in the dRail layout for a single VV_{dd}

made have no overhead in terms of area, performance or power. Furthermore, as the power and ground connections are only shrunk, the cells can be used in a traditional placement flow by routing continuous M1 rails across the top and bottom of the cell row to revert back to a conventional placement. This is advantageous as duplicates of the original standard cells are not needed and, once the modifications have been made to the standard cells for compatibility with a dRail layout approach, they can be used for both dRail and conventional layout. This also has the added advantage of being able to mix both dRail placement with conventional placement and will be shown in Section 5.3.2.

The modified standard cells enables both power gated and non-power gated cells to be brought together in the physical layout because automatic power and ground sharing has been eliminated. However, the use of M1 rails is forbidden as the switched and unswitched supplies of both types of cells would become shorted. Instead, a method of dual supply rail routing is used which enables both supplies to be routed to all standard cells. To demonstrate this technique, a simple example is given with a single switched VV_{dd} in Fig. 5.6. Power gated cells are shown as shaded in Fig. 5.6 and notice that the elimination of sharing between V_{dd} pins with the new modified standard cells allows it to be placed abutted with always-on cells. For dual power supply rail routing, firstly, instead of using conventional double back placement between adjacent placement rows, as in Fig. 5.2, the placement rows are separated and a routing channel is introduced. This provides space for both the V_{dd} and VV_{dd} to be routed to the site rows so that both supplies are available for connecting to anywhere in the physical layout. One supply is routed over the standard cells while one is routed between the rows. The supply that is routed over the standard cells, which in this example is V_{dd}, is positioned over the V_{dd} pins of the standard cells and is routed in Metal2 (M2) to stop the shorting of the standard cell M1 V_{dd} pins. The supply that is routed between the placement rows, VV_{dd} in this case, is routed on both M1 and M2 as there is no concern of the switched and unswitched supplies becoming shorted. It is equally feasible for the power supply rails

to have been routed with the VV_{dd} over the standard cells and the V_{dd} between the placement rows.

With both supplies available for connection in the physical layout the standard cells must then be 'hooked up' to their respective rail. Depending on which power supply a standard cell connects to, the power hook-up is done with either (1) a Via between the M1 V_{dd} pin of the cell and the M2 rail that is routed over the standard cells, or (2) an M1 stub between the V_{dd} pin of the cell and the rail between the placement rows. In Fig. 5.6, the VV_{dd} is routed between the placement rows and therefore the shaded power gated cell is connected to it with an M1 stub. The V_{dd} supply is conversely routed over the cells and the always-on cells connect to it with Vias. In this example only a switched VV_{dd} is considered and therefore the V_{ss} is common to both the power gated and non-power gated cells. This enables the versatility of the new modified cells, Fig. 5.5, to be capitalised on by using conventional double-back placement and M1 rail for the V_{ss} supply as shown in Fig. 5.6. It should be noted though, that the dual power supply rail routing and power hook-up given here for a switched VV_{dd} supply is fully applicable when using a switched VV_{ss}. Therefore, if both V_{dd} and V_{ss} are switched, as is the case in the symmetric virtual rail clamping proposed in Chapter 4, the power routing and hook-up can be employed on both sides of the site row.

5.2.2 dRail Design Flow

To achieve the dRail layout shown in Fig. 5.6 with both power gated and non-power gated cells being placed together, a number of modifications are required to a conventional voltage area physical design flow introduced in Chapter 1, Section 1.4.1.1. The voltage area physical design flow is duplicated in Fig. 5.7 and is shown together with the dRail physical design flow for comparison purposes. Before considering the changes that are required within the physical design flow to achieve dRail, first it should be noted that the UPF observes a minor change in comparison to when using a voltage area physical layout. Power domains, supply nets and isolation are defined in the same way as you would for a conventional power gating design flow (Chapter 1, Section 1.4.1.1), however, the power gates must be defined within the default 'Top' power domain. This is because the power gates would conventionally be defined for the power domain with which they are related to and therefore can only be instantiated in the EDA place and route tool once the voltage area corresponding to the power domain exists. In a dRail physical layout though, the voltage area is never created and therefore trying to instantiate the power gates would cause an error.

As can be seen in Fig. 5.7(b), the dRail physical design flow shows three key differences to a conventional power gating design flow, Fig. 5.7(a), which are highlighted and directly relate to the three changes discussed in Section 5.2.1. The first change, which modifies the standard cells in the gate library must occur before the design can be

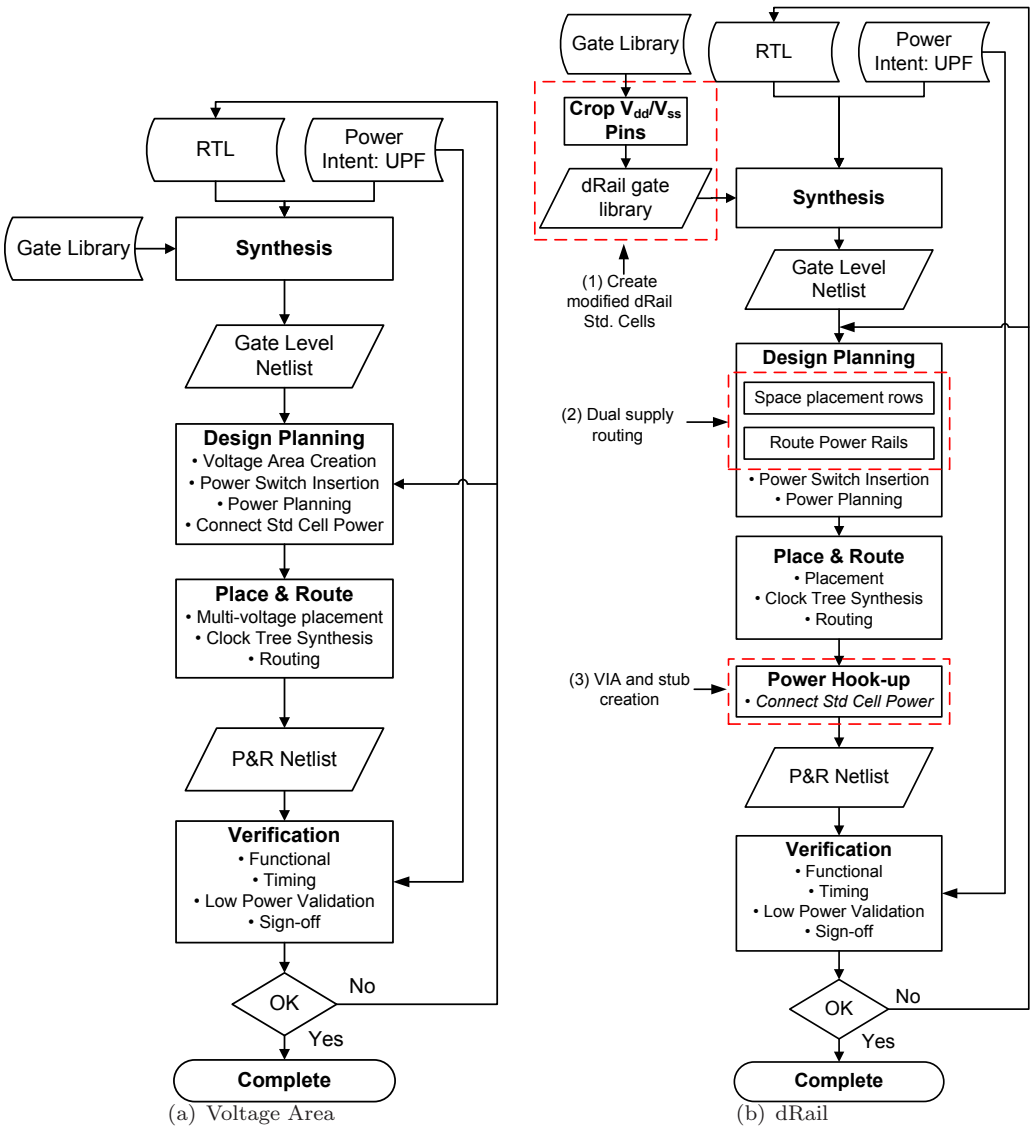


Figure 5.7: Power gating physical design flow for (a) traditional voltage area (b) dRail

synthesised. Once this step is done though, this flow step does not need to be repeated for future designs because, as discussed in Section 5.2.1, the modifications made are compatible with both dRail and conventional layout approaches. An example of how the modifications can be made to a gate library will be given with the experimental results in Section 5.3. The second and third changes required in a dRail layout conversely happen within the physical layout steps of the design flow. Most notably, the second change which spaces the site rows for the routing channel and then routes the power on Metal2 replaces the voltage area creation used in a conventional power gating flow as one is not required in a dRail physical layout. The creation of the spaced site rows and dual supply routing required in dRail can be fully automated in the implementation scripts using commands available in the EDA tool and an example of these steps will be given in Section 5.3.1 when discussing the implementation of a sub-clock power gated ARM

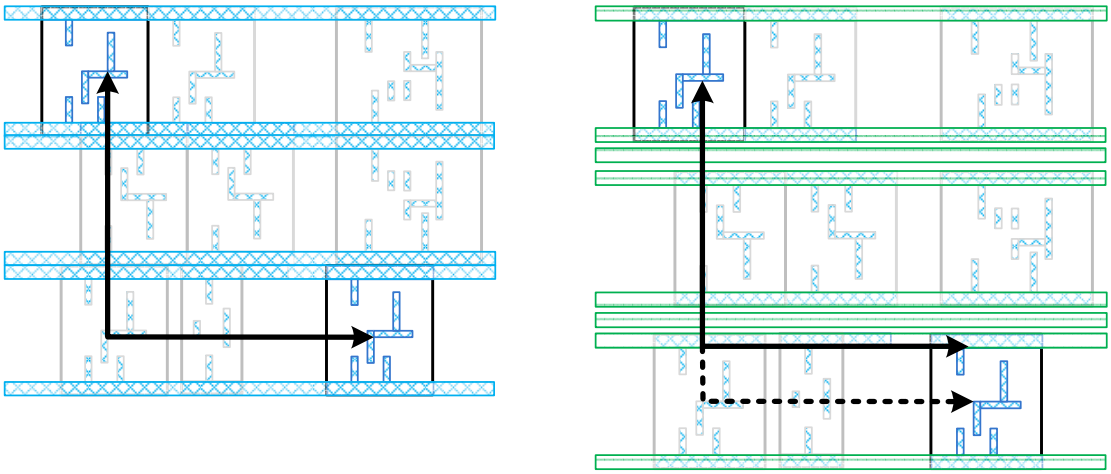


Figure 5.8: Spreading of standard cells due to inclusion of routing channel

Cortex-M0 microprocessor using dRail layout. The third power hook-up change which creates the Via and stub connections between the standard cells and the power rails, as shown in Fig. 5.6, is performed after place and route. The reason this step happens after the place and route has completed is because the EDA tool may move the location of the standard cells during the place and route stage. In the case of a dRail layout the Via and stub connections are bespoke for each standard cell and must therefore be done once their locations are finalised and known. The power hook-up performed for each of the standard cells can be fully automated in the implementation scripts.

5.2.3 Design Considerations

The dRail layout technique introduces four design considerations during its implementation. Firstly, the N-well and P-well of power gated and non-power gated cells are joined and are therefore assumed to be common. This means that individual control of the wells in power gated and non-power gated cells cannot be achieved in a dRail layout. Secondly, the utilisation of M2 for power rail routing can introduce routing blockage, which can deter the EDA routing tool, such as Synopsys IC Compiler, from using M2 for signal routing resulting in greater use of higher metal layers. This can have a negative impact as higher metal layers exhibit higher capacitance, increasing the output load of logic gates [2] which in turn increases dynamic power. Thirdly, the use of a routing channel between site rows for an additional supply rail introduces area overhead in the form of placement blockage as it cannot be used for standard cell placement. In the case of a single switched power supply rail only one routing channel is required for every two placement rows as the opposing non-switched power supply can still be placed in traditional double back placement as shown in Fig. 5.6. However, the area overhead is doubled if both power supply rails are switched. This area overhead is dependent on the process technology used and an explanation of the area overhead calculation will be given in Section 5.3.1 during the implementation of the ARM Cortex-M0 processor

with sub-clock power gating. Finally, the routing channel introduces a further effect in the form of standard cell spreading. In the case of conventional double-back placement, the standard cells cannot be placed any closer together as the placement rows are abutted. In a dRail layout the placement rows are separated by a finite distance which means standard cells in different placement rows experience a greater distance between each other due to this separation. This is demonstrated diagrammatically in Fig. 5.8 where the distance between two standard cells in different rows is shown to increase because of the extra vertical distance introduced by the routing channels in the dRail layout. This can have a negative impact as signal routing length is increased and more or higher drive strength gates may be required to compensate for the additional distance the signals have to traverse.

5.3 Experimental Results

Two case studies are used to investigate how eliminating the placement constraint associated with the voltage area approach affects energy efficiency. Firstly, the dRail technique is used on the sub-clock power gated Cortex-M0 microprocessor implemented in the test chip in Chapter 4. Secondly, dRail is used on an ARM Cortex-A5 microprocessor to demonstrate the proposed dRail physical layout technique on a traditional power gating strategy and show how bounded use of dRail can be used in a larger design to improve energy efficiency. All dRail implementations have been synthesised and fully place and routed using the dRail design flow described in Fig. 5.7(b), Section 5.2.1, whereas voltage area implementations used for comparison have been implemented using the conventional voltage area design flow, Fig. 5.7(a). All implementations have been done using the Synopsys EDA tool suite with UPF. In line with the first additional step of the dRail physical layout design flow, Fig. 5.7(b), the TSMC 65LP ARM Artisan^{TM} gate library used for the test chip in Chapter 4 has been modified by shrinking the V_{dd} and V_{ss} pins as described in Section 5.2. It is observed that in a conventional double-back placement, the standard cells in the TSMC 65nm process technology have a 0.32μm wide M1 rail connection to the power rails. Therefore, to achieve similar current density along the rails in a dRail physical layout they needed to be a minimum of 0.16μm in width. This is because M1 and M2 have similar current densities and both are used to route the power to each standard cell in dRail layout, as opposed to just M1. This enabled the V_{dd} and V_{ss} pins to be cropped from the edge of the standard cell boundary by 0.16μm in the vertical direction whereas the pins are cropped by 0.045μm in the horizontal direction in accordance with the M1 spacing rules.

To implement the changes to the V_{dd} and V_{ss} pins of each standard cell, the library exchange format (LEF) of the TSMC 65nm gate library is manipulated directly. The LEF file is an industry standard specification developed by Cadence for representing the physical layout of the standard cells in the gate library and can be used by commercial

```
01:  MACRO INV_X0P5M_A12TR
02:    CLASS CORE ;
03:    ORIGIN 0 0 ;
04:    FOREIGN INV_X0P5M_A12TR 0 0 ;
05:    SIZE 0.6 BY 2.4 ;
06:    SYMMETRY X Y ;
07:    SITE sc12_cln65lp ;
08:    PIN A
09:      DIRECTION INPUT ;
10:      USE SIGNAL ;
11:      ANTENNAPARTIALMETALSIDEAREA 0.1643 LAYER M1 ;
12:      ANTENNAMODEL OXIDE1 ;
13:        ANTENNAGATEAREA 0.0489 LAYER M1 ;
14:        ANTENNAMAXSIDEAREACAR 3.35991825 LAYER M1 ;
15:      PORT
16:        LAYER M1 ;
17:          RECT 0.25 0.98 0.35 1.41 ;
18:      END
19:    END A
20:    PIN VDD
21:      DIRECTION INOUT ;
22:      USE POWER ;
23:      SHAPE ABUTMENT ;
24:      PORT
25:        LAYER M1 ;
26:          RECT 0.13 1.545 0.23 2.72 ;
27:          RECT -0.045 2.08 0.645 2.72 ;
28:      END
29:    END VDD
30:    PIN VSS
31:      DIRECTION INOUT ;
32:      USE GROUND ;
33:      SHAPE ABUTMENT ;
34:      PORT
35:        LAYER M1 ;
36:          RECT 0.13 -0.32 0.23 0.855 ;
37:          RECT -0.045 -0.32 0.645 0.32 ;
38:      END
39:    END VSS
40:    PIN Y
41:      DIRECTION OUTPUT ;
42:      USE SIGNAL ;
43:      ANTENNAPARTIALMETALSIDEAREA 0.5053 LAYER M1 ;
44:      ANTENNADIFFAREA 0.142625 LAYER M1 ;
45:      PORT
46:        LAYER M1 ;
47:          RECT 0.41 0.46 0.51 0.87 ;
48:          RECT 0.41 1.5 0.51 1.91 ;
49:          RECT 0.45 0.77 0.55 1.6 ;
50:      END
51:    END Y
52:  END INV_X0P5M_A12TR
```

```
PIN VDD
  DIRECTION INOUT ;
  USE POWER ;
  SHAPE ABUTMENT ;
  PORT
    LAYER M1 ;
      RECT 0.13 1.545 0.23 2.24 ;
      RECT 0.045 2.08 0.555 2.24 ;
  END
END VDD
PIN VSS
  DIRECTION INOUT ;
  USE GROUND ;
  SHAPE ABUTMENT ;
  PORT
    LAYER M1 ;
      RECT 0.13 0.16 0.23 0.855 ;
      RECT 0.045 0.16 0.555 0.32 ;
END
END VSS
```

Figure 5.9: Changes to LEF definition of an inverter logic gate in the TSMC 65nm ARM Artisan™ Library [155] for dRail technique

EDA tools to derive the standard cell abstractions used in the place and route tools. The script used to crop the V_{dd} and V_{ss} pins is listed in full in Appendix C.2 but a summary is given here to understand how the changes are made in the LEF file using an inverter gate example which is listed in Fig. 5.9. The LEF file is read line by line using a Perl script due to its efficiency with text file manipulation. A line beginning with the keyword *SIZE*, such as line 5 in Fig. 5.9, defines the width and height of the cell currently being defined. In the inverter given, the width is 0.6 and the height is 2.4. These dimensions are important to know for calculating the allowable dimensions of the new cropped V_{dd} and V_{ss} pins and are therefore captured into *width* and *height* variables. The Perl script

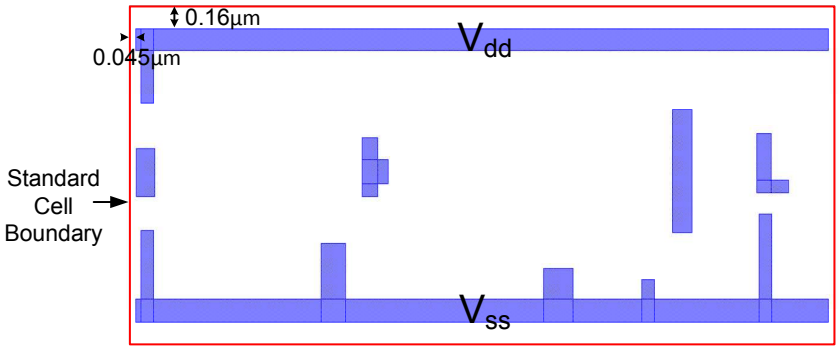


Figure 5.10: D-type flip-flop from Fig. 5.1 modified for dRail

also sets a *pin* variable to track if and which pin is currently being defined in the LEF file. For example, in the inverter LEF definition shown in Fig. 5.9, when line 20 "*PIN VDD*" is reached, the script is aware that the *VDD* pin is being defined. When line 29 "*END VDD*" is reached the pin variable is cleared so that the script knows the *VDD* pin is no longer being defined. This is important because if a V_{ss} or V_{dd} pin is not being defined the script simply prints out the current line without attempting any changes. If a V_{dd} or V_{ss} pin is being defined then the script moves forward, printing each line until the layer the pin is being defined on is known, e.g. line 25 Fig. 5.9. If the layer is M1 the script then determines on the subsequent lines if the shapes defining the pin need to be cropped. Each pin is made up of a set of rectangles which are defined using the *RECT* keyword as shown in lines 26 and 27 Fig. 5.9. Each *RECT* line defines in order: the left co-ordinate, bottom co-ordinate, right co-ordinate and top co-ordinate of a rectangle shape relative to the bottom left of the standard cell placement boundary, which is considered to be the co-ordinate (0,0). If any of these coordinates fall outside of the boundary then the script modifies the value to a value that resides within the boundary of the standard cell. For example, line 27 Fig. 5.9 defines a rectangle on the M1 layer and it can be seen that the left co-ordinate (-0.045) is negative corresponding to a value outside of the boundary, the right co-ordinate (0.645) is greater than the width of the cell (0.6) and the top co-ordinate (2.72) is greater than the height of the cell (2.4). Therefore, the two *RECT* statements are modified as shown in Fig. 5.9 by cropping the pins by 0.16μm and 0.045μm in the vertical and horizontal dimensions. An example of the D-type flip-flop standard cell that was shown in Fig. 5.1 modified for use with dRail is shown in Fig. 5.10.

5.3.1 Case Study 1: ARM Cortex-M0 with SCPG

The first case study used to investigate the advantages of an unconstrained placement in the physical layout of power gating is the sub-clock power gated ARM Cortex-M0 microprocessor used in Chapter 4. This test case is chosen because the sub-clock power gating technique necessitates the separation of the combinational and sequential logic

so that the combinational logic can be power gated, Fig. 4.4. As a result, in the physical layout of the sub-clock power gating technique the combinational and sequential logic must be constrained into separate voltage areas, as was shown in Fig. 4.8, when using a conventional voltage area power gating design flow. The sub-clock power gating technique therefore lends itself well to the dRail layout technique as it would enable the combinational logic cells to be power gated whilst simultaneously being allowed to intermingle with the sequential logic cells. In Chapter 4, the Cortex-M0 was implemented with the registers on a switched power supply rail and was done to enable the entire Cortex-M0 to be powered down when investigating other experiments on the test chip. In this section, to simplify the implementation of the Cortex-M0 the registers are reverted to the always-on power supply while the combinational logic is still power gated using symmetric virtual rail clamping (Fig. 4.4). As explained in Section 5.2.2, the power gates need to be defined in the top level power domain in the UPF to enable them to be instantiated in a dRail physical design flow. Therefore, the power gates defined for the combinational domain, the *domain* attribute of the *create_power_switch* command changes from the combinational power domain *M0_Comb_PD* as used in Chapter 4, Section 4.3 to the top power domain *Top_PD* as follows:

```
create_power_switch VVDD_M0_Comb_sw0 -domain Top_PD \
    -input_supply_port VDD VDD \
    -output_supply_port VVDD_M0_Comb VVDD_M0_Comb \
    -control_port sleep M0/HEAD_SLEEP[0] \
    -on_state on_state VDD !sleep
```

To provide an example of the dual supply rail routing in the dRail physical layout flow, the following was done to minimise the area overhead associated with the routing channel needed in the dRail layout. As explained, the standard cell V_{dd} and V_{ss} pins are cropped vertically by 0.16μm from the placement boundary, Fig. 5.10. In the physical layout using Synopsys IC Compiler, the placement rows must be snapped to a grid of routing tracks with a pitch of 0.1μm. Therefore, one placement site row is snapped to one routing track, *Tr1* in Fig. 5.11, the additional rail between the site rows is placed on the next routing track (*Tr2*) and the next placement row is on the third routing track (*Tr3*). For sufficient current density, the rail in the middle of the placement rows is 0.16μm in width and so extends in both directions by 0.08μm, Fig. 5.11. The M2 rail that is placed over the standard cell pins is placed 0.16μm away from the edge of the site row and so the space left between parallel M2/M1 rails (marked as Δ in Fig. 5.11) is 0.18μm. In the TSMC 65nm process used in the test cases this space is greater than the design rule spacing requirement of 0.1μm and so in this case the 0.2μm routing channel space in the layout is acceptable. If the design rule was violated then the space between the placement rows would have to be increased. The figure shown to the right of Fig. 5.11 demonstrates the complete site row spacing and rail creation used in the dRail physical layout.

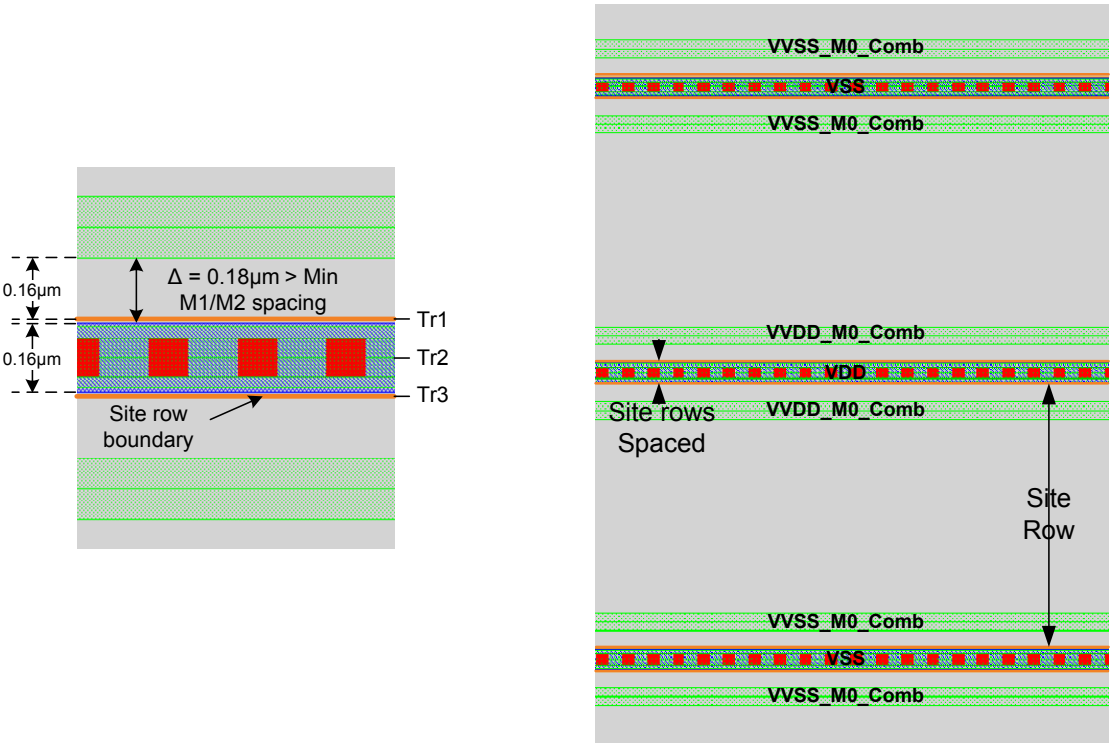


Figure 5.11: Site row spacing and Metal1 Metal2 rail creation

Fig. 5.12 shows a screen capture of the final placement and power connection of the standard cells in the dRail layout taken from Synopsys IC Compiler after the 'Power Hook-up' step in Fig. 5.7(b) has been completed. As can be seen, the standard cells are connected up to their respective power supplies using Vias and stubs in accordance with the architecture of a circuit using sub-clock power gating, Fig. 4.4. The register is connected to the always-on V_{dd} and V_{ss} using stubs between its power and ground pins and the supply rails that run between the site rows. Similarly the isolation gates which provide clamping between the combinational and sequential domains ('ISOL' in Fig. 4.4) also connect to the true supplies in the same way. The combinational gates on the other hand can be seen to connect to the VV_{dd} and VV_{ss} supply rails through the use of Vias. In addition to the standard cell power hook-up, the screen capture also shows a footer power gate to demonstrate how it provides power to the virtual supply rails in the physical layout. To verify the complete dRail layout does not introduce any spurious design rule spacing violations, the final layout is passed through Calibre by Mentor Graphics for design rule checks.

Fig. 5.13 shows a comparison of the sub-clock power gated ARM Cortex-M0 layout using a voltage area and dRail. On the left of Fig. 5.13 is the conventional voltage area layout approach of sub-clock power gating and as can be seen the combinational logic is constrained to a central voltage area such that it can be power gated with the sequential logic on its periphery. Furthermore, it can be seen that the isolation gates have been grouped around the boundary of the combinational and sequential voltage

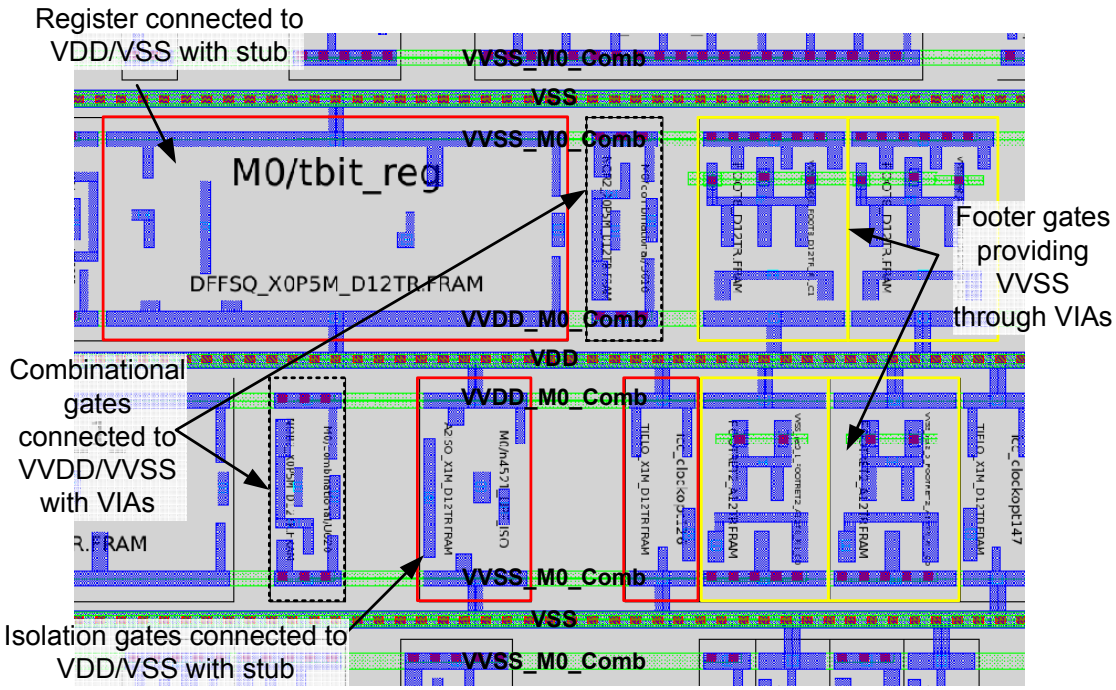


Figure 5.12: Power hook-up in the dRail layout

areas by the EDA tool to optimise placement. To the right of Fig. 5.13 is the same Cortex-M0 microprocessor but using the dRail layout technique. As can be seen, the use of dRail frees the EDA placement tool to place the combinational and sequential logic gates anywhere in the physical layout. In addition to this the isolation gates between the combinational and sequential logic are now interspersed within the layout.

A comparison of the areas in both voltage area and dRail layout is given in Table 5.1. By removing the placement constraint of the logic cells in the physical layout, the standard cell area of the entire design comes down from 43172μm^2 in the voltage area layout to 42047μm^2 in the dRail layout representing a 2.6% reduction. To appreciate the total placement area of the two physical layouts though, the power gating area cost of each technique must be taken into consideration. As discussed in Section 5.1, the voltage area layout requires a guard band which introduces an area overhead and in the implementation of the Cortex-M0 this guard band accounts for an overhead of 1601μm^2. In the case of the dRail layout the routing channels contribute to the power gating area overhead. For the sub-clock power gated Cortex-M0 implementation there are a total of 75 routing channels each of which is 0.2μm in width and 305.2μm in length. This amounts to an area overhead of 4578μm^2. As can be seen in Table 5.1, combining the standard cell areas of the two layouts with their area overheads, the voltage area has a total placement area of 44773μm^2 whereas the dRail layout has a total placement area of 46625μm^2 representing a 4.1% increase in total placement area. To observe the impact enabling an unconstrained placement has on routing length, the signal lengths on each metal layer are shown in Table 5.2 and represented graphically in Fig. 5.14. The EDA tool is configured to optimise routing for utilisation of lower metal layers as higher metal

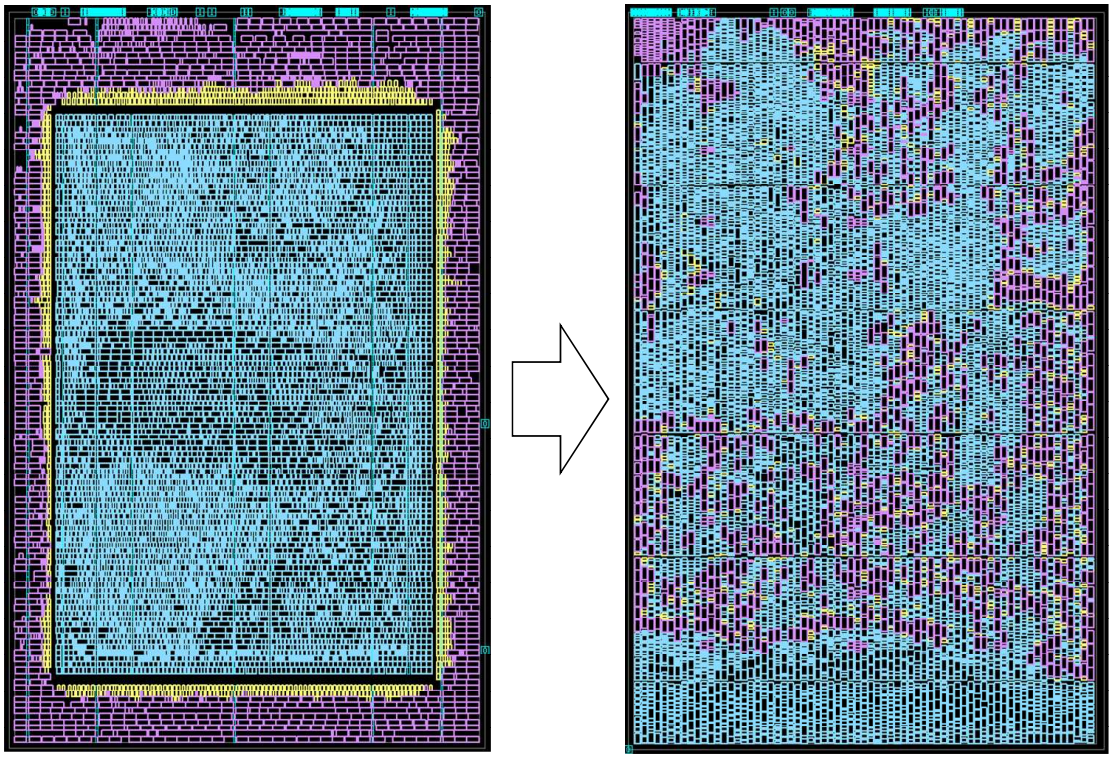


Figure 5.13: Physical layout of ARM Cortex-M0 with sub-clock power gating using traditional voltage area layout (left) and dRail (right)

Table 5.1: Area comparison of voltage area and dRail layout, Cortex-M0

	Voltage Area	**Proposed dRail**	**Difference (%)**
Std. Cell Area (μm^2)	43,172	42,047	-2.6
PG Area Cost (μm^2)	1,601	4,578	186
Total Placement Area (μm^2)	44,773	46,625	4.1

layers have greater capacitance resulting in increased capactive wire load [2]. As can be seen in Table 5.2, the total signal routing length has come down by 18.7% in the dRail case due to the ability to place combinational logic, sequential logic and isolation gates closer together. However, the routing blockage caused by the additional power routing on M2 discussed in Section 5.2.3, causes a reduction of 4% in the percentage of M2 utilisation for total signal routing. Most importantly though, the highest reductions in the total amount of routing performed on any metal layer are observed on Metal6 (24.3%) and Metal7 (17.6%). Higher metal layers have lower resistance but higher capacitance than lower metal layers making them ideal for the power network distribution but undesirable for signal routing as they increase the capacitive wire load on the outputs of logic gates [2]. The use of these metal layers can increase both signal transition delay and dynamic power and so their use should ideally be minimised within the physical layout.

To quantify how the reduction in standard cell area and total signal routing impact

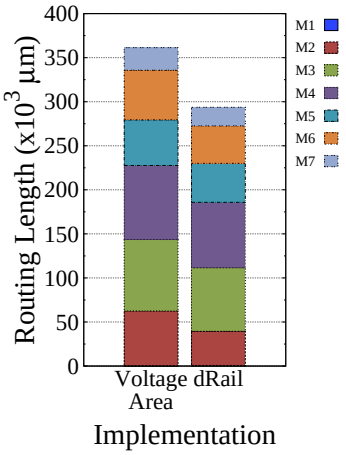


Table 5.2: Distribution of signal routing in voltage area and dRail layout, Cortex-M0

Metal Layer	Voltage Area (μm)	% total routing	Proposed dRail (μm)	% total routing	Diff. (%)
M1	381	0.1	639	0.2	67.7
M2	62,173	17.2	38,806	13.2	-37.6
M3	81,076	22.4	72,198	24.6	-11.0
M4	84,015	23.3	74,185	25.3	-11.7
M5	51,792	14.3	44,217	15.0	-14.6
M6	56,323	15.6	42,620	14.5	-24.3
M7	25,543	7.1	21,058	7.2	-17.6
Total	361,303	-	293,723	-	-18.7

Figure 5.14: Distribution of signal routing in sub-clock power gated Cortex-M0 using voltage area and dRail

Table 5.3: Sub-clock power gated Cortex-M0 average power of voltage area and dRail layouts, V_{dd}=0.7V

Clock Frequency (kHz)	Voltage Area (μW)	Proposed dRail (μW)	Saving (%)
1	4.37	4.20	3.9
10	4.54	4.38	3.5
50	5.34	5.20	2.6
100	6.33	6.18	2.4
250	9.12	8.90	2.4

energy efficiency of the sub-clock ARM Cortex-M0, both the voltage area and dRail layouts have been simulated using the Dhrystone benchmark and simulation flow presented in Chapter 3, Fig. 3.7. The sub-clock power gating mode of operation with symmetric virtual rail clamping (Chapter 4, Section 4.2) is simulated with a 200ns low period in the clock duty-cycle at a supply voltage of 0.7V over a range of clock frequencies and the results are presented in Table 5.3. As can be seen, improvements in energy efficiency are observed across all simulated frequency points. The reductions in both standard cell area and routing length enable up to 3.9% improvement in the power dissipation of the sub-clock power gated ARM Cortex-M0. This demonstrates that removing the placement constraint associated with using a voltage area can enable better energy efficiency to be achieved in the sub-clock power gating technique through reduction of standard cell area and signal routing.

5.3.2 Case Study 2: ARM Cortex-A5 Data Engine

The sub-clock power gated ARM Cortex-M0 test case used in the previous section is an extreme example of a placement constraint imposed by voltage areas in the physical layout. For this reason, the second test case, an ARM Cortex-A5 processor, investigates

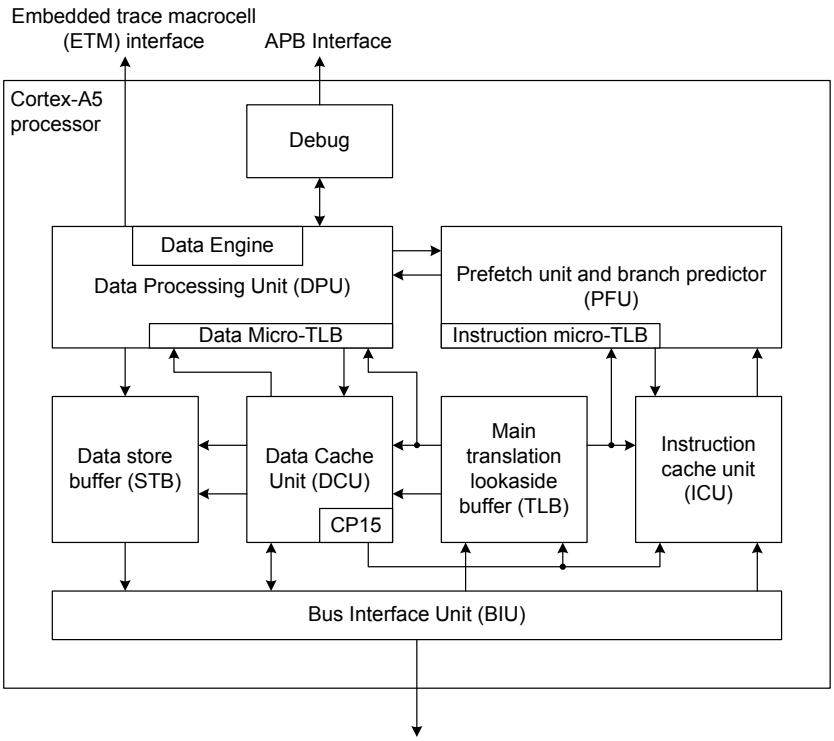


Figure 5.15: Top level block diagram of an ARM Cortex-A5 processor core [165]

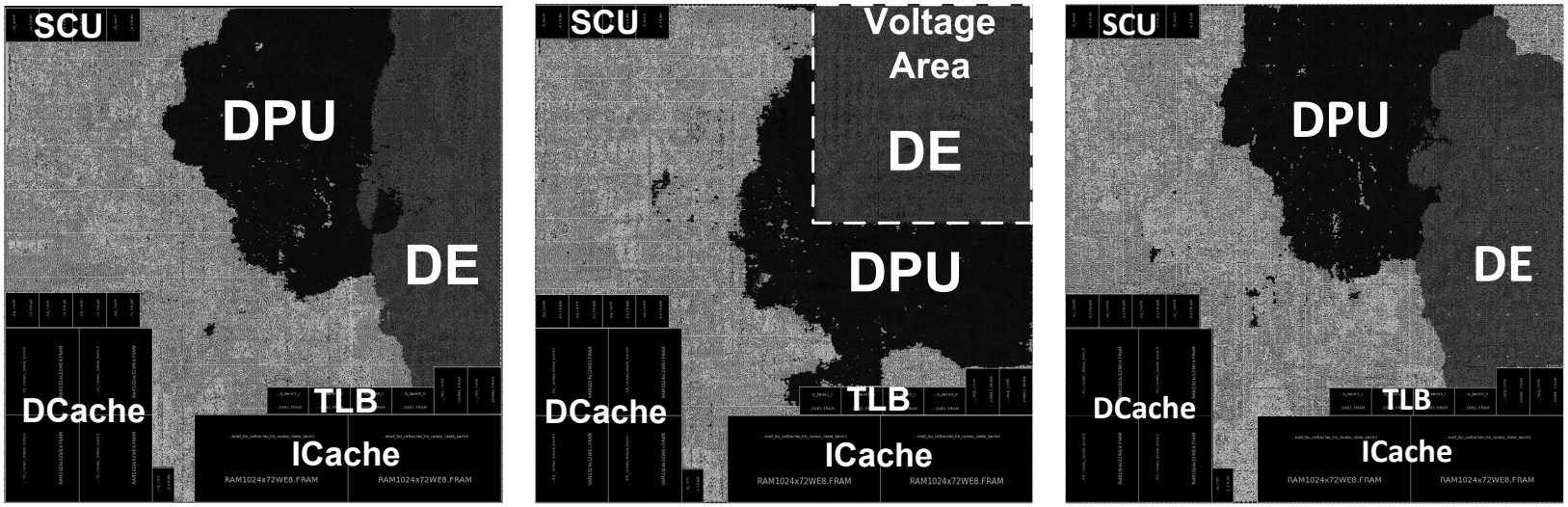
the advantages of an unconstrained placement in the physical layout of a conventional power gating example but also enables investigation of the utility of dRail in larger power gating designs. The Cortex-A5 is a high performance multi-core processor based on the 32-bit ARMv7^{TM} architecture supporting up to four individual cores each with an individual level-1 cache, which are kept coherent with the use of a Snoop Control Unit (SCU) [165]. A top level block diagram showing the basic functionality of a single Cortex-A5 core is shown in Fig. 5.15. As can be seen the Cortex-A5 employs a Harvard memory structure. The instruction cache unit (ICU) and prefetch unit (PFU) work together to obtain instructions which are then passed to the data processing unit (DPU) for processing. The processor is capable of executing 32-bit ARM and 16-bit and 32-bit Thumb instructions. The data cache unit (DCU) controls access from the DPU to the data RAMs while the store buffer (STB) holds store operations committed by the DPU. The bus interface unit (BIU) provides the interface between the processor core and external components. The data processing unit (DPU) forms the main decode and execution space of the processor and includes the general purpose registers and status registers as well as the arithmetic and logic unit. Part of the DPU is made up with the data engine (DE) which is an optional block of logic consisting of a floating point unit (FPU) and Neon^{TM} media processing unit. These two units are specifically designed to improve the performance of audio, video, 3D graphics, image and speech processing [165].

The ARM Cortex-A5 test case used in this section has been configured with a single

core, as opposed to multiple cores, with 16kB level-1 data (DCache) and instruction (ICache) cache and Translation Lookaside Buffer (TLB) cache. Although the snoop control unit (SCU) is used for cache coherency between multiple cores it is still present in unicore variants of the Cortex-A5 and so an SCU cache RAM is also included in the implementations. It has been shown that the floating point unit in a processor exhibits great potential for power gating during active execution [120], and consequently, in the implementation of the ARM Cortex-A5 the data engine has been made power gateable with a single switched VV_{dd} supply which is assumed to be controlled externally. The data engine consists of both combinational and sequential logic and is therefore unlike the sub-clock power gated ARM Cortex-M0 test case used in the previous section. However, as it forms part of the data processing unit, the data engine has a tight interaction with the data processing unit as will be shown later in the section. It should be noted, retention registers are not used in the implementation as the data engine is primarily an executional unit and it is assumed the data engine would only be shut down once the current execution task has been completed. All the implementations described, target and achieve the same 400MHz performance target and to ensure equal comparison, the location of the cache RAMs in the floorplan and silicon core area (1245μm x 1244.2μm) are kept fixed in all implementations. The site row creation for the dRail implementations reported in this section are created in the same way as was described in Section 5.3.1. Two experiments are carried out to investigate the dRail physical layout technique: the first compares voltage area and dRail layouts of the Cortex-A5 with power gating against the implementation of a Cortex-A5 without power gating; the second capitalises on the large size of the ARM Cortex-A5 and exploits the versatility of the proposed dRail standard cells and layout technique to demonstrate how bounded use of dRail within the physical layout can be leveraged to reduce the routing channel area overhead and improve energy efficiency further.

5.3.2.1 dRail Vs Voltage Area

To measure the impact of using the voltage area and dRail layouts on area, routing and power in a Cortex-A5 employing power gating, an implementation was created without power gating for comparison purposes. The results of this implementation are presented in Table 5.4 under 'No Power Gating'. Table 5.4 shows the total standard cell area of the implementation and the power gating area cost of the implementation, which in the case of no power gating is 0. This is followed by the effective total placement area calculated by taking the sum of the total standard cell area and power gating area cost. The remainder of the table shows the total signal routing length of the implementation and finally the power normalised to the implementation without power gating. The power value shown in the final row was obtained by first capturing switching vector data from the simulation of the post layout RTL netlist of the Cortex-A5 with the data engine powered, when running the Dhrystone benchmark at 400MHz. This vector



(a) (b) (c)

Figure 5.16: Floorplan of A5 with interaction of Data Engine and Data Processing Unit (a) no power gating (b) DE power gated with voltage area (c) DE power gated with dRail

Table 5.4: Area, routing and power in no power gating and power gating with voltage area [3], and proposed dRail, Cortex-A5

	No Power Gating	**Voltage Area**	**Diff (%)**	**Proposed dRail**	**Diff (%)**
Total Cell Area (μm^2)	1,246,592	1,286,710	3.22	1,258,415	0.95
PG Area Cost (μm^2)	0	2180	-	85,326	-
Total Placement Area (μm^2)	1,246,592	1,288,890	3.39	1,343,741	7.79
Routing Length (μm)	6,819,157	7,329,862	7.49	6,783,361	-0.52
Normalised Total Power	1.00	1.09	-	1.01	-

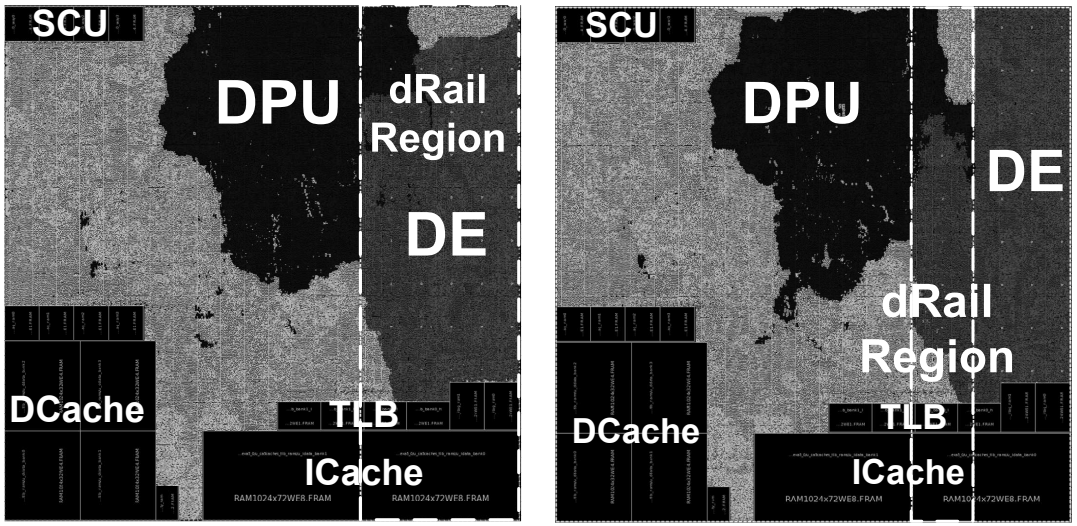
data was then used in Synopsys' Primetime-PX power simulator in conjunction with post-layout parasitic information to report an average active power value over a period of time. The floorplan of the implementation without power gating is shown in Fig. 5.16(a) and what is apparent is how closely the data processing unit (DPU) and data engine (DE) interact with one another. This is important to note as it implies the EDA placement tool assumes better results when the two clusters of logic can be placed freely and intermingled within the physical layout. It can be observed from Fig. 5.16(a) that the data engine is largely confined to the right of the floorplan and so a suitable location for the voltage area implementation is found to be at the top right so as to not block output pin and cache RAM access. This voltage area layout is shown in Fig. 5.16(b). As can be seen, the DPU is 'pulled' towards the DE by the EDA placement tool and is done so to reduce the distance between logically connected gates and maintain the 400MHz performance target. However, the voltage area shows a clear boundary between the two clusters of logic and the separation of the DPU and DE consequently has an impact on area, routing and power as shown in Table 5.4 under 'Voltage Area'. The difference with respect to the implementation with no power gating is also shown. Standard cell area is increased by 3.2% due to the confinement of the data engine into a voltage area. Furthermore the guard band that is required around the voltage area introduces a power gating area cost of $2180\mu m^2$. The combination of these two area overheads results in a 3.39% increase in total placement area with respect to no power gating. It can also be observed in Table 5.4 that total signal routing length increases by 7.5%. The combination of more/larger standard cells in the voltage area physical layout and increased signal routing causes the average active power to also increase by 9%.

Fig. 5.16(c) shows the floorplan of the Cortex-A5 when using the dRail layout. Both the unswitched and switched power supplies are made available everywhere in the physical layout and so, unlike traditional voltage area layout, the EDA tool has the freedom to place all standard cells anywhere in the floorplan. As can be seen in Fig. 5.16(c), the placement freedom given to the EDA tool results in a similar tightly coupled layout as the design without power gating, Fig. 5.16(a). As a result, the increase in total standard cell area is subsequently lower when compared to using a voltage area layout as shown in Table 5.4. The dRail layout, however, suffers a large power gating area cost of $85{,}326\mu m^2$ due to the inclusion of the routing channels and total placement

area is increased by 7.79% as opposed to 3.39% in the voltage area layout. This is similar to what was observed in sub-clock power gated ARM Cortex-M0 test case used in the previous section. Interestingly, routing length is reduced even when compared to no power gating and can be explained by a reduction in routing congestion from the introduction of the routing channels. Despite the increased total placement area, the reductions in standard cell area and signal routing have a positive impact on the total power of the dRail implementation and, as can be seen in Table 5.4, power is comparable to the implementation without power gating. This demonstrates that just as in the first test case, the use of dRail to remove the placement constraint imposed by a voltage area in the physical layout of power gating has helped to improve the energy efficiency of the circuit.

5.3.2.2 Bounded dRail

The first experiment used dRail with no consideration of the impact including routing channels can cause, and as such resulted in an increase in total placement area when compared to the voltage area layout. However, it can clearly be seen from Fig. 5.16(a) and 5.16(c) that the EDA placement tool locates the data engine entirely to the right of the physical layout and this observation can be capitalised on by exploiting the versatility of the proposed dRail standard cells. All standard cells placed in the left two thirds of the floorplan shown in Fig. 5.16(c) do not need to be power gated and so the inclusion of a switched power supply in that part of the physical layout is redundant and wasteful of area due to the inclusion of the accompanying routing channel. Instead, the switched power supply is only needed in approximately the right third of the floorplan to be able to power gate the data engine standard cells. Therefore, with the ability to use the dRail standard cells for both dRail and conventional double-back layout (Section 5.2.1) it is possible to bound the use of dRail to only the right third of the floorplan. The resulting layout of the Cortex-A5 is shown in Fig. 5.17(a). In the dRail region of Fig. 5.17(a) both V_{dd} and VV_{dd} are made available to the standard cells using the dRail technique whereas outside the region only V_{dd} is available to the standard cells using a conventional double-back placement. While it is necessary to enclose the data engine entirely in the dRail region, the availability of both supply rails provides freedom to the EDA placement tool to bring in non power gated standard cells as required. This removes the major constraint of a voltage area layout where the highlighted region would be exclusive to data engine cells only demonstrating the strength of the proposed dRail layout technique. Table 5.5 shows the results achieved with this bounded 'Partial dRail' implementation. The results are presented in the same way as Table 5.4 and for comparison purposes are presented with the results from no power gating and voltage area layout. The standard cell area comes down to a value close to the implementation without power gating and the improvement over a blanket use of dRail can be attributed to reduced standard cell spreading in the non-power gated region of the layout discussed



(a) (b)

Figure 5.17: Floorplan of A5 with interaction of Data Engine and Data Processing Unit (a) DE power gated with partial dRail (b) DE power gated with dRail on interface

Table 5.5: Area, routing and power in no power gating and power gating with voltage area [3], partial dRail, and interface dRail, Cortex-A5

	No Power Gating	**Voltage Area**	**Diff (%)**	**Proposed Partial dRail**	**Diff (%)**	**Proposed Interface dRail**	**Diff (%)**
Total Cell Area (μm^2)	1,246,592	1,286,710	3.22	1,236,267	-0.83	1,236,528	-0.81
PG Area Cost (μm^2)	0	2180	-	35,752	-	18,359	-
Total Placement Area (μm^2)	1,246,592	1,288,890	3.39	1,272,019	2.04	1,254,887	0.67
Routing Length (μm)	6,819,157	7,329,862	7.49	6,574,952	-3.58	6,506,849	-4.58
Normalised Total Power	1.00	1.08	-	1.00	-	1.00	-

in Section 5.2.3. Furthermore, the power gating area cost associated with this layout is reduced compared to the dRail case resulting in a total placement area lower than the voltage area layout. Routing length similarly benefits from the elimination of the placement constraint and overall, it can be seen in Table 5.5, the further reductions in standard cell area and routing length results in the power of the Partial dRail layout matching the power of the Cortex-A5 without power gating. It should be noted that, this type of bounded layout was not possible in the sub-clock power gated Cortex-M0 test case because of its small size.

An interesting observation in the 'Partial dRail' layout in Fig. 5.17(a) is that the interaction of the data processing unit and data engine is largely isolated to the boundary of the two clusters of logic. For this reason a second bounded implementation is created, and shown in Fig. 5.17(b), where dRail is only used on the interface of the two clusters of logic to further minimise the area overhead associated with the dRail layout. In this 'Interface dRail' layout the left two thirds of the floorplan have only a V_{dd} supply just like the 'Partial dRail' layout but the far right of the floorplan only has a VV_{dd} supply rail. The dRail technique is used in the middle of these two regions of the layout with both V_{dd} and VV_{dd} being made available to the standard cells, allowing this region to be used for intermingling of power gated and non-power gated cells. The results from the 'Interface dRail' layout are show in Table 5.5. As can be seen, the area, routing and power are very similar to the 'Partial dRail' implementation but the power gating area cost has been reduced further. In this case an improvement of 38% is achieved over the voltage area layout when comparing total placement area, whilst the 9% increase in active power has simultaneously been eliminated.

To compare the five different implementations more easily, the area, routing and power of each layout normalised to the implementation without power gating can be seen in Fig. 5.18, 5.19 and 5.20 respectively. As can be seen, the inclusion of a voltage area to implement power gating in the Cortex-A5 increases area and routing which in turn reduces the energy efficiency of the circuit. By eliminating the placement constraint imposed by a voltage area through the use of the dRail layout technique it is possible to eliminate almost all of the standard cell area, routing and power overheads but at

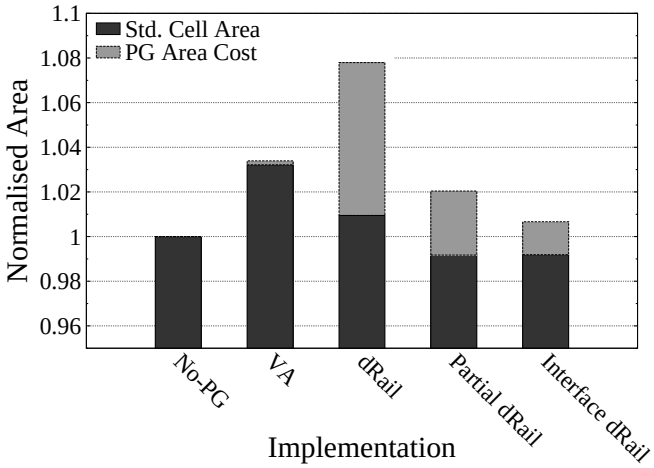


Figure 5.18: Normalised total area with cell area and PG area overhead shown

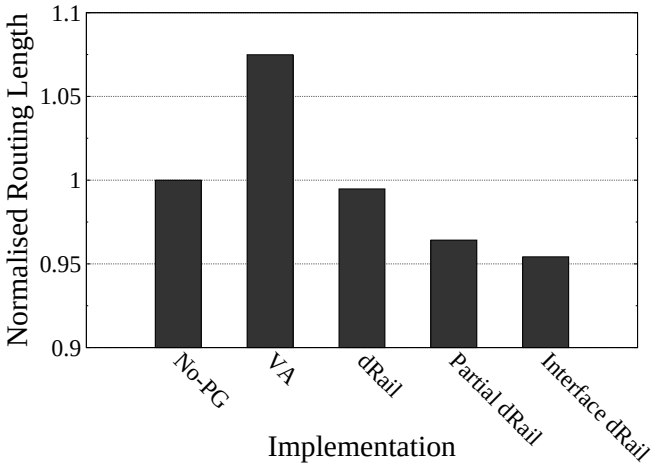


Figure 5.19: Normalised total routing length

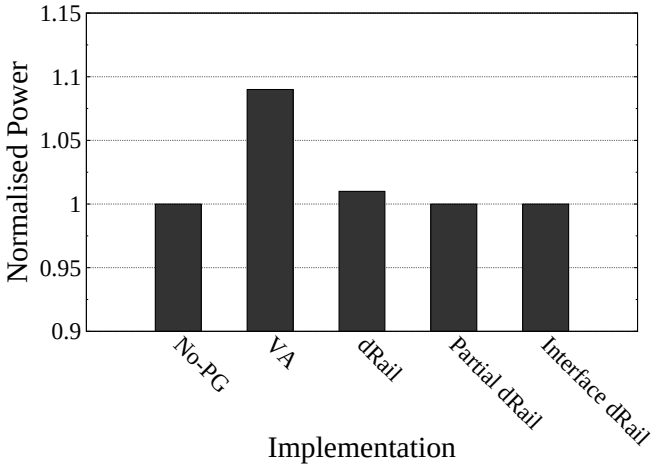


Figure 5.20: Normalised dynamic and leakage power at 400MHz

the cost of an increased total placement area. Through the use of bounded dRail layout though, it can be seen that area, routing and power overheads are eliminated and the total placement area is brought down to a comparable level to the original Cortex-A5 without power gating.

5.4 Concluding Remarks

Physical layout is an important step in the implementation of leakage power minimisation techniques and this chapter has considered the importance of a dedicated physical layout technique for power gating. In the physical layout of power gating, the voltage area approach is well supported by EDA tools and is the most common technique used to achieve placement of both power gated and non-power gated cells. The primary advantage of using a voltage area approach is the ability to use conventional layout and placement techniques without concern of the switched and unswitched power supplies from becoming shorted. However, it has been shown that the use of a voltage area forces the separation of power gated and non-power gated cells in the physical layout introducing placement constraints on the standard cells within the design. This chapter has demonstrated that this separation can introduce standard cell area and signal routing length overheads which reduce the energy efficiency of a circuit using power gating, and has considered how removing the placement constraint associated with a voltage area approach can improve energy efficiency. To enable both power gated and non-power gated cells to be placed together in the physical layout to reduce area and routing overheads and improve energy efficiency a new technique called dRail has been proposed.

The dRail technique proposed in this chapter is achieved with three changes to the physical layout. Firstly, the standard cells of a commercially available gate library are modified such that abutted gates in the physical layout do not automatically share their power and ground connections. This is done by cropping the V_{dd} and V_{ss} pins of the standard cells. Secondly, a method for dual power supply rail has been presented which enables routing of both a switched and unswitched power to be achieved to enable connection to either anywhere in the layout. This is done by introducing a routing channel and routing one supply over the standard cells and one between the placement rows. Finally, each cell must be hooked up to its respective power rail and is achieved either with either a Via to the rail over the cells or a stub to the rail between the rows. The dRail technique has been used in the sub-clock power gated ARM Cortex-M0 from Chapter 4 and compared against traditional voltage area approach. It is shown that standard cell area and routing length are reduced and simulation results from the post layout netlist show an improvement in energy efficiency of the sub-clock power gated circuit at all frequency points. dRail has also been used in the implementation of a conventional power gating example using an ARM Cortex-A5 processor and similar conclusions are

drawn. However, it is shown that the larger size of the test case and the versatility of the modified dRail standard cells can be capitalised on to implement bounded use of the dRail technique. Through bounded use of dRail it is shown that the same standard cell area and routing length reductions are attainable but at a reduced area cost associated with the dRail layout. The dRail technique has been fully incorporated into a power gating design flow using commercially available gate libraries and EDA tools.

Chapter 6

Conclusion and Future Work

Technology scaling has enabled increased integration, lower manufacturing cost and higher performance with each new technology node but power economy has progressively become an important design goal. As a result, over the last twenty years there has been considerable research effort in the area of low power design techniques. While dynamic power has dominated in the past, leakage power has become a greater concern in modern technology processes due to device scaling. Reduction in threshold voltage has increased subthreshold leakage current, reduction of gate oxide thickness has increased gate leakage current and increased doping concentrations to combat short channel effects has increased band-to-band tunneling current. The rapid rise of these three components of leakage has led to leakage power dissipation in digital circuits becoming a major concern and in 65nm technology processes and below leakage has become as significant as dynamic power dissipation. This has led to the development of many different leakage power minimisation techniques that aim to reduce the rising magnitude of leakage power dissipation in digital circuits. The contributions presented in this thesis provides new techniques for leakage power minimisation and their physical layout for embedded processors which are summarised in the next section and is followed by possible areas for future work.

6.1 Thesis Contributions

Recently there has been a shift from performance driven to power and energy driven design goals in some current and emerging applications such as wireless sensor nodes and biomedical sensors. In these types of applications the speed performance of the digital circuit is either not of concern or is an unnecessary design goal due to the low demands on the processor often requiring performance between 10-100s of kHz. However, many of these applications rely on an untethered power source making energy a primary constraint and many of the target applications require power budgets on the order

of 10-100s μW. To meet the energy and power constraints in these low performance applications leakage power minimisation is key to maximising energy efficiency of the digital circuit.

The first objective of this thesis[1] is met by the sub-clock power gating technique proposed in Chapter 3. Due to the low performance requirements of the processors in the target applications it is observed that there can be considerable combinational idle time within the clock period resulting in large amounts of the active mode power becoming dominated by the leakage power of the circuit. This is because, unlike dynamic power, leakage power is independent of the clock frequency and so while average dynamic power is lowered with low clock frequency, average leakage power remains the same. The proposed sub-clock power gating technique takes advantage of the low frequency of operation and idle combinational logic to power gate the combinational logic within the clock period during the active mode of operation. By cutting the power of idle combinational logic within the clock period the proposed technique is able to make the component of leakage power of a digital circuit dependent on clock frequency. The sub-clock power gating technique is the first study into the application of power gating within the clock period and is achieved by splitting the combinational and sequential logic in the digital circuit enabling the sequential logic to remain always-on while the combinational logic can be power gated within the clock period. Instead of using a traditional power gating controller, the proposed technique achieves control of the PMOS power gate, used for the combinational logic, with the clock signal. This means the combinational logic is power gated when the clock is high and is activated when the clock is low. Provided the low phase of the clock is kept long enough for the power rail to recharge and next logic state to evaluate, the high phase of the clock can be extended to capitalise on all the combinational idle time within the clock period. Control of the isolation gates between the power gated combinational logic and sequential logic also presents a challenge and is achieved with a small circuit that is also controlled by the clock but is additionally made adaptive to the state of the combinational virtual supply rail. This ensures outputs are clamped as soon as the clock goes high but are only unclamped once the combinational logic is powered up.

To incorporate the proposed sub-clock power gating technique in a digital design it is shown in Chapter 3 how three additional steps are required in a traditional power gating design flow meeting the first half of the second objective of this thesis. For compatibility with a traditional Unified Power Format (UPF) power gating design flow, the design targeted for augmenting with sub-clock power gating is first synthesised to a generic gate library and is secondly parsed with a Perl script to split the combinational logic gates into one Verilog module and the sequential logic into another Verilog module. The final additional step wraps up the newly created Verilog modules with the sub-clock power gating control circuitry before synthesising the design with a normal power

[1] All objectives are listed in Section 2.5

gating design flow. Using the presented design flow the sub-clock power gating technique is implemented in three test cases using a 90nm gate library: a 16-bit parallel multiplier, the ARM Cortex-M0 and a recently proposed processor for wireless sensor networks, the Event Processor. Post layout simulation of the transistor level netlists using HSpice are presented for the three test cases meeting the remainder of the second objective of this thesis. Results show that the proposed sub-clock power gating can enable up to 74%, 52% and 24% savings in active power for the multiplier, Cortex-M0 and Event Processor respectively through leakage power savings. Furthermore, using an example 100kHz performance target, it is shown the sub-clock power gating technique enables 3.5x, 2x and 1.3x improvement in energy efficiency for the multiplier, Cortex-M0 and Event Processor.

The sub-clock power gating technique proposed in Chapter 3 has shown encouraging results and Chapter 4 proposed a new power gating technique called symmetric virtual rail clamping to investigate its utility in the sub-clock power gating technique. The symmetric virtual rail clamping technique is proposed to reduce wake-up energy cost and improve the energy efficiency and applicable frequency range of the sub-clock power gating technique. Instead of shutting down the power gated logic completely, as is the case in conventional power gating, the proposed symmetric virtual rail clamping technique reduces the power gated logic's supply by $2V_{th}$ to a value less than the threshold voltage of the transistors in the digital circuit. This is achieved by placing a pair of NMOS and PMOS transistors at the head and foot of the power gated logic. The technique exploits the built-in threshold voltages of the NMOS transistor at the head and PMOS transistor at the foot of the power gated logic to clamp the VV_{dd} and VV_{ss} rails by V_{thn} and V_{thp} respectively, when the logic is put into sleep mode. By maintaining a voltage across the power gated logic it is possible to hold valid logic outputs of the power gated logic gates which eliminates signal glitching when waking up from sleep mode. In addition to this, the charge stored in the VV_{dd} rail is recycled to the VV_{ss} rail when entering the sleep mode and reduces the amount of energy required to restore the supply rails to a nominal value. Reverse body biasing is also capitalised on in the proposed technique and the symmetric nature of the clamping used enables a lower clamped voltage to be achieved when compared to asymmetric clamping. The symmetric virtual rail clamping technique is incorporated with the sub-clock power gating technique and is implemented for fabrication in a silicon test chip. Experimental results are obtained from the fabricated test chip for both the symmetric virtual rail clamping and sub-clock power gating techniques meeting the third objective of this thesis. It is shown that the lower wake-up energy of the symmetric virtual rail clamping technique improves the energy efficiency of sub-clock power gating across all frequency points and enables sub-clock power gating to be used up to 417kHz as opposed to just 1kHz when using conventional shut down power gating. This demonstrates a 400x improvement in the applicable clock frequency range. An on chip clock modulator is used in the silicon test chip and experimental results validate the ability to achieve better energy efficiency with

a modified clock duty cycle. As a result, for all sub-clock power gating with symmetric virtual rail clamping results, a 200ns low phase of clock is maintained while the high phase is modulated to vary the clock frequency. Using a real energy harvester tuning program it is shown that sub-clock power gating with symmetric virtual rail clamping can enable 1.4x improvement in energy efficiency at a clock frequency of 200kHz.

The two proposed techniques (sub-clock power gating, Chapter 3, and symmetric virtual rail clamping, Chapter 4) have focused on the reduction of leakage power using power gating. During the physical layout of the techniques the popular voltage area layout technique is used as it is both a common approach to implementing power gating and is well supported by EDA tools. In Chapter 5, the fourth objective of this thesis is met by firstly gaining a deeper understanding of why a voltage area is required in the physical layout and secondly proposing a new physical layout technique called dRail. It is shown that to save space and capitalise on the availability of multiple metal layers for signal routing, standard cells in modern gate libraries are designed such that they can automatically share their power and ground supplies when they are abutted with one another. This leads to the use of Metal1 power and ground *rails* in the physical layout which connect all the standard cells placed in a row to the same power and ground supplies, simplifying the power routing. However, in the implementation of power gating this layout causes an issue due to the need for a subset of cells to connect to a switched power supply. Therefore, to enable the use of conventional placement techniques (power and ground sharing with Metal1 rails) all the cells connecting to the switched power supply are grouped and separated into a voltage area in the physical layout. This leads to a placement constraint on a subset of the cells within the design increasing standard cell area and signal routing length overheads. Energy efficiency of the power gated circuit is subsequently reduced as more/larger standard cells exhibit higher leakage and dynamic power and increased signal routing increases the output capacitive load of logic gates increasing dynamic power. To reduce the standard cell area and signal routing overheads, Chapter 5 proposes a new physical layout technique called dRail. The proposed technique enables both power gated and non-power gated standard cells to be placed together in the physical layout unlike voltage area layout which separates the two. Three challenges arise with the proposed technique. Firstly, the standard cells must be modified to stop automatic sharing of power and ground between abutted standard cells and is achieved by cropping the V_{dd} and V_{ss} pins of the standard cells in a conventional standard cell library. Secondly, a method to route both a switched and unswitched power supply for availability anywhere in the placement row is required which is done by introducing a routing channel and then routing one supply over the standard cells and one supply between the placement rows. Finally power gated and non-power gated cells must be able to connect to their respective power supplies and is achieved by either creating a Vi to the power supply routed over the cells or a stub to the supply routed between the rows.

In line with the second objective of this thesis, it is shown that to incorporate dRail in a traditional physical design flow the first change required in dRail is performed before the digital design can be synthesised whereas the second occurs within design planning and the third after place and route is completed. dRail is compared with voltage area layout in the sub-clock power gated ARM Cortex-M0 used in Chapter 4 and experimental results show that standard cell area and routing length are reduced by 3% and 19% respectively due to the freedom given to the EDA tool for standard cell placement. These two reductions subsequently show an improvement in energy efficiency of up to 4% in the sub-clock power gating mode of operation due to the reduction of dynamic and leakage power associated with fewer/smaller logic gates and reduction of dynamic power associated with reduced wiring capacitance. The dRail technique is also investigated in an ARM Cortex-A5 processor for a conventional application of power gating. Experimental results show that dRail can help to improve energy efficiency in this test case too due to almost complete elimination of area and routing overheads observed in voltage area layout. However, the size of the Cortex-A5 and the versatility of the modified dRail standard cells is capitalised on to propose bounded use of dRail which enables the same area and routing length reductions to be achieved through placement freedom but also reduces area cost associated with the routing channels required in the dRail technique from 7.79% to 0.67%.

The contributions presented in this thesis provide novel techniques for reducing the leakage power of embedded processors and their physical layout that are fully compatible with existing design flows using commercially available EDA tools and gate libraries. The conclusions drawn in this thesis are supported by analysis on a range of fully synthesised test cases and simulation on post layout transistor level netlists using power accurate simulators such as HSpice, as well as experimental validation through fabricated silicon. Some of the work presented in this thesis has already been picked up by industry [166] and it is hoped that the techniques proposed in this thesis will make useful contributions towards the development of future embedded processors.

6.2 Future Work Directions

6.2.1 Improved Sub-Clock Power Gating

Sub-clock power gating has demonstrated that power gating within the clock cycle can reduce the leakage power of a digital circuit but currently the technique powers up all of the combinational logic regardless of whether all the registers need to be updated. Chapter 2 showed that a number of recent works have developed techniques for combining clock gating and power gating [128, 131]. In the sub-clock power gating technique the clock gate enables could be capitalised on to keep combinational gates shut down if the registers that they form the fan-in cone to do not need to be updated. This would

save energy associated with recharging the virtual rail and leakage of the combinational gates that can be kept off over multiple clock cycles. The application of this technique would require careful consideration of the design partitioning for individual control of multiple power domains. To be able to achieve this a method of tracing the inputs or outputs from each clock gated register would be required that was fully compatible with current EDA tools. Furthermore, there would need to be further investigation of an efficient physical layout strategy that could be used with commercially available EDA tools that could be employed for the implementation of potentially dozens of voltage areas in a single digital circuit.

6.2.2 Further Applications of Symmetric Virtual Rail Clamping

Shut down power gating offers high leakage power saving but when used in short periods it is restricted by the energy needed to restore the power gated logic from sleep mode to active mode as shown in the experimental results of Chapter 4. The symmetric virtual rail clamping technique proposed in Chapter 4 on the other hand showed great benefit in reducing the wake-up energy cost associated with moving between the power gated modes of operation. When used in the sub-clock power gating technique symmetric virtual rail clamping enabled a much greater range of operation. A number of recent papers have suggested a variety of different techniques for using power gating during the active mode for high performance systems capitalising on aspects such as functional unit idle time [122, 123]. Instead of using shut down power gating in these techniques, symmetric virtual rail clamping could be used and would enable power gating to be used more frequently and over shorter time periods as the energy cost of waking up is reduced. As many of these active mode power gating techniques are targeted for high performance systems, one concern would be that the inclusion of both a header and footer power gate would introduce a too large IR drop to the power gated logic which could have an adverse impact on performance. In these applications the utilisation of super cut-off CMOS (SCCMOS) [98, 99] (Chapter 2, Section 2.2) could be investigated to reduce the IR drop associated with the application of symmetric virtual rail clamping whilst maintaining low area cost from inclusion of power gating transistors.

6.2.3 Physical Layout for Body Biasing

Chapter 5 showed that the implementation of power gating in the physical layout introduces challenges and as a result the voltage area layout technique is a solution to this but comes at the cost of placement constraint increasing standard cell area and signal routing length which reduces energy efficiency. It would be interesting to investigate how the implementation of the body biasing technique introduces physical layout challenges as the problem is very similar to power gating. Gates that are to have their

threshold voltages modulated through body biasing need to have independent control of their wells and therefore need to be grouped together to stop shorting of wells between modulated and unmodulated gates. This separation is identical to the separation that occurs between power gated and non-power gated cells using a voltage area and will introduce standard cell area and routing length overheads which reduces the digital circuit's energy efficiency. Therefore, investigating a new layout technique which enables greater amalgamation of the modulated and unmodulated cells would help to reduce area and routing overheads and improve energy efficiency.

Appendix A

Microprocessor Details

A.1 ARM Cortex-M0

The information in this section gives an overview of the ARM Cortex-M0 microprocessor used in Chapters 3, 4 and 5 of this thesis. The information presented in this section is gathered from the ARM Cortex-M0 Generic User Guide available from ARM [160] and The Definitive Guide to the ARM Cortex-M0 by Joseph Yiu [167].

The ARM Cortex-M0 was chosen as a test case because of its relevance to low performance energy constrained applications. The ARM Cortex-M0 is, at time of writing, the smallest microprocessor available from ARM with a reported 12,000 logic gates in its minimum configuration. A simplified block diagram of the Cortex-M0 processor is shown in Fig. A.1. The processor is made up of a number of essential units and optional units. The first optional unit is the Wake-up Interrupt Controller (WIC) which enables support for a power management unit. Using this optional unit the processor can be powered down while there is nothing to be executed and the WIC can monitor incoming interrupts. If an interrupt is detected the WIC can interface with a power management unit to enable the processor for execution. The optional debug hardware provides access to the system's bus. It contains functional blocks to handle breakpoints and watchpoints in developer code so that when a debug event occurs, the processor can be put in a halted state while the state of the processor is examined. Throughout this thesis the debug hardware has been omitted from the processor to reduce area and power. The Nested Vector Interrupt Controller (NVIC), Processor Core and Bus Matrix make up the essential blocks within the processor. The NVIC can be configured to have up to 32-bit interrupt request signals in addition to a single nonmaskable interrupt (NMI). The purpose of the NVIC is to judge the priority of incoming interrupts so that it can automatically handle the execution of nested interrupts. When an interrupt is requested the NVIC interfaces with the processor so that it can call and execute the correct interrupt service routine. The purpose of the NMI is an interrupt for high priority

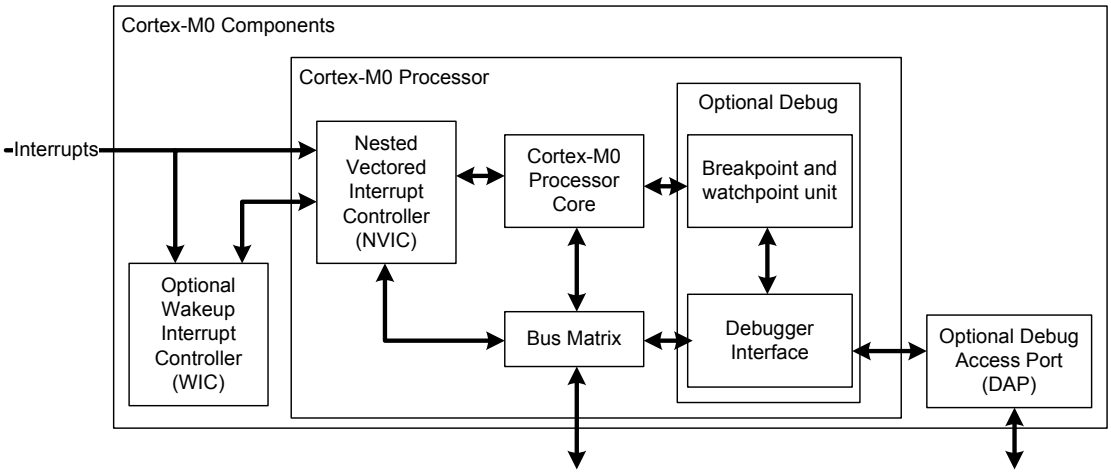


Figure A.1: Block diagram of the Cortex-M0 processor

tasks that may, for example, be safety critical and cannot be disabled unlike the other interrupts in the processor. When the NMI is requested, it has the highest priority out of all interrupts and guarantees the interrupt service routine will be executed for the requested interrupt.

The processor core in Fig. A.1 makes up the bulk of the processor and contains the register banks, the ALU, datapath and control logic. The register bank is made up of 12 general purpose registers, a stack pointer register used for the address to the stack, a link register used for storing the return address of a subroutine and the program counter which stores the address of the current instruction being executed. In addition to the main register bank there are three main special registers: program status register which contains information about the program execution and ALU flags, the Primask register which is used for blocking all of the interrupts except the NMI in the NVIC and finally the Control register which is used for control flow. The ALU is made up of an addition and logical unit, a multiplier, and a shift and permute unit as was shown in Chapter 3, Fig. 3.15. Depending on the application requirements, the multiplier in the Cortex-M0 can be configured to be a large, fast single cycle multiplier for higher performance systems, or a small 32 cycle multiplier for low performance systems.

The entire system uses a von Neumann architecture with a single unified 32-bit wide data and instruction bus. The bus used by the Cortex-M0 is the Advanced Microcontroller Bus Architecture (AMBA) AHB-Lite specification developed by ARM. The microprocessor is based on the ARMv6-M^{TM} architecture and is based on the classic load-store architecture with a 3 stage pipeline consisting of instruction fetch, instruction decode and instruction execute. The processor supports most of the Thumb-2 instruction set introduced in 2003. The Thumb-2 instruction set is a combination of the original Thumb instruction set consisting of only 16-bit instructions with an addition of a number of 32-bit Thumb instructions. The purpose of this was to enable the Thumb instruction set to be capable of all the instructions that could previously only be carried out by the

ARM instruction set. The result of this is a smaller code footprint, thanks to the 16-bit instructions, but approximately the same level of performance as when using the ARM instruction set. In the ARM Cortex-M0, the entirety of the 16-bit Thumb instruction set is supported but only a subset of the 32-bit Thumb instructions are supported. In total, 56 base instructions are available in the Cortex-M0: 50, 16-bit Thumb and 6, 32-bit Thumb instructions. Despite, the Cortex-M0's small size and limited instruction set the capability of the processor makes it much more energy efficient than equivalent 8-bit and 16-bit processors. Throughput is reported to be 3x better than the popular Texas Instruments MSP430 and 2x better than the PIC24 and so for a given performance the Cortex-M0 could operate with much lower clock frequency and power compared to its 8-bit and 16-bit counterparts [167].

Appendix B

Benchmarks and Simulation

B.1 Dhrystone Benchmark

In Chapters 3, 4 and 5, the active power of the ARM Cortex-M0 microprocessor has been obtained and compared using the Dhrystone benchmark. In this appendix background information for the benchmark is given.

The Dhrystone benchmark program is a synthetic computing benchmark first developed by Reinhold P. Weicker in 1984 and is representative of integer operations in modern CPUs. The benchmark was developed as an alternate to the then popular Whetstone benchmark program which predominantly used floating point operations. Version 1 of the benchmark was originally written in Ada in 1984 but evolved to address some issues until finally v2.1 was released in 1988. The Dhrystone benchmark has since been ported to the C programming language and Dhrystone v2.1 in C remains the most common version currently used in industry [168]. The advantages of Dhrystone include its open source nature making it freely available to the public and more importantly its compact code size of 1-1.5kB meaning it can be used on large high performance processors as well as small low power embedded processors [168]. In the case of the ARM C Compiler targeted for the Cortex-M0 the Dhrystone benchmark compiles to a size of 1.2kB. The primary purpose of the Dhrystone benchmark is to offer a performance metric in terms of number of Dhrystones per second. This metric corresponds to the number of times the main loop in the Dhrystone benchmark is completed per second and offers an improved metric over MIPS (millions of instructions per second) which varies between RISC and CISC computers and can be a distorted metric of performance. DMIPS is another common performance metric reported from the Dhrystone benchmark and compares the Dhrystone performance of a machine against the nominal 1MIPS VAX11/780 machine [161]. The VAX11/780 can perform 1757 Dhrystones/sec therefore the DMIPS value for a processor is calculated by dividing its reported Dhrystones/sec by 1757. For example, a processor with a DMIPS of 200 means it is 200 times faster than the VAX11/780.

Table B.1: Statistics of statements and operands in Dhrystone benchmark

(a) Statement Types

Statement Type	**Num.**
V1=V2	9
V = Constant	12
Assignment with array element	7
Assignment with record component	6
X = Y + | - | '&&' | '|' Z	5
X = Y + | - | '==' Constant	6
X = X + | - 1	3
X = Y × | / Z	2
X = Expression, two operators	1
X = Expression, three operators	1
if with else	7
if without else	7
for	7
while	4
do while	1
switch	1
break	1
P(..) Procedure call	11
X = function call F(..)	6
Total	97

(b) Operators

Operator Type	**Num.**	**%**
Arithmetic		
+	21	33.3
-	7	11.1
*	3	4.8
/	1	1.6
Comparison		
==	9	14.3
/ =	4	6.3
>	1	1.6
<	3	4.8
>=	1	1.6
<=	9	14.3
Logic		
&& (AND-THEN)	1	1.6
| (OR)	1	1.6
! (NOT)	2	3.2
Total	63	100.1

(c) Operand Type (Counted per reference)

Operand Type	**Num.**	**%**
Integer	175	72.3
Character	45	18.6
Pointer	12	5.0
String30	6	2.5
Array	2	0.8
Record	2	0.8
Total	242	100.0

(d) Operand Locality

Operand Locality	**Num.**	**%**
local variable	114	47.1
global variable	22	9.1
parameter	45	18.6
value	23	9.5
reference	22	9.1
function result	6	2.5
constant	55	22.7
Total	242	100.0

The Dhrystone benchmark's main focus is on integer operations and consists of 8 main procedures and 3 functions using general arithmetic and logic functions, control flows and C structures. The Dhrystone benchmark also relies on the C library functions memcpy(), strcpy() and strcmp() for performing a series of string manipulation tasks which have been measured to take up 23% of the execution time in the VAX 11/785 but has been reported to take up to 65% of the execution time on the ARM9E [161]. A breakdown of the number of statement types, operator types, operand types and operand locality statistics that are reported within the Dhrystone benchmark are given in Tables B.1(a) to B.1(d).

The program first sets up the variables used in the main loop of the Dhrystone benchmark

and then loops around the main execution loop for a given number of iterations. The benchmark completes by printing out a series of statements confirming the benchmark executed without any errors. When the Dhrystone benchmark has been used in this thesis, power has only been measured whilst executing the main loop and not when performing the setup at the start or printf functions towards the end of the C program.

While the primary reason many people use the Dhrystone benchmark is to measure and compare performance between different processors, this is not the reason the benchmark has been chosen as a test case in this thesis. Performance has always been considered as a secondary metric after power and instead, the small size of the Dhrystone benchmark and its primary focus on integer core functions (arithmetic and logic) means it is well tailored to general purpose microprocessors and their typical applications. For example, in a wireless sensor node, the bulk of the processing done by the node is data manipulation on the collected data [57–59, 63]. Therefore, the Dhrystone benchmark provides a representative workload of general processing execution in a microprocessor as a means to comparing active power.

B.2 Energy Harvester Tuning Program

```
1
2  #define NUM_SAMPLES     2
3  #define MIN_PERIOD 16
4  #define MAX_PERIOD 32
5
6
7  /* Global ADC variable: */
8  const uint16_t ADC[1000] = {1088, 1135, 1192, 1251, 1313, 1379, 1445, 1508,
9  1570, 1623, 1673, 1713, 1745, 1767, 1777, 1777, 1769, 1746, 1713, 1673, 1622,
10 1566, 1500, 1431, 1360, 1284, 1210, 1136, 1068, 1004, 945, 895, 855, 822,
11 802, 790, 791, 802, 825, 859, 899, 950, 1012, 1076, 1147, 1218, 1294, 1365,
12 1438, 1506, 1572,  1629, 1676, 1716, 1748, 1767, 1775, 1775, 1759, 1737,
13 1700, 1659, 1603, 1544, 1476, 1407, 1336, 1261, 1187, 1115, 1046, 982, 928,
14 881, 842, 814, 794, 788, 793, 808, 830, 871, 915, 968, 1032, 1098, 1168, 1242,
15 1317, 1390, 1460, 1528, 1590, 1645, 1690, 1727, 1755, 1770, 1775, 1770, 1752,
16 1721, 1685, 1640, 1585, 1522, 1456, 1385, 1310, 1237, 1162, 1093, 1028, 965,
17 913, 869, 834, 808, 796, 792, 798, 818, 846, 886, 935, 991, 1054, 1121, 1194,
18 1267, 1340, 1414, 1483, 1549, 1607, 1660, 1703, 1735, 1758, 1772, 1773, 1762,
19 1741, 1711, 1669, 1621, 1564, 1500, 1432, 1359, 1286, 1210, 1139, 1071, 1004,
20 947, 897, 856, 824, 802, 793, 791, 804, 826, 859, 900, 954, 1012, 1076, 1147,
21 1219, 1292, 1366, 1438, 1505, 1570, 1627, 1675, 1716, 1745, 1766, 1774, 1771,
22 1758, 1735, 1699, 1655, 1601, 1543, 1477, 1408, 1334, 1259, 1187, 1114, 1048,
23 985, 928, 882, 842, 816, 798, 791, 796, 810, 833, 873, 918, 974, 1034, 1099,
24 1170,  1242, 1317, 1389, 1461, 1529, 1590, 1645, 1690, 1727, 1754, 1769, 1774,
25 1767, 1750, 1719, 1685, 1638, 1584, 1520, 1455, 1383, 1309, 1234, 1162, 1091,
26 1026, 964, 914, 869, 833, 809, 795, 792, 800, 818, 847, 887, 934, 991, 1055,
27 1123, 1195, 1269, 1343, 1416, 1486, 1551, 1609, 1661, 1704, 1738, 1760, 1772,
28 1773, 1763, 1742, 1711, 1670, 1620, 1563, 1499, 1429, 1358, 1283, 1208, 1137,
29 1069, 1005, 946, 896, 855, 823, 801, 792, 793, 804, 826, 859, 902, 953, 1013,
30 1078, 1148, 1221, 1294, 1368, 1440, 1508, 1572, 1629, 1678, 1717, 1746, 1765,
31 1773, 1771, 1757, 1734, 1697, 1653, 1601, 1540, 1475, 1405, 1333, 1259, 1185,
32 1113, 1045, 980, 926, 879, 841, 814, 796, 791, 795, 810, 835, 873, 919, 974,
```

```
33  1036, 1102, 1173, 1246, 1321, 1393, 1464, 1530, 1594, 1646, 1692, 1728, 1754,
34  1770, 1774, 1768, 1750, 1719, 1683, 1636, 1581, 1519, 1453, 1381, 1305, 1231,
35  1159, 1090, 1022, 963, 909, 865, 831, 807, 794, 791, 800, 818, 849, 889, 935,
36  993, 1059, 1126, 1198, 1272, 1346, 1418, 1488, 1553, 1611, 1664, 1705, 1740,
37  1761, 1772, 1773, 1763, 1741, 1709, 1667, 1618, 1561, 1496, 1428, 1355, 1281,
38  1207, 1134, 1064, 1001, 944, 893, 852, 822, 801, 791, 792, 805, 828, 860, 904,
39  956, 1014, 1082, 1151, 1222, 1298, 1371, 1442, 1511, 1574, 1631, 1680, 1718,
40  1750, 1767, 1775, 1771, 1756, 1735, 1698, 1653, 1601, 1539, 1473, 1403, 1329,
41  1255, 1182, 1110, 1042, 980, 924, 878, 838, 812, 795, 789, 794, 809, 836, 875,
42  919, 973, 1036, 1103, 1173, 1247, 1322, 1395, 1466, 1533, 1594, 1649, 1693,
43  1729, 1755, 1771, 1775, 1769, 1751, 1719, 1683, 1637, 1582, 1517, 1451, 1379,
44  1307, 1232, 1159, 1088, 1021, 960, 908, 865, 832, 808, 793, 792, 799, 819,
45  848, 889, 937, 993, 1057, 1126, 1198, 1272, 1347, 1419, 1488, 1554, 1612,
46  1664, 1705, 1739, 1761, 1772, 1773, 1763, 1740, 1709, 1668, 1618, 1559, 1495,
47  1426, 1352, 1279, 1205, 1134, 1065, 999, 943, 892, 852, 821, 801, 790, 793,
48  805, 827, 863, 904, 957, 1018, 1081, 1152, 1224, 1299, 1373, 1443, 1511, 1575,
49  1632, 1681, 1719, 1750, 1767, 1773, 1771, 1755, 1733, 1696, 1652, 1597, 1538,
50  1470, 1400, 1326, 1254, 1180, 1108, 1039, 978, 924, 876, 839, 813, 795, 790,
51  796, 811, 840, 875, 922, 975, 1039, 1106, 1177, 1249, 1324, 1398, 1468, 1534,
52  1595, 1649, 1694, 1729, 1756, 1770, 1774, 1769, 1750, 1716, 1680, 1633, 1577,
53  1516, 1448, 1378, 1303, 1229, 1156, 1085, 1019, 960, 908, 864, 829, 807, 794,
54  791, 799, 820, 851, 891, 939, 997, 1060, 1129, 1202, 1275, 1350, 1419, 1490,
55  1554, 1614, 1665, 1707, 1737, 1761, 1772, 1773, 1761, 1738, 1707, 1666, 1616,
56  1557, 1493, 1424, 1353, 1277, 1205, 1132, 1064, 999, 943, 893, 853, 823, 801,
57  795, 795, 806, 829, 864, 907, 959, 1018, 1083, 1154, 1225, 1300, 1373, 1444,
58  1511, 1573, 1631, 1676, 1717, 1745, 1765, 1774, 1769, 1754, 1730, 1693, 1650,
59  1596, 1537, 1470, 1400, 1327, 1252, 1180, 1109, 1042, 981, 926, 880, 841,
60  818, 797, 791, 798, 814, 844, 878, 924, 977, 1042, 1107, 1177, 1251, 1325,
61  1399, 1466, 1535, 1597, 1649, 1694, 1729, 1756, 1769, 1773, 1766, 1748, 1717,
62  1681, 1634, 1578, 1514, 1447, 1375, 1301, 1226, 1153, 1083, 1016, 956, 905,
63  861, 827, 804, 791, 789, 797, 818, 850, 890, 940, 996, 1062, 1131, 1203,
64  1279, 1353, 1426, 1496, 1561, 1618, 1670, 1712, 1743, 1766, 1778, 1775, 1765,
65  1742, 1710, 1667, 1616, 1558, 1491, 1422, 1350, 1275, 1199, 1128, 1060, 993,
66  936, 888, 845, 816, 798, 787, 788, 802, 825, 861, 904, 956, 1018, 1083, 1154,
67  1227, 1302, 1378, 1447, 1515, 1579, 1636, 1686, 1723, 1753, 1770, 1777, 1773,
68  1759, 1735, 1697, 1652, 1599, 1538, 1471, 1399, 1326, 1251, 1177, 1105, 1038,
69  975, 920, 873, 834, 808, 791, 786, 791, 808, 837, 874, 919, 975, 1039, 1106,
70  1178, 1252, 1329, 1407, 1472, 1541, 1601, 1656, 1700, 1736, 1763, 1777, 1780,
71  1772, 1753, 1723, 1684, 1636, 1578, 1515, 1447, 1374, 1298, 1225, 1149, 1080,
72  1012, 953, 901, 857, 824, 802, 787, 786, 795, 815, 848, 889, 939, 996, 1061,
73  1130, 1203, 1278, 1353, 1424, 1494, 1561, 1618, 1669, 1711, 1743, 1766, 1776,
74  1775, 1764, 1741, 1709, 1666, 1617, 1558, 1492, 1422, 1350, 1276, 1203, 1130,
75  1061, 996, 940, 890, 850, 821, 800, 791, 793, 805, 828, 864, 906, 960, 1019,
76  1085, 1155, 1230, 1301, 1375, 1445, 1513, 1575, 1632, 1680, 1717, 1747, 1765,
77  1772, 1768, 1753, 1728, 1692, 1647, 1594, 1532, 1467, 1398, 1325, 1250, 1178,
78  1107, 1041, 980, 924, 879, 842, 817, 801, 794, 801, 819, 847, 882, 928, 982,
79  1045, 1110, 1183, 1255, 1328, 1400, 1469, 1535, 1596, 1649, 1692, 1728, 1752,
80  1766, 1770, 1762, 1742, 1712, 1674, 1628, 1572, 1509, 1440, 1371, 1298, 1225,
81  1152, 1083, 1017, 958, 907, 865, 831, 809, 795, 795, 804, 823, 855, 895, 945};
82
83  int main (void)
84  {
85          int freq;
86          int new_position;
87
88
89          /* Initializations */
90
91          // Call CMSIS System Initialisation
```

```
92          SystemInit();
93
94          // Call IK Specific Initialisation
95          uSTEPInit();
96
97
98          printf ("\nStarting Tuning!\n");
99
100
101         while(1)
102         {
103                 freq = Tuning_Get_Frequency();
104         //Include to check correct functionality
105                 //printf("Resonant Frequency = %d\n", freq);
106                 if (freq == 0)
107                 {
108                         freq = 1;
109                 }
110                 new_position = Tuning_Calculate_Position (freq);
111         //Include to check correct functionality
112                 //printf("New Stepper Position = %d\n", new_position);
113         }
114
115
116     return 0;
117 }
118
119 int Tuning_Get_Frequency (void) {
120
121         int cycle_count, freq;
122         int zero_value;
123         int up_zero_time, down_zero_time;
124         int up_time, down_time;
125         int up_time_new, down_time_new;
126         int max_value, min_value;
127         int crossed_down, crossed_up;
128         int prev_accel;
129         int count;
130         uint32_t samples[NUM_SAMPLES];
131
132         // for debug only
133         uint32_t accel_samples [1000];
134
135         cycle_count = 0;
136         freq = 0;
137         zero_value = 0;
138         up_zero_time = 0;
139         down_zero_time = 0;
140         up_time = 0;
141         down_time = 0;
142         up_time_new = 0;
143         down_time_new = 0;
144         max_value = 0;
145         min_value = 100000;
146         crossed_down = 0;
147         crossed_up = 0;
148         prev_accel = 0;
149         count = 0;
150
```

```
151
152         // stick in this loop until all samples are acquired
153         while(count < 1000){
154
155         samples[0] = ADC[count];
156                 samples[1] = ADC[count];
157
158                 //generator_array[count] = samples[0];
159                 accel_samples[count] = samples[1];
160
161                 // find initial zero-point
162                 if (count < (MAX_PERIOD * 2))
163                 {
164                         if (samples[1] > max_value)
165                         {
166                                 max_value = samples[1];
167                         }
168                         if (samples[1] < min_value)
169                         {
170                                 min_value = samples[1];
171                         }
172                 }
173
174                 if (count == (MAX_PERIOD * 2))
175                 {
176                         zero_value = (max_value + min_value) / 2;
177                 }
178
179                 if ((count > (MAX_PERIOD * 2)) && (count <= (MAX_PERIOD * 4)))
180                 {
181                         if ((samples[1] >= zero_value) && (prev_accel < zero_value ))
182                         {
183                                 up_zero_time = count;
184                         }
185                         if ((samples[1] <= zero_value) && (prev_accel > zero_value ))
186                         {
187                                 down_zero_time = count;
188                         }
189                 }
190
191
192                 if (count == (MAX_PERIOD * 4))
193                 {
194                         up_time = up_zero_time;
195                         down_time = down_zero_time;
196                         cycle_count = 0;
197                 }
198
199                 if (count > (MAX_PERIOD * 4))
200                 {
201                         if (samples[1] < min_value)
202                                 min_value = samples[1];
203                         if (samples[1] > max_value)
204                                 max_value = samples[1];
205
206                         if ((count >= up_time + (MIN_PERIOD * 2))
207                     && (count <= up_time + (MAX_PERIOD * 2)))
208                         {
209                                 if ((samples[1] >= zero_value) && (prev_accel < zero_value )
```

```
210                     && (crossed_up == 0))
211                             {
212                                     up_time_new = count;
213                                     crossed_up = 1;
214                                     zero_value = (min_value + max_value) / 2;
215                                     max_value = 0;
216                             }
217                     }
218
219             if ((count >= down_time + (MIN_PERIOD * 2))
220                     && (count <= down_time + (MAX_PERIOD * 2)))
221                     {
222                             if ((samples[1] <= zero_value) && (prev_accel > zero_value )
223                     && (crossed_down == 0))
224                             {
225                                     down_time_new = count;
226                                     crossed_down = 1;
227                                     zero_value = (min_value + max_value) / 2;
228                                     min_value = 100000;
229
230                             }
231                     }
232
233                     if (count == up_time + (MAX_PERIOD * 2))
234                     {
235                             if (up_time == up_time_new)
236                             {
237                                     up_time = 99999;
238                             }
239                             else
240                             {
241                                     up_time = up_time_new;
242                                     cycle_count++;
243                                     crossed_up = 0;
244                             }
245                     }
246
247                     if (count == down_time + (MAX_PERIOD * 2))
248                     {
249                             if (down_time == down_time_new)
250                             {
251                                     down_time = 99999;
252                             }
253                             else
254                             {
255                                     down_time = down_time_new;
256                                     cycle_count++;
257
258                                     crossed_down = 0;
259                             }
260                     }
261             }
262             prev_accel = samples[1];
263             count++;
264     }
265
266     if ((down_time == 99999) || (up_time == 99999)
267         || (cycle_count <= 20) || (cycle_count >= 50))
268     {
```

```
269                 freq = 0;
270         }
271         else
272         {
273                 down_time -= down_zero_time;
274                 up_time -= up_zero_time;
275
276                 freq = (200000 * cycle_count);
277                 freq /= (up_time + down_time);
278                 freq += 5;
279                 freq /= 10;
280         }
281         return (freq);
282 }
283
284 int Tuning_Calculate_Position (int freq) {
285         float h, h2;
286         int position;
287
288         h2 = freq;
289         h2 *= h2;
290         h2 *= 0.1076;
291
292         h = freq;
293         h *= 134.39;
294
295         h -= h2;
296         position = h;
297         position -= 35193;
298
299         return(position);
300 }
```

B.3 HSpice Simulation

HSpice simulation for measuring and comparing the power of different circuits is key for obtaining the sub-clock power gating results presented in Chapters 3, the ring oscillator results in 4 and the comparison of sub-clock power gating dRail and voltage area layouts in Chapter 5. It was chosen because of its precision and accuracy when measuring the power dissipation of the digital circuits using power gating as it provides transient simulation of a transistor level netlist inclusive of parasitic resistance and capacitance. HSpice is particularly useful for the analysis of sub-clock power gating since the power gating is used within the clock period and the contribution of energy overheads of moving between different power modes is important in the overall comparison of power dissipation. The flow for simulating and obtaining the power results for the digital circuits was shown in Chapter 3, Fig. 3.7. To better understand how the HSpice simulator is used, this section gives a more detailed breakdown of each step in the experimental flow, Fig. 3.7, to explain how these power numbers are obtained.

B.3.1 Prerequisites

To be able to perform an HSpice simulation on a digital circuit, the technology library that is used for synthesis and place and route must contain transistor level spice netlists of the logic gates. These logic gate netlists may be contained in a .sp or .spi file. The netlist lists every resistance and capacitance between nodes in the logic gate such that accurate power and timing of the logic gate can be simulated by HSpice. An example of an inverter from the Synopsys 90nm EDK library, which was used primarily in Chapter 3, is given below:

```
.SUBCKT INVX0 VDD VSS IN QN

*|NET QN
Cg1 QN 0 2.27785e-17
Cg2 M2:SRC 0 1.09209e-17
Cg3 M1:SRC 0 5.05047e-18
R1 QN M2:SRC 10.2254
R2 QN M1:SRC 10.3248

*|NET VDD
C4 M2:BULK QN 2.59792e-17
C5 M2:BULK M2:SRC 1.02811e-17
Cg6 M2:BULK 0 2.86883e-17
C7 VDD QN 4.86946e-17
C8 VDD M2:SRC 1.76669e-18
Cg9 VDD 0 5.88306e-17
C10 M2:DRN M2:SRC 1.51919e-18
Cg11 M2:DRN 0 9.38633e-18
R3 M2:BULK VDD 5.75539
R4 VDD M2:DRN 10.8827

*|NET IN
C12 M2:GATE VDD 6.28583e-18
C13 M2:GATE QN 7.25325e-18
C14 M2:GATE M2:SRC 1.32772e-17
Cg15 M2:GATE 0 3.23643e-17
C16 M1:GATE QN 1.52067e-18
C17 M1:GATE M1:SRC 9.77115e-18
Cg18 M1:GATE 0 2.32996e-17
C19 IN QN 5.43184e-17
C20 IN VDD 5.48736e-17
C21 IN M2:BULK 1.41622e-17
C22 IN M2:DRN 2.13681e-17
R5 M2:GATE M1:GATE 121.902
R6 M2:GATE IN 24.7454
R7 M1:GATE IN 70.8714

*|NET VSS
C23 M1:BULK M2:GATE 6.25494e-19
```

```
C24 M1:BULK M2:BULK 2.03571e-17
C25 M1:BULK VDD 7.71549e-17
C26 M1:BULK M1:GATE 1.8666e-19
C27 M1:BULK QN 2.21551e-17
C28 M1:BULK M2:DRN 2.79905e-18
C29 M1:BULK IN 6.44154e-17
C30 M1:BULK M1:SRC 9.56875e-18
Cg31 M1:BULK 0 1.8654e-17
C32 VSS IN 3.69918e-17
C33 VSS QN 8.74073e-17
C34 VSS VDD 1.60284e-17
C35 VSS M2:BULK 6.70277e-18
C36 VSS M2:GATE 8.66673e-18
C37 VSS M1:SRC 4.34729e-18
Cg38 VSS 0 5.90599e-17
C39 M1:DRN M1:GATE 3.59226e-18
C40 M1:DRN IN 6.5268e-18
C41 M1:DRN M2:DRN 8.9675e-19
C42 M1:DRN QN 1.44866e-18
C43 M1:DRN M2:BULK 2.63115e-18
C44 M1:DRN M1:SRC 2.95164e-18
Cg45 M1:DRN 0 1.10511e-17
R8 M1:BULK VSS 5.70998
R9 VSS M1:DRN 10.5874

*
* Instance Section
*
MM2 M2:DRN M2:GATE M2:SRC M2:BULK p12 ad=0.165p as=0.17p l=0.1u pd=1.7u ps=1.72u w=0.55u
MM1 M1:DRN M1:GATE M1:SRC M1:BULK n12 ad=0.072p as=0.072p l=0.1u pd=1.08u ps=1.08u w=0.24u

.ENDS
```

As can be seen, each input or output signal - IN, QN, VDD and VSS - has a set of capacitances and resistances associated with it given by lines starting with C and R. A Spice netlist is written such that the name and type of a fundamental component such as a resistance or capacitance is given by the first term on the line, e.g. 'C39' represents a capacitance. The two following strings name the two nodes in the netlist that the component fits between and finally the value or size of the component is given by the number at the end of the line. The 'MM' devices shown at the end of the netlist are the transistors in the logic gate. The models for the transistors are non-standard Spice components and are provided separately to the logic gate netlists and are normally obtained from the foundry. These models are kept in a .LIB file. The transistors generally use a BSIM 54 model and have a number of parameters that are set when the transistors are instanced in the logic gate netlists provided in the gate library. The transistors have four terminals, as opposed to two like the capacitance and resistances corresponding to the source, drain, gate and bulk of a transistor.

B.3.2 Extract RC netlist

Once a digital circuit has been synthesised and fully place and routed, a tool such as Synopsys Star-RC can be used to extract the full RC spice netlist of the layout. The EDA tool can use the spice netlist file of the technology library logic gates to correctly assemble a top level netlist for the digital circuit and this is done by creating instances of each logic gate. An example of how gates are instanced is given below and is taken from the isolation control circuit (Chapter 3, Fig. 3.4) in a real netlist of a sub-clock Cortex-M0:

```
...
Xiso/U1 iso/U1:VDD iso/U1:VSS iso/U1:IN iso/U1:QN INVX0
Xiso/U2 iso/U2:VSS iso/U2:VDD iso/U2:IN2 iso/U2:IN1 iso/U2:Q AND2X1
Xiso/U3 iso/U3:VSS iso/U3:VDD iso/U3:IN2 iso/U3:IN1 iso/U3:Q OR2X1
...
```

The name of the device instance is given first, followed by the connecting nets and finally the type of device being instanced. Just as was the case in the logic gates, the top level netlist of the digital circuit layout also has capacitance and resistances to model the parasitics on signal nets between logic gates and the power grid. It must be noted that the larger the netlist and the more parasitics the netlist models, the longer the simulation run-time will be when using HSpice as it must simulate and iterate around many more elements. The Cortex-M0 simulations in Chapter 3, for example have simulation times on average of 24 hours.

B.3.3 Simulation Vectors

The simulation done in HSpice needs some stimulus to the inputs of the digital circuit. In all simulations done in this thesis a digital vector (.vec) file was used in HSpice. This vector file defines a set of input signals to the digital circuit, and controls their value through the simulation time. An advantage of using the digital vector file is its ability to also define and check the outputs of the digital circuit as it is being simulated. This means the digital vector file will produce an error report file (.err) with signal value mismatches and at what time if an error is detected. An example of a digital vector file used in the simulation of the 16-bit multiplier in Chapter 3 is given below:

```
;define inputs and outputs
RADIX 4444 4444 44444444

;names of vectors
VNAME input1[[15:12]] input1[[11:8]] input1[[7:4]] input1[[3:0]]
input2[[15:12]] input2[[11:8]] input2[[7:4]] input2[[3:0]]
```

```
result_reg[[31:28]] result_reg[[27:24]] result_reg[[23:20]] result_reg[[19:16]]
result_reg[[15:12]] result_reg[[11:8]] result_reg[[7:4]] result_reg[[3:0]]

;input or output
IO IIII IIII OOOOOOOO

;Time units
TUNIT NS

;delay output checking by half clock cycle
tdelay HalfClkPeriod 0000 0000 FFFFFFFF

;specify high voltage for input signals
vih Supply FFFF FFFF 00000000

;specify low voltage for input signals
vil 0 FFFF FFFF 00000000

;threshold voltage between high and low states
vth HalfSupply 0000 0000 FFFFFFFF

;start of vector data with period of simulation clock period
PERIOD ClkPeriod
F0F0 0F0F 00000000
0F0F F0F0 00000000
A0CD 39FC 0E2C2E10
...
```

The digital vector file begins by defining the bus width (radix) in number of bits for each signal. The maximum size is 4 since a 4 bit signal can easily be assigned a hexadecimal value when the value of the signal is defined. The input and output signal names for the digital circuit are defined next and match the names in the spice netlist, this is followed by whether the signal is an input (I) or an output (O). The time unit is defined next and is followed by a few parameters which are only applied to a signal if its corresponding bit position is asserted i.e. a F means it is applied and a 0 means it is not. The output checking done by the digital vector file is delayed by half a clock period to ensure the outputs have stabilised. The maximum, minimum and threshold values of the logic signals are given next. Finally the digital vector file defines the values for the inputs and outputs for the simulation. The values can be set at a given time or can be listed and stepped through according to a period. In this example, a period is given and is set to the clock period of the simulation and means the first value is applied/checked on the inputs and outputs at time 0 and subsequent values are stepped through in time with the clock period.

In the case of the 16-bit multiplier circuit used in Chapter 3, the values of the inputs and outputs are easily computed and entered into the digital vector file. However, in the case of the ARM Cortex-M0 microprocessor used in the same chapter the value of these signals must be captured from a true functional simulation. Any Verilog simulator such as Mentor Modelsim or Synopsys VCS can be used to acquire the simulation vectors. As mentioned in Chapter 3, the Dhrystone benchmark was used and the values of the inputs and outputs were written out to a file at each clock tick. As simulation time was on average 24 hours with a circuit as large as the Cortex-M0 the HSpice simulations were started from the middle rather than reset. This meant the values of the registers also had to be captured from the functional simulation. These register values could then be captured into the registers before the vector file is read and the method used for this will be shown in the next section.

B.3.4 Netlist Simulation with HSpice

With the spice netlist for the layout obtained, the technology library spice files available and finally the simulation stimulus, the digital circuit can be simulated with HSpice. A transient simulation was used to measure power over a period of time. An example spice 'deck' used in the simulation of a the 16-bit multiplier in Chapter 3 is shown below:

```
**Analysis of power gated multiplier
.OPTIONS LIST NODE POST
.OPTION PROBE=1
.OPTION CONVERGE=1
.PROBE i(vsupply) v(clock) v(nreset) v(xmultiplier.vdd:20)
+v(xmultiplier.vddv:20) v(xmultiplier.vss:20) v(vdd) v(input1*)
+v(input2*) v(result_reg*) v(xmultiplier.result_reg*:d)
+ v(xmultiplier.input1_reg*:q) v(xmultiplier.input2_reg*:q)

**Simulation - Type, Resolution & Duration
.TRAN .1N 9U
.TEMP 100

*Transistor Models
.LIB '../../../lib/SAED90nm.lib' TT_12
.LIB '../../../lib/SAED90nm.lib' TT_12_HVT

*Spice netlists of Technology library and layout
.INCLUDE '../../../lib/saed90nm.spi'
.INCLUDE '../../../../data/pipelined_multiplier-netlist.spf'

*Digital vector file stimulus
.VEC './vectors'
```

```
**Power Supply
Vsupply VDD 0 0.6V

** Input signals
Vnreset nReset 0 PWL(0NS 0V  0.15US 0V  0.151US 0.6V)
Vclock clock 0 PULSE(0V 0.6v 0.5US 0.2NS 0.2NS 0.5US 1US)
Vnoverride nOverride 0 PWL(0NS 0.6v)

XMultiplier clock nReset input1[15] input1[14] input1[13] input1[12]
+input1[11] input1[10] input1[9] input1[8] input1[7] input1[6]
+input1[5] input1[4] input1[3] input1[2] input1[1] input1[0] input2[15]
+input2[14] input2[13] input2[12] input2[11] input2[10] input2[9]
+input2[8] input2[7] input2[6] input2[5] input2[4] input2[3] input2[2]
+input2[1] input2[0] result_reg[31] result_reg[30] result_reg[29]
+result_reg[28] result_reg[27] result_reg[26] result_reg[25] result_reg[24]
+result_reg[23] result_reg[22] result_reg[21] result_reg[20] result_reg[19]
+result_reg[18] result_reg[17] result_reg[16] result_reg[15] result_reg[14]
+result_reg[13] result_reg[12] result_reg[11] result_reg[10] result_reg[9]
+result_reg[8] result_reg[7] result_reg[6] result_reg[5] result_reg[4]
+result_reg[3] result_reg[2] result_reg[1] result_reg[0] nOverride VDD 0
+route_opt

*Measure average power
.meas tran ADDER_AVG AVG p(XMultiplier)

.END
```

To begin with the deck sets up some options for the simulation followed by the transient length of time and temperature of the simulation. The transistor models are sourced in the following lines from the .lib files. The spice netlists of the logic gates and the layout are gathered in the next lines and the final file to be sourced is the digital vector file which provides the stimulus to the simulation. Unlike the multiplication inputs and outputs defined in the digital vector file, the *nReset* and *nOVERRIDE* signal do not continuously change throughout the simulation and the *clock* simply pulses at the clock period. These three stimulus inputs are defined in the following lines; the *nReset* signal is pulsed at the start of the simulation to initialise the registers, the *nOverride* is forced off and the clock has a frequency of 1MHz in this example. To vary the clock duty cycle to maximise power savings with sub-clock power gating, as is done in Chapter 3, the pulse stimulus used by the *Clock* can be altered to change the length of time spent at logic 1 and 0. An instance of the multiplier layout is created next with a name of *XMultiplier*. The final line of the spice deck, before the .END, sets up the simulation to perform the power measurement. The measure command in this example simply performs an average on the entire transient of the power of the multiplier instance *XMultiplier*.

In the case of the ARM Cortex-M0 microprocessor simulations of sub-clock power gating in Chapter 3, the simulation is started at non-zero time due to the length of time required for an HSpice simulation of 10 vectors (Chapter 3, Section 3.3.2). The registers in the microprocessor must therefore be initialised to the correct values corresponding to a point midway through the benchmark program. To achieve this, the value of each register is captured from a functional netlist simulation and the following is done in the HSpice simulation to force the known value into a register. Each of the 'D' inputs to the registers is attached to a voltage supply through a voltage controlled resistor. At t=0 the voltage controlled resistors are effectively a short circuit forcing the 'D' inputs to a known value set by the voltage source. This value is then registered with a pulse of the clock and the voltage controlled resistor is then set to effectively an open circuit to stop it from interfering with the rest of the simulation. An example of how this is done in the spice deck is given below:

```
*Voltage to control Voltage controlled resistors
Vinitialise initialise 0 PWL(0NS 0.6V 0.6US 0.6V 0.601US 0V)

*Register initialiser
VTbo2a4_reg sw5 0 PWL(0NS 0.6V 0.75US 0.6V 0.751US 0V)
gsw5 xminiswift.m0/Tbo2a4_reg:d sw5 VCR PWL(1) initialise 0 0,1e8 0.6,0
```

In this example, the *initialise* control signal is asserted for 0.6μs during which time, the 'D' input of the register Tbo2a4 is connected to logic 1. In these 0.6μs, the value is clocked in with a rising edge of the clock after which the *initialise* signal is deasserted disconnecting the supply from the 'D' input of the register.

Appendix C

Scripts

C.1 Test Chip ARM Cortex-M0 UPF

```
#----------------------------------------
#Top Power Domain
#----------------------------------------
create_power_domain Top_PD -include_scope

create_supply_port VDD
create_supply_net VDD -domain Top_PD
connect_supply_net VDD -ports VDD

create_supply_port VSS
create_supply_net VSS -domain Top_PD
connect_supply_net VSS -ports VSS

create_supply_port VVSS_M0_Comb -direction out
create_supply_net VVSS_M0_Comb -domain Top_PD -resolve parallel
connect_supply_net VVSS_M0_Comb -ports VVSS_M0_Comb

create_supply_port VVDD_M0_Comb -direction out
create_supply_net VVDD_M0_Comb -domain Top_PD -resolve parallel
connect_supply_net VVDD_M0_Comb -ports VVDD_M0_Comb

set_domain_supply_net Top_PD \
-primary_power_net VDD \
-primary_ground_net VSS

#----------------------------------------
#M0 Top Power Domain
#----------------------------------------
create_power_domain M0_Top_PD -elements {M0}
```

```
create_supply_net VDD -domain M0_Top_PD -reuse
create_supply_net VSS -domain M0_Top_PD -reuse
create_supply_net VVDD_M0_Top -domain M0_Top_PD -resolve parallel

set_domain_supply_net M0_Top_PD \
-primary_power_net VVDD_M0_Top \
-primary_ground_net VSS

#----------------------------------------
#M0 Top VVDD Power Switch
#----------------------------------------
create_power_switch VVDD_M0_Top_sw \
-domain M0_Top_PD \
-input_supply_port {VDD VDD} \
-output_supply_port {VVDD_M0_Top VVDD_M0_Top} \
-control_port {sleep OFF} \
-on_state {on_state VDD {!sleep}}

#----------------------------------------
#M0 Top Outputs Isolation Stategy
#----------------------------------------

set_isolation M0_Top_isol \
-domain M0_Top_PD \
-isolation_power_net VDD \
-isolation_ground_net VSS \
-clamp_value 0

set_isolation_control M0_Top_isol \
-domain M0_Top_PD \
-isolation_signal ISO \
-isolation_sense high \
-location parent

set_isolation M0_Top_noIso \
-domain M0_Top_PD \
-elements M0/VIRTUALRAIL \
-no_isolation

#----------------------------------------
#M0 Combinational Logic Power Domain
#----------------------------------------
create_power_domain M0_Comb_PD -elements {M0/combinational}

create_supply_net VDD -domain M0_Comb_PD -reuse
create_supply_net VSS -domain M0_Comb_PD -reuse
```

```
create_supply_net VVSS_M0_Comb -domain M0_Comb_PD -resolve parallel -reuse

#Combinational logic drowsy power rail
create_supply_net VVDD_M0_Comb -domain M0_Comb_PD -resolve parallel -reuse

set_domain_supply_net M0_Comb_PD \
-primary_power_net VVDD_M0_Comb \
-primary_ground_net VVSS_M0_Comb

#-----------------------------------------
#M0 Combinational Logic VVDD Power Switch & Retention switch
#-----------------------------------------
create_power_switch VVDD_M0_Comb_sw0 -domain M0_Comb_PD \
-input_supply_port {VDD VDD} -output_supply_port {VVDD_M0_Comb VVDD_M0_Comb} \
-control_port {sleep M0/HEAD_SLEEP[0]} -on_state {on_state VDD {!sleep}}
create_power_switch VVDD_M0_Comb_sw1 -domain M0_Comb_PD \
-input_supply_port {VDD VDD} -output_supply_port {VVDD_M0_Comb VVDD_M0_Comb} \
-control_port {sleep M0/HEAD_SLEEP[1]} -on_state {on_state VDD {!sleep}}
create_power_switch VVDD_M0_Comb_sw2 -domain M0_Comb_PD \
-input_supply_port {VDD VDD} -output_supply_port {VVDD_M0_Comb VVDD_M0_Comb} \
-control_port {sleep M0/HEAD_SLEEP[2]} -on_state {on_state VDD {!sleep}}
create_power_switch VVDD_M0_Comb_sw3 -domain M0_Comb_PD \
-input_supply_port {VDD VDD} -output_supply_port {VVDD_M0_Comb VVDD_M0_Comb} \
-control_port {sleep M0/HEAD_SLEEP[3]} -on_state {on_state VDD {!sleep}}

create_power_switch VVDD_M0_Comb_ret0 -domain M0_Comb_PD \
-input_supply_port {VDD VDD} -output_supply_port {VVDD_M0_Comb VVDD_M0_Comb} \
-control_port {ret M0/HEAD_RET[0]} -on_state {on_state VDD {ret}}
create_power_switch VVDD_M0_Comb_ret1 -domain M0_Comb_PD \
-input_supply_port {VDD VDD} -output_supply_port {VVDD_M0_Comb VVDD_M0_Comb} \
-control_port {ret M0/HEAD_RET[1]} -on_state {on_state VDD {ret}}
create_power_switch VVDD_M0_Comb_ret2 -domain M0_Comb_PD \
-input_supply_port {VDD VDD} -output_supply_port {VVDD_M0_Comb VVDD_M0_Comb} \
-control_port {ret M0/HEAD_RET[2]} -on_state {on_state VDD {ret}}
create_power_switch VVDD_M0_Comb_ret3 -domain M0_Comb_PD \
-input_supply_port {VDD VDD} -output_supply_port {VVDD_M0_Comb VVDD_M0_Comb} \
-control_port {ret M0/HEAD_RET[3]} -on_state {on_state VDD {ret}}

#-----------------------------------------
#M0 Combinational Logic VVSS Power Switch & Retention switch
#-----------------------------------------
create_power_switch VVSS_M0_Comb_sw0 -domain M0_Comb_PD \
-input_supply_port {VSS VSS} -output_supply_port {VVSS_M0_Comb VVSS_M0_Comb} \
-control_port {sleep M0/FOOT_SLEEP[0]} -on_state {on_state VSS {sleep}}
create_power_switch VVSS_M0_Comb_sw1 -domain M0_Comb_PD \
-input_supply_port {VSS VSS} -output_supply_port {VVSS_M0_Comb VVSS_M0_Comb} \
-control_port {sleep M0/FOOT_SLEEP[1]} -on_state {on_state VSS {sleep}}
```

```
create_power_switch VVSS_M0_Comb_sw2 -domain M0_Comb_PD \
-input_supply_port {VSS VSS} -output_supply_port {VVSS_M0_Comb VVSS_M0_Comb} \
-control_port {sleep M0/FOOT_SLEEP[2]} -on_state {on_state VSS {sleep}}
create_power_switch VVSS_M0_Comb_sw3 -domain M0_Comb_PD \
-input_supply_port {VSS VSS} -output_supply_port {VVSS_M0_Comb VVSS_M0_Comb} \
-control_port {sleep M0/FOOT_SLEEP[3]} -on_state {on_state VSS {sleep}}

create_power_switch VVSS_M0_Comb_ret0 -domain M0_Comb_PD \
-input_supply_port {VSS VSS} -output_supply_port {VVSS_M0_Comb VVSS_M0_Comb} \
-control_port {ret M0/FOOT_RET[0]} -on_state {on_state VSS {!ret}}
create_power_switch VVSS_M0_Comb_ret1 -domain M0_Comb_PD \
-input_supply_port {VSS VSS} -output_supply_port {VVSS_M0_Comb VVSS_M0_Comb} \
-control_port {ret M0/FOOT_RET[1]} -on_state {on_state VSS {!ret}}
create_power_switch VVSS_M0_Comb_ret2 -domain M0_Comb_PD \
-input_supply_port {VSS VSS} -output_supply_port {VVSS_M0_Comb VVSS_M0_Comb} \
-control_port {ret M0/FOOT_RET[2]} -on_state {on_state VSS {!ret}}
create_power_switch VVSS_M0_Comb_ret3 -domain M0_Comb_PD \
-input_supply_port {VSS VSS} -output_supply_port {VVSS_M0_Comb VVSS_M0_Comb} \
-control_port {ret M0/FOOT_RET[3]} -on_state {on_state VSS {!ret}}


#----------------------------------------
#M0 Comb Outputs Isolation Stategy
#----------------------------------------

set_isolation M0_Comb_isol \
-domain M0_Comb_PD \
-isolation_power_net VVDD_M0_Top \
-isolation_ground_net VSS \
-clamp_value 0

set_isolation_control M0_Comb_isol \
-domain M0_Comb_PD \
-isolation_signal M0/ISOLATE \
-isolation_sense high \
-location parent

set_isolation M0_Comb_noIso \
-domain M0_Comb_PD \
-elements M0/combinational/VIRTUALRAIL \
-no_isolation

#----------------------------------------
#Port States
#----------------------------------------
add_port_state \
VDD                          -state {on 1.08 1.2 1.32}
```

```
add_port_state \
VSS                              -state {on 0 0 0}
add_port_state \
VVDD_M0_Top_sw/VVDD_M0_Top    -state {on 1.08 1.2 1.32} -state {off off}
add_port_state \
VVSS_M0_Comb_sw0/VVSS_M0_Comb -state {on 0 0 0} -state {off off}
add_port_state \
VVDD_M0_Comb_sw0/VVDD_M0_Comb -state {on 1.08 1.2 1.32} -state {off off}

create_pst M0_Top_pst -supplies {VSS VDD VVDD_M0_Top VVSS_M0_Comb VVDD_M0_Comb }
add_pst_state all_on        -pst M0_Top_pst -state  { on  on  on  on  on}
add_pst_state all_off       -pst M0_Top_pst -state  { on  on  off  off off }
add_pst_state scpg          -pst M0_Top_pst -state  { on  on  on  on off }
add_pst_state drowsy        -pst M0_Top_pst -state  { on  on  on  off  off }
```

C.2 dRail LEF Modification Script

```
1  eval 'exec perl -w -S $0 ${1+"$@"}'
2    if 0;
3
4  my $cellh = "";       # Cell height
5  my $cellw = "";       # Cell width
6  my $minx  = -0.045;  # Min X M1 spacing coordinate
7  my $olx   = -0.045;  # Overlap of bounding box in X
8  my $oly   = -0.160;  # Overlap of bounding box in Y
9
10 my $layer = "";
11 my $pin = "";
12
13 while (<>) {
14
15    ($cellw, $cellh) =  /\s*SIZE\s*(\S+)\s*BY\s*(\S+)/ if /SIZE/;
16
17    ($layer) = /\s*LAYER\s*(\S+)/ if /LAYER/;
18
19    ($pin) =  /\s*PIN\s*(\S+)/ if /PIN/;
20
21    ($pin) =  "" if /END\s$pin/;
22
23    if (($pin eq "VDD" || $pin eq "VSS") && $layer eq "M1" && /RECT /) {
24      ($rect, $left, $bottom, $right, $top) = /(\s*)RECT\s*(\S+)\s*(\S+)\s*(\S+)\s*(\S+)/;
25
26      # Truncated rails in X direction and shrink in Y direction
27      if ($left == $minx && $top > $cellh) { # VDD
28      printf "%s RECT %s %s %s %s ;\n", $rect, 0.0-$olx, $bottom, $cellw+$olx, $cellh+$oly;
29      if ($left == $minx && $bottom < 0) {   # VSS
30      printf "%s RECT %s %s %s %s ;\n", $rect, 0.0-$olx, 0.0-$oly, $cellw+$olx, $top;
31      }
32
33      # Truncate stubs in Y direction
34      if ($left != $minx && $top > $cellh) { # VDD
35      printf "%s RECT %s %s %s %s ;\n", $rect, $left, $bottom, $right, $cellh+$oly;
36      }
```

```
37        if ($left != $minx && $bottom < 0) {    # VSS
38        printf "%s RECT %s %s %s %s ;\n", $rect, $left, 0.0-$oly, $right, $top;
39        }
40
41        next; # line
42      };
43
44      print;
45
46  } # while
```

References

[1] K. C. Pokhrel. Physical and Silicon Measures of Low Power Clock Gating Success: An Apple to Apple Case Study. In *Synopsys User Group (SNUG), San Jose 2007*, 2007.

[2] K. Roy, S. Mukhopadhya, and H. Mahmoodi-Meimand. Leakage Current Mechanisms and Leakage Reduction Technique in Deep-Submicrometer CMOS Circuits. *The IEEE*, 91:305–327, 2003.

[3] M. Keating, D. Flynn, R. Aitken, A. Gibbons, and K. Shi. *Low Power Methodology Manual*. Springer, 2007.

[4] S. Pillai and S. Baswant. Considerations and Challenges for Optimizing Leakage-Power at Sub-45nm. In *Synopsys User Group (SNUG), Austin*, 2011.

[5] K. Shi and D. Howard. Sleep Transistor Design and Implementation - Simple Concepts Yet Challenges To Be Optimum. In *International Symposium on VLSI Design, Automation and Test*, 2006.

[6] S. Hanson, B. Zhai, K. Bernstein, D. Blaauw, A. Bryant, L. Chang, K.K. Das, W. Haensch, E.J. Nowak, and D.M. Sylvester. Ultralow-Voltage Minimum-Energy CMOS. *IBM Journal of Research and Development*, 50:469–490, 2006.

[7] L. Wei, Z. Chen, K. Roy, M.C. Johnson, Y. Ye, and V.K. De. Design and Optimization of Dual-Threshold Circuits for Low-Voltage Low-Power Applications. *IEEE Transactions On Very Large Scale Integration (VLSI) Systems*, 7:16–24, 1999.

[8] S. Mukhopadhyay, C. Neau, R. Cakici, A. Agarwal, C.H. Kim, and K. Roy. Gate Leakage Reduction for Scaled Devices Using Transistor Stacking. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 11:716–730, 2003.

[9] Y. Shin, S. Paik, and H. Kim. Semicustom Design of Zigzag Power-Gated Circuits in Standard Cell Elements. *IEEE Transactions On Computer-Aided Design of Integrated Circuits and Systems*, 28:327–339, 2009.

[10] S. Kim, S. V. Kosonocky, D. R. Knebel, K. Stawiasz, and M. C. Papaefthymiou. A Multi-Mode Power Gating Structure for Low-Voltage Deep-Submicron CMOS

ICs. *IEEE Transactions On Circuits and Systems-II:Express Briefs*, 54:586–590, 2007.

[11] H. Singh, K. Agarwal, D. Sylvester, and K.J. Nowka. Enhanced Leakage Reduction Techniques Using Intermediate Strength Power Gating. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 15:1215 –1224, 2007.

[12] N. S. Kim, K. Flautner, D. Blaauw, and T. Mudge. Circuit and Microarchitectural Technique for Reducing Cache Leakage Power. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 12:167–184, 2004.

[13] A. Agarwal, S. Mukhopadhyay, A. Raychowdhury, K. Roy, and C.H. Kim. Leakage Power Analysis and Reduction for Nanoscale Circuits. *IEEE Micro*, 26, 2006.

[14] N. Seki, Lei Zhao, J. Kei, D. Ikebuchi, Yu. Kojima, Yohei Hasegawa, H. Amano, T. Kashima, S. Takeda, T. Shirai, M. Nakata, K. Usami, T. Sunata, J. Kanai, M. Namiki, M. Kondo, and H. Nakamura. A Fine-Grain Dynamic Sleep Control Scheme in MIPS R3000. In *IEEE International Conference on Computer Design, 2008*, 2008.

[15] J. Seomun, I. Shin, and Y. Shin. Synthesis of Active-Mode Power-Gating Circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 31:391–403, 2012.

[16] C. Kim and K. Roy. Dynamic Vth Scaling Scheme for Active Leakage Power Reduction. In *Design, Automation and Test in Europe (D.A.T.E.) Conference*, 2002.

[17] ARM. *Cortex-M0 Devices Generic User Guide*. ARM, 2009.

[18] M. Hempstead, G.Y. Wei, and D. Brooks. System Design Considerations for Sensor Network Applications. In *International Symposium on Circuits and Systems*, 2008.

[19] TSMC. *TSMC 65nm CLN65LP HVT 1.20 Volt SC12 High Performance Standard Cell Library Databook Rev. 1.0*. ARM, 2010.

[20] ARM. *Cortex-A5 Technical Reference Manual, Revision:r0p0*. ARM, 2009.

[21] Chenming C. Hu. *Modern Semiconductor Devices for Integrted Circuits*. Prentice Hall, 2009.

[22] N.H.E. Weste and D. Harris. *CMOS VLSI Design: A Circuits and Systems Perspective*. Pearson Education, 2005.

[23] Y. Taur and T.H. Ning. *Modern VLSI Devices*. Cambridge University Press, 2009.

[24] J. Kao, A. Chandrakasan, and D. Antoniadis. Transistor Sizing Issues And Tool For Multi-threshold Cmos Technology. In *Proceedings of the 34th Design Automation Conference, 1997*, 1997.

[25] D. Lee, D. Blaauw, and D. Sylvester. Gate Oxide Leakage Current Analysis and Reduction for VLSI Circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 12, 2004.

[26] B. J. Sheu, D. L. Scharfetter, P. Ko, and M. Jeng. BSIM: Berkeley Short-Channel IGFET Model for MOS Transistors. *IEEE Journal Of Solid-State Circuits*, 22:558–566, 1987.

[27] N. S. Kim, T. Austin, D. Blaauw, T. Mudge, K. Flautner, J. S. Hu, M. J. Irwin, M. Kandemir, and V. Narayanan. Leakage Current: Moore's Law Meets Static Power. *Computer*, 36, 2003.

[28] M.Y. Qadri, H.S. Gujarathi, and K.D. McDonald-Maier. Low Power Processor Architectures and Contemporary Techniuqes for Power Optimization - A Review. *Journal of Computers*, 4, 2009.

[29] S. Rusu, S. Tam, H. Muljono, D. Ayers, J. Chang, B. Cherkauer, J. Stinson, J. Benoit, R. Varada, J. Leung, R.D. Limaye, and S. Vora. A 65-nm Dual-Core Multithreaded Xeon Processor With 16-MB L3 Cache. *IEEE Journal of Solid-State Circuits*, 42, 2007.

[30] M. K. Gowan, L. L. Biro, and D. B. Jackson. Power Considerations in the Design of the Alpha 21264 Microprocessor. In *Proceedings of the 35th Annual Design Automation Conference*, 1998.

[31] S. Segars. Low Power Design Techniques for Microprocessors. In *International Solid-State Circuits Conference Tutorial 2001*, 2001.

[32] R. Gonzalez and M. Horowitz. Energy Dissipation In General Purpose Microprocessors. *IEEE Journal Of Solid-State Circuits*, 31:1277–1284, 1996.

[33] S. Huda, M. Mallick, and J.H. Anderson. Clock Gating Architectures for FPGA Power Reduction. In *International Conference on Field Programmable Logic and Applications 2009*, 2009.

[34] Synopsys. *Synopsys Design Compiler User Guide Version B-2008.09*. Synopsys Inc., 2008.

[35] D. Rabe and W. Nebel. Short Circuit Power Consumption of Glitches. In *International Symposium on Low Power Electronics and Design*, 1996.

[36] S. Kim, J. Kim, and S.Y. Hwang. New Path Balancing Algorithm for Glitch Power Reduction. *IEEE Proceedings - Circuits, Devices and Systems*, 148, 2001.

[37] M. Hashimoto, H. Onoedera, and K. Tamaru. Input Reordering for Power and Delay Optimization. In *ASIC Conference and Exhibit 1997*, 1997.

[38] E. Athanasopoulou and C.N. Hadjicostis. Bounds on FSM Switching Activity. *Journal of Signal Processing Systems*, 53, 2008.

[39] Y. Huang, P. Chen, and T. Hwang. Switching-Activity Driven Gate Sizing and V_{th} Assignment for Low Power Design. In *Asia and South Pacific Conference on Design Automation*, 2006.

[40] T. D. Burd, T. A. Pering, A. J. Stratakos, and R. W. Brodersen. A Dynamic Voltage Scaled Microprocessor System. *IEEE Journal Of Solid-State Circuits*, 35:1571–1580, 2000.

[41] M. Gligor, N. Fournel, and F. Pétrot. Adaptive Dynamic Voltage and Frequency Scaling Algorithm for Symmetric Multiprocessor Architecture. In *Euromicro Conference on Digital System Design/Architectures, Methods and Tools*, 2009.

[42] S.Y. Bang, K. Bang, S. Yoon, and E.Y. Chung. Run-Time Adaptive Workload Estimation for Dynamic Voltage Scaling. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 28:1334–1347, 2009.

[43] K. Funaoka, A. Takeda, S. Kato, and N. Yamasaki. Dynamic Voltage and Frequency Scaling for Optimal Real-Time Scheduling on Multiprocessors. In *International Symposium on Industrial Embedded Systems 2008*, 2008.

[44] D. Sengupta and R.A. Saleh. Application-Driven Voltage Island Partitioning for Low-Power System-on-Chip Design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 28:316–326, 2009.

[45] H. Wu, M.D.F. Wong, and I. Liu. Timing Constrained and Voltage Island Aware Voltage Assignment. In *Proceedings of the 43rd Annual Design Automation Conference*, 2006.

[46] S. Borkar. Design Challenges of Technology Scaling. *IEEE Micro*, 19:23–29, 1999.

[47] R.H. Denard, F. H. Gaensslen, V. L. Rideout, E. Bassous, and A. R. Leblanc. Design of Ion-Implanted MOSFETs with Very Small Physical Dimensions. *IEEE Journal of Solid-State Circuits*, 98:668–678, 1974.

[48] D. Duarte, N. Vijaykrishnan, M.J. Irwin, H-S Kim, and G. McFarland. Impact of Scaling on The Effectiveness of Dynamic Power Reduction Schemes. In *IEEE International Conference on Computer Design*, 2002.

[49] M. Venkatasubramanian and V. D. Agrawal. Subthreshold Voltage High-k CMOS Devices Have Lowest Energy and High Process Tolerance. In *IEEE 43rd Southeastern Symposium on System Theory (SSST)*, 2011.

[50] B. Chu-Kung, S. Corcoran, G. Dewey, M.K. Hudait, J.M. Fastenau, J. Kavalieros, W.K. Liu, D. Lubyshev, M. Metz, K. Millard, N. Mukherjee, W. Rachmady,

U. Shah, and R. Chau. Advanced High-K Gate Dielectric for High-Performance Short-Channel $In_{0.7}Ga_{0.3}As$ Quantum Well Field Effect Transistors on Silicon Substrate for Low Power Logic Applications. In *IEEE International Electron Devices Meeting*, 2009.

[51] ITRS, White Paper - More-than-Moore last accessed July 2011. http://www.itrs.net/Links/2010ITRS/Home2010.htm.

[52] B.H. Calhoun and D. Brooks. Can Subthreshold and Near-Threshold Circuits Go Mainstream? *IEEE Micro*, 30:80 –85, 2010.

[53] Nvidia. Nvidia Tegra Multi-processor Architecture. 2011.

[54] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoko, S. Shigematsu, and J. Yamada. 1-V Power Supply High-Speed Digital Circuit Technology with Multithreshold-Voltage CMOS. *IEEE Journal of Solid-State Circuits*, 30:847–854, 1995.

[55] R. Jotwani, S. Sundaram, S. Kosonocky, A. Schaefer, V.F. Andrade, A. Novak, and S. Naffziger. An x86-64 Core in 32nm SOI CMOS. *IEEE Journal of Solid-State Circuits*, 46:162 –172, 2011.

[56] S. Kim, S. V. Kosonocky, and D. R. Knebel. Understanding and Minimizing Ground Bounce During Mode Transition of Power Gating Structures. In *International Symposium on Low Power Electronics and Design*, 2003.

[57] Y. Kikuchi, M. Takahashi, T. Maeda, M. Fukuda, Y. Koshio, H. Hara, H. Arakida, H. Yamamoto, Y. Hagiwara, T. Fujita, M. Watanabe, H. Ezawa, T. Shimazawa, Y. Ohara, T. Miyamori, M. Hamada, M. Takahashi, and Y. Oowaki. A 40nm 222mW H.264 Full-HD Decoding, 25 Power Domains, 14-Core Application Processor With x512b Stacked DRAM. *IEEE Journal of Solid-State Circuits*, 46:32 –41, 2011.

[58] K. Shi and D. Howard. Challenges in Sleep Transistor Design and Implementation in Low Power Design. In *Design Automation Conference*, 2006.

[59] IEEE 1801. *Unified Power Format (UPF) Standard*. IEEE, last accessed August 2012. http://standards.ieee.org/findstds/standard/1801-2009.html.

[60] Youngsoo Shin, Jun Seomun, Kyu-Myung Choi, and Takayasu Sakurai. Power Gating: Circuits, Design Methodologies, and Best Practice for Standard-Cell VLSI Designs. *ACM Transactions on Design Automation and Electronic Systems*, 15:28:1–28:37, 2010.

[61] J.D. Meindl and J.A. Davis. The Fundamental Limit on Binary Switching Energy for Terascale Integration (TSI). *IEEE Journal of Solid-State Circuits*, 35:1515 –1516, 2000.

[62] Bo Zhai, D. Blaauw, D. Sylvester, and K. Flautner. The Limit of Dynamic Voltage Scaling and Insomniac Dynamic Voltage Scaling. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 13:1239 –1252, 2005.

[63] A. Wang and A. Chandrakasan. A 180-mV Subthreshold FFT Processor Using Minimum Energy Design Methodology. *IEEE Journal of Solid-State Circuits*, 40:310–319, 2005.

[64] Steven C. Jocke, Jonathan F. Bolus, Stuart N. Wooters, Travis N. Blalock, and Benton H. Calhoun. A 2.6μW Sub-Threshold Mixed-Signal ECG SoC. In *Proceedings of the 14th ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED)*, 2009.

[65] J. Kwong, Y.K. Ramadass, N. Verma, and A.P. Chandrakasan. A 65nm Sub-Vt Microcontroller with Integrated SRAM and Switched Capacitor DC-DC Converter. *IEEE Journal of Solid-State Circuits*, 44:115–126, 2009.

[66] Bo Zhai, S. Pant, L. Nazhandali, S. Hanson, J. Olson, A. Reeves, M. Minuth, R. Helfand, T. Austin, D. Sylvester, and D. Blaauw. Energy-Efficient Subthreshold Processor Design. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 17:1127 –1137, 2009.

[67] L. T. Clark, E. J. Hoffman, J. Miller, M. Biyani, L. Luyn, S. Strazdus, M. Morrow, K. E. Velarde, and M. A. Yarch. An Embedded 32-b Microprocessor Core for Low-Power and High-Performance Applications. *IEEE Journal of Solid-State Circuits*, 36, 2001.

[68] S. V. Kosonocky, A. J. Bhavnagarwala, K. Chin, G. D. Gristede, A.-M. Haen, W. Hwang, M. B. Ketchen, S. Kim, D. R. Knebel, K. W. Warren, and V. Zyuban. Low-Power Circuits and Technology for Wireless Digital Systems. *IBM Journal of Research and Development*, 47, 2003.

[69] K. Tran, P. Nguyen, H. Lu, C. Phan, Q. Phan, H. Kudo, H. Masuda, S. Negishi, M. Yamamoto, K. Hirose, and Y. Okamoto. A Low-Power Processor for Portable Navigation Devices: 456mW at 400MHz and 24mW in Software Standby Mode. In *IEEE Asian Solid-State Circuits Conference*, 2008.

[70] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless Sensor Networks for Habitat Monitoring. In *International Workshop on Wireless Sensor Network and Applications*, 2002.

[71] K. Martinez, P. Padhy, A. Riddoch, H.L.R Ong, and J.K. Hart. Glacial Environment Monitoring using Sensor Networks. In *Proceedings of Real-World Wireless Sensor Networks*, 2005.

[72] M. Zakrzewski, S. Junnila, A. Vehkaoja, H. Kailanto, A. Vainio, I. Defee, J. Lekkala, J. Vanhala, and J. Hyttinen. Utilization of Wireless Sensor Network for Health Monitoring in Home Environment. In *International Symposium on Industrial Embedded Systems*, 2009.

[73] Texas Instruments. *MSP430 User's Guide*. TI, 2009.

[74] P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi. Hardware Design Experiences in ZebraNet. In *International Conference on Embedded Networked Sensor Systems*, 2004.

[75] U. Bilstrup and P. Wiberg. An Architecture Comparison between a Wireless Sensor Netowrk and an Active RFID System. In *International Conference on Local Computer Networks*, 2004.

[76] B.A. Warneke and K.S.J. Pister. An Ultra-Low Energy Microcontroller for Smart Dust Wireless Sensor Networks. In *IEEE International Solid-State Circuits Conference*, 2004.

[77] X. Liu, Y. Zheng, M. W. Phyu, F. N. Endru, V. Navaneethan, and B. Zhao. An Ultra-Low Power ECG Acquisition and Monitoring ASIC System for WBAN Applications. *IEEE Journal On Emerging and Selected Topics in Circuits and Systems*, 2, 2012.

[78] R. F. Yazicioglu, P. Merken, R. Puers, and C. V. Hoof. A 200μW Eight-Channel EEG Acquisition ASIC for Ambulatory EEG Systems. *IEEE Journal On Emerging and Selected Topics in Circuits and Systems*, 2, 2012.

[79] X. Liu, Y. J. Zheng, M. W. Phyu, B. Zhao, M. Je, and X. J. Yuan. A Miniature On-Chip Multi-Functional ECG Signal Processor with 30μW Ultra-Low Power Consumption. In *32nd Annual International Conference of the IEEE EMBS*, 2010.

[80] Internet of Things Initiative, last accessed October 2012. http://www.iot-i.eu/public/front-page.

[81] L. Atzori, A. Iera, and G. Morabito. The Internet of Things: A Survey. *Computer Networks*, 54:2787–2805, 2010.

[82] Debasis Bandyopadhyay and Jaydip Sen. Internet of Things: Applications and Challenges in Technology and Standardization. *Wireless Personal Communications*, 58:49–69, 2011.

[83] A. P. Chandrakasan and S. Sheng an R. W. Brodersen. Low-Power CMOS Digital Design. *IEEE Journal of Solid-State Circuits*, 27, 1992.

[84] Bipul C. Paul, Amit Agarwal, and Kaushik Roy. Low-Power Design Techniques for Scaled Technologies. *Integration the VLSI Journal*, 39:64–89, 2006.

[85] M. Rahman and C. Sechen. Post-synthesis Leakage Power Minimization. In *Design, Automation Test in Europe (D.A.T.E.) Conference, 2012*, 2012.

[86] Rodolfo P. Santos, Gabriela S. Clemente, Abel Silva-Filho, Cristiano Araújo, Adriano Sarmento, Manoel Lima, and Edna Barros. An Optimization Mechanism Intended for Static Power Reduction using Dual-V_{th} Technique. *Journal of Electrical and Computer Engineering*, 2012:1:1–1:12, 2012.

[87] Yifang Liu and Jiang Hu. A New Algorithm for Simultaneous Gate Sizing and Threshold Voltage Assignment. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29:223 –234, 2010.

[88] T. Luo, D. Newmark, and D. Z. Pan. Total Power Optimization Combining Placement, Sizing and Multi-Vt Through Slack Distribution Management. In *Asian and South Pacific Design Automation Conference, 2008*, 2008.

[89] N. Sirisantana and Kaushik Roy. Low-Power Sesign using Multiple Channel Lengths and Oxide Thicknesses. *Design Test of Computers, IEEE*, 21:56 – 63, 2004.

[90] B. Chung and J.B. Kuo. Gate-level Dual-Threshold Static Power Optimization Methodology (GDSPOM) Using Path-Based Static Timing Analysis (STA) Technique for SOC Application. *Integration, the VLSI Journal*, 41:9 – 16, 2008.

[91] A. Srivastava. Simultaneous Vt Selection and Assignment for Leakage Optimization. In *Proceedings of the 2003 International Symposium on Low Power Electronics and Design, 2003. ISLPED '03.*, 2003.

[92] P. Royannez, H. Mair, F. Dahan, M. Wagner, M. Streeter, L. Bouetel, J. Blasquez, H. Clasen, G. Semino, J. Dong, D. Scott, B. Pitts, C. Raibaut, and Uming Ko. 90nm Low Leakage SoC Design Techniques for Wireless Applications. In *IEEE International Solid-State Circuits Conference, 2005. Digest of Technical Papers*, 2005.

[93] P. Gupta, A.B. Kahng, P. Sharma, and D. Sylvester. Gate-Length Biasing for Runtime-Leakage Control. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25:1475 – 1485, 2006.

[94] T. Fukai, Y. Nakahara, M. Terai, S. Koyama, Y. Morikuni, T. Suzuki, M. Nagase, A. Mineji, T. Matsuda, T. Tamura, F. Koba, T. Onoda, Y. Yamada, M. Komori, Y. Kojima, Y. Yama, M. Ikeda, T. Kudoh, T. Yamamoto, and K. Imai. A 65nm-node CMOS Technology with Highly Reliable Triple Gate Oxide Suitable for Power-Considered System-on-a-Chip. In *Symposium on VLSI Technology, 2003. Digest of Technical Papers*, 2003.

[95] A.K. Sultania, D. Sylvester, and S.S. Sapatnekar. Gate Oxide Leakage and Delay Tradeoffs for Dual-Tox Circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 13:1362 –1375, 2005.

[96] Dongwoo Lee, H. Deogun, D. Blaauw, and D. Sylvester. Simultaneous State, Vt and Tox Assignment for Total Standby Power Minimization. In *Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings*, 2004.

[97] D. Bol, J. De Vos, C. Hocquet, F. Botman, F. Durvaux, S. Boyd, D. Flandre, and J. Legat. A 25MHz 7μW/MHz Ultra-Low-Voltage Microcontroller SoC in 65nm LP/GP CMOS for Low-Carbon Wireless Sensor Nodes. In *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2012.

[98] Ping Liu, J. Wang, M. Phan, M. Garg, R. Zhang, A. Cassier, L. Chua-Eoan, B. Andreev, S. Weyland, S. Ekbote, M. Han, J. Fischer, G.C.-F. Yeap, Ping-Wei Wang, Q. Li, C.S. Hou, S.B. Lee, Y.F. Wang, S.S. Lin, M. Cao, and Y.J. Mii. A Dual Core Oxide 8T SRAM Cell with Low Vccmin and Dual Voltage Supplies in 45nm Triple Gate Oxide and Multi Vt CMOS for Very High Performance Yet Low Leakage Mobile SoC Applications. In *Symposium on VLSI Technology (VLSIT)*, 2010.

[99] S. Narendra, V. De, S. Borkar, D.A. Antoniadis, and A.P. Chandrakasan. Full-Chip Subthreshold Leakage Power Prediction and Reduction Techniques for Sub-0.18μm CMOS. *IEEE Journal of Solid-State Circuits*, 39:501 – 510, 2004.

[100] S. Narendra, V. De, D. Antoniadis, A. Chandrakasan, and S. Borkar. Scaling of Stack Effect and its Application for Leakage Recution. In *International Symposium on Low Power Electronics and Design*, 2001.

[101] S. Hanson, Mingoo Seok, Yu-Shiang Lin, Zhi Yoong Foo, Daeyeon Kim, Yoonmyung Lee, N. Liu, D. Sylvester, and D. Blaauw. A Low-Voltage Processor for Sensing Applications With Picowatt Standby Mode. *IEEE Journal of Solid-State Circuits*, 44(4):1145 –1155, 2009.

[102] E. Rotem, A. Naveh, D. Rajwan, A. Ananthakrishnan, and E. Weissmann. Power-Management Architecture of the Intel Microarchitecture Code-Named Sandy Bridge. *IEEE Micro*, 32:20 –27, 2012.

[103] G. Dhiman and T.S. Rosing. System-Level Power Management Using Online Learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 28:676 –689, 2009.

[104] Youngsoo Shin, Sewan Heo, Hyung-Ock Kim, and Jung Yun Choi. Supply Switching With Ground Collapse: Simultaneous Control of Subthreshold and Gate Leakage Current in Nanometer-Scale CMOS Circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 15:758 –766, 2007.

[105] M. Horiguchi, T. Sakata, and K. Itoh. Switched-Source-Impedance CMOS Circuit for Low Standby Subthreshold Current Giga-Scale LSI's. *IEEE Journal of Solid-State Circuits*, 28:1131 –1135, 1993.

[106] H. Homayoun, A. Sasan, A.V. Veidenbaum, Hsin-Cheng Yao, S. Golshan, and P. Heydari. MZZ-HVS: Multiple Sleep Modes Zig-Zag Horizontal and Vertical Sleep Transistor Sharing to Reduce Leakage Power in On-Chip SRAM Peripheral Circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 19:2303 –2316, 2011.

[107] H. Kawaguchi, K. Nose, and T. Sakurai. A Super Cut-Off CMOS (SCCMOS) Scheme for 0.5-V Supply Voltage with Picoampere Stand-by Current. *IEEE Journal of Solid-State Circuits*, 35:1498 –1501, 2000.

[108] M. Drazdziulis, P. Larsson-Edefors, and L. Svensson. Overdrive Power-Gating Techniques for Total Power Minimization. In *IEEE Computer Society Annual Symposium on VLSI, 2007*, 2007.

[109] A. Valentian and E. Beigne. Automatic Gate Biasing of an SCCMOS Power Switch Achieving Maximum Leakage Reduction and Lowering Leakage Current Variability. *IEEE Journal of Solid-State Circuits*, 43:1688 –1698, 2008.

[110] K.J. Nowka, G.D. Carpenter, E.W. MacDonald, H.C. Ngo, B.C. Brock, K.I. Ishii, T.Y. Nguyen, and J.L. Burns. A 32-bit PowerPC System-on-a-Chip with Support for Dynamic Voltage Scaling and Dynamic Frequency Scaling. *Solid-State Circuits, IEEE Journal of*, 37:1441 – 1447, 2002.

[111] S. Shigematsu, S. Mutoh, Y. Matsuya, Y. Tanabe, and J. Yamada. A 1-V High-Speed MTCMOS Circuit Scheme for Power-Down Application Circuits. *IEEE Journal of Solid-State Circuits*, 32:861–869, 1997.

[112] H. Iwaki, H. Yoshida, H. Suzuki, T. Yamada, and S. Kurosawa. A Novel Powering-Down Scheme for Low Vt CMOS Circuits. In *VLSI Circuits 1998. Digest of Technical Papers*, 1998.

[113] Z. Zhang, X. Kavousianos, K. Chakrabarty, and Y. Tsiatouhas. A Robust and Reconfigurable Multi-mode Power Gating Architecture. In *24th International Conference on VLSI Design (VLSI Design)*, 2011.

[114] K. Flautner, N.S. Kim, S. Martin, D. Blaauw, and T. Mudge. Drowsy Caches: Simple Techniques for Reducing Leakage Power. In *International Symposium on Computer Architecture, 2002*, 2002.

[115] D.A. El-Dib, Z. Abid, and H.A. Shawkey. Investigating an Aggressive Mode for Drowsy Cache Cells. In *Canadian Conference on Electrical and Computer Engineering*, 2008.

[116] A. Abdollahi, F. Fallah, and M. Pedram. Leakage Current Reduction in CMOS VLSI Circuits by Input Vector Control. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 12:140 –154, 2004.

[117] Y. Xu, Z. Luo, Z. Chen, and X. Li. Minimum Leakage Pattern Generation Using Stack Effect. In *International Conference on ASIC 2003*, 2003.

[118] Lin Yuan and Gang Qu. Simultaneous Input Vector Selection and Dual Threshold Voltage Assignment for Static Leakage Minimization. In *IEEE/ACM International Conference on Computer-Aided Design, 2007*, 2007.

[119] H. Rahman and C. Chakrabarti. An Efficient Control Point Insertion Technique for Leakage Reduction of Scaled CMOS Circuits. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 52:496 – 500, 2005.

[120] C. Neau and K. Roy. Optimal Body Bias Selection for Leakage Improvement and Process Compensation Over Different Technology Generations. In *International Symposium on Low Power Electronics and Design (ISLPED)*, 2003.

[121] L.T. Clark, M. Morrow, and W. Brown. Reverse-Body Bias and Supply Collapse for Low Effective Standby Power. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 12:947 –956, 2004.

[122] HeungJun Jeon, Yong-Bin Kim, and Minsu Choi. Standby Leakage Power Reduction Technique for Nanoscale CMOS VLSI Systems. *IEEE Transactions on Instrumentation and Measurement*, 59:1127 –1133, 2010.

[123] S. Narendra, A. Keshavarzi, B. A. Bloechel, S. Borkar, and V. De. Forward Body Bias for Microprocessors in 130nm Technology Generation and Beyond. *IEEE Journal of Solid-State Circuits*, 38:696–701, 2003.

[124] J. W. Tschanz, S. G. Narendra, Y. Ye, B. A. Bloechel, S. Borkar, and V. De. Dynamic Sleep Transistor and Body Bias for Active Leakage Power Control of Microprocessors. *IEEE Journal Of Solid-State Circuits*, 38:1838–1845, 2003.

[125] Yung-Chih Liang, Ching-Ji Huang, and Wei-Bin Yang. A 320-MHz 8bit x 8bit Pipelined Multiplier in Ultra-Low Supply Voltage. In *IEEE Asian Solid-State Circuits Conference, 2008*, pages 73 –76, 2008.

[126] Z. Hu, A. Buyuktosunoglu, V. Srinivasan, V. Zyuban, H. Jacobson, and P. Bose. Microarchitectural Techniques for Power Gating of Execution Units. In *International Symposium on Low Power Electronics and Design (ISLPED)*, 2004.

[127] M. Sjalander, M. Drazdziulis, P. Larsson-Edefors, and H. Eriksson. A Low-Leakage Twin-Precision Multiplier using Reconfigurable Power Gating. In *IEEE International Symposium on Circuits and Systems, 2005*, 2005.

[128] K. Usami, M. Nakata, T. Shirai, S. Takeda, N. Seki, H. Amano, and H. Nakamura. Implementation and Evaluation of Fine-Grain Run-time Power Gating for a Multiplier. In *IEEE International Conference on IC Design and Technology, 2009*, 2009.

[129] T.T. Hoang and P. Larsson-Edefors. Data-Width-Driven Power Gating of Integer Arithmetic Circuits. In *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2012.

[130] K. Usami, M. Nakata, T. Shirai, S. Takeda, N. Seki, H. Amano, and H. Nakamura. Implementation and Evaluation of Fine-Grain Run-Time Power Gating For A Multiplier. In *International Conference on IC Design and Technology*, 2009.

[131] Chi-Hoon Shin, Myeong-Hoon Oh, Sung Nam Kim, and Seong Woon Kim. Fine-Grained Power Gating of Datapath using FSM. In *Networked Embedded Systems for Enterprise Applications*, 2011.

[132] Bin Liu, Yici Cai, Qiang Zhou, Jinian Bian, and Xianlong Hong. FSM Decomposition for Power Gating Design Automation in Sequential Circuits. In *Internation Conference on Application Specific Integrated Circuits*, 2005.

[133] L. Bolzani, A. Calimera, A. Macii, E. Macii, and M. Poncino. Enabling Concurrent Clock and Power Gating in an Industrial Design Flow. In *Design Automation and Test in Europe (D.A.T.E.)*, 2009.

[134] K. Usami and N. Ohkubo. A Design Approach for Fine-grained Run-Time Power Gating Using Locally Extracted Sleep Signals. In *International Conference on Computer Design 2006*, 2006.

[135] K. Usami and H. Yoshioka. A Scheme to Reduce Active Leakage Power by Detecting State Transitions. In *The 2004 47th Midwest Symposium on Circuits and Systems, 2004*, 2004.

[136] M. Hamada, T. Kitahara, N. Kawabe, H. Sato, T. Nishikawa, T. Shimazawa, T. Yamashita, H. Hara, and Y. Oowaki. An Automated Runtime Power-Gating Scheme. In *25th International Conference on Computer Design*, 2007.

[137] J. Seomun, I. Shin, and Y. Shin. Synthesis and Implementation of Active Mode Power Gating Circuits. In *Design Automation Conference 2010*, 2010.

[138] E. Macii, L. Bolzani, A. Calimera, A. Macii, and M. Poncino. Integrating Clock Gating and Power Gating for Combined Dynamic and Leakage Power Optimization in Digital CMOS Circuits. In *11th Euromicro Conference on Digital System Design Architectures, Methods and Tools, 2008*, 2008.

[139] K. Nose, M. Hirabayashi, H. Kawaguchi, L. Seongsoo, and T. Sakurai. Vth-Hopping Scheme to Reduce Subthreshold Leakage for Low-power Processors. *IEEE Journal Of Solid-State Circuits*, 37:413–419, 2002.

[140] A. Wang and A. P. Chandrakasan. Optimal Supply and Threshold Scaling for Subthreshold CMOS Circuits. In *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, 2002.

[141] S.M. Martin, K. Flautner, T. Mudge, and D. Blaauw. Combined Dynamic Voltage Scaling and Adaptive Body Biasing for Lower Power Microprocessors Under Dynamic Workloads. In *IEEE/ACM International Conference on Computer Aided Design, 2002*, 2002.

[142] L. Yan, Jiong Luo, and N.K. Jha. Joint Dynamic Voltage Scaling and Adaptive Body Biasing for Heterogeneous Distributed Real-Time Embedded Systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24:1030 – 1041, 2005.

[143] N. Mehta and B. Amrutur. Dynamic Supply and Threshold Voltage Scaling for CMOS Digital Circuits Using In-Situ Power Monitor. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 20:892 –901, 2012.

[144] Hao Xu, Wen-Ben Jone, and R. Vemuri. Novel Vth Hopping Techniques for Aggressive Runtime Leakage Control. In *23rd International Conference on VLSI Design, 2010*, 2010.

[145] Hao Xu, Wen-Ben Jone, and R. Vemuri. Aggressive Runtime Leakage Control Through Adaptive Light-Weight Hopping With Temperature and Process Variation. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 19:1319 –1323, 2011.

[146] Hao Xu, R. Vemuri, and W.-B. Jone. Selective Light Vth Hopping (SLITH): Bridging the Gap Between Runtime Dynamic and Leakage. In *Design, Automation Test in Europe (D.A.T.E.) Conference*, 2009.

[147] B.H. Calhoun, S. Khanna, R. Mann, and J. Wang. Sub-Threshold Circuit Design with Shrinking CMOS Devices. In *IEEE International Symposium on Circuits and Systems*, 2009.

[148] R.G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge. Near-Threshold Computing: Reclaiming Moore's Law Through Energy Efficient Integrated Circuits. *Proceedings of the IEEE*, 98:253 –266, 2010.

[149] Yu Pu, J. Pineda de Gyvez, H. Corporaal, and Yajun Ha. An Ultra-Low-Energy Multi-Standard JPEG Co-Processor in 65nm CMOS With Sub/Near Threshold Supply Voltage. *IEEE Journal of Solid-State Circuits*, 45:668 –680, 2010.

[150] D. Fick, R.G. Dreslinski, B. Giridhar, Gyouho Kim, Sangwon Seo, M. Fojtik, S. Satpathy, Yoonmyung Lee, Daeyeon Kim, N. Liu, M. Wieckowski, G. Chen, T. Mudge, D. Sylvester, and D. Blaauw. Centip3De: A 3930DMIPS/W Configurable Near-Threshold 3D Stacked System with 64 ARM Cortex-M3 Cores. In

IEEE International Solid-State Circuits Conference Digest of Technical Papers, 2012.

[151] Jianli Chen and Wenxing Zhu. An Analytical Placer for VLSI Standard Cell Placement. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 31:1208 –1221, 2012.

[152] Chien-Yen Wang, Chaomin Luo, and G.E. Jan. An efficient Full-and-Elimination approach for Floorplan Area Minimization. In *International Conference on Microelectronics (ICM)*, 2009.

[153] J.Z. Yan, C. Chu, and Wai-Kei Mak. SafeChoice: A Novel Approach to Hypergraph Clustering for Wirelength-Driven Placement. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 30:1020 –1033, 2011.

[154] A. Sathanur, L. Benini, A. Macii, E. Macii, and M. Poncino. Row-Based Power-Gating: A Novel Sleep Transistor Insertion Methodology for Leakage Power Optimization in Nanometer CMOS Circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 19:469–482, 2011.

[155] Hyo-Sig Won, Kyo-Sun Kim, Kwang-Ok Jeong, Ki-Tae Park, Kyu-Myung Choi, and Jeong-Taek Kong. An MTCMOS Design Methodology and its Application to Mobile Computing. In *Proceedings of the 2003 International Symposium on Low Power Electronics and Design (ISLPED)*, 2003.

[156] Chao Wang, Yit-Chow Tong, and Yu Shao. VLSI Design and Analysis of a Critical-Band Processor for Speech Recognition. In *Proceedings of the IEEE International SOC Conference, 2004*, 2004.

[157] Synopsys. *Digital Standard Cell Library SAED_EDK90_CORE Databook Rev. 1.8*. Synopsys, 2009.

[158] W.M. Elgharbawy and M.A. Bayoumi. Leakage Sources and Possible Solutions in Nanometer CMOS Technologies. *IEEE Circuits and Systems Magazine*, 5, 2005.

[159] J.N. Rodrigues, T. Olsson, L. Sornmo, and V. Owall. Digital Implementation of a Wavelet-Based Event Detector for Cardiac Pacemakers. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 52:2686 – 2698, 2005.

[160] R.J.M. Vullers, R.V. Schaijk, H.J. Visser, J. Penders, and C.V. Hoof. Energy Harvesting for Autonomous Wireless Sensor Networks. *IEEE Solid-State Circuits Magazine*, 2, 2010.

[161] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava. Power Management in Energy Harvesting Sensor Networks. *ACM Transactions on Embedded Computing Systems (TECS)*, 6, 2007.

[162] Richard York. Benchmarking in Context: Dhrystone. 2002.

[163] D. Juan, Y. Chen, M. Lee, and S. Chang. An Efficient Wake-Up Strategy Considering Spurious Glitches Phenomenon for Power Gating Designs. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 18:246–255, 2010.

[164] E. Pakbaznia, F. Fallah, and M. Pedram. Charge Recycling in Power-Gated CMOS Circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 27:1798–1811, 2008.

[165] A. Weddell, D. Zhu, G. V. Merrett, S. P. Beeby, and B. M. Al-Hashimi. A Practical Self-Powered Sensor System with a Tunable Vibration Energy Harvester. In *International Workshop on Micro- and Nano-Technology for Power Generation and Energy Conversion Applications 2012*, 2012.

[166] David Flynn. An ARM perspective on Addressing Low-Power Energy-Efficient SoC Designs. In *Proceedings of the 2012 ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED)*. ACM, 2012.

[167] Joseph Yiu. *The Definitive Guide to the ARM Cortex-M0*. Elsevier, 2011.

[168] Alan R. Weiss. Dhrystone Benchmark: History, Analysis, "Scores" and Recommendations. 2002.