CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Low power predictable memory and processing architectures
Authors	Chen, Jiaoyan
Publication date	2013
Original Citation	Chen, J. 2013. Low power predictable memory and processing architectures. PhD Thesis, University College Cork.
Type of publication	Doctoral thesis
Rights	© 2013, Jiaoyan Chen. - http://creativecommons.org/licenses/by-nc-nd/3.0/
Download date	2026-11-17 12:18:18
Item downloaded from	https://hdl.handle.net/10468/1174



Ollscoil na hÉireann



LOW POWER PREDICTABLE MEMORY AND PROCESSING ARCHITECTURES

Jiaoyan Chen

A Thesis Submitted to the National University of Ireland, Cork, in Fulfilment of the Requirements for the Degree of Doctor of Philosophy

Supervisor: Dr. Emanuel M. Popovici

Head of Department: Prof. Nabeel A. Riza

Department of Electrical and Electronic Engineering

University College Cork

April 2013

ACKNOWLEDGEMENT

To come to this point, I have been away from home since September, 2007. During the time, I missed some special moments and also experienced something I did not expect. The biggest lesson I have learned through this journey is to keep faith in adversity. Because those setbacks make me grow like it is Rain that grows flowers not just sunshine.

In May, 2008, while I was taking the exams for my master courses, my dear grandpa passed away, as his only grandson, I could not get back home in time. I wish I could have shared my proud and happiness with him and I know he would have been more proud and cheerful than I am.

I want to pay tribute to my dear parent for all their sacrifices and supports all the way. I cannot ask for more from them. It is them who make all this happen, to make my dream come true. I love you!

My supervisor Emanuel Popovici is the best supervisor I have met and could have met in my mind. His optimism, encouragement, guidance not only helps me with my research but also affects my attitude when I face difficulties. I am very grateful for what he has done for me in the last 5 years and wish him and his family all the best in the future.

Also, I want to thank all the staff and colleagues in our department for their kindness and help. Although I cannot name all of them here, I will keep their face in my mind and wish them the best luck.

At last, I would like to express my gratitude to the Science Foundation Ireland under Grant number 07/IN.1/I977 who funded me for my PhD study and also Synopsys, Ireland for their generous support in this project.

Chen Jiaoyan

January, 2013

ABSTRACT

Great demand in power optimized devices shows promising economic potential and draws lots of attention in industry and research area. In embedded system, mobile devices and wireless senor network in particular, which usually operate at relatively low frequency domain, battery-life is the prime interest. Many techniques and technologies have been implemented to achieve low power consumption. Due to the continuously shrinking CMOS process, not only dynamic power but also static power has emerged as a big concern in power reduction. Other than power optimization, average-case power estimation is quite significant for power budget allocation but also challenging in terms of time and effort.

In this thesis, we will introduce a methodology to support modular quantitative analysis in order to estimate average power of circuits, on the basis of two concepts named Random Bag Preserving and Linear Compositionality. It can shorten simulation time and sustain high accuracy, resulting in increasing the feasibility of power estimation of big systems. One block cypher and a reversible ripple carry adder are built to demonstrate the theory.

For power saving, firstly, we take advantages of the low power characteristic of adiabatic logic and asynchronous logic to achieve ultra-low dynamic and static power. We will propose two memory cells, which could run in adiabatic and non-adiabatic mode. About 90% dynamic power can be saved in adiabatic mode when compared to other up-to-date designs; while in non-adiabatic mode, our SRAM cells could still save up to 50% energy. With aggressive technology scaling, process variation is also taken into account during the simulation along with temperature variation. About 90% leakage power is saved. Both of the proposed designs improve write ability and good read ability compared to the conventional 6T SRAM cell.

Secondly, a novel logic, named Asynchronous Charge Sharing Logic (ACSL), will be introduced. In conventional delay-insensitive asynchronous circuits, high power consumption caused by dual-rail logic and comprehensive completion detectors sometimes restricts its popularity. ACSL addresses this by using charge sharing technology which has not been implemented in asynchronous logic before. Additionally, the realization of completion detection is simplified considerably. A class of processing units, such as carry look-ahead adder, multiplier and Booth multiplier, is designed and built to demonstrate the high energy

efficiency of ACSL. Not just the power reduction improvement, ACSL brings another promising feature in average power estimation called data-independency where this characteristic would make power estimation effortless and be meaningful for modular quantitative average case analysis.

Finally, a new asynchronous Arithmetic Logic Unit (ALU) with a ripple carry adder implemented using the logically reversible/bidirectional characteristic exhibiting ultra-low power dissipation with sub-threshold region operating point will be presented. The proposed adder is able to operate multi-functionally. 4 different sizes of ALU are built using the proposed adders and the domino adders with other static logic units. It manages to save about 10% to 26% average power for addition operations and 20% to 75% power for logical operations. At last, an online testing technology for reversible circuits, called reversible BILBO, is introduced. 100% fault coverage is reported with high efficiency.

Contents

List of Figures ..... **v**

List of Tables ..... **ix**

1 Introduction **1**

1.1 Power Dissipation in CMOS Technology 2

1.1.1 Dynamic power 3

1.1.2 Static power 4

1.2 Principles of Power Reduction 5

1.2.1 Lowering dynamic power 5

1.2.2 Other dynamic power reduction techniques 6

1.2.3 Lowering static power 7

1.3 Challenges for Modern ICs 10

1.3.1 Optimization and estimation 10

1.3.2 Challenges in controlling static power 10

1.3.3 Controlling memory power 11

1.3.4 Challenges in processing units 11

1.4 Research Work in this Thesis 12

1.5 Thesis Organization 13

2 Adiabatic Logic and Asynchronous Logic **14**

2.1 Adiabatic Logic 14

2.1.1 Adiabatic logic family 15

2.1.2 Power Clock Generator 16

2.1.3 Complex Adiabatic Circuits 20

2.2 Asynchronous Logic 24

2.2.1 Synchronous Logic versus Asynchronous Logic 24

2.2.2 Fundamental Protocols of Asynchronous Logic 26

2.2.3 Function Blocks 29

2.3 Conclusion 30

3 Average-case Power Estimation 31

3.1 Motivation 31

3.2 Average-Case Power Estimation 32

3.2.1 Random Bag Preservation 33

3.2.2 Linear Compositionality 34

3.2.3 Modular Quantitative Average-case Power Analysis 35

3.3 Experiment Demonstration 36

3.4 Conclusion 41

4 Ultra Low Power Memory cell Design 43

4.1 Motivation 43

4.2 Related Works 44

4.2.1 Conventional 6T SRAM 45

4.2.2 9T SRAM 45

4.2.3 NC-SRAM 46

4.3 Two Novel SRAM Cells 46

4.3.1 Proposed 8T SRAM 47

4.3.2 Proposed 9T SRAM 49

4.4 Power Saving Strategy 50

4.5 Write and Read Operations 51

4.5.1 Write Ability 52

4.5.2 Read Ability 53

4.6 Performance Evaluation 56

4.6.1 Simulation Waveforms 56

4.6.2 32x32 SRAM Array Dynamic Power Comparison 57

4.6.3 Static Power and Process Variation 59

4.7 Conclusion 62

5 Asynchronous Charge Sharing Logic 63

5.1 Introduction 63

5.2 Asynchronous Charge Sharing Logic 64

5.2.1 Charge Sharing Technology 64

5.2.2 Basic Block 66

5.2.3 General Operation of ACSL 67

5.2.4 Circuit Design and ACSL Architecture 67

5.2.5 Contributions of ACSL 73

5.3 Arithmetic Units Based on ACSL 74

5.3.1 One-bit Full Adder Comparison 74

5.3.2 Carry Look-ahead Adder 76

5.3.3 Array-based Multiplier 78

5.3.4 Booth Multiplier 81

5.4 ACSL Power Estimation 86

5.5 Conclusion 89

6 Asynchronous Sub-threshold Bidirectional ALU Design 91

6.1 Introduction 91

6.2 Background 92

6.2.1 Sub-threshold Logic 92

6.2.2 Reversible Logic 92

6.3 Proposed ALU Design 93

6.3.1 Proposed Full Adder Design 94

6.3.2 Proposed ALU Design 96

6.4 Performance Evaluation 98

6.5 Testing For Reversible Circuits 107

6.6 Conclusion 108

7 Contribution and Future Work 110

7.1 Future Work 110

References 114

List of Figures

Figure 1.1 Generic Architecture of an Embedded System 2

Figure 1.2 Dynamic Power Consumption Caused by Two Switching Events 4

Figure 1.3 Total Chip Dynamic and Static Power Dissipation Trends Based on the International Technology Roadmap for Semiconductors [14] 8

Figure 1.4 Mobile System on Chip (SOC) Power Consumption Trends [36] 8

Figure 2.1 Adiabatic Switching (a) Evaluation Phase (b) Energy Recuperation Phase [17] 15

Figure 2.2 (a) General Schematic for PFAL [59] (b) General Schematic for 2N-2N2P [64] (c) General Schematic for ECRL [64] 17

Figure 2.3 Energy Consumption per Switching versus Frequency for a CMOS Inverter, an ECRL inverter, a 2N-2N2P Inverter and a PFAL Inverter [55] 17

Figure 2.4 Asynchronous Resonant Adiabatic Power Clock Generators (a) 2N Asynchronous (b) 2N2P Asynchronous (c) 2N Synchronous (d) 2N2P Synchronous 19

Figure 2.5 (a) Switched Capacitor Clock Generator (b) Ideal Waveform of Stepwise Charging [54] . 20

Figure 2.6 Simulation Results of QSERL Inverter Chain versus Static-CMOS Inverter Chain [70] ... 21

Figure 2.7 Energy Dissipation per Cycle vs. Frequency for Multipliers Including Self-Test Logic [71] 21

Figure 2.8 Synchronous Architecture versus Asynchronous Architecture [77] 25

Figure 2.9 (a) Bundled Data Channel (b) 4-phase Protocol (c) 2-phase Protocol [79] 27

Figure 2.10 4-phase Dual-rail Protocol [77] 27

Figure 2.11 Two Completion Detection Structure [77] 28

Figure 2.12 Generic DDCVSL Gate [84] 29

Figure 3.1 A Linear Composed System Built from Randomness Preserving Components 35

Figure 3.2 Full Adder using Reversible Majority Gate and Sum Gate 37

Figure 3.3 Modular Four Bit Ripple Carry Adder 38

Figure 3.4 Modularity in DES 40

Figure 4.1 Microcontroller Power Consumption Distribution (a) A 32-bit MCU with 8k-byte SRAM (b) A cache with 8k-byte SRAM (c) Cell array with miscellaneous [110] 44

Figure 4.2 SRAM Cell Designs 45

Figure 4.3 Two proposed SRAM cells 47

Figure 4.4 Write-Trip-Point Comparison 52

Figure 4.5 (a) Static-Noise-Margin Comparison (b) SNM of the Proposed SRAM 54

Figure 4.6 Reading Operation of the Proposed 8T SRAM cell and the Proposed 9T SRAM Cell 55

Figure 4.7 Simulation waveforms of the Proposed 8T and 9T SRAM Cells 55

Figure 4.8 32x32 SRAM Arrays Power Comparison 58

Figure 4.9 Leakage Current in Idle Mode of the Proposed 9T SRAM cell 59

Figure 4.10 Monte Carlo -Temperature Variation 60

Figure 4.11 Monte Carlo –Process Variation 61

Figure 5.1 General Operation of ACSL 66

Figure 5.2 Simulation Waveforms for ACSL 66

Figure 5.3 Two Stage Architecture of ACSL Circuits 68

Figure 5.4 Schematics for (a) C-element (b) Dynamic AND gate (c) Sharing Detector (d) VPC_Shr (e) VPC_Ctrl 69

Figure 5.5 Signal Transition Diagram for (a)Four Phase Handshaking Protocol (b) ACSL Handshaking Protocol....69

Figure 5.6 (a) Generic PFAL Function Block (b) PFAL Full Adder....70

Figure 5.7 Schematic for (a) Dual-control Lines Latch (b) IACSL Memory....71

Figure 5.8 Simulation Waveforms for (a) Dual-control Lines latch (b) IACSL memory....73

Figure 5.9 Dynamic Power Comparison of Adder Designs....75

Figure 5.10 Parallel-prefix Adders (a) Kogge–Stone Adder (b) Sklansky Adder [127]....77

Figure 5.11 Block Diagram – 8x8-bit Multiplier....80

Figure 5.12 Block Diagram of a 16x16 Radix-4 Booth Multiplier....82

Figure 5.13 Two Booth Partial Product Generators....83

Figure 5.14 Block Diagram of Proposed PPG....83

Figure 5.15 Block Diagram of ACSL Booth Multiplier....84

Figure 6.1 Reversible Gates (a) Feynman Gate (b) Toffoli Gate....93

Figure 6.2 The CMDK Adder [143]....94

Figure 6.3 PTL Implementation of (a) Feynman Gate (b) Toffoli Gate [48]....95

Figure 6.4 Proposed Bidirectional Full Adder....96

Figure 6.5 (a) Forward Computation (b) Reverse Computation....97

Figure 6.6 Proposed ALU Block Diagram....97

Figure 6.7 Schematic of Sense Amplifier....99

Figure 6.8 Schematic of C-element....99

Figure 6.9 Schematic of Proposed Latch 100

Figure 6.10 Two-stage Execution Flow 100

Figure 6.11 Architecture of a Conventional ALU 101

Figure 6.12 Full Adders Power Comparison 103

Figure 6.13 Power Comparison of Addition Operation 106

Figure 6.14 Power Comparison of Logic Operation 106

Figure 6.15 Stuck-at-0 Fault Detection 108

Figure 6.16 Reversible BILBO Test Procedure 108

List of Tables

Table 2.1 Comparison of Adiabatic PCGs @ 100MHz …..……20

Table 2.2 Comparison of Adiabatic and Static CMOS Arithmetic Units at Different Running Frequency……..…….22

Table 3.1 Truth Table for *XOR* Gate and *AND* Gate…….33

Table 3.2 Truth Table for Majority Gate and Sum Gate……38

Table 3.3 4-bit Full Adder Power Consumption……39

Table 3.4 Power Summary of Sum of Each Components, Single Round DES…….....41

Table 5.1 Power, Speed Comparison of Four Full-Adders…….....75

Table 5.2 Comparison of Energy, Area and Static Power…….....78

Table 5.3 Power, Speed, Area Comparison of Four Multipliers…….....79

Table 5.4 ACSL PPG versus Standard PPG…….....83

Table 5.5 Performance Comparison with the Existing Multipliers.....86

Table 5.6 Power Summary of 8-bit ACSL Carry Look-ahead Adder.....88

Table 6.1 ALU Modes of Operation.....100

Table 6.2 Transistor Count Comparison.....101

Table 6.3 Dynamic and Leakage Power of Full Adders.....102

Table 6.4 Power Comparison of ALUs.....105

List of Publications

- **J.Chen**, D.Vasudevan, E.Popovici, M.Schellekens, "**Ultra Low Power Asynchronous Charge Sharing Logic**", Journal of Low Power Electronics, August.2012, vol. 8,pp.526-534.
- M.Schellekens, D.Early, D.Vasudevan, **J.Chen**, A.Gao, E.Popovici, "**MOQA logic Units For Modular Power Analysis**", Submitted to The Journal of Logic and Algebraic Programming, January.2012.
- **J.Chen**, D.McCarthy, A.Amaricai, E.Popovici, "**Novel Asynchronouos Charge Sharing Logic for Ultra Low Power and Easy-predictable Arithmetic Operations**", Submitted to International Symposium on Low Power Electronics and Design (ISLPED), March, 2013.
- **J.Chen**, D.Vasudevan, E.Popovici, M.Schellekens,"**Ultra Low Power Booth Multiplier Using Asynchronous Logic**", IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC), 2012, pp.81-88.

- **J.Chen**, D.Vasudevan, E.Popovici, M.Schellekens, "**Design of a low power, Sub-Threshold, Asynchronous Arithmetic Logic Unit Using a Bidirectional Adder**", 14th IEEE Euromicro Conference on Digital System Design (DSD), 2011, pp. 301-308.
- **J.Chen**, D.Vasudevan, E.Popovici, M.Schellekens, P.Gillen, "**Design and analysis of a novel 8T SRAM cell for adiabatic and non-adiabatic operations**", 17th IEEE International Conference on Electronics, Circuits, and Systems (ICECS), 2010, pp. 434-437.
- T.Ye, **J.Chen**, D.Vasudevan, E.Popovici, M.Schellekens, "**Static Average Case Power Estimation Technique for Block Ciphers**", 13th IEEE Euromicro Conference on Digital System Design (DSD), 2010, pp.689-696.
- **J.Chen**, D.Vasudevan, E.Popovici, M.Schellekens, "**Reversible Online BIST using a Bidirectional BILBO**", ACM International Conference on Computing Frontiers (CF), 2010,pp. 257-266.
- D.Vasudevan, M.Goudarzi, **J.Chen** ,E.Popovici, M.Schellekens, "**A Reversible MIPS Multi-cycle Control FSM Design**", the IEEE Asia Symposium on Quality

Electronic Design (ASQED), 2009, pp. 336-342.

- M.Goudarzi, **J.Chen** , D.Vasudevan, E.Popovici, M.Schellekens, "**Reversing Deterministic Finite State Machines**", IET 20th Irish Signals and Systems Conference (ISSC), 2009, pp.19-23.
- **J.Chen** , D.Vasudevan, E.Popovici, "**A Memory Cell**", 1416239, European Patent Office, 2011.

1 INTRODUCTION

Embedded systems [1] are found in many applications including consumer electronics, industrial control, medical electronics (mHealth), appliances, home automation, automotive, wireless sensor networks, energy metering, entertainment, sports, etc. All these applications have a dramatic impact on improving our quality of life. In general, embedded systems are referred to systems which have a computing component (processor or microcontroller or DSP), memory along with other hardware designed for a specific application [2], shown in Fig.1.1. The key characteristic is that these systems are optimized to handle a particular task at the lowest cost. Very often, these systems, which include real-time system such as real-time DSP (Digital Signal Processing), have soft constraints (average case) and cost constraint rather than hard constraints (worst case). Therefore, the system is usually less complex than a general-purpose computer for example.

In the broad space of the embedded systems, a special case is attributed to mobile devices. These are driven by some specific constraints on power consumption. In this context, some of the most stringent power consumption constraints are present in the context of wireless sensor networks (WSN) [3], which consists of a class of miniature wireless devices called nodes. These usually battery-powered nodes are spatially distributed to monitor various data, such as environmental, medical, structural etc., and communicated through low power radio transceivers [4]. The main characteristic of these applications is the requirement for long battery life. While for mobile devices, the battery life lasts from several hours to couple of days depending customer usage, in the world of WSN, the battery life span could be days (medical applications), months (environmental monitoring) or even years (structural health). In order to maintain very long battery lifespan, energy harvesting [5] is sometimes employed.

Most modern mobile embedded systems including the nodes in WSN integrate one or several processors, memory, sensors, interface blocks and the trend is to merge more of these units into a single System in a Package or System on a Chip with a view to decreasing the power consumption and reduce cost. CMOS technology scaling [6] and developed power management techniques [7-9] play a major role in suppressing the increase of energy consumption. However, with the shrinking CMOS process, especially when down to so-called deep sub-micron regime (technology nodes below 100nm) [10], some new issues arise,

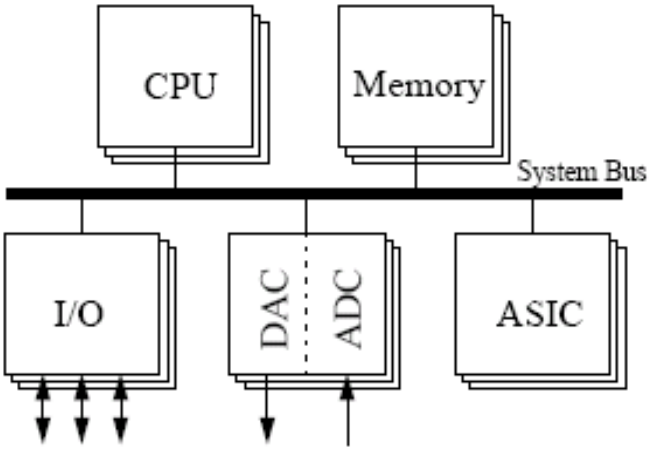


Figure 1.1 Generic Architecture of an Embedded System

such as increased static power, reliability issues, mismatch, performance variation, etc.

Therefore, power dissipation in the above mentioned applications is becoming the prime design concern, which demands optimizations at all levels from software down to physical-implementation. An interesting characteristic of many embedded systems is that the clock speed requirements are relatively low while compared to high performance computing systems (orders of magnitude lower clock frequencies). This allows a niche opportunity to achieve power optimization. However, in order to minimize cost, these systems are usually optimized for the average case. An efficient optimization algorithm, in turn, requires efficient performance estimation for the average case which is a hard problem [11, 12]. Finally, the design of completely predictable embedded systems which can be analysed statically is also a hard problem [13]. In this thesis, some novel low power architectures and techniques will be introduced along with a new average power estimation technique.

In the following sections, the basic components of power consumption are introduced first. A number of methodologies to achieve low power dissipation are explained. A list of challenges for current power optimization methods is identified. At last, but not least, the aim of this research and thesis outline is given.

1.1 Power Dissipation in CMOS Technology

As low power designs have drawn a significant attention in both industry and academia,

numerous techniques and technologies have been considered during the past decade. There are three sources to the total power consumption: the dynamic power, the static power and power dissipated due to the short-circuit current. The dynamic power was the major component for power consumption for sub-micron CMOS process such as 0.35μm, 0.18μm, 0.13μm, etc. However, for deep sub-micron CMOS technologies (45nm, 28nm, etc), static (leakage) power starts to be increasingly significant or even dominates the total power consumption [14] especially for applications which run at relatively low clock frequency. This trend inspires our research in this area, with both dynamic and static power needed to be minimized. The total power dissipated in a CMOS circuit can be calculated as follows:

$$P_{total} = P_{dynamic} + P_{static} + P_{short-circuit} \tag{1.1}$$

It should be noted that the short circuit current could be minimized by matching the rise/fall time of input and output signals and is usually neglected in most cases due to its quite short time [15]. It is also worthwhile to distinguish the definition of energy consumption and power consumption. Energy consumption by electronic circuits is calculated as the product of average power consumption and operation time, which has a direct influence on the battery lifetime while power consumption is critical to dimensioning the power supply, and managing the cooling systems. Hence, the average power consumption and total operation time need to be both targeted to minimize the total energy consumption. The operation time minimization is a result of enhancing the performance of the circuits, which often conflicts with power reduction. This two-dimensional optimization problem greatly complicates CMOS circuits design.

1.1.1 Dynamic power

The dynamic power, also regarded as switching power, is consumed when the state changes occur in CMOS circuits due to charging and discharging of load capacitance. The equation of dynamic power consumption of CMOS switching event is given by [16]:

$$P_{dynamic} = f * C_L * V_{DD}^{\ 2} \tag{1.2}$$

where f is the product of the switching activity of transition times input rate, C_L is the capacitive load at the transition node and V_{DD} is the power supply. From (1.2), it can be easily



(a) Charging **(b) Discharing**

Figure 1.2 Dynamic Power Consumption Caused by Two Switching Events

concluded that there are three ways to reduce the dynamic power by lowering the switching activity, load capacitance and the supply voltage. Additionally, it can be concluded that the dynamic power consumption is not directly related to transistor size, or switching delay, while it is data dependent, reflected as the switching activity. Fig.1.2 shows two switching events in a standard CMOS inverter, one is to charge the output load C_L from 0 to V_{DD}, the other is to discharge the capacitive load from V_{DD} to 0. A fixed energy defined by $½\ C_L V_{DD}^2$ gets dissipated in a logic gate's MOSFETs whenever its output toggles. The total energy dissipation of these two events is then $C_L V_{DD}^2$ [16, 17].

1.1.2 Static power

The static power in CMOS circuits is due to the leakage current. It exists even if a transistor is in a stable state once it is powered-on. Leakage increases exponentially as the CMOS technology scales down, especially as the thickness of the insulating region decreases. In the deep sub-micron technologies, the leakage power has emerged as one of the primary concerns in power optimization, which also limits the further-increase of circuits' performance [18].

Static power consumption is mainly caused by the sub-threshold leakage current and gate oxide leakage current where sub-threshold leakage usually dominates. Sub-threshold leakage is caused by unwanted current flow from drain to source while the transistor is operating in the weak inversion region.The sub-threshold leakage current is expressed as [19]:

$$I_{sub} = K_1 W e^{\frac{V_G - V_{th}}{mV_T}} \left(1 - e^{\frac{-V_{DS}}{V_T}}\right) \tag{1.3}$$

where K_1 and m are experimentally derived parameters (the factor of length L is involved), W is the gate width, V_T is thermal voltage, roughly being 26mV at room temperature, V_G is the gate voltage, V_{DS} is known as drain-source voltage while V_{th} is the threshold voltage. Sub-threshold leakage increases exponentially with $(V_G - V_{th})$. To lower the dynamic power without paying speed penalty, the threshold voltage V_{th} reduces as long as the successive decrease of the power, it meanwhile leads to an exponential raise in sub-threshold conduction, I_{sub}. On the other hand, gate oxide leakage current is defined by the current flow from the gate electrode through the thin gate insulating layer into the substrate. Static power is also caused by leakage current through reverse biased diodes (between diffusion regions and wells). In modern technology processes, the diode leakage is very small compared to the other two components, so it may be neglected during power calculations. In this thesis, main focus will be on how to reduce sub-threshold leakage current while gate-oxide leakage current would also be considered in the SRAM cell designs.

1.2 Principles of Power Reduction

In the 1990s, when low power CMOS designs started to draw significant research attention, the work was focused almost exclusively on reducing dynamic power. CMOS constant field scaling has increased function density and reduced transistor dynamic power by orders of magnitude since then. However, static power cannot be neglected with the development of CMOS technology, because the stand-by power is becoming significant for the battery-powered devices. This section presents the general principles of both dynamic and static power reduction along with several up-to-date low power CMOS design techniques.

1.2.1 Lowering dynamic power

Based on equation 1.2, among all factors, it is obvious that decreasing the power supply V_{DD} has the biggest impact because of its comparatively low design effort and easy controllability even though the effect by lowering V_{DD} in deep sub-micron process is not as significant as that in sub-micro region. Nevertheless, it is still an important method to reduce the dynamic power and several techniques have been proposed. Dynamic voltage and frequency scaling (DVFS) technique [20] is used in some computer architectures, particularly embedded microprocessors, by powering circuits with different levels of voltage depending upon the requirement of data throughput. Additionally, Sub-threshold logic [21] is an extreme

methodology to reduce V_{DD} close to or lower than the threshold voltage of MOS transistors, while they work in weak-inversion region. In this way, the dynamic power could be suppressed to an extremely low level at the expense of performance.

Besides manipulating the power supply, we could also achieve the power reduction by lowering the capacitive load, C_L, which is normally decided by the function of fan-out, wire length, transistor sizes. In order to optimize the capacitance, careful selection of gate size and layout are crucial. CAD tools could be used to prevent busy signals from driving the heavier-loaded gate inputs according to the switching activity information.

The last factor is the switching activity, which depends on the clock frequency and input data of the circuit. To decrease this part in particular, circuit-level techniques, Operand Isolation [22] and Precomputation [23], Register Retiming [24], Bus Encoding [25], Routing and Placement techniques, etc, are all effective.

1.2.2 Other dynamic power reduction techniques

Clock gating [26] reduces the dynamic power by inserting more logic to a circuit to prune the clock tree. By doing so, the gated clock prevents the flip-flop from dissipating unnecessary power on the clock edge. Clock gating is widely used in today's synchronous system designs and can be automatically implemented by CAD tools, like Synopsys Power Complier. This reliable and straightforward power optimization technique can save significant die area as well as power, due to replacing a significant number of multiplexers (MUXes) with clock gating logic. Moreover, as a large proportion of dynamic power in modern ICs is due to the clock tree, applying clock gating technique could also reduce the power in this area [27]. However, it may have a few minor drawbacks, such as reduced scan test coverage and increased clock skew.

As for the design of digital circuits, different logic styles are considered to satisfy various constraints and for different purposes. Among them, asynchronous logic [28] and adiabatic logic [29] are two unconventional logic styles which both target on power saving with different strategies. While asynchronous logic is used for average case performance synthesis and optimization, the adiabatic logic was shown to perform well in the low clock frequency domains (below 200MHz in particular).

For asynchronous logic, its main advantage is that it dispenses with the global clock signal employed to synchronous digital circuitry. It operates relying on data changes to trigger and propagate circuit activity, through a so-called handshake protocol. Hence, dynamic power is only consumed when data changes occur. Moreover, it allows the system to run as quick as possible unlike that synchronous systems are restricted by safety margin (worst-case), an asynchronous MIPS R3000 microprocessor [30] proves that asynchronous logic is also suitable for high-performance applications with efficient design. All the clock tree problems of synchronous designs are also eliminated at an expense of more customized layout design. The system becomes more robust to the variability which affects deep sub-micron CMOS. A number of asynchronous implementations of CPUs and application-specific processors have been reported [31-33]. This logic has also been used to improve the performance of the interconnect [34] and it is also present in some modern high performance architectures such as GALS [35].

Adiabatic logic was proposed initially in the context of reversible computing and reversible logic and it is proved to have ultra-low power characteristic. It powers the circuit by AC supply rather than DC supply for the evaluations, reducing the current flow through transistors and thereby to achieve considerable dynamic power reduction. This logic has been a hot topic during the past couple of years as a potential technology for embedded systems. Adiabatic logic shows great potential in low-throughput, energy-constrained applications. The details of asynchronous logic and adiabatic logic will be introduced in Chapter 2.

1.2.3 Lowering static power

At the 2002 International Electron Devices Meeting, Intel chairman Andrew Grove cited off-state current leakage in particular as a limiting factor in future microprocessor integration [14]. This situation gets exacerbated following the process scaling trend. Fig.1.3 [14] represents the total chip dynamic and static power dissipation trends based on the international roadmap for semiconductors. It can be seen that sub-threshold leakage power and gate-oxide leakage power approached and then exceeded the dynamic power around 2005 when the transistor size (represented by gate length) dropped below 65nm. Although gate-oxide leakage current might get lowered by using high-k material, it is yet inevitable that the static power eventually surpasses the dynamic power. It is also shown in Fig.1.4 [36] that the increase of static power, especially in memory, surpasses the dynamic power.

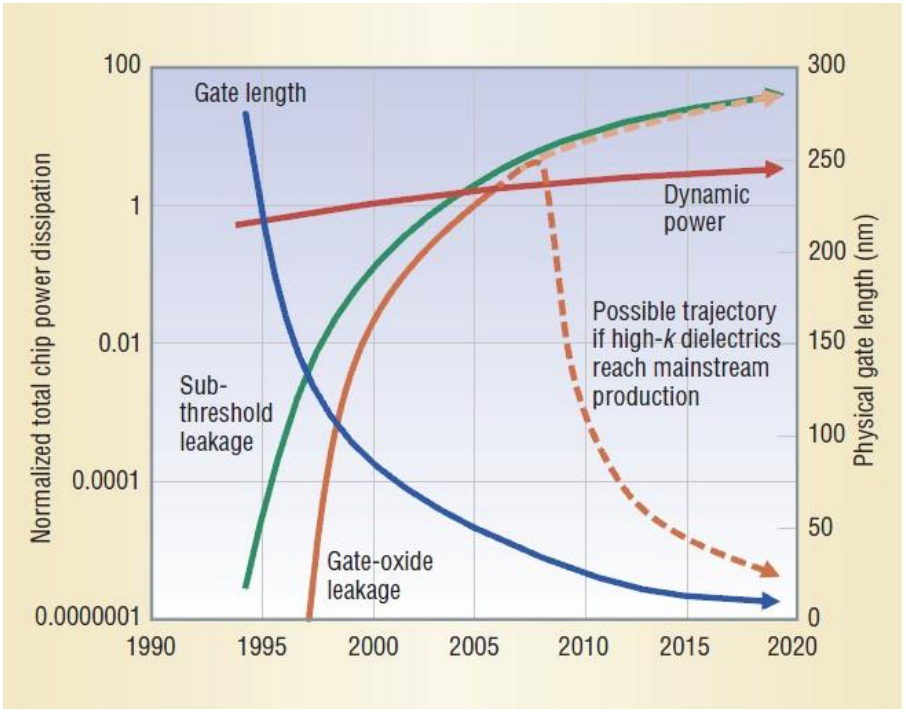


Figure 1.3 Total Chip Dynamic and Static Power Dissipation Trends Based on the International Technology Roadmap for Semiconductors [14]

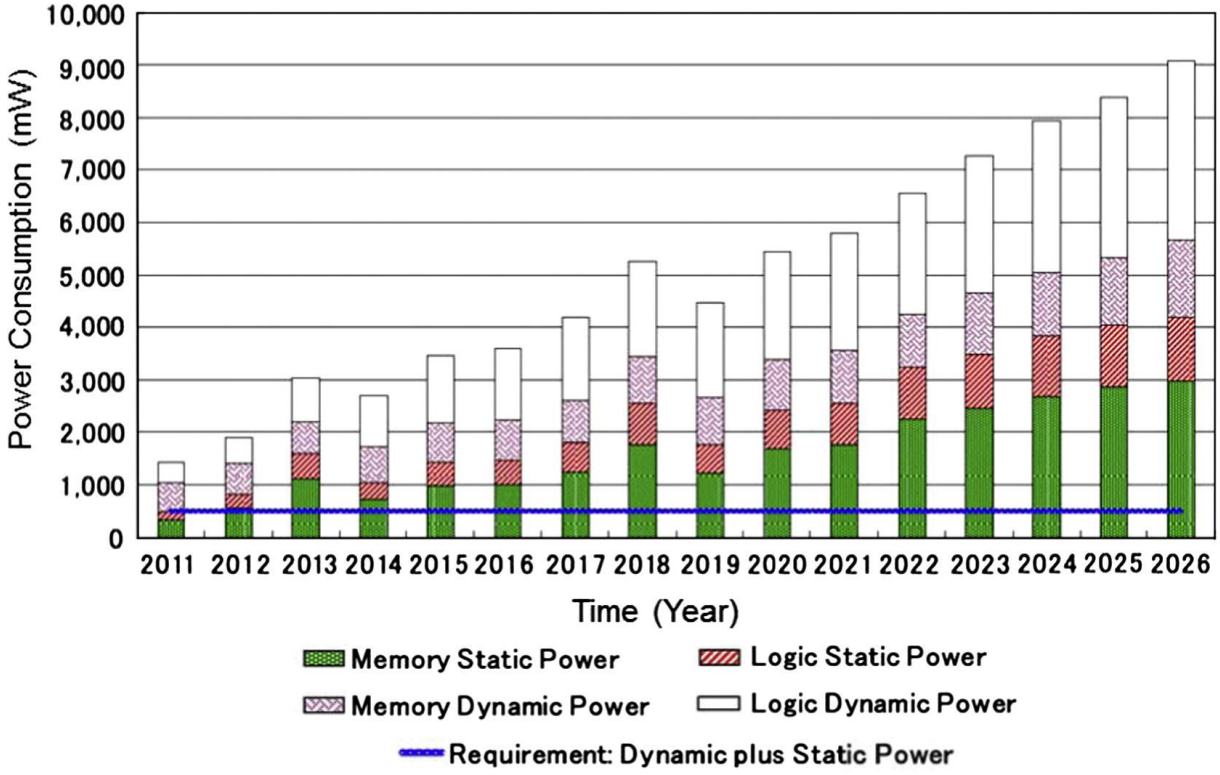


Figure 1.4 Mobile System on Chip (SOC) Power Consumption Trends [36]

As shown in equation 1.3, there are two direct ways to reduce the sub-threshold leakage current, which are lowering the supply voltage and increasing the threshold voltage. Unfortunately, the speed would have been degraded with the increase of the threshold voltage, because it affects the I-V characteristic of a transistor. Therefore, Multiple Threshold Voltage techniques offer designers various choices of $V_{th,}$ where fast devices are selected on timing-critical paths and High-V_{th} devices are used elsewhere to reduce leakage [8]. Multi-V_{th} transistors require modifications to the fabrication process such as adjusting the gate oxide thickness, gate oxide dielectric constant or adding additional photolithography and ion implantation steps, which now is available in many modern deep sub-micro CMOS technology. Another method is to apply reverse bias voltage to the base or bulk terminal of the transistors [37].

Multithreshold CMOS (MTCMOS) is commonly implemented in the context of Power Gating [38] in the form of "sleeping" transistors, creating a virtual power rail. The sleep transistors in the stack isolate the real power supplies from the virtual power rail and thereby leakage current in the unused circuit blocks is almost eliminated when high threshold voltage devices are employed.

Coarse-grained (block-wise) power gating and fine-grained approach are two alternatives to implement MTCMOS. For the former, logic blocks are partitioned to determine when a block could be safely turned off, which is inflexible and also requires carefully sizing of sleep transistors. Power management circuitry is necessary and should be always active. By contrast, fine-grained power gating provides high flexibility, allowing certain gates or circuits to be independently powered up or shut down, at the expense of significant sleep transistor area overhead.

Compared to sub-threshold leakage, gate-oxide leakage is less well understood [14]. Using high-k material [39, 40] in semiconductor manufacturing processes to replace the traditional extremely thin gate dielectric layer is the most effective approach so far. In 2007 and 2008, Intel and IBM started to deploy hafnium based high-k dielectrics for their products [41]. Continuing research in this area is likely to identify even better materials allowing thicker oxide layers while also reducing leakage current.

1.2.4 Summary

In my work, main focus is on lowering both dynamic and static power consumption by taking advantages of asynchronous logic and adiabatic logic while avoiding the overhead introduced by these two logic styles. Sub-threshold logic is also used in reversible circuit design.

1.3 Challenges for Modern Embedded System Design

With the increasing complexity at both IC design level and system level, many challenges arise. Power consumption has become a primary constraint in microprocessor and memory design in particular.

1.3.1 Optimization and estimation

The process of circuit design ranges from complex electronic systems all the way down to the individual transistors within an integrated circuit. For big systems, they are optimized for performance and/or cost rather than power while small circuits could be optimized for either speed or power. This is reflected in the presence of a myriad of CAD tools for speed/area optimization while there are very few options when it comes to systematic power estimation and optimization. With increasing complexity of systems comes the challenging and time consuming process of estimating the complete power consumption through extensive simulation. To achieve the goals of estimating accurate power dissipation of the circuits during system design, a number of power estimation techniques [11, 16, 42, 43] for worst-case and average-case have been introduced. Worst-case estimation targets on the performance under worst scenario such as peak current occurred in the circuit, i.e. clock tree in synchronous system, which is regarded as a safe and pessimistic analysis specific to real-time systems, while the average-case estimation intends to analyze the performance under general usage which is well-noted for its difficulty and time consumption due to large space of possible scenarios. To clarify, worst-case is more meaningful for safety and lifespan concerns. On the other hand, the general performance, such as battery life, speed, can be predicted based on average-case study.

1.3.2 Challenges in controlling static power

Although there are a few techniques and technologies that have already been used to suppress

the leakage power, such as MTCOMS, high-k materials, its practical application faces significant technological and cost challenges. In addition, as low threshold voltages are usually assigned to performance-critical circuits, it incurs high sub-threshold leakage for those parts.

Besides, as CMOS technology continues to scale down, process variations in gate length, oxide thickness and doping concentration becomes more significant. The impact of gate length variations on sub-threshold leakage is exponential. New approaches are desired to counter this issue.

The gate-oxide leakage raises new challenges for leakage power reduction [44] as it starts to overtake the sub-threshold leakage at some stage. For example, as the sub-threshold leakage decreases following the drop of operating temperature – in standby mode, the gate-oxide leakage could be more dominant, because it is less dependent on temperature than the sub-threshold leakage current.

1.3.3 Controlling memory power

On-chip memories constitute the major portion of the system area budget and account for a considerable share of total power, i.e. 35%, especially leakage. In some cases [14, 45, 46], leakage power dominates the entire cache power budget (about 70%). Therefore, ultra-low static power memory is desired. For a typical memory cell, sub-threshold leakage current occurs on both bit-lines and within the cell. Additionally, gate-oxide leakage current is flowing through the transistor gates. Some circuit, control and compiler techniques are implemented to address this problem. The overhead in terms of performance, die size and extra power needs to be carefully managed.

1.3.4 Challenges in processing units

Processing units, such as data-paths and Arithmetic Logic Units (ALUs), are very critical for microprocessors, which consume a large amount of power including the associated clock tree, i.e. 25% out of total. High switching activity and difficulty in controlling variations could lead to mismatching in delay, resulting in glitches hence additional power which could further increase the share of the total power consumption. Also, with the developing density and complexity along with the increase of clock rate, distributed global clock signal needs a

special attention, timing issues becoming more critical as less variation being tolerated if the circuit is to function properly. Apart from functionality, unwanted switches and leakage power are two big concerns as well, not only in processing units but also in the clock tree. To increase the quality of the clock signal, buffers are often inserted into the clock tree, dissipating significant power.

1.4 Research Work in this Thesis

There are two main aims of our study: efficient and accurate power estimation methodologies for the average case for some classes of embedded systems and power optimization of data-path and memory for embedded systems.

Firstly, a novel power estimation technique is proposed for a class of architectures, including block ciphers, reversible circuits and Modular Quantitative Analysis (MOQA) gates. It is a static approach, which requires significantly less timing and effort to predict the average dynamic power of the designs.

Secondly, two new memory cells are designed to achieve ultra-low dynamic and static power, thanks to the adiabatic logic and novel topology of the cells. As mentioned above, leakage power is a big concern in memory circuits due to its high density, especially in deep sub-micron processes. Two process nodes, namely 65nm and 45nm, are utilized to analyze the circuits. Process variations also are taken into account by carrying out the Monte Carlo simulation.

Finally, a new logic named Asynchronous Charge Sharing Logic (ACSL) is proposed, with high power efficiency compared to traditional asynchronous logic types and lower area. This new logic is then applied to a number of blocks which compose a typical data-path or can be found in the processor ALU. The new ACSL logic has easy, fast and accurate predictability for average power consumption. In the quest to design fully predictable systems, I explore some reversible principle by using a modified reversible adder to serve as a multi-function gate to build an ALU by taking advantage of bidirectional characteristic of reversible logic. Sub-threshold logic is applied to realize ultra-low dynamic power. Besides, a testing technique for reversible circuits is briefly introduced.

1.5 Thesis Organization

The rest of this thesis is organized as follows.

- Chapter 2 provides the background information about reversible logic, adiabatic logic and asynchronous logic, which are the cornerstones of my research.
- Chapter 3 includes several techniques and designs for power estimation.
- Chapter 4 presents two ultra-low power memory cell designs, able to operate in adiabatic mode.
- Chapter 5 explores a novel logic, ACSL, which could be implemented into data-processing units.
- Chapter 6 introduces power optimization of a reversible ALU and a testing approach for reversible circuits.
- Chapter 7 summarizes the contributions of the thesis, followed by a synopsis of potential future work.

2 ADIABATIC LOGIC AND ASYNCHRONOUS LOGIC

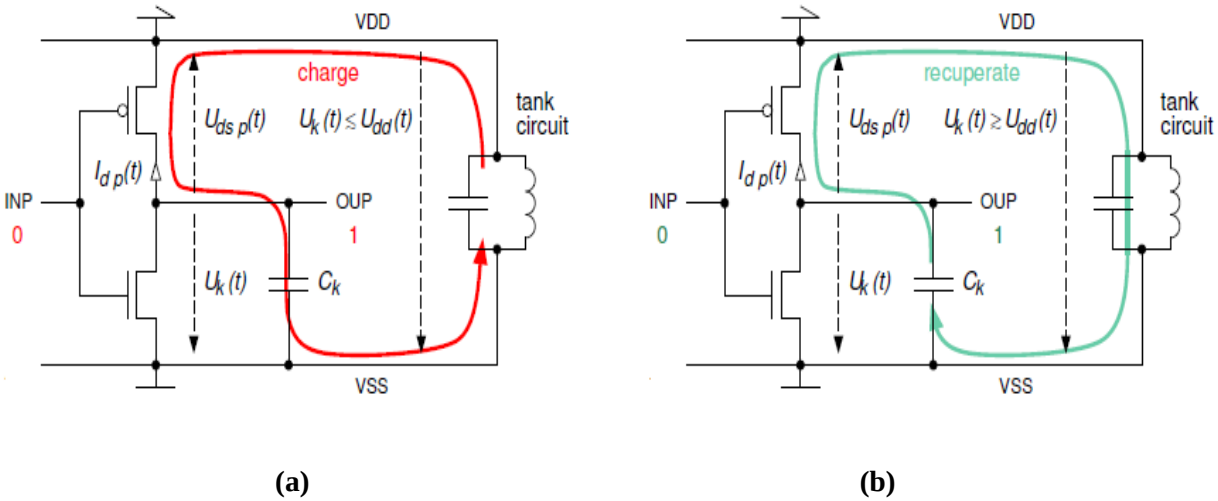
2.1 Adiabatic Logic

Some of the theoretical studies for adiabatic computers were reported in [47-49]. These devices were initially studied in the context of reversible computing and reversible logic, the main idea being that for such systems most of the energy during the computing is being recycled/re-used. Reversible computing is a model of computing where the computational process to some extent is reversible, physical reversibility and logical reversibility in particular [50]. A process is regarded as physically reversible if it causes no increase in physical entropy. These circuits are also referred to as charge recovery logic or adiabatic logic. While the future of reversible logic is still uncertain with quantum computing being mentioned as the main underlying technology [51, 52], as mentioned in Chapter 1, adiabatic logic has become a promising methodology and good alternative to achieve low power/energy property due to its special characteristics.

Dissimilar to conventional CMOS circuits where energy is dissipated during a switching event (refer to equation 1.2), Adiabatic circuits recycle the energy after evaluation through power clock which usually is LC resonant circuit [53] or switch-capacitor tank [54]. Only losses due to the resistance of the switches needed for the logic operation still occur [55], which results in dramatic energy saving. Several Adiabatic logic families [56-59] have been proposed, which target on efficiency and compactness. Moreover, Adiabatic logic may benefit in popularity from future devices thanks to being insusceptible to Hot Carrier Injection and showing less impact of Bias Temperature Instability than static CMOS circuits [60].

The characteristic of adiabatic logic encompasses two aspects which are energy recycling and slow and smooth and low current flow through transistors, realized through energy-conserving charge and discharge processes provided by the supply voltage source which could vary over time. The energy consumption in adiabatic logic is given by [58]:

$$E_{AL} = 2\frac{RC}{T}CV_{DD}^{2} \tag{2.1}$$



(a) **(b)**

Figure 2.1 Adiabatic Switching (a) Evaluation Phase (b) Energy Recuperation Phase [17]

where *R* is the resistance in the charging path of the circuit, consisting of the on-resistance of transistors in the charging path and the sheet resistance of the signal line. *C* is the capacitor and *T* is the transition time. From (2.1) we can observe that when *T* increases, the energy decreases (*T* is determined by Power Clock Generator (PCG)). Thereby, adiabatic logic is favored in low frequency applications. The simplified adiabatic switching including evaluation phase and energy recuperation phase through a suitably designed resonant tank circuit (one kind of PCGs) is shown in Fig.2.1 [17]. Because of the importance of PCG, major concerns over the feasibility and efficiency of power clock generator are inevitable [49]. A number of power clock architectures were reported in the literature [53, 61, 62].

Furthermore, there are two principles to meet in order to qualify for adiabatic logic [63]:

1) Never turn on a transistor when there is a voltage potential difference between the source and the drain.
2) Never turn off a transistor when current is flowing through it.

Adiabatic logic plays a great role in our work in terms of memory designs and arithmetic unit designs. The advantages, drawbacks and limitations of adiabatic logic are investigated in this Chapter.

2.1.1 Adiabatic logic family

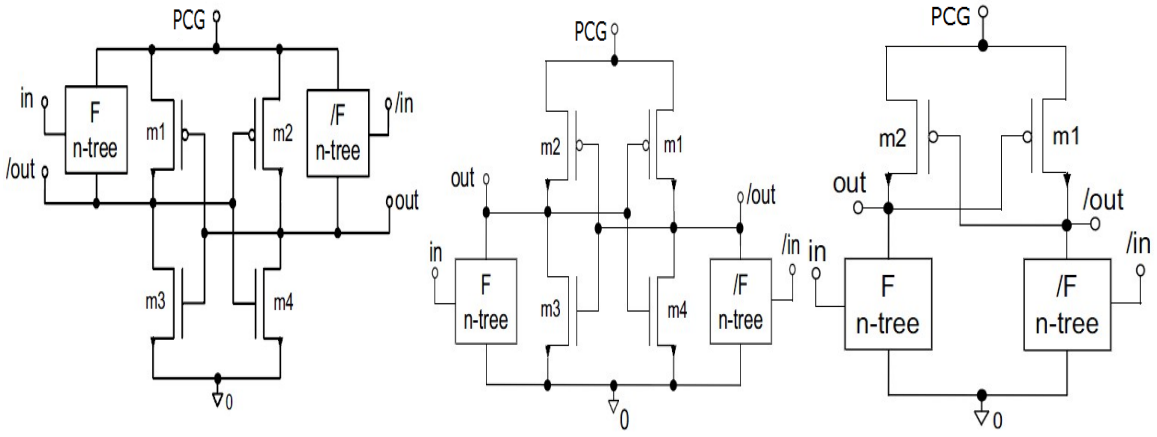
First, it should be noted that differential signals are applied in adiabatic logic. Unlike standard

CMOS circuits, adiabatic circuits do not contain V_{DD} (DC supply) which is replaced by AC supply. Three main styles in adiabatic logic family are depicted in Fig.2.2, which are Positive Feedback Adiabatic Logic (PFAL) [59], 2N-2N2P [64], Efficient Charge Recovery Logic (ECRL) [57]. Besides, other adiabatic logic styles are reported in [65, 66]. In this section, we only focus on the three typical logic families. All three structures are charged and discharged through *PCG*. When inputs are ready, *PCG* starts to evaluate the circuits by charging up to a certain value, usually V_{DD}. Meanwhile, the differential outputs are set at '1' or '0' depending on the function of the *n-tree*. After the outputs are read, *PCG* then recycles the energy stored in the circuits by discharging itself to zero. In PFAL, *n-tree* blocks in the circuit are in parallel with the transmission PMOS transistors *m_1* and *m_2* in Fig.2.2 (a), which results in smaller equivalent resistance and thus lower energy consumption. Moreover, 2N-2N2P structure was derived from ECRL in order to reduce the coupling effect. The major superiority of 2N-2N2P over ECRL is due to the existence of cross-coupled NMOS transistors *m_3* and *m_4* in Fig.2.2 (b) result in non-floating outputs during recovery phase. In [55], it also has been reported that PFAL has the lowest power dissipation and the best consistency of V_{DD} scaling in contrast to 2N-2N2P and ECRL. These structures could be implemented into logic/arithmetic units, such as inverter, NAND, NOR, ADDER and etc.

Fig2.3 shows the energy consumption per switching operation versus frequency for three inverters based on the mentioned logic families along with the conventional CMOS inverter where V_{DD} is 1.8V and capacitive load at each output node is 20fF [55]. It can be seen that PFAL always consumes the lowest energy dissipation of them all when the frequency is higher than 2kHz. Also, it is worth mentioning that the CMOS inverter nearly consumes the constant energy regardless of the frequency (>10 kHz) while the energy consumption of the other three inverters increases following the growth of the frequency. When the frequency is greater than 100MHz, adiabatic logic loses its advantage in power reduction. Nevertheless, it can be concluded that adiabatic logic excels CMOS logic in relatively low frequency region.

2.1.2 Power Clock Generator

As mentioned above, PCG is arguably one of the most important components in adiabatic circuits. It partially determines performance, power consumption and area of the circuits. To date, most PCGs are composed of pass transistors, inductors and capacitors which are known as LC oscillators. Sinusoidal waveforms are generated by these oscillators, where the



(a) **(b)** **(c)**

Figure 2.2 (a) General Schematic for PFAL [59] (b) General Schematic for 2N-2N2P [64] (c) General Schematic for ECRL [64]

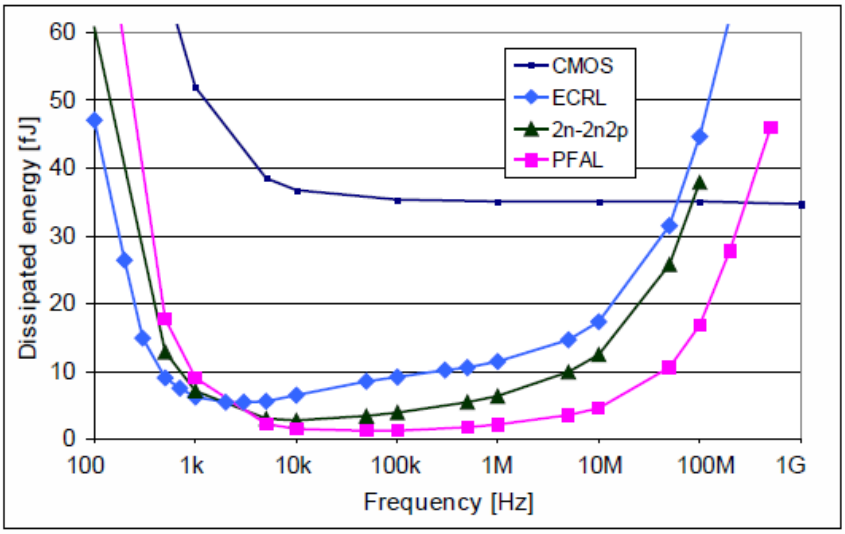


Figure 2.3 Energy Consumption per Switching versus Frequency for a CMOS Inverter, an ECRL inverter, a 2N-2N2P Inverter and a PFAL Inverter [55]

frequency is determined by the value of inductor, *L*, and capacitor, *C*, the equation is given by [62]:

$$f = \frac{1}{2\pi\sqrt{LC}} \tag{2.2}$$

Fig.2.4 exhibits four different architectures of PCGs [53] which could be divided into two groups, asynchronous style and synchronous style. For asynchronous one, see Fig.2.4 (a) and (b), it uses feedback loops to self-oscillate and thus create the needed waveforms. On the other hand, for synchronous PCGs, the external clock signals, *Clk1* and *Clk2*, are inserted to control the pass transistors, see Fig.2.4 (c) and (d). The frequency of these clock signals should be matched to the frequency of the oscillator, the waveforms of these signals are also illustrated in Fig.2.4 (c). Several issues are associated to the asynchronous design. The primary one is the instability of its oscillation frequency caused by capacitive load variation of the circuits. Also, it has been proved that this type of design is not able to generate 4, 8 and more phase shifted power clocks [53] which are needed in some Adiabatic circuits [61, 67]. However, more phase PCGs would not guarantee always high power efficiency [68]. At last, asynchronous PCGs may not suit large systems with high requirement of synchronization. Table 2.1 [53] lists the comparison of all four architectures in terms conversion efficiency and charge recovery rate. Various PCGs were laid out in a standard 0.18μm CMOS technology and simulated in a uniform test environment. The ratio of power dissipation of the whole system and the power consumed by load determines the conversion efficiency of the PCGs. Charge recovery is calculated by comparing the power dissipation of adiabatic circuits controlled by non-adiabatic power clock drivers and controlled by charge-recovery PCGs respectively.

Based on these issues, it is observed that synchronous style shows more power efficiency, especially, the 2N2P model. Despite this, there are still many obstacles to overcome when it comes to power clock generator designs, such as its high area consumption and usually requiring off-chip inductor, its feasibility in large systems and also in asynchronous design. Due to the nature of LC resonant oscillator, even though there is no activity needed for the connecting adiabatic circuits, it still consumes energy. Despite some research been done on this problem [69], there is still no stable solution yet. Last but not least, the energy efficiency of PCGs degrades exponentially along with the increase of the frequency, which to a great extent decides the practicability of adiabatic circuits.

Other than LC oscillator, capacitor-based clock generator was also used in several adiabatic designs. It realizes ramp-like charging in a stepwise fashion, which is not genuine but rather quasi-Adiabatic. The architecture and the ideal waveform of this power clock are shown in

Fig.2.5 [54].

Compared to the LC topologies, there are several attractions of capacitor array generator. First, it can be implemented using on-chip capacitors, and the off-chip inductor is eliminated. Secondly, it is allowed to be utilized in a modular fashion, which is beneficial to design reuse. Finally, it is static during "Idle" and "Hold" phases, see Fig.2.5 (b). However, the value of capacitors should be carefully selected and the clock signals, which control the capacitor tank, also require special attention.

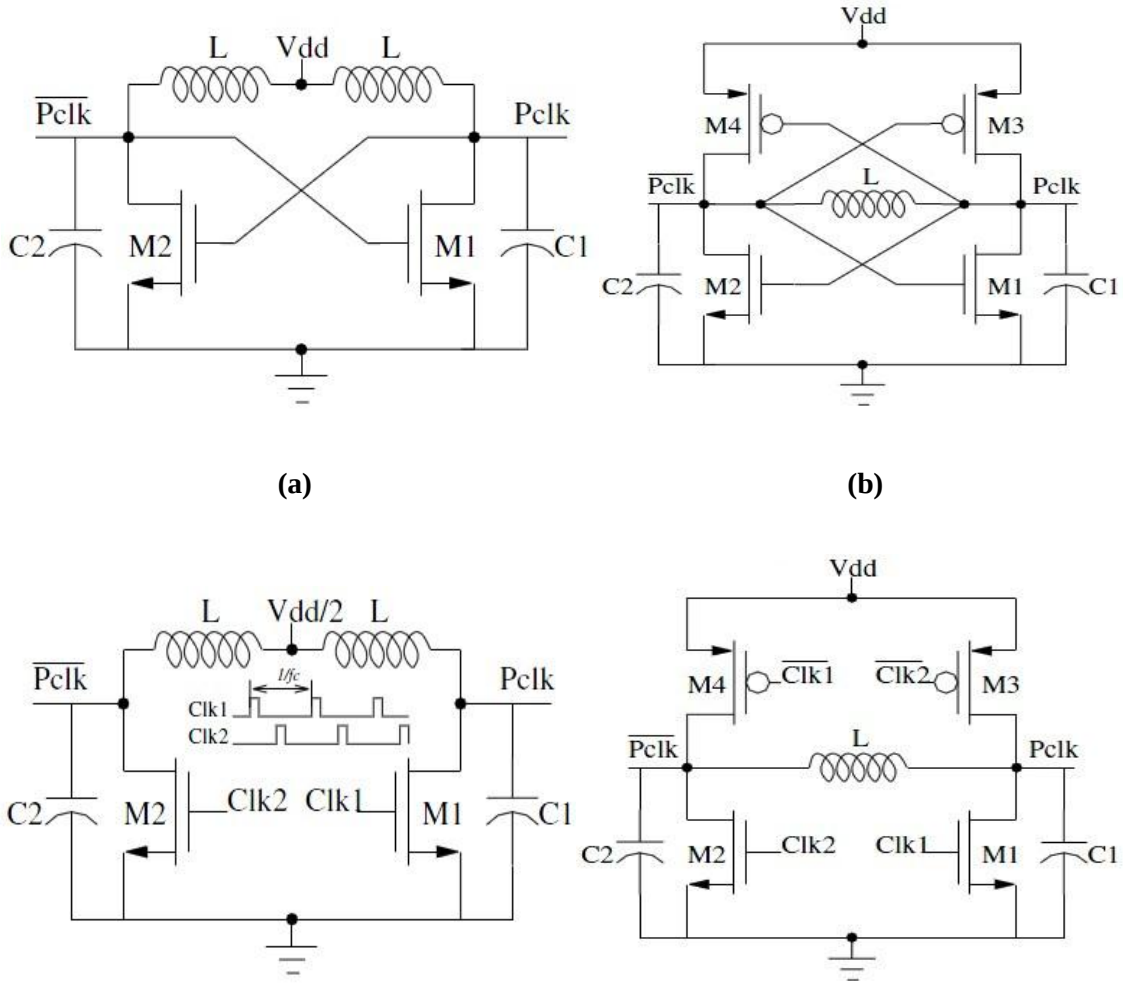
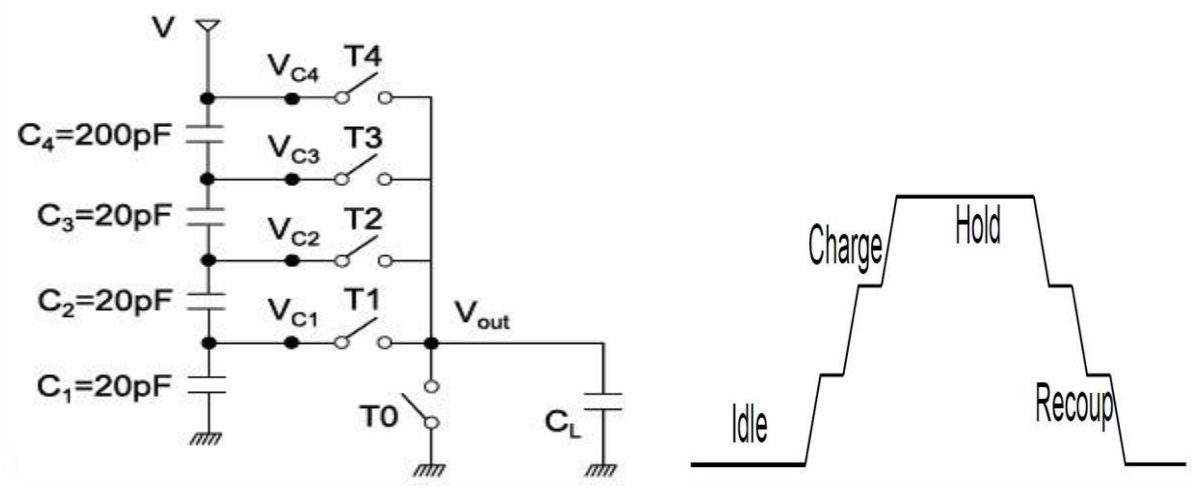


Figure 2.4 Asynchronous Resonant Adiabatic Power Clock Generators (a) 2N Asynchronous (b) 2N2P Asynchronous (c) 2N Synchronous (d) 2N2P Synchronous

Table 2.1 Comparison of Adiabatic PCGs @ 100MHz [53]

PCG	*PCG Conversion Efficiency (%)*	*Charge Recovery (%)*
2N Asynchronous	39.9	21.1
2N2P Asynchronous	43.2	44.4
2N Synchronous	60.1	27.3
2N2P Synchronous	62.0	60.7



(a) **(b)**

Figure 2.5 (a) Switched Capacitor Clock Generator (b) Ideal Waveform of Stepwise Charging [54]

2.1.3 Complex Adiabatic Circuits

Most of the work on adiabatic circuits focus on the design of new logic families as well as some arithmetic units. In this section, a comparison between adiabatic circuits and CMOS circuits is presented. The main focus is to show the energy efficiency of adiabatic logic and assess its performance limit.

A popular model, which is used to demonstrate the energy efficiency of adiabatic logic is the inverter chain. Almost every adiabatic logic style was implemented into this architecture,

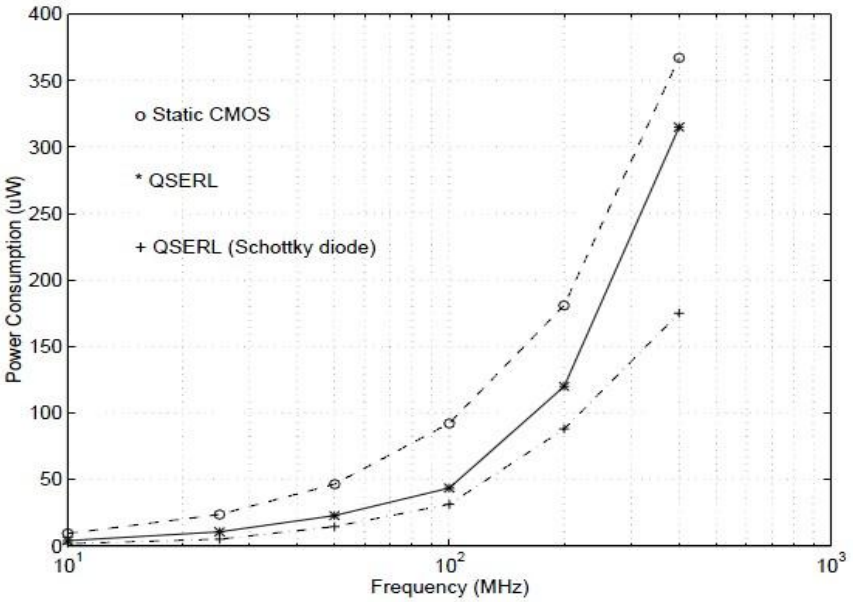


Figure 2.6 Simulation Results of QSERL Inverter Chain versus Static-CMOS Inverter Chain [70]

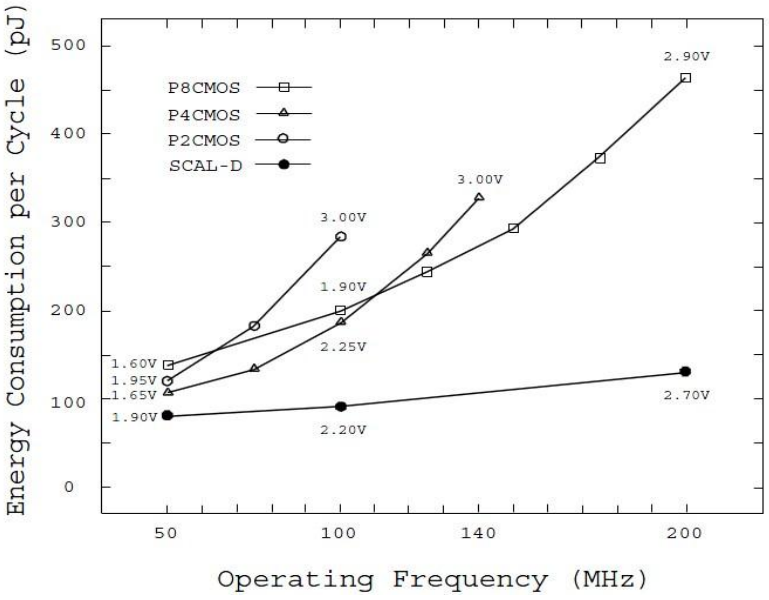


Figure 2.7 Energy Dissipation per Cycle vs. Frequency for Multipliers Including Self-Test Logic [71]

which would give the researchers a clear vision from a power point of view, with simple functionality. Fig.2.6 depicts an 8-inverter chain using QSERL (Quasi-Static Energy Recovery Logic) compared to static CMOS inverter chain [70] using MOSIS 0.5μm CMOS NWELL process. QSERL uses two complementary sinusoidal supply clocks and possesses

Table 2.2 Comparison of Adiabatic and Static CMOS Arithmetic Units at Different Running Frequency [72]

Arithmetic Units		*Running Frequency (MHz)*			
		10	*20*	*30*	*100*
4-bit CLA	*Adiabatic (pJ)*	*4.16*	*5.00*	*5.76*	*10.31*
	CMOS(pJ)	*71.56*	*71.56*	*71.56*	*71.56*
	Gain	17.20	14.31	12.42	6.94
8-bit CLA	*Adiabatic (pJ)*	*10.12*	*13.04*	*15.69*	*30.31*
	CMOS (pJ)	*210.80*	*210.80*	*210.80*	*210.80*
	Gain	20.84	16.16	13.43	6.95
16-bit CLA	*Adiabatic (pJ)*	*22.50*	*28.90*	*37.20*	*71.90*
	CMOS(pJ)	*503.20*	*503.20*	*503.20*	*503.20*
	Gain	22.63	17.41	13.53	7.00
4-bit Multiplier	*Adiabatic (pJ)*	*8.02*	*9.72*	*12.17*	*23.39*
	CMOS (pJ)	*76.28*	*76.28*	*76.28*	*76.28*
	Gain	9.51	7.85	6.27	3.26
8-bit Multiplier	*Adiabatic (pJ)*	*29.00*	*38.63*	*55.40*	*133.33*
	CMOS (pJ)	*762.00*	*762.00*	*762.00*	*762.00*
	Gain	26.28	19.73	13.75	5.72

several positive characteristics of static CMOS logic. It can be seen that QSERL inverter chain saves more than 50% of energy at 100MHz. While the frequency increases to 400MHz, the saving percentage is down 14% while the scalability is not demonstrated in [70]. However, this could be improved by using Schottky diodes instead.

In [71], a true single-phase 8-bit adiabatic multiplier was built based on SCAL-D (Source Coupled Adiabatic Logic) using MOSIS 0.5μm CMOS NWELL process compared to three pipelined static CMOS multiplier which are 2- stage, 4-stage and 8-stage, respectively. Simulation was performed using 64 randomly generated input vectors which independent with each other. The probability of being HIGH (and LOW) for each input in each clock cycle is 0.5, and the probability that an input switches in the following cycle is also 0.5. It achieves energy efficiency across a broad range of power clock frequencies by using a pair of cross-coupled transistors, a pair of diode-connected transistors, and an individually tunable current source at each gate. The simulation results are shown in Fig.2.7. Three operating frequency were chosen for the comparison, which are 50MHz, 100MHz and 200MHz. It is clear that SCAL-D is more energy efficient than the pipelined static CMOS designs across the entire chosen frequency range. It is also worth mentioning that energy consumption rises following the increase of operation frequency is possibly due to more glitches occurred because of high frequency.

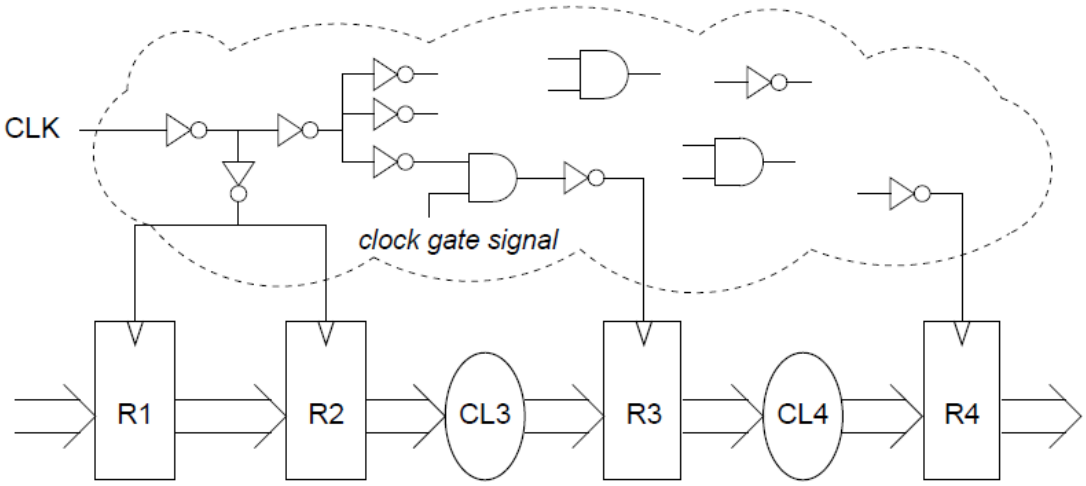
A more comprehensive comparison is given in [72]. One adiabatic logic called PFAL, mentioned in Section 2.1, and static CMOS were implemented into two main arithmetic models, which are Carry Look-Ahead adder (CLA) and multiplier. A module generator, a C++ program [72], is used to synthesize the architectures. Both PFAL and static CMOS circuits have been simulated after the parasitic parameters are extracted from the layout, with V_{DD}=5V. The input test pattern is a long sequence of random values that gives a figure of the average power consumption. The energy consumption per operation and the adiabatic gain of the PFAL circuits working at 10, 20, 30 and 100MHz are reported in Table 2.2 [72]. It is obvious that the adiabatic gain varies along with the operating frequency. The higher the frequency is, the lower the gain is. It indicates that adiabatic logic is not very efficient when high performance is required. Nevertheless, the energy saving is very impressive for relatively low frequency region.

2.2 Asynchronous Logic

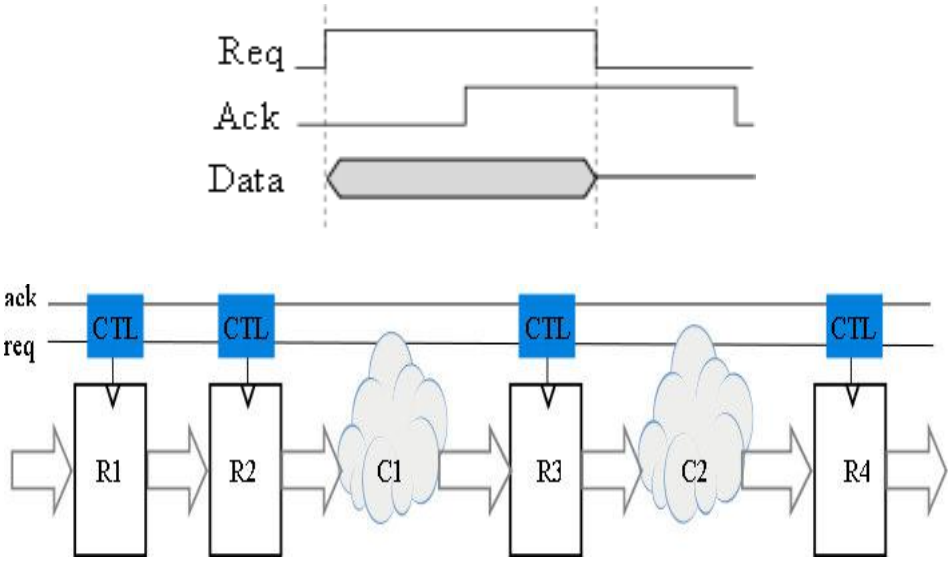
It is widely accepted that a single clock (global clock) scheme would not adjust to the nano-scaled very large scale integration (VLSI) systems and, thus asynchronous architectures (or hybrid) emerge as potential alternatives [28]. Due to the uncontrollable parameter variations across a chip, it would be unreasonable to match the delay of the clock and other signals during processing, although this usually works under worst-case design. While embracing the benefits in terms of potential low power consumption, high robustness to delay and mismatch, design reuse, electromagnetic compatibility and more tolerance to process variations and external voltage fluctuations compared to synchronous designs, asynchronous logic is re-enacted in today's deep sub-micron CMOS technology [73, 74]. Also, asynchronous logic is well suited for applications where messages are generated at irregular intervals, for example wireless sensor network. The general comparison and discussion between synchronous logic and asynchronous logic is carried out in the next section.

2.2.1 Synchronous Logic versus Asynchronous Logic

Synchronous logic, which is still the mainstream of most digital circuit design, is defined by a global clock signal distributed throughout the system. As shown in Fig.2.8 (a), each memory block, represented by *R* (Register), is controlled by the global clock signal, *CLK*, which is distributed through the clock tree, which means the whole system is under control of the global clock signal. However, when it comes to physical design, particularly, when delay is introduced, the system is forced to run at the worst-case speed in order to avoid malfunction. Other than this, the essential clock tree (buffers, AND gates, etc.) results in large overhead in area and power consumption [75]. The main advantage of synchronous circuits is low engineering effort in terms of design, cell libraries, layout, test and debug, which is well supported by most existing EDA tools. These EDA tools may also guarantee the functionality and stability of the chips by setting specific constraints, timing or power. Nevertheless, the continuously increasing speed and non-negligible wire delay across the chip make it more and more difficult to match the timing constraint, not to mention the serious process variation and other fluctuations. Moreover, from the power consumption point of view, the clock tree as a critical part of the synchronous circuits contributes about 40 – 50% of the total power [76]. Also, because of simultaneous switching of many registers (flip-flops, latches), peak



(a) Synchronous data-path with clock tree



(b) Asynchronous data-path

Figure 2.8 Synchronous Architecture versus Asynchronous Architecture [77]

current could be quite high, which would eventually increase the risk of system breakdown and other stability issues. Fig.2.8 (a) also presents the general structure of the clock tree [77].

Rather than using the global clock signal, asynchronous circuits use a protocol called handshaking [78] instead. The basic structure of an asynchronous circuit is exhibited in Fig.2.8 (b) where the asynchronous circuit follows the instructions from the pipeline controllers *CTL* which are communicated through handshaking signals represented by *ack*

and *req* signals. Unlike the conventional synchronous logic whose operation speed is determined by global worst-case latency, in asynchronous designs the speed depends on actual local latencies. In other words, an asynchronous circuit has the potential to run at the highest possible speed. Even if worst case delay is considered in synchronous circuits, glitches are difficult to be completely avoided due to variations in the delays of various paths. This issue would typically result in wasting a significant amount of power.

2.2.2 Fundamental Protocols of Asynchronous Logic

In synchronous circuits, race hazard is a typical flaw where inputs arrivals may vary. Therefore, timing is critical in synchronous circuits. In asynchronous systems, such hazards are eliminated through the use of a handshaking type protocol for synchronization. Under this circumstance, protocols based on data processing become intuitive. Different handshake protocols have their own advantages over area, speed, power consumption and robustness (at run time and process variations).

2.2.2.1 Bundled-data Protocol

Among several existing asynchronous communication protocols, the bundled-data protocol is one of the most popular, where separate request and acknowledge wires are bundled with the data signals [79], Fig.2.9. There are two main types of this protocol depending on the number of phases demanded, which are 4-phase protocol and 2-phase protocol. The 4-phase bundled-data protocol, which requires superfluous return-to-zero transitions, could lead to extra time and energy cost. On the other hand, 2-phase protocol, introduced under the name Micro-pipelines by Ivan Sutherland in his 1988 Turing Award lecture, regards both ‘0’ to ‘1’ and ‘1’ to ‘0’ transitions on request and acknowledge wires as equal signal event, which could ideally get rid of those unnecessary switching cost compared to 4-phase bundled-data protocol. Because the response of signal events in practical implementation is more complex, there is no solid answer to which of them is better.

It is worth mentioning that bundled-data in most cases could be referred to single-rail as only one rail of data is needed. Moreover, all the bundled-data protocols rely on delay matching, as they are not truly delay-insensitive. Therefore, extra care is required, such as using tile-based data-path structure, adding buffers, having a safety margin between data sender and

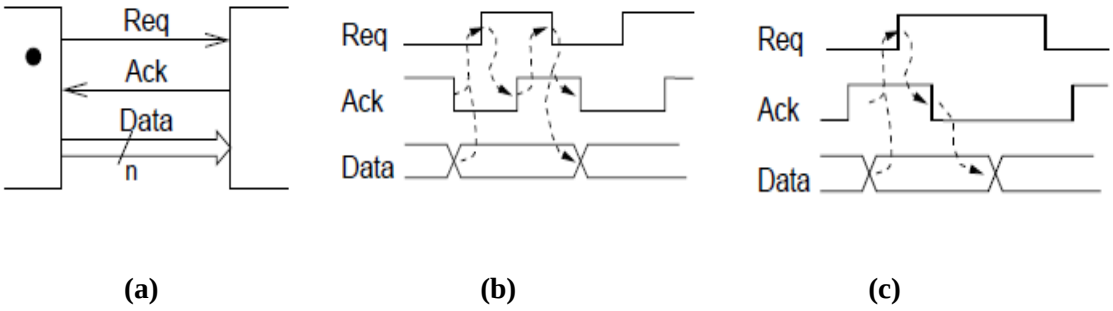


Figure 2.9 (a) Bundled Data Channel (b) 4-phase Protocol (c) 2-phase Protocol [79]

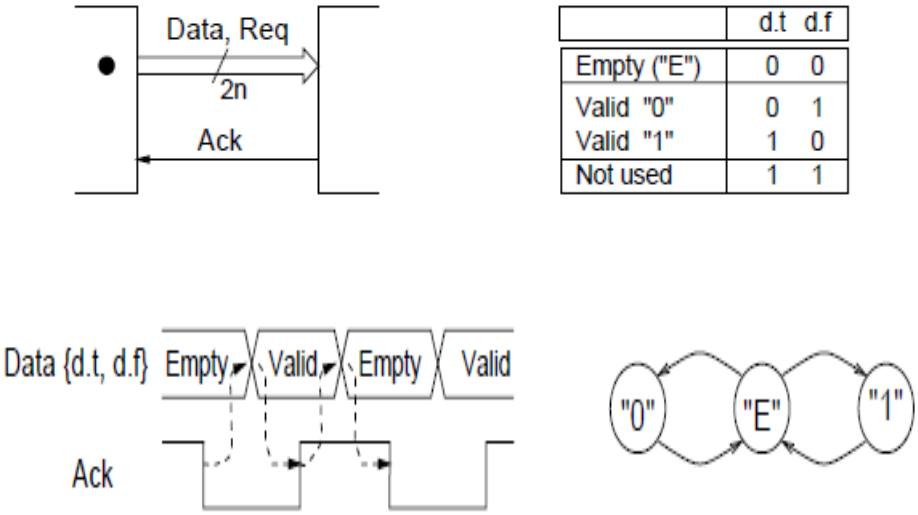


Figure 2.10 4-phase Dual-rail Protocol [77]

receiver. However, the increasing variability of progressive technology needs extended delay margin for safety as it could still further deteriorate the quasi-delay-insensitive property of this handshake protocol.

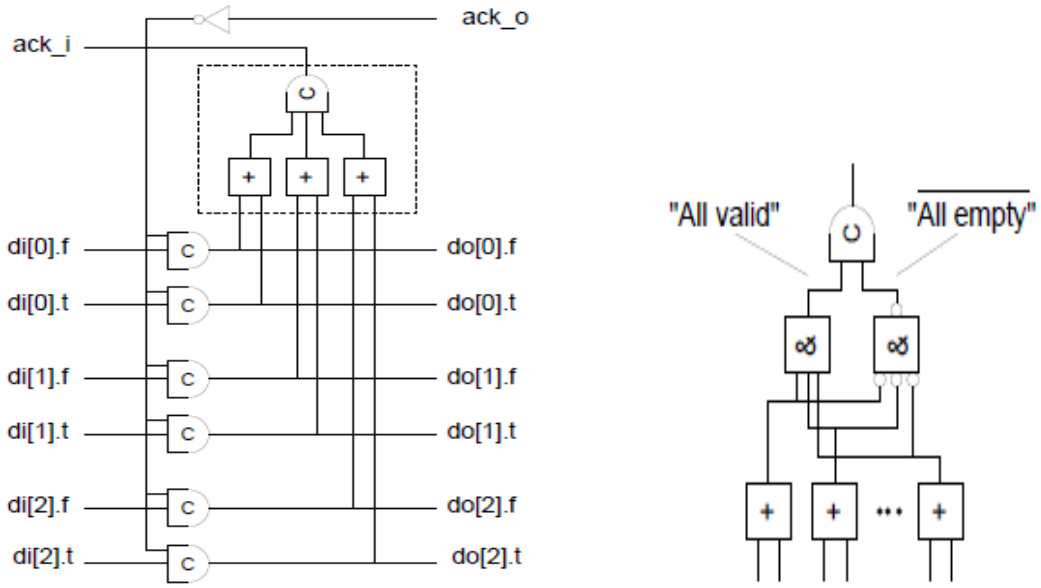
2.2.2.2 4-phase Dual-rail Protocol

Different from the bundled-data (single-rail) protocol, the 4-phase dual-rail protocol, the classic approach rooted in David Muller's pioneering work in the 1950s, [80], takes the advantage of two complementary data to represent more information other than data itself, such as the Request signal. Fig.2.10 depicts the general model of this type of protocol [77]. This feature ensures that the communication between two parties are always reliable regardless of delays on the connecting wires, and thus to enhance the robustness of the

system significantly. In some papers, this protocol is also called delay-insensitive or delay-independent [81].

This characteristic is more useful in bit-parallel channels, where there are N-bit data, represented by N-bit wires. Only after all data becomes valid, the receiver would be activated. It is well-suited in some array or tree based computations, such as the carry look-ahead adders, multipliers, etc.

Although this protocol is renowned for its high stability, it faces high area and power consumption issue due to the dual-rail structure. Especially, when it comes to the parallel channel case, the cost of the circuits for completion detection becomes very expensive. Fig.2.11 [77] shows an N-bit latch with its completion detection. It can be seen that it is composed of several *Muller C-elements* [82] and *OR* gates, alternatively. It still consists of couples of *OR* and *AND* gates plus one *C-element*. The circuit could get more complex when there are more channels in parallel [83].



(a) C-elements and OR gates **(b) OR, AND gates and C-element**

Figure 2.11 Two Completion Detection Structure [77]

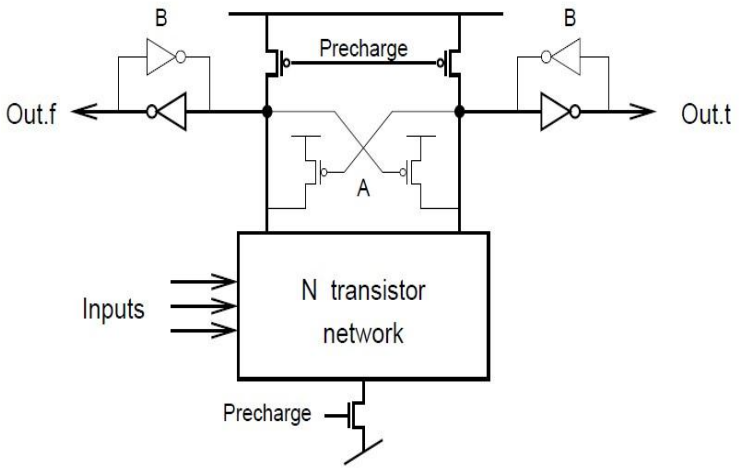


Figure 2.12 Generic DDCVSL Gate [84]

2.2.3 Function Blocks

Function blocks in asynchronous circuits are equivalent of combinational circuits in synchronous counterparts. There are several choices to implement these function blocks depending on the requirements of different communication protocols.

2.2.3.1 Bundled-data Function Blocks

As discussed above, single-rail logic is sufficient for the bundled-data protocol, so any traditional static or dynamic logic could be implemented in this case. For dynamic logic, the request signal could also be regarded as the pre-charge signal. Besides, the delay elements should be inserted to match the worst-case latency of the critical path in the circuit.

2.2.3.2 Dual-rail Function Blocks

In order to fulfill the delay-insensitive property, dual-rail logic is favored for its straightforward completion detection, essentially produced by pre-charged differential logic. Domino Differential Cascade Voltage Switch Logic (DDCVSL) [84], which is a modification of conventional dual-rail Domino logic, is very popular in several asynchronous designs [85-87] due to its lower power consumption and higher speed than convention dual-rail Domino logic. Fig.2.12 describes the topologies of DDCVSL [84]. It can be seen that the pre-charging operation is conducted by *Precharge* signal which could be connected to the *Request* signal when it is applied into asynchronous logic. When the evaluation is finished, two

complementary outputs would be detected by the Completion Detector and thus the Acknowledge signal is generated to the previous stage and also the Request signal to the next stage. Once the data is received by the latch of the next stage, the DDCVSL circuit would then be pre-charged. Meanwhile, both outputs are equal, which triggers the Acknowledge signal to become low again to indicate that the circuit is ready for the next computation.

2.3 Conclusion

Adiabatic logic is now well-recognized as a promising low power design candidate, especially in low throughput cases such as embedded systems even some quasi-adiabatic designs could also get benefits from it. Significant, energy could be saved by following its principles compared to the static CMOS counterparts. Yet, power-clock generators, which play a significant role in adiabatic circuits, still require careful design for both timing control and energy-recycling efficiency issues.

Meanwhile, asynchronous logic, with so many features beneficial to deep sub-micron technology, shows a bright future in today's CMOS IC designs. The practicality of combining these two techniques together with the synchronous is worth considering in order to further-decrease the power consumption. In [88], a method to realize this association is reported, but there are still some serious obstacles, in particular related to modifying PCGs to meet the new architecture constraints in particular.

In Chapter 4 and Chapter 5, two new memory cell designs and a novel logic style ACSL are proposed respectively. Those memory cells are able to operate in both adiabatic and non-adiabatic mode depending on the performance requirement. ACSL implements an adiabatic logic family, PFAL, into asynchronous circuits eliminating PCG but maintaining energy efficiency.

3 AVERAGE-CASE POWER ESTIMATION

3.1 Motivation

With the shrinking of the technology nodes, the amount of logic, memory and interconnect integrated per mm square of silicon is increasing allowing more functionality on the same chip. With the increased functionality, one of the major emerging constraints is the power consumption. Power consumption estimation and optimization plays a major role in the system design and test flows [89, 90]. Before we explore power reduction methodologies, we will focus on power estimation which will enable the designer to take some early decisions on the power budget. An efficient power budget will save design time (and therefore the time to market) as well as cost. In our work we look at estimating efficiently the average power consumption of a digital circuit where high level estimation and gate level estimation are two methodologies. This can be later used to also estimate the energy. But high-level estimation which represents the circuit by the Boolean equations is usually not accurate while gate-level estimation is time consuming. In contrast, worst-case estimation is given by the highest instantaneous power consumed by the circuit.

A typical power estimation methodology uses the input data sequence and input switching probabilities [11, 91]. With the increasing complexity of the system comes the challenging and time consuming process of estimating the complete system power consumption through extensive simulation. The estimation process becomes quickly infeasible as the simulation effort grows exponentially with the size of the input. The power consumption has an input pattern dependence [92] which leads to a number of problems such as selecting the representative input data set and switching probabilities (data profiling). Often, the power consumption of a functional block needs to be estimated when the rest of the chip has not yet been designed. Under this circumstance, little information may be known about the inputs to this already-designed block.

A number of power estimation methods and techniques have been introduced, using probabilistic estimation, such as CREST [93], DENSIM [94], BDD [95], and statistical estimation, including McPower [96], MED [97]. For probabilistic estimation, it uses probabilities to describe the set of all possible logic signals, in other words, it assumes a

typical behavior of the circuit inputs in terms of probabilities. The issues raised by this methodology are what probabilities are exactly required, how they can be obtained, and most

significantly, what kind of analysis should be performed. On the other hand, statistical estimation is more straightforward, which uses traditional simulation models to invoke the circuits and monitor the power. Again, if the system is very large, the simulation effort and sample size is a growing problem.

The development of a new power estimation technique is attractive if it could provide fast (ideally a formula) and accurate power estimation for each of the constituent blocks of a circuit given the input data profile. Such a method (if it existed), could help designers to have a clear view of the power consumed by different blocks and thus implement specific power optimization techniques to corresponding blocks. This is a very hard problem for both power and timing estimation [98] and in order to tackle it only some classes of circuits and only some input data profiles were restrained. Two characteristics called random bag preservation and linear compositionality [98, 99], involved in the proposed methodology, increase not only the feasibility of power estimation of large systems but also the flexibility for future redesign of systems. The simulation time of the system can be also dramatically reduced while maintaining high accuracy. I demonstrate the applicability of the proposed method with experiments carried out on two designs. The first is a modular adder design and the second is a Data Encryption Standard (DES) encryption block. Both of these designs are modular systems. I will expand then the applicability of the proposed methodology for other circuits and systems.

3.2 Average-Case Power Estimation

Average-case analysis in power estimation is more beneficial than worst-case and best-case study as it is directly related to the energy. Often it is also because both worst-case and best-case scenarios do not occur frequently in the system. An average case estimation is also useful for so called soft constrained embedded systems which are not safety critical. Usually to do the average-case analysis, for example, the total power of the system *S* with respect to the inputs *I* is given by [100]:

$$P_S^T(I) = \sum_{i \in I} P_S(I) \tag{3.1}$$

The average power of *S* with respect to all possible combinations of *I* is defined as [100]:

$$\overline{P}_S(I) = \frac{P_S^T(I)}{|I|} = \frac{\sum_{i \in I} P_S(i)}{|I|} \tag{3.2}$$

If the system has a large number of inputs, it would be very costly and time consuming to record and process during the simulation all possible combinations and sequences of the data inputs. If the system is built on the basis of the property of randomness preservation (uniform data distribution or uniform switching probabilities for the inputs) and satisfies the principle of linear compositionality (linear cascaded system), then the average power estimation of the total system can be reduced to the power estimation of its individual blocks and thus improve the efficiency of the average-case analysis.

3.2.1 Random Bag Preservation

Firstly, the notion of "random bag preserving" gate of block is introduced [98]. The standard notion of a *bag* is to track the number of bits are that repeated in the inputs or outputs of a gate via multiplicities. For example, the bag {1,0,1,1,0} is identical to the bag {1,1,1,0,0} but is not the same as the bag {1,1,0,0,0}. In the bag {1,0,1,1,0}, the multiplicity of the element 1 is 3 and the multiplicity of the element 0 is 2 while in the bag {1,1,0,0,0}, the corresponding numbers are 2 and 3 respectively. As for a *bag*, it can be defined as a *random bag* if the elements of the bag happen to be random structures of binary numbers.

Therefore, it can be understood that a gate is random bag preserving if it is able to transform a random structure into a uniform random bag. In other words, a gate is random bag preserving if the data distribution at its input is preserved at the output of the circuit. For example, note the truth table 3.1 of the *XOR* gate; it is random bag preserving because the number of 1s is equal to the number of 0s at both input and output. The probability of each element to occur is exactly same. In contrast, consider for instance the *AND* gate also displayed in Table 3.1; the inputs are represented by the random structure while the output bag has no random bag preserving property because the multiplicities of 0 and 1 are not equal.

Table3.1 Truth Table for *XOR* Gate and *AND* Gate

XOR	
Input	***Output***
00	***0***
01	***1***
10	***1***
11	***0***

AND	
Input	***Output***
00	***0***
01	***0***
10	***0***
11	***1***

It can be understood that not every gate maintain this property which limits its applications. However, reversible logic [47] is able to translate every logic gate into a randomness preserving gate. It also can be applied the other property namely linear compositionality introduced below.

3.2.2 Linear Compositionality

First, we discuss the case of the linear composition of arbitrary logic gates G_1 and G_2. All outputs from G_1 are used as inputs by G_2. Say G_1 operates on an input bag I, the output produced by G_1 on the bag I is denoted by $O_{G1}(I)$. The average power measure for the linear composition G_1 ; G_2 of any two logic gates G_1, G2's modular is given by [101]:

$$\overline{P}_{G_1;G_2}(I) = \frac{\sum_{i\in I} P_{G1;G2}(i)}{|I|}$$

$$= \frac{\sum_{i\in I} P_{G1}(i) + \sum_{j\in O_{G1}(I)} P_{G2}(j)}{|I|}$$

$$= \overline{P}_{G_1}(I) + \overline{P}_{G_2}(O_{G_1}(I)) \tag{3.3}$$

Consider if gate G_1 maintains the property of random bag preserving, the output bag of G_1 is still a random bag and fed into gate G_2. Intuitively, for a system built from full serial composed random bag preserving elements, the randomness is thus preserved all the way through the path. The overall average power of the system can be defined in terms of the individual average power of its components as shown in Fig.3.1. Given the *A*, *B*, *C* and *D*

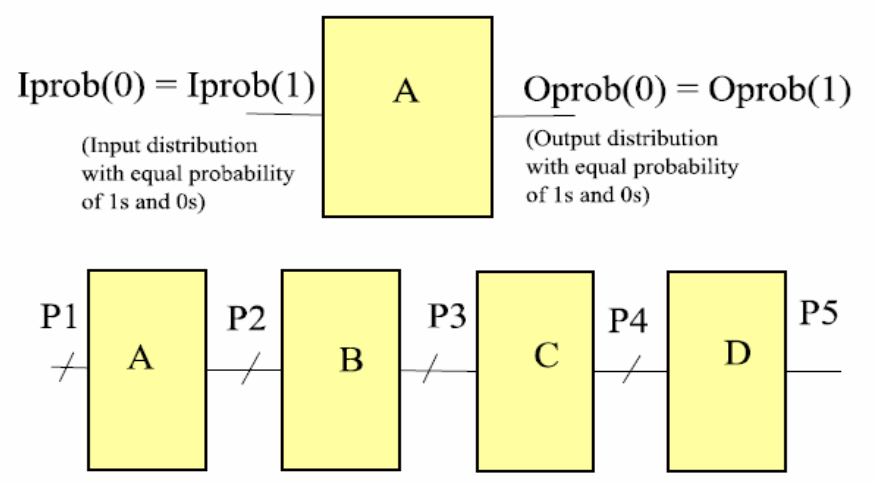


Figure 3.1 A Linear Composed System Built from Randomness Preserving Components

are randomness preserving blocks, the average power of the whole system will be composed linearly of the average power of the individual components. With these conditions, the central meaning of the new average-case power estimation technique based on the two mentioned properties is that the whole average power of the complete (or a part of the) system with random input bag, indicated by *R*, must be equal to the additions of the power consumption of all (or some) of the components in the specific systems. The new equation is given by [101]:

$$\overline{P}_{A;B;C;D}(R) = \overline{P}_A(R_A) + \overline{P}_B(R_B) + \overline{P}_C(R_C) + \overline{P}_D(R_D) \quad (3.4)$$

3.2.3 Modular Quantitative Average-case Power Analysis

Based on the concepts of randomness preservation, a new notion on random bag preserving gate to support Modular Quantitative Average-case (MOQA) [102] power analysis has been introduced. These gates can be referred as MOQA logic units. The linear compositionality of power measure was derived for MOQA circuits to aid the modular power estimation. This modular behavior has the potential to significantly reduce power simulation costs, replacing a potentially exponential number of tests with a static power derivation [101, 103, 104]. Some of the simplest MOQA units are the inverter and the XOR/XNOR gates used in classical logic design.

3.3 Experiment Demonstration

In this section, the capacity of the MOQA modular power derivation by experimenting with two digital designs namely Adder and Data Encryption Standard, a block cypher. An algorithm called Loop Input Set [103] is used in a test bench to generate all possible combination of the inputs for different function blocks and thus to get the average power of each, the pseudo code for this algorithm is shown below. Therefore, to get the results of the whole average power consumption of components or systems, the $2^n \cdot 2^n$ patterns for n bits inputs should be considered. For example, a gate with 3-bit input, this algorithm will generate the input vectors starting from 000, 000, 001, 010…111 and then change to 001, 000, 001, 010….111; after all, the last set will be 111, 000, 001, 010…111. Compared to the complete IO-set which requires $2^n \cdot 2^n!$ combinations, this new algorithm reduces the number of patterns by $(2^n \cdot 2^n!)–(2^n \cdot 2^n)$ for an n bit design. It not only saves on simulation time but also on demanding memory size. All the experiments in this section use this algorithm to provide input vectors.

Loop Input Set Algorithm

```
Loop Input Set(Struct input [1:M], struct
output[1:M])
Process{  Struct temp[1:M] := "intial value =0";
          Struct variable[1:M] = "intial value =0";
          Struct I = "initial value = 0";
          for each i ∈ 2^M loop
                output = temp;
                 wait for {activity time};
                output = variable +i;
                wait for {activity time};
          end loop;
          temp = temp +1;
          wait for delay time;}
 end process;
```

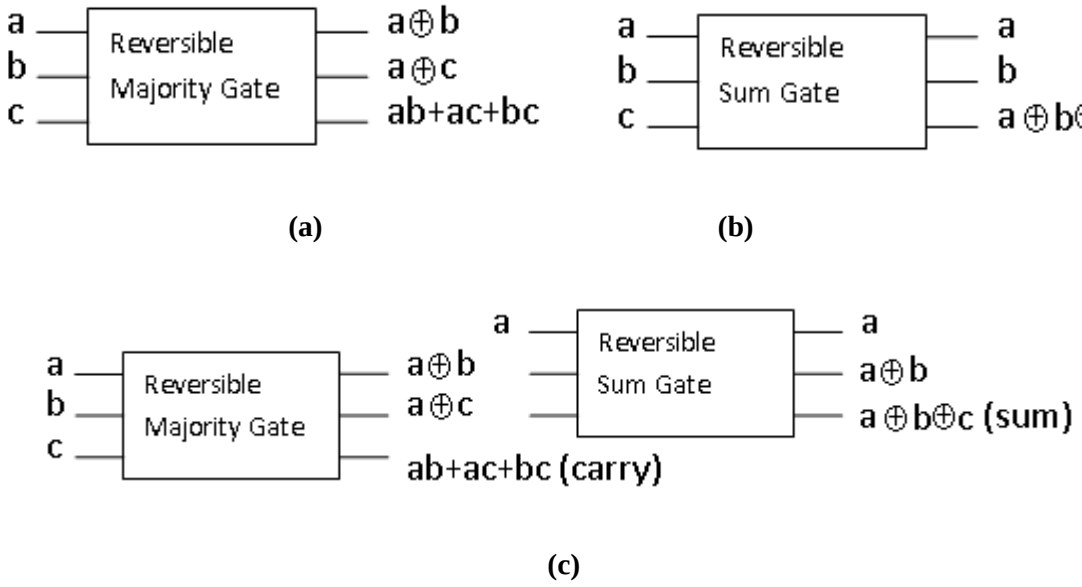


Figure 3.2 Full Adder using Reversible Majority Gate and Sum Gate

3.3.1.1 Ripple Carry Adder

In this adder example, a 4 bit ripple carry adder circuit is considered, depicted in Fig.3.3, built using modular single bit full adders in Fig.3.2. Each single bit full adder is built using two gates namely Reversible Majority Gate [105] and Reversible Sum Gate. These two gates are random bag preserving in nature, as their input to output mapping is bi-jective, truth table listed in Table 3.2. It is worth mentioning that all the reversible gates are included in the class of MOQA logic units. To illustrate the linear compositional nature of power across this 4 bit full adder design, actual power consumption of the design was calculated by implementing design at gate level. The implementation used the Synopsys CAD tool design flow and Primetime based on 65nm CMOS technology typical model. The full adder power was measured by exercising all the possible input combinations of the design. The experimental results for the design are shown in Table 3.3 [101]. The first column gives the names of the design. The following four columns, give the switching, internal logic, leakage and total power of the designs respectively. The final column gives the percentage of power contributed by each block. The first row gives 100 % as it is the complete 4 bit full adder design's total power. From the table it is evident that the first three full adders consume exactly 25.7 % when the last one consumes 22.9% which is down to the load capacitance at carry output is smaller than that of other three adders (no adder uses this carry out). Also the sum of the average power of each of these full adders gives the power equal to the total

power of the 4 bit full adder. This is given in the first row of Table 3.3 as 100 % confirming the linear compositionality of the power measure.

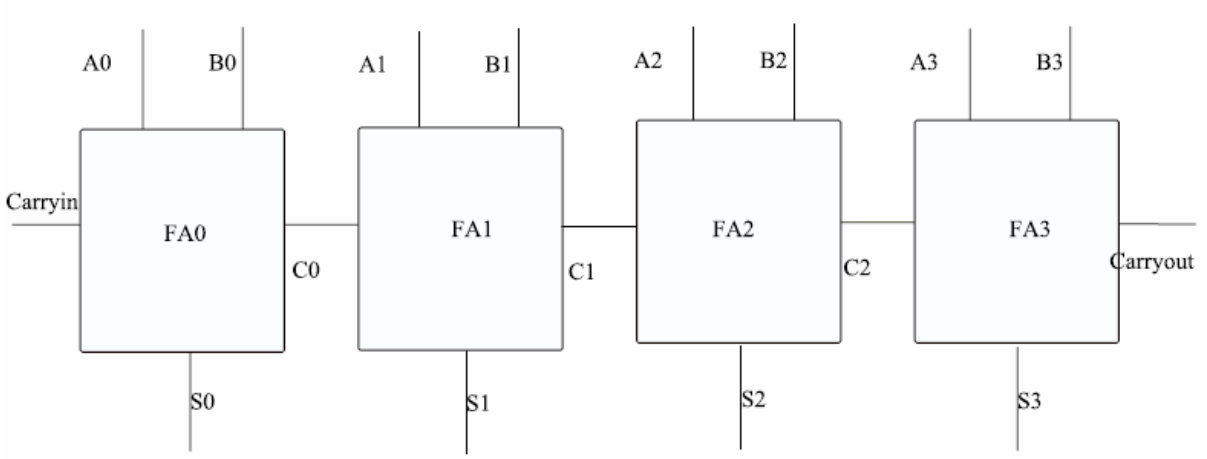


Figure 3.3 Modular Four Bit Ripple Carry Adder

Table 3.2 Truth Table for Majority Gate and Sum Gate

Majority Gate	
Input	***Output***
000	***000***
001	***010***
010	***100***
011	***111***
100	***110***
101	***101***
110	***011***
111	***001***

Sum Gate	
Input	***Output***
000	***000***
001	***001***
010	***011***
011	***010***
100	***101***
101	***100***
110	***110***
111	***111***

Table 3.3 **4-bit Ripple Carry Adder Power Consumption**

Hierarchy	***Switching Power (µW)***	***Internal Power (µW)***	***Leakage Power (nW)***	***Total Power (µW)***	***%***
4-bit Adder	4.4	10.5	318	15.2	100
FA3	0.79	2.63	79	3.50	22.9
FA2	1.21	2.62	79	3.91	25.7
FA1	1.21	2.62	79	3.91	25.7
FA0	1.21	2.64	79	3.92	25.7

A similar analysis can be also performed on a classical adder. Here one can consider that the one bit adder is built of a parity checker (for generating the sum) and a majority voter circuit (for generating the carry). Both the parity checker and the majority voter are MOQA units as described above.

3.3.1.2 Data Encryption Standard

The DES architecture is composed of modular components of 16 rounds as shown in Fig.3.4. We can extend the above technique in estimating the average power for the DES statically [103]. DES inputs are plaintext blocks of 64 bit with a secret key of effective size K=56 bits and outputs are cipher-text blocks of size 64 bit. Initially, the input is subject to an initial permutation after which the 64 bit input is dived into a 32 bit *L0* set and a 32 bit *R0* set. Both of those two 32 bit sets undergoes 16 rounds/iterations involving the function *f* and keys K1 to K16. In the function f, the 32 bit input data is combined with a 48 bit permutation of the key at the corresponding round/iteration. All the operations involved in a DES round are randomness preserving, making the whole round itself randomness preserving. This property allows that for the dynamic power estimation of one round, we need the dynamic power of the constituent blocks which is easier to simulate.

In Fig.3.4, 16 rounds of the DES is shown on the left as 16 boxes. The circuit/design for each round is expanded and shown on the right. The function block F (marked by an arrow) is

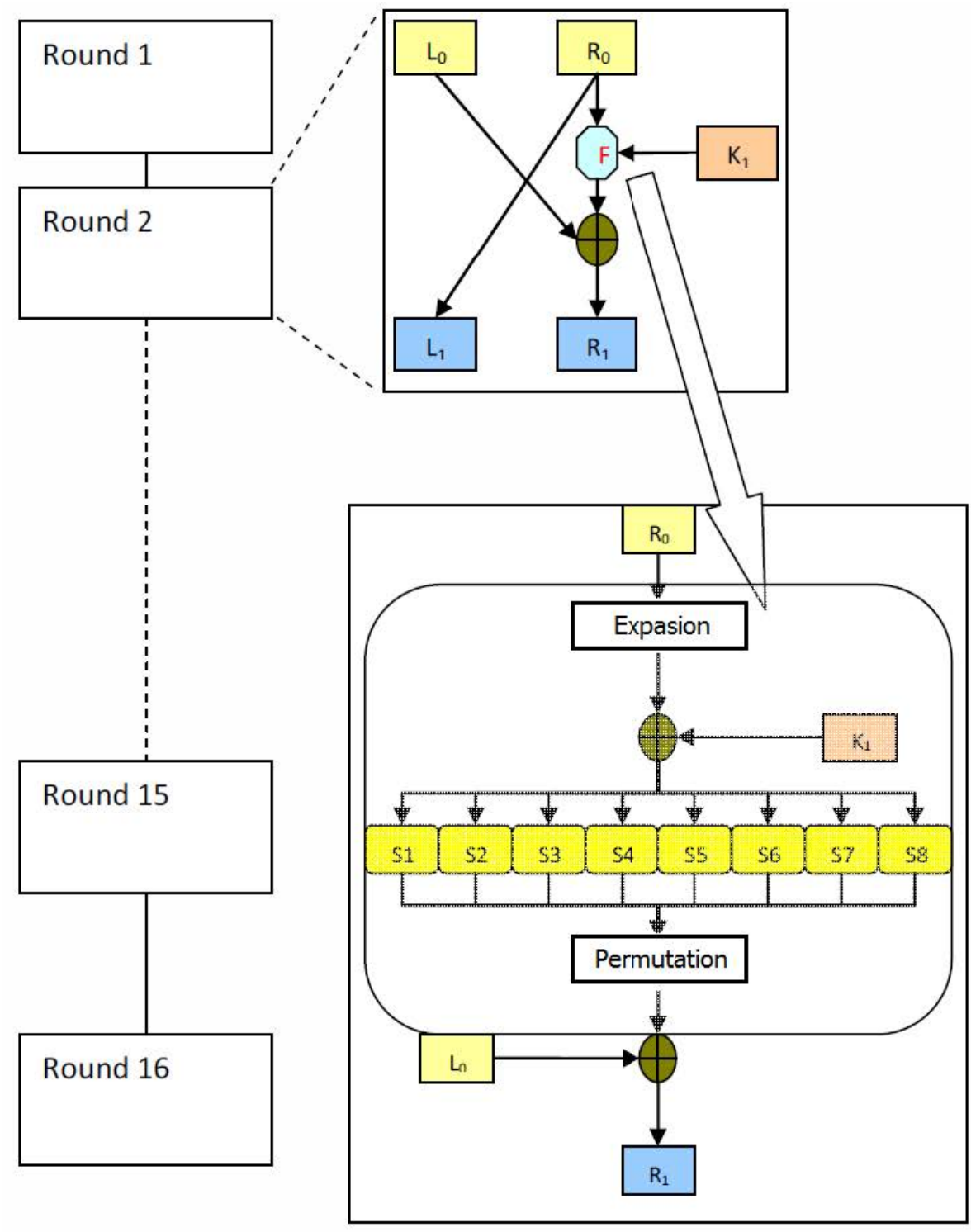


Figure 3.4 Modularity in DES

completely modular in nature. Since each component of this block is composed of XOR gates as their atomic gates. XOR gates are random bag preserving gates as mentioned earlier.

Table 3.4 Power Summary of Sum of Each Components, Single Round and DES

Power (mW)	*Input*	*Logic*	*Clock*	*Output*
Expansion	1.83	0	0	0
48-bit XOR	0	0.13	0	0
8 S-boxes	0	0.87	0	0
32-bit XOR	0	0.32	0	0
Permutation	0	0	0	0
2 Registers	0	0.135	1.21	94.98
Sum of all	1.83	1.455	1.21	94.98
Single Round	1.83	1.438	1.22	94.92
DES (16 round)	1.83	25.045	5.106	94.68

Hence, using the mentioned algorithm, the linear combination of the power measure of each of these blocks measured separately gives the power measure approximately equal to the total power consumption of the single round. Xilinx tool has been used to obtain the power data for DES. This is evident from the experimental result shown in Table 3.4. The first 6 rows give the power measure for the individual components. The seventh row gives the power measure obtained by summing the powers in first 6 rows. The eighth row gives the power consumption of the single round measured as a while. It can be concluded that the total power of the single round block is equal to the sum of the individual power measures of each components of the single round in terms of input power, logic power, clock power and output power within 2% error. The same experimental procedure can be demonstrated for the whole DES design comprising of sixteen rounds. The ninth row presents the corresponding power consumption of the entire DES. I also make the observation that the LFSR is another type of MOQA unit and hence the average power estimation method can be applied to Built-In-Self-Test of randomness preserving circuits as will be presented in Chapter 6.

3.4 Conclusion

As an important factor in power budget allocation, accurate power estimation is crucial but difficult due to the increasing complexity of systems. Usually extensive simulation is required to acquire the average power of systems, and it takes a lot of time and storage

memory to process the data.

In this chapter, a new method was introduced to estimate the average power consumption. It is simple(static) and time-saving if the systems meet several conditions, like random bag preserving and linear compositionality. A new algorithm called Loop Input Set has been proposed to generate test vectors for the circuits, which is very efficient. In contrast to the conventional complete IO-set, as many as $(2^n \cdot 2^n!)–(2^n \cdot 2^n)$ patterns can be saved without losing accuracy. Two circuits have been used to demonstrate the proposed method, which are 4-bit reversible ripple carry adder and DES. The structure and the elements of them match two mentioned conditions. The simulation results show great accuracy of our theory and methodology. While the presented method is very accurate and efficient, it still does not cover all possible circuit topologies and/or input data profiles, for example, this methodology cannot be applied to the control path of the circuits. The generalization of this methodology to encompass more generic circuits is very difficult to achieve. In Chapter 5 I will introduce some new logic which could help to design power estimation algorithms which are independent of the input data profile. However, in the next chapters I will focus on the optimization of the power consumption of two major components of an embedded system, namely the memory block and the processing unit.

4 ULTRA LOW POWER MEMORY CELL DESIGN

4.1 Motivation

Research in low power, robust memory cell design [106-108] has attracted considerable attention during recent years due to application's requirements, the technology scaling to deep submicron feature size and the increasing density of the transistors in integrated circuits (ICs). Portable devices with limited battery-life require low standby power processors and memory. Often, embedded static random access memory (SRAM) arrays [14] [109] can be the dominant part of the whole static power consumption and also occupied chip area, thus minimization of memory power is a crucial area of concern for today's IC designers.

As introduced in Chapter 1, two main contributors to power dissipation are dynamic power and static power. Many designs and techniques were presented during the last decade, targeting particularly the dynamic power. Also, several emerging logic style have been implemented into SRAM cell design, known as adiabatic logic [58] and sub-threshold logic [21] in order to minimize particularly the dynamic power. For the sub-threshold SRAMs the power optimization comes at an expense of timing degradation as the memories could hardly run in the MHz domain which could affect the total performance of the system. Therefore, adiabatic logic is favored in my design, whose characteristics have been discussed in Chapter 2. On the other hand, apart from dynamic power optimization, more emphasis of this design was on the static power optimization. For many embedded systems applications the long standby time could still lead to high leakage power dissipation; hence, static power optimization is of great importance.

The total power consumption distribution in a 32-bit microcontroller unit (MCU), a cache, SRAM and SRAM respectively are shown in Fig.4.1 [110]. According to the break-down, it is obvious that cache takes the biggest portion of the power consumption in the MCU by 35% while CPU and clock take 10% and 15% respectively. Hence SRAM dominates the power consumption where the cell array contributes 85% of the SRAM power. Therefore, reducing the power of the cell array could significantly improve the power efficiency of a processor.

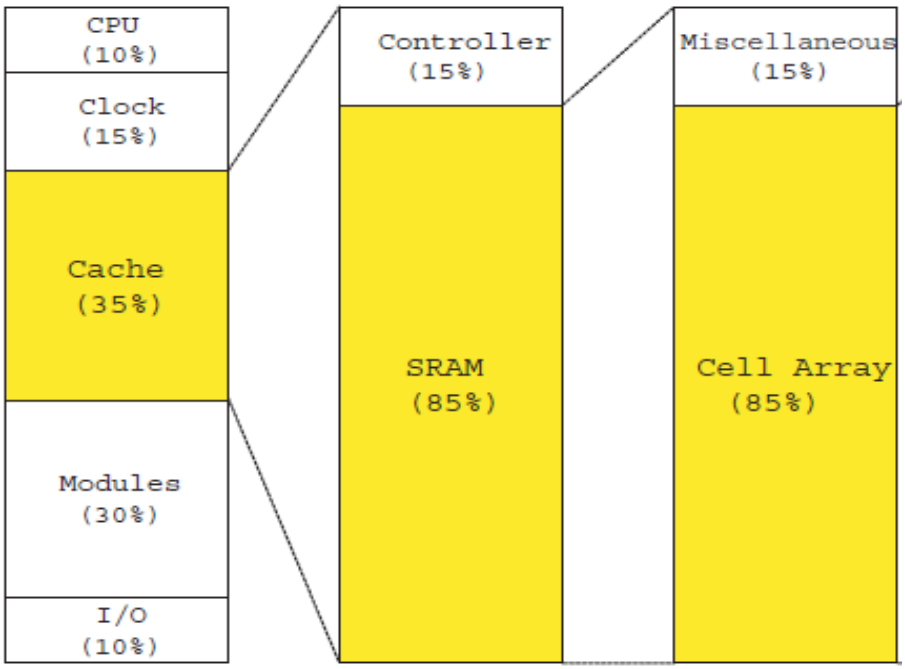
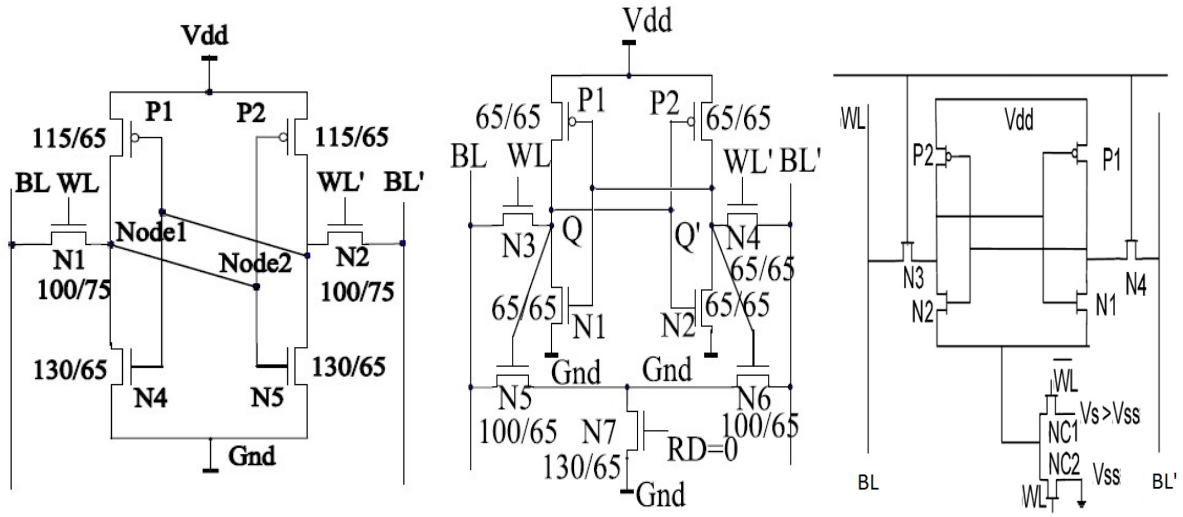


Figure 4.1 Microcontroller Power Consumption Distribution (a) A 32-bit MCU with 8k-byte SRAM (b) A cache with 8k-byte SRAM (c) Cell array with miscellaneous [110]

Besides the low power constraint, high stability to variations and low error during read and write operations are also highly desirable in SRAM design. The static noise margin (SNM) is considered to be the standard measure of the stability of a SRAM cell while write-trip-point (WTP) [111] characterizes the quality of the write operation of an SRAM cell.

4.2 Related Works

There are many memory cell architectures which are reported in the literature with different optimization goals. Our main optimization constraint is the power consumption, particularly in static mode. Several SRAM designs, which are dedicated to achieving low leakage power, are reported in [112, 113] . The conventional 6T SRAM cell, the 9T SRAM cell [106] and the NC-SRAM cell introduced in [108] are described in this section.



(a) Conventional 6T SRAM **(b) 9T SRAM [106]** **(c) NC-SRAM [108]**

Figure 4.2 SRAM Cell Designs

4.2.1 Conventional 6T SRAM

SRAM cells are conventionally composed of 6 transistors as shown in Fig.4.2 (a), sized for performance in the 65nm process. Transistors *P1*, *P2*, *N4* and *N5* consist of a pair of cross-coupled inverters to retain the value. Transistors *N1* and *N2* are both access transistors to take care of writing and reading operations. This type of design is widely used in many designs and applications. The memory suffers from poor stability because the stored data will be disturbed during the reading operation. The transistors of 6T SRAM cell must be carefully sized to meet the requirements of both write and read [111]. Otherwise, errors may occur. Additionally, the power consumption of this type of SRAM cell is also high. One advantage is that it uses only 6 transistors to build a memory cell resulting a lower area and relatively simple layout.

4.2.2 9T SRAM

The 9T SRAM cell is presented in Fig.4.2 (b) [106], also using 65nm technology. Compared with the standard 6T SRAM cell, it can be seen that the 9T SRAM cell is using a smaller transistor sizing especially in the upper part (transistors *N1*, *N2*, *N3*, *N4*, *P1* and *P2*). The design has two separate sections for the writing and reading operation respectively. The upper part, which consists of two cross-coupled inverters, is only accessible in writing mode

through the transistor *N3* and *N4*. The lower sub-circuit serves the reading operation. The biggest improvement for this design is to enhance the SNM by 2 times. Also, the sizing of transistors is not as restricted as that in 6T SRAM. The idle 9T SRAM cells are placed into a super cut-off sleep mode which resulting in reducing the leakage power consumption by 22.9% as compared to the standard 6T SRAM cells in 65nm CMOS technology [106]. However, according to the simulation results, this 9T SRAM cell does not show much improvement in power dissipation compared to the 6T SRAM cell.

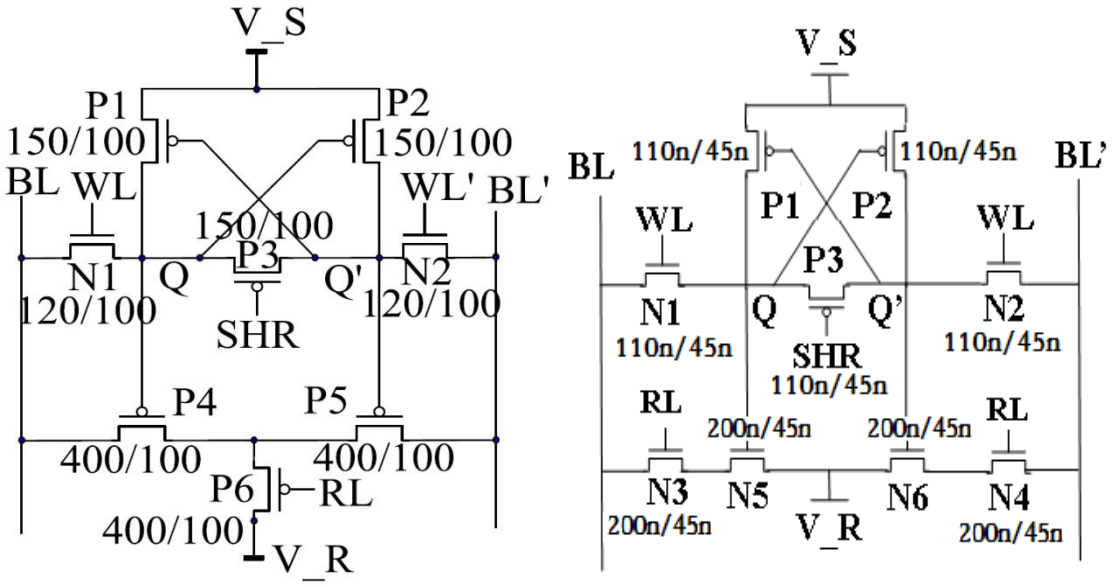
4.2.3 NC-SRAM

The N-Controlled SRAM (NC-SRAM) is illustrated in Fig.4.2 (c) [108]. For writing and reading mode, this SRAM cell works similarly with the 6T SRAM cell. While in idle mode, the source of the NMOS transistors *N1* and *N2* can be switched to a positive voltage *Vs* through the pass transistor *NC1* instead of *Vss*. In doing so, it reduces the leakage power of the cell. Compared to a conventional 6T SRAM cell, the NC-SRAM cell decreases the total gate leakage current by 66% and the idle power by 58% [108]. But it inherits the poor SNM from the conventional 6T SRAM. Also, the power consumed during writing and reading operations is not improved.

4.3 Two Novel SRAM Cells

Two novel SRAM cells with good power saving property, which could run in both adiabatic and non-adiabatic mode, are described in this section. Writing and reading are two basic functions of an SRAM cell. Both these two operations consume most of the dynamic power of an SRAM cell. As the novel SRAM cells are able to operate in adiabatic mode, the proposed SRAM can save the dynamic power during these two operations.

One cell architecture is composed of 8 transistors while the other is built by 9 transistors. The structures of these two designs are similar, using two separate parts to realize the data input and read respectively. Of these two cells, the proposed 9T SRAM cell also has better reading performance, which is very important for an SRAM cell.



(a) Proposed 8T SRAM **(b) Proposed 9T SRAM**

Figure 4.3 Two proposed SRAM cells

4.3.1 Proposed 8T SRAM

4.3.1.1 Circuit Description

The architecture of a novel 8T SRAM cell [114] is depicted in Fig.4.3 (a) sized in 90nm CMOS technology. Transistors *N1*, *N2, P1*, *P2* and *P3* are used for writing operation and data retention. The writing access transistors, *N1* and *N2*, are controlled by write signal *WL*. Two PMOS transistors, *P1* and *P2*, are combined to form a ring to store the data at internal nodes *Q* and *Q'*. The PMOS transistor, *P3*, controlled by *SHR* signal, is placed between nodes *Q* and *Q'* in order to share the energy between *Q* and *Q'* and thus meet the adiabatic principles mentioned in Chapter 2. The three transistors below *P4*, *P5* and *P6* control the reading operation, while *P4* and *P5* follow the control of *Q* and *Q'* respectively. Transistor *P6* is mastered by a read-line signal RL

4.3.1.2 Write Operation

A concept of "default value", which is set as $V_{DD}/2$, is introduced in order to take advantage of using adiabatic logic. It is similar to the split-level adiabatic logic with each bit-line always starting to change from $V_{DD}/2$. In order to meet the principle of adiabatic operation introduced in section 1, there is a crucial step before the write signal *WL* goes high (to turn on

the transistors *N1* and *N2*) which is to share the energy between the nodes *Q* and *Q'*. This can be completed by setting the signal *SHR* to low and thus to turn on the PMOS *P3*. In this way, *Q* and *Q'* reach the same final voltage, $V_{DD}/2$. Moreover, before the write operation starts, the value of two bit-lines *BL* and *BL'* are held at $V_{DD}/2$ as well. Once the signal *WL* goes high, the two NMOS transistors *N1* and *N2* are activated. If *BL* is charged from $V_{DD}/2$ to V_{DD}, the node *Q* is switched to high value from $V_{DD}/2$. On the other hand, *Q'* follows *BL'* to decrease from $V_{DD}/2$ to zero. The slower *BL* and *BL'* change the more power can be reduced, referring to equation (2.1). After the nodes *Q* and *Q'* have been overwritten, the signal *WL* can be switched to zero to turn off the transistors *N1* and *N2*. *BL* and *BL'* are then back to the default value, $V_{DD}/2$, which could be easily done by sharing the energy between *BL* and *BL'*. This means the SRAM now is turned into the idle mode and ready for reading or overwriting. In addition, to improve the stability of the proposed SRAM cell, based on 65nm and 45nm technologies we used for the simulations, after the writing operation is completed, *V_S* is switched to 0.8V to reduce the leakage current and thus achieve high stability during intensive reading operations.

4.3.1.3 Read Operation

In common SRAM designs, the lines *BL* and *BL'* need to be pre-charged to full V_{DD} before reading starts [17]. In our design, *BL* and *BL'* lines are both returned back to $V_{DD}/2$ after the writing operation finishes, which could be considered as pre-charging, so that more power can be saved. In read mode, assuming node *Q* is '1' and *Q'* is set to '0', which means transistor *P4* is off and *P5* is on. When the read signal *RL* goes down, it thus turns on the transistor *P6*. *BL'* is then discharged by the power source *V_R* through the path consisting of transistors *P5* and *P6*. Meanwhile, BL stays at the default value. In non-adiabatic mode, the power source *V_R*, which is connected to the source of transistor *P6*, is set to ground level. It should be noted that *BL'* could not be discharged to zero due to the property of PMOS transistor. In adiabatic mode, *V_R* sweeps down from $V_{DD}/2$ to a lower level, which makes *BL'* follows this change. After the data of *BL* and *BL'* is used by sense amplifier, *V_R* would recover to $V_{DD}/2$ again along with *BL'*. The reading operation is thus completed and the transistor *P6* can be switched off.

4.3.2 Proposed 9T SRAM

The schematic of the proposed 9T SRAM cell is exhibited in Fig.4.3 (b). Similar to the proposed 8T SRAM cell, the transistors *N1*, *N2*, *P1*, *P2* and *P3* are also served for the writing operation and data retention, which could be sized to minimal ratio, where the writing access transistors *N1* and *N2* follow the control of the write signal *WL* and transistors *P1* and *P2* are used to keep the data at nodes *Q* and *Q'*. Also, the PMOS transistor *P3*, which is located between nodes *Q* and *Q'*, is dedicated to share the energy between these two nodes to meet the adiabatic principle and improve the write ability as well. Dissimilar to the proposed 8T SRAM cell, although the lower part of the circuit still takes charge of the reading operation, NMOS transistors are chosen here instead of PMOS transistors in order to improve the reading performance and strengthen the stability. The transistors *N5* and *N6* are mastered by nodes Q and Q' respectively. The two read access transistors *N3* and *N4* are activated by the read-line signal *RL*.

4.3.2.1 Write Operation

As introduced earlier in this section, the "default value" concept is also applied into this 9T SRAM cell. Before WL goes high to overwrite nodes *Q* and *Q',* transistor *P3* is turned on by signal *SHR* to share the voltage between these two nodes. In this way, *Q* and *Q'* will finally reach the same value, $V_{DD}/2$, after a short period of time. During a writing operation, assuming '1' is going to be written into node *Q*, signal *WL* goes high firstly, after that *BL* is charged from $V_{DD}/2$ to full V_{DD}, and *BL'* is discharged from $V_{DD}/2$ to ground. Once writing is finished, *Q* and *Q'* hold their new values '1' and '0' respectively. The signal *WL* is then decreased to zero to turn off the two write access transistors *N1* and *N2*. Additionally, both *BL* and *BL'* have to be recovered to the default value, $V_{DD}/2$, after writing is finished. Meanwhile, *V_S* is set to 0.8V.

4.3.2.2 Read Operation

As mentioned before, after writing, both *BL* and *BL'* are set to $V_{DD}/2$, which means the memory cell is ready to be read. As seen in Fig.4.3 (b), in the read mode, assuming the nodes *Q*='1' and *Q'*='0', the transistor *N5* is on and transistor N6 is off. When the read signal *RL* goes high, it turns on the transistors *N3* and *N4*. BL is then discharged by the power source

V_R gradually through the path consisting of *N3* and *N5*. At the same time, *BL'* stays at $V_{DD}/2$. *V_R* sweeps gradually from $V_{DD}/2$ to 0V. After the data of *BL* and *BL'* has been sensed by the sense amplifier, the signal *RL* can be switched to zero and thus to cut off *N3* and *N4*. *BL* then needs to be re-charged to $V_{DD}/2$ to prepare for future reading or writing operations. It is worth mentioning that, in non-adiabatic mode, the power source *V_R* can be set to a constant level, which is lower than 0.5V, i.e. 0.3V, because when reading is completed, the 0.2V voltage difference between *BL* and *BL'* is sufficient for the sense amplifier to produce the correct results (the minimal requirement is 0.1V).

4.3.2.3 Advantages Over the Proposed 8T SRAM

The proposed 8T SRAM cell has two big disadvantages. Firstly, as only PMOS transistors (*P4*, *P5* and *P6* in Fig.4.3 (a)) are used for reading data, the discharge on bit-line is very slow due to the poor performance of PMOS when it needs to pass a zero. Moreover, the voltage difference between two the bit-lines *BL* and *BL'* may not be large enough to be detected by the sense amplifier, which could generate a reading fault. Secondly, when 45nm (or lower) technology node is used, after hundreds of times of reading the data from this memory cell, one internal node which holds '0' would eventually be charged up by the leakage current, which could cause errors. This is because the leakage current is high in the deep submicron region, caused by the shrinking length of transistors and also the reduction of the thickness of gate oxide. The gate oxide leakage current is much higher when a transistor is On than when it is Off. If using PMOS transistors for the reading sub-circuit, the internal node which holds a low voltage value always turns on a PMOS transistor whose gate is connected to this internal node. Under this circumstance, gate oxide leakage current is high. The proposed 9T design addresses this issue by using NMOS transistors instead and thus the memory could be still readable even after thousands of cycles. Moreover, the reading speed is much faster than that of the proposed 8T architecture. However, after to some IC companies, fabrication seems the only way to prove the feasibility of these structures.

4.4 Power Saving Strategy

The dynamic and static power have been two major sources of the total power consumption of a circuit. In an SRAM array, the dynamic power is consumed not only by the SRAM cell but also by the capacitor of the bit-lines. Due to the high capacitance, usually the major part

of dynamic power is dissipated on bit-lines. As mentioned in Chapter 2, adiabatic logic can dramatically save the dynamic power by limiting the current flow at the expense of performance. Furthermore, once the transistor is turned on, the energy flows through it in a gradual and smooth manner not abruptly switching from 0 to V_{DD} (and vice-versa) as in static CMOS logic. In this way, the current will be very small, which results in low power consumption. In order to implement this type of logic into the memory design, the two proposed SRAM cells are divided into two separate parts.

In writing mode, thanks to the introduction of the PMOS *P3* in Fig.4.2, I ensure that the voltage both inside and outside of the memory cell is balanced before a write operation starts although there is a small amount of power consumed during sharing. Hence, the adiabatic principles are satisfied. Regarding the reading operation, firstly, full-swing pre-charging is avoided and with a dramatic energy saving. Secondly, *V_R* provides adiabatic reading behavior when voltage distribution is applied. Even in the non-adiabatic mode, since the proposed SRAM cell works with the half-swing principle, it can still save power for both write and read operations.

On the other hand, static power, which is often neglected in sub-micron (below 1μm) CMOS processes, starts to dominate the total power consumption in some applications which use deep submicron underlying processes, while memory blocks usually have the most leakage power consumption due to their high density/area. Static power is mainly caused by sub-threshold current and gate leakage current, which are proportional to V_{DS} (Drain Source Voltage) and gate oxide thickness respectively. Having no control over the process parameters, reducing V_{DS} is my main approach to achieve low static power. It can be observed from the two proposed designs (Fig.4.3) that there is no direct ground connection in the SRAM cells along with the balanced voltage distribution in idle mode, hence the standby power can be reduced to an extremely low level. The power results will be discussed next in Section 4.6.

4.5 Write and Read Simulations

Besides the low power constraint, high stability to variations and low error during write and read operations are also highly desirable in SRAM design. The static noise margin (SNM) is considered to be the standard measure of the stability of a SRAM cell while write-trip-point

(WTP) characterizes the quality of write operation of an SRAM cell. Based on the discussion in last section, it could be predicted that the write-ability of both proposed SRAM cells should be close while the 9T design would surpass the 8T one regarding the read-ability.

4.5.1 Write Ability

The write ability of a SRAM cell is characterized by write-trip-point (WTP), which describes how much power is needed to write into a cell. The lower the WTP, the higher is the pulling down energy required and vice-versa [111]. Since we use the sharing in our SRAM cells, the nodes *Q* and *Q'* (in Fig.4.3) are level before new data is written to the cell. It makes it much easier to overwrite the proposed SRAM cells in contrast to other designs. Fig.4.4 shows the

WTP comparison of a standard 6T, 9T and the proposed 9T SRAM cell at V_{DD}=1V. The WTPs of 6T, 9T and the proposed 9Tcells are 0.33V, 0.29V and 0.49V respectively, which indicates that our 9T SRAM has the best write ability. Because the WTP of our SRAM cell is the highest, it indicates that lower power is consumed for overwriting. In other words, it is much easier to write the data into our SRAM cell. It is also interesting to see that the conventional 6T SRAM cell has higher WTP than the 9T one, which is due to bigger transistor sizing.

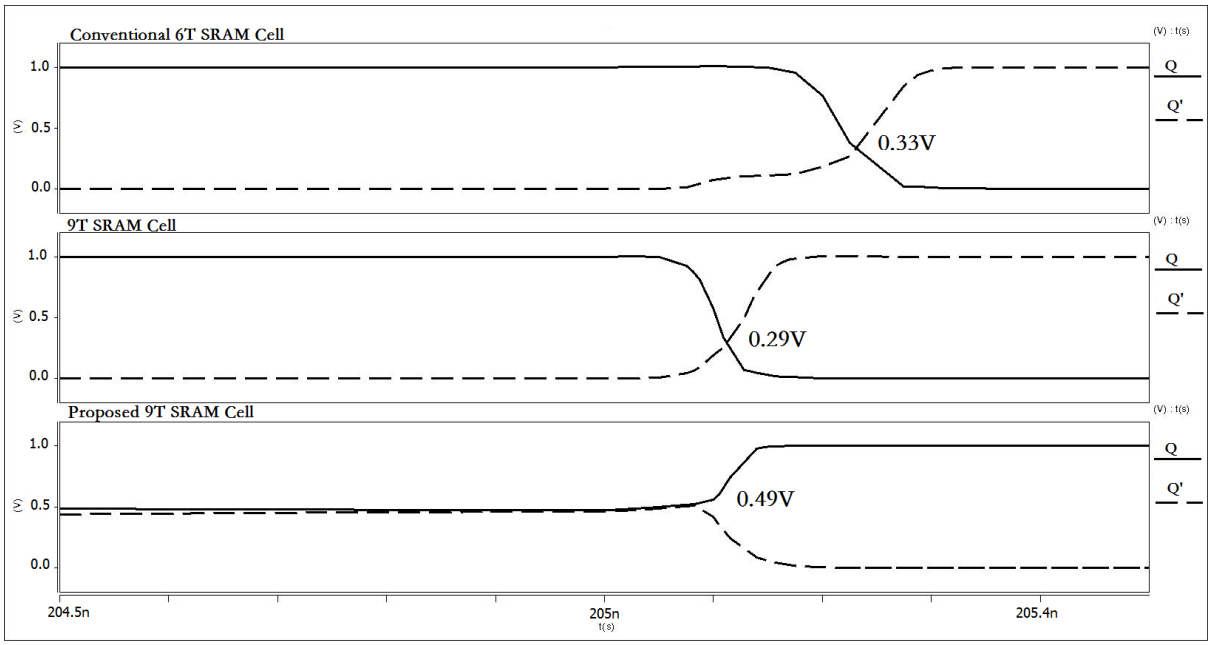
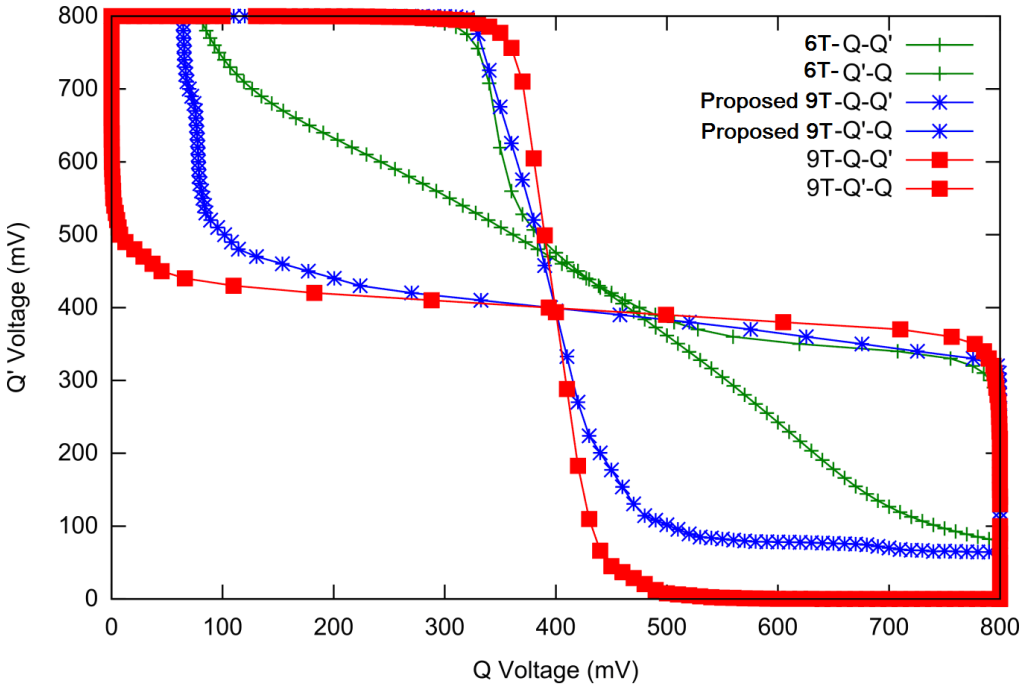


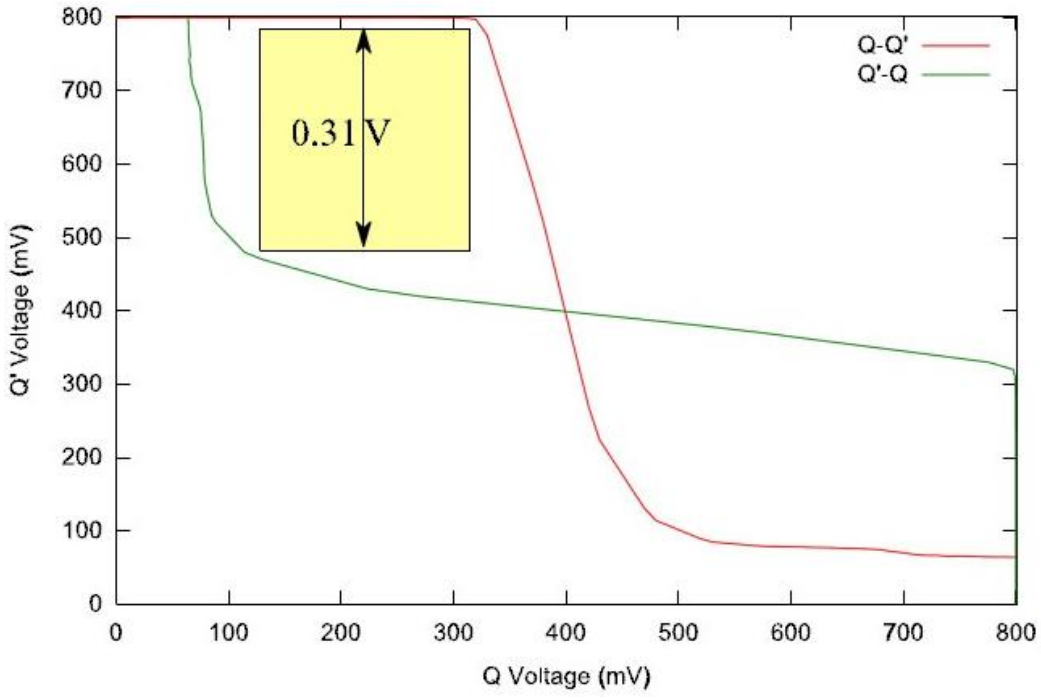
Figure 4.4 Write-Trip-Point Comparison

4.5.2 Read Ability

Static noise margin (SNM) is the standard parameter to characterize the read stability of a SRAM cell. Since data retention nodes *Q* and *Q'* (in Fig.4.3) are isolated in read mode, the SNM is enhanced compared with the standard 6T SRAM cell. However, the SNM of our SRAM cells are not as good as that of 9T one [106] as our designs are without a pull down transistors and a ground connection. Fig.4.5 (a) exhibits the SNM comparison of 6T, 9T and our 9T SRAM cell at V_{DD} = 0.8V. It is because our SRAM uses *V_S* = 0.8V during reading and data retention at 65nm and 45nm nodes. Also, it should be noted that our 8T and 9T SRAM cell have almost equal SNM. The SNM of our SRAM cell is about 310mV, shown in Fig.4.5 (b), which is almost 110mV higher than that of 6T SRAM and is approximately 60mV lower than that of a 9T SRAM. As mentioned above, the proposed 8T SRAM cell is very slow for data reading and would have errors after hundreds of cycles of reading. Our 9T SRAM cell uses four NMOS transistors to replace the three PMOS transistors as the lower part of the circuit. During a reading operation, assuming *Q* ='1' and *Q'* ='0', because the gate oxide leakage current is much lower when a transistor is Off than when it is On, our design could be read thousands of times. Fig.4.6 shows the simulation waveforms of the proposed 8T SRAM cell and the proposed 9T SRAM cell being read after hundreds of times in non-adiabatic mode where the temperature is 100°C. It can be seen that the reading operation of the 8T design (dotted line) is quite slow compared to our design (solid line), and the discharging level gets insufficient to be detected by sense amplifiers at the end while the discharging of the 9T design is much more effective and fast. Therefore, it is clear that the 9T SRAM cell has much better reading performance than the 8T one. Moreover, we tested the reading reliability of our 9T SRAM cell by accessing it 10,000 times in 1mS. Because there is no ground connection in our SRAM cell, which means that one of the internal nodes *Q* or *Q'* is floating and would be charged up by leakage current to a certain value. During the simulation, assuming *Q'*='0', it could finally drift to the value as high as the threshold voltage of a NMOS transistor. Even though, as *V_R* is set to 0.3V, the voltage difference between the Gate and Source node of transistor N6 in Fig.4.3 (b) is still smaller the threshold voltage of transistor N6 which indicates it is always off and thus it will not affect the reading ability.



(a)



(b)

Figure 4.5 (a) Static-Noise-Margin Comparison (b) SNM of the Proposed SRAM

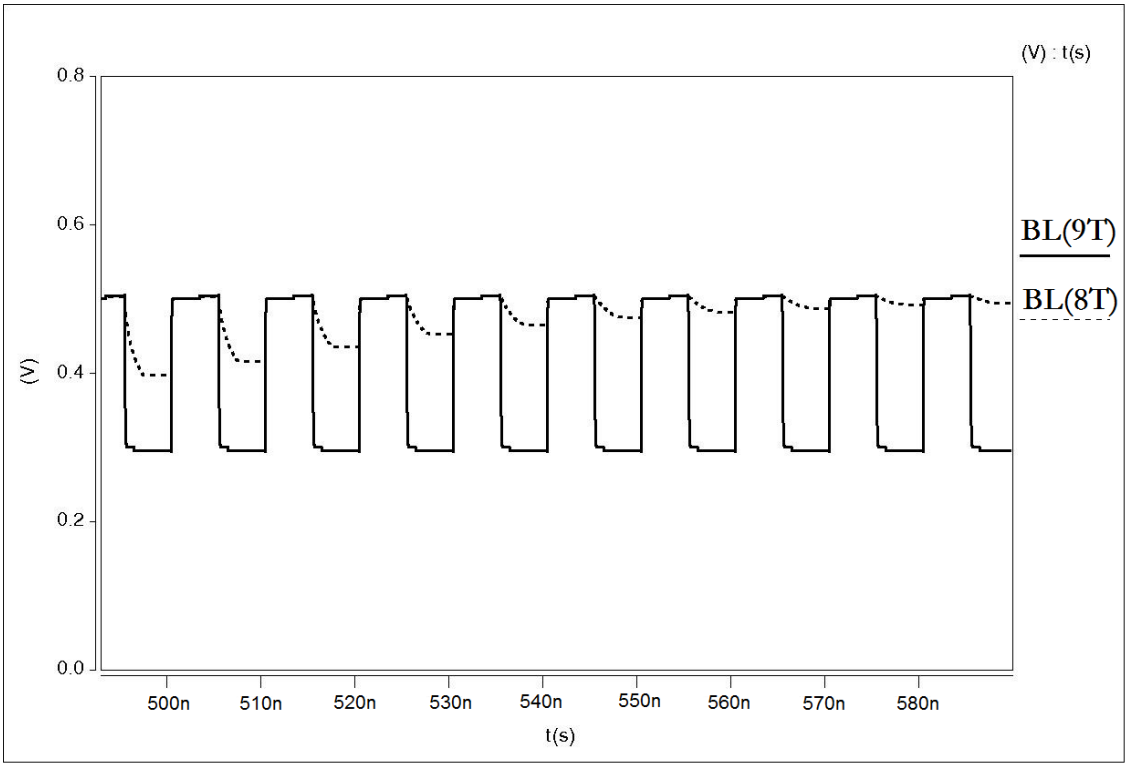


Figure 4.6 Reading Operation of the Proposed 8T SRAM cell and the Proposed 9T SRAM Cell

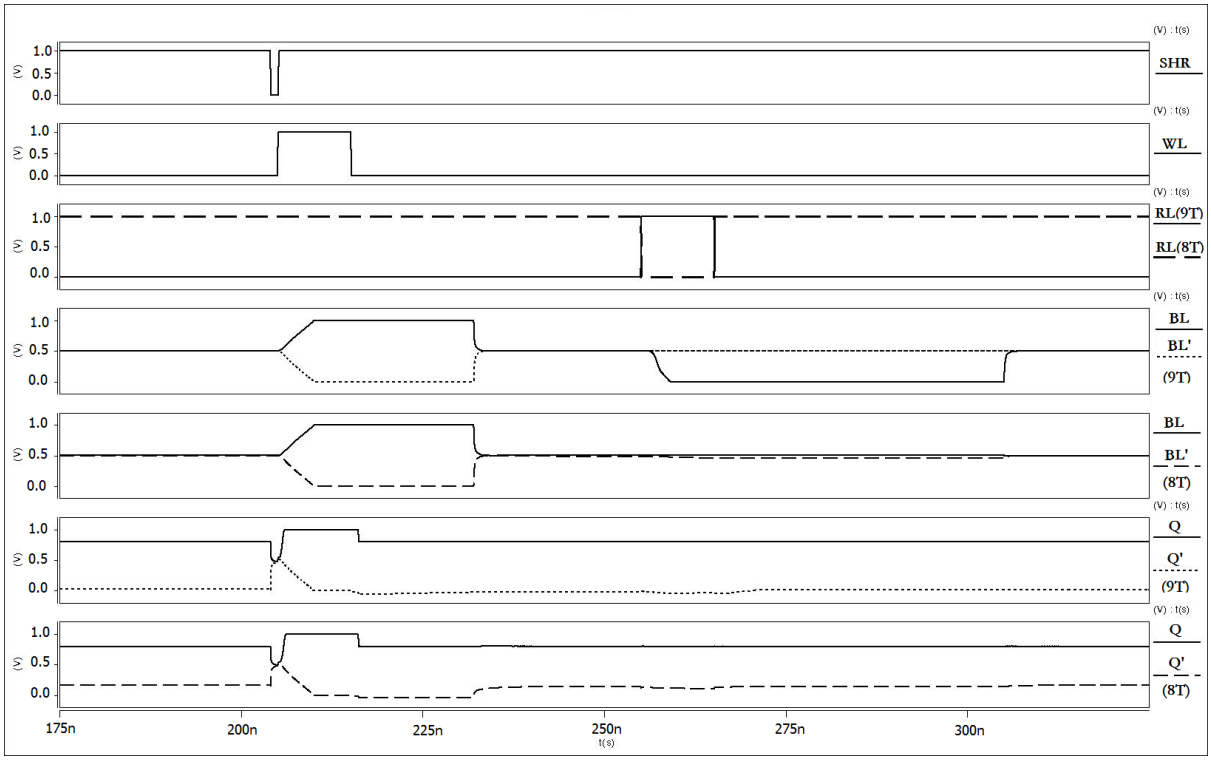


Figure 4.7 Simulation waveforms of the Proposed 8T and 9T SRAM Cells

4.6 Performance Evaluation

As discussed above, the novel 9T SRAM cell is better than the 8T design in terms of reading performance and stability. In this section, the simulation waveforms of the proposed 8T and 9T SRAM cells are described and explained. Because the 9T architecture proved to be more practical, only the 9T design is chosen to be compared with other designs mentioned in Section 4.2 from a power consumption point of view. All the simulations are done by HSPICE using 45nm High-VT SNSP (Slow NMOS Slow PMOS) technology (library available commercially) with the switching probability is 0.5. Monte Carlo simulation is used to determine the leakage power and to perform the comparison.

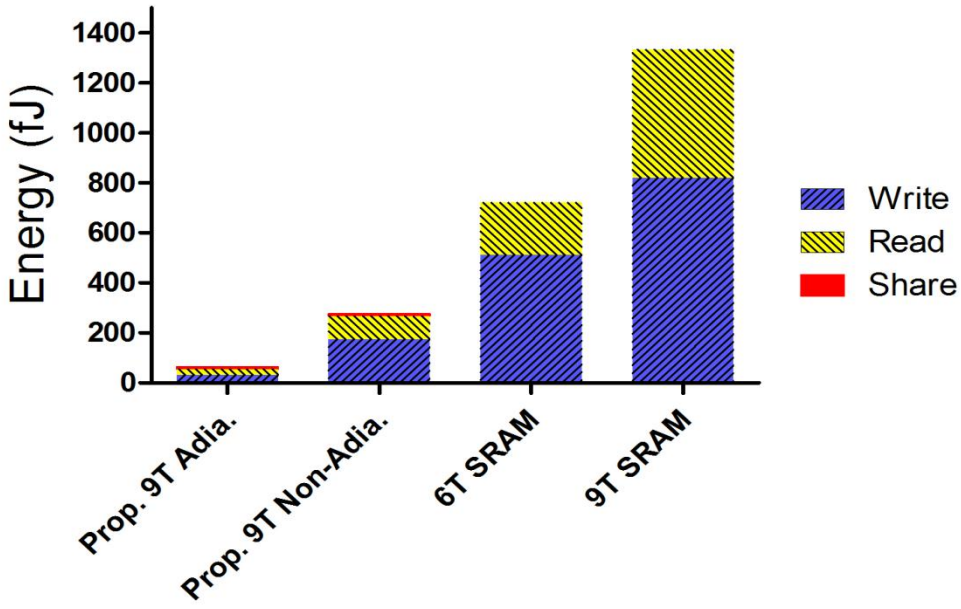
4.6.1 Simulation Waveforms

The simulation waveforms of the two proposed SRAM cell running in adiabatic mode are shown in Fig.4.7. As introduced in Section 4.3, there is a step called sharing needed to be done before the write operation. When the signal *SHR* became low, *P3* was on (in Fig.4.3) and thus the energy of nodes *Q* and *Q'* (both 8T and 9T SRAM) was shared. After a very short period of time, these two nodes were finally even at 0.5V. The next step was to write the data into the cell. Two write access transistors *N1* and *N2* were turned on when *WL* was set high. *Q* followed *BL* and *Q'* followed *BL'* subsequently. In this case, *Q* switched to high and *Q'* was turned to zero. After the writing was completed, *BL* and *BL'* would return to the default value $V_{DD}/2$. It should also be noticed that after the writing was finished *V_S* in Fig.4.3 was switched to 0.8V instead of 1V, so did Q in this case. This could effectively reduce the leakage current in the PMOS ring combined with *P1*, *P2* and *P3*. When *RL* went high (*RL* went low for the 8T cell), *BL* was then discharged gradually through the path consisting of *N3* and *N5* by *V_R* in our 9T SRAM cell while it was *BL'* which went low following the *V_R* in the 8T design. *RL* became low again (*RL* went high for the 8T cell) after the values of *BL* and *BL'* are used by sense amplifier. It could be easily seen that the discharge on *BL'* in the 8T model is quite small and slow due to the poor passing "zero" ability of PMOS transistors. Therefore, compared with these two cells, the 9T one is doubtlessly superior to the 8T model.

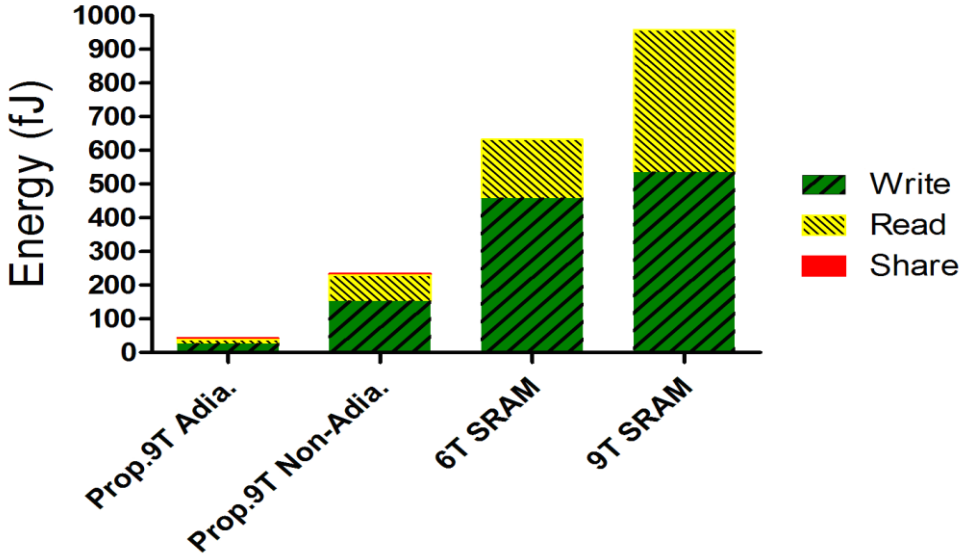
4.6.2 32x32 SRAM Array Dynamic Power Comparison

Three 32x32 SRAM arrays, which were built by the standard 6T, the 9T [106] and the proposed 9T cell, were implemented in 65nm and 45nm technology running at 100MHz. Both adiabatic and non-adiabatic modes were considered for our 32x32 SRAM array. Figures 4.8 (a) and (b) illustrate dynamic power comparison among the three types of designs using 65nm and 45nm process respectively, in the event of three basic operations, writing, reading and sharing. As shown in Fig.4.8 (a), the adiabatic writing and reading cost about 31.5fJ and 23fJ energy for 65nm process, which are more than 90% saving compared to 6T and 9T SRAM array, which consume 512fJ, 209fJ and 820fJ, 513fJ respectively. In non-adiabatic mode, our design can still save 70%-80% energy in writing and more than 50% in reading compared to the conventional 6T design. It is due to the half swing operation is applied. Power dissipation of pre-charging is not counted here since there are various methods to realize this procedure. It is interesting to see that 9T SRAM consumes much higher power than 6T SRAM. At 65nm, the sharing energy dissipation is about 10fJ in both adiabatic and non-adiabatic mode, which is a low penalty given the savings during write and read operations. When considering the 45nm technology node (Fig.4.7 (b)), the proposed 9T memory design shows consistently ultra-low power property. In adiabatic and non-adiabatic modes, the energy consumed during writing operation is 25.8fJ and 152fJ respectively, which is more than 94% and 67% less energy compared to 6T and 9T design, which are 458fJ and 535fJ. Also more than 90% energy is saved during reading operation due to adiabatic logic. Due to the half swing operation, greater than 50% power reduction is achieved in the non-adiabatic case despite about 5fJ being dissipated during the sharing operation.

These results show that our design provides an ultra-low dynamic power property compared to the conventional 6T and the 9T memory cell, while the adiabatic mode brings further a 30% to 40% reduction than the non-adiabatic mode. This part of saving is mainly obtained from bit-lines charging and discharging due to the high capacitance on them.



(a) Implemented in 65nm Process



(b) Implemented in 45nm Process

Figure 4.8 32x32 SRAM Arrays Power Comparison

4.6.3 Static Power and Process Variation

As feature size shrinks, the main contribution of power consumption is down to leakage power especially when the system is in the idle mode. Significant efforts, such as supply voltage scaling [115], idle mode implementation [116, 117], and body biasing [118], have been made to decrease the leakage current of the SRAMs since the memory spends a relatively long time in the idle mode.

The main leakage current components of the proposed 9T design during idle mode are shown in Fig.4.9. These are the gate oxide leakage current and sub-threshold leakage current. Due to the exponential relation between sub-threshold current and process parameters, such as effective gate length, oxide thickness and doping concentration, process variations can severely affect both power and timing yields of the designs [119, 120]. Therefore, the power consumption of the design with respect to process variation is investigated further.

Four designs, the conventional 6T, 9T, NC-SRAM and the proposed 9T, were implemented in 65nm and 45nm CMOS processes. The fluctuations in the threshold voltage, doping concentration and gate oxide thickness with normal distribution were assumed. The sigma variation of 10% was considered for each parameter [119]. Fig.4.10 (a) and (b) show the

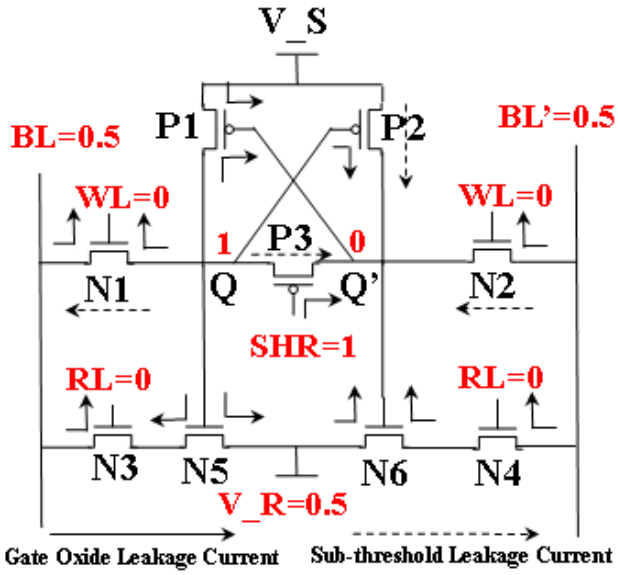
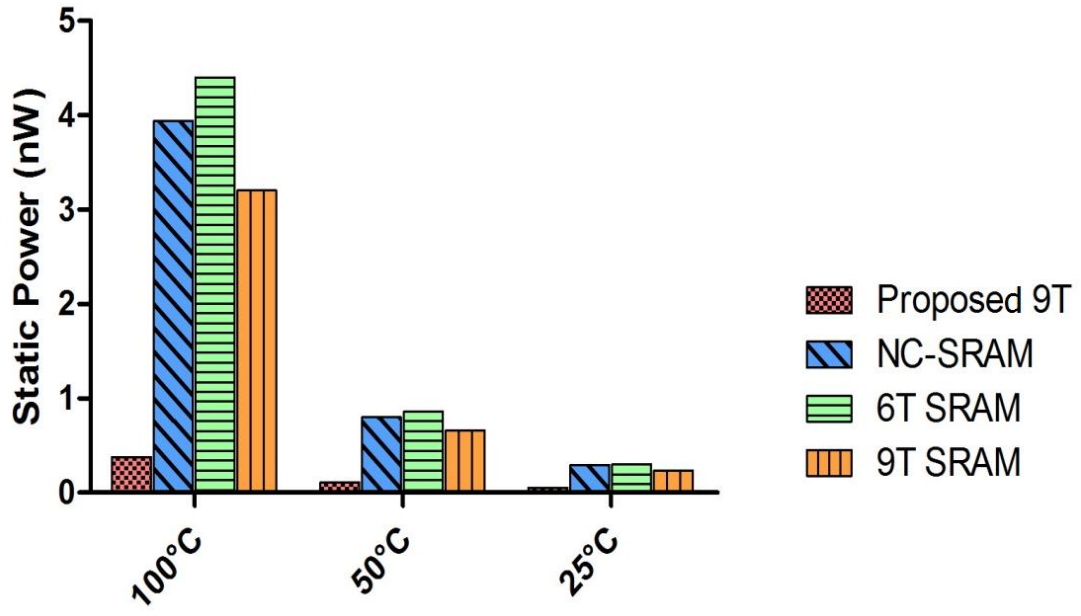
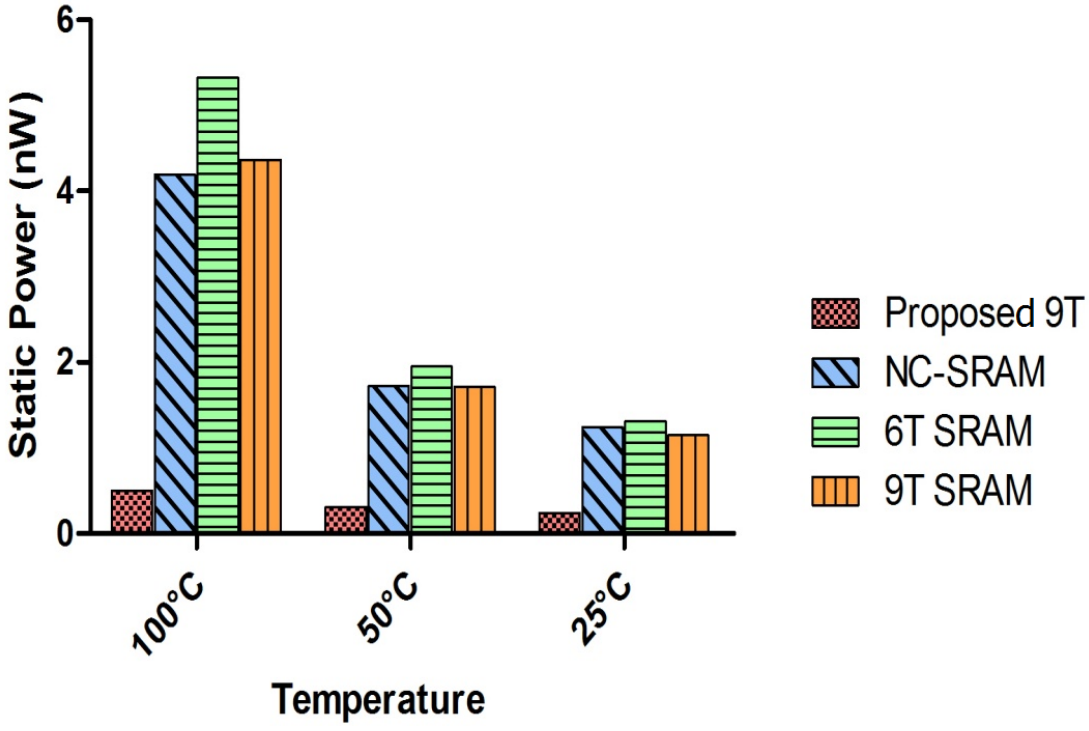


Figure 4.9 Leakage Current in Idle Mode of the Proposed 9T SRAM cell



(a) Memory cells implemented in 65nm CMOS Process



(b) Memory cells implemented in 45nm CMOS Process

Figure 4.10 Monte Carlo -Temperature Variation

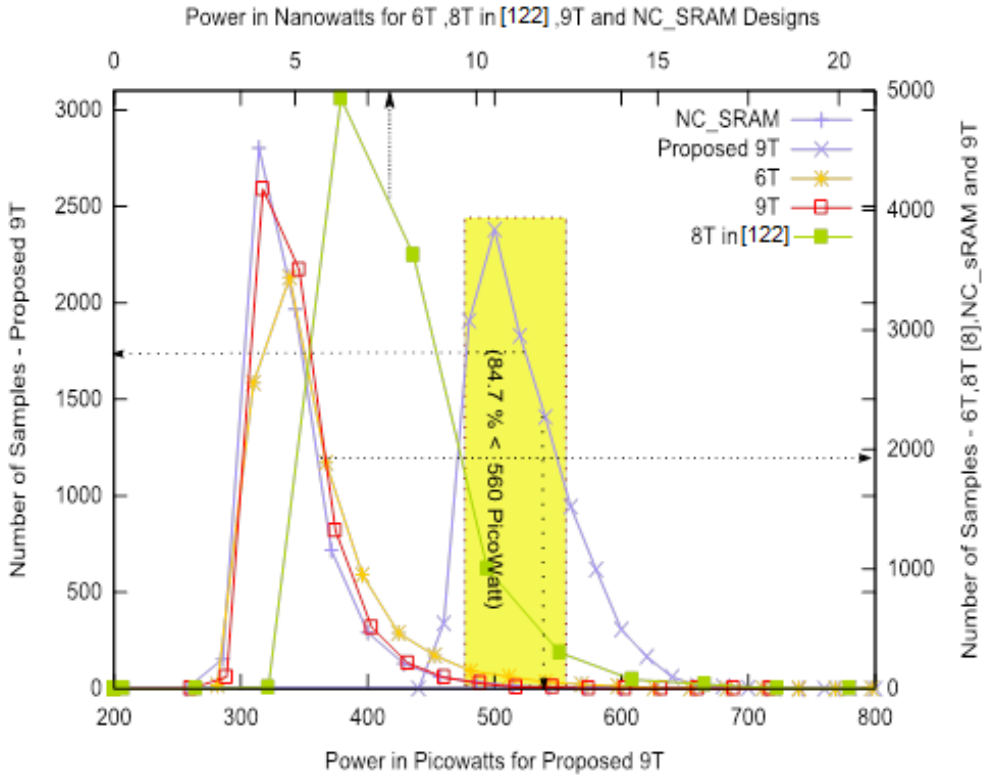


Figure 4.11 Monte Carlo –Process Variation

Monte Carlo simulations [121] of 30 samples by varying temperature from 25 to 100°C. It should be noticed that the figures only shows the mean values for each design. Our 9T SRAM cell has the lowest leakage power at each temperature node. Around 90% power is saved compared to the other three designs at 100°C, which are 0.375nW for 65nm and 0.5nW for 45nm. When the temperature decreases to 50°C and 25°C, the power reduction of our design is still up to 80%.

Monte Carlo simulations for process variation (45nm) of 10000 samples with five memory cells in idle mode were carried out. The average power consumption of the 6T, 9T, NC-SRAM, the 8T in [122] and the proposed 9T memory cells are plotted and shown in Fig.4.11, using the same representation as the one presented in [106]. The bottom x-axis gives the average power in picoWatt (with the scale 200 - 800 picoWatt) for the proposed design while the top x-axis gives the average power in nanoWatt (with the scale 2-22 nanoWatt) for the other designs. The left y-axis represents the number of samples for the proposed 9T design while the right y-axis exhibits the sample numbers of the other designs. The results of the proposed 9T design are shown in blue (x point). It can be seen that approximately 8200

samples (85%) (yellow block) are located at the point less than 550 pW, while the distribution of 9T and NC-SRAM cells both had around 7600 samples (76%) with the power between 4 and 6 nW. There are more than 7000 samples (70%) of the conventional 6T design spreading from 3.85 to 6.85 nW. Hence, power reduction of about 90%, is achieved by the proposed 9T cell.

Based on the simulation results, our proposed SRAM designs exhibit impressive capability in lowering static power, at different temperature nodes, or under intensive Monte Carlo simulation, with nearly 90% power decrease being accomplished.

4.7 Conclusion

SRAM, being an important part of an embedded system (such as wireless sensor networks, has to meet stringent low power requirements. Two new SRAM cells have been proposed which are composed of 8 transistors and 9 transistors respectively. Both of the proposed designs have improved write ability and good read ability compared to the conventional 6T SRAM cell.

Adiabatic logic has been used in our designs in order to achieve ultra-low dynamic power with 90% savings when compared to other popular models. Even in non-adiabatic mode, our SRAM cells could also save up to 50% energy for both write and read operations. Both modes can run at 100MHz, for non-adiabatic mode, the frequency could be even higher. With aggressive technology scaling, process variation has also been taken into account during the simulation along with temperature variation. About 90% leakage power is saved.

The 9T version has proved to be better than the 8T one in terms of the stability and reading speed despite a little more leakage power dissipation. Optimized layout need to be done in the future to compare thoroughly with state-of-art designs.

5 ASYNCHRONOUS CHARGE SHARING LOGIC

5.1 Introduction

With the growing complexity of digital circuits, it is more difficult to optimize their power consumption and the energy. New design techniques are required to achieve this goal. Average case power and timing optimization are tackled within the asynchronous design for some time. In large system on a chip, a single clock (global clock) distribution scheme poses significant power and timing issues and asynchronous architectures or hybrid emerge as potential alternatives due to their localized communication. Also, due to the uncontrollable parameter variations across a chip, the asynchronous schemes are more robust than the fully synchronous solutions which are optimized for the worst case. While embracing the benefits in terms of potential low power consumption, high robustness, design reuse, electromagnetic compatibility and more tolerance to process variations and external voltage fluctuations compared to synchronous designs, asynchronous logic is re-enacted in today's deep sub-micron CMOS technology [17]. Also, these promising features ensure asynchronous logic an important role in embedded systems [31, 123] where modularity is highly preferred and global signals are usually inefficient. Finally, I choose asynchronous logic features in our quest to derive average case power efficient embedded systems.

A novel logic, named Asynchronous Charge Sharing Logic (ACSL) is introduced in this Chapter along with the design of all crucial components. The general comparison and discussion between synchronous logic and asynchronous logic has already been presented in Chapter 2 as well as the basic structure, fundamental protocols and function blocks of asynchronous circuits. The 4-phase dual-rail protocol is preferred in ACSL due to its high stability. However, high power consumption caused by dual-rail logic and comprehensive completion detectors sometimes restrict its popularity. ACSL addresses this by using charge sharing technology which has not been implemented in asynchronous logic before. Additionally, the realization of completion detection is simplified considerably. Apart from the power reduction, ACSL brings another promising feature in average power estimation called input data-independency where this characteristic would make power estimation

effortless and be meaningful for modular quantitative average case analysis as presented in Chapter 3.

5.2 Asynchronous Charge Sharing Logic

The cornerstone of Asynchronous Charge Sharing Logic (ACSL) is to successfully integrate the charge sharing technology with asynchronous logic. To realize this, Positive Feedback Adiabatic Logic (PFAL) is used to build the function blocks in ACSL because PFAL meets the principle of charge sharing technology and matches well with asynchronous logic. However, as the Power Clock Generator (PCG) in adiabatic logic is a big obstacle for asynchronous logic, the charge sharing technology is thus chosen to replace the PCG. Energy is preserved by transferring it to the next stage of the circuits, rather than recycling by the PCG in adiabatic logic. There are other advantages of ACSL, such as ultra-low leakage power due to the naturally incorporated power gating technology and simplified completion detection. In this section, the detail of charge sharing technology and power estimation are given. The basic blocks which make up ACSL are introduced. The general operation of ACSL is also explained. Finally, the circuit design and ACSL architecture are shown.

5.2.1 Charge Sharing Technology

Dual-rail differential dynamic logic is often used in asynchronous design due to its high reliability. The switching activity of this type of logic is high, in fact the switching probability equals one, because the pair of two complementary parts is pre-charged to V_{DD} and one output node would be always discharged to ground during evaluation. Therefore, high power consumption is inevitable. The dynamic energy dissipation of a cycle of pre-charging and evaluation in dual-rail dynamic logic is given by:

$$E_{Dynamic} = \frac{1}{2}CV_{DD}{}^2 + \frac{1}{2}CV_{DD}{}^2 = CV_{DD}{}^2 \tag{5.1}$$

where C is capacitance of the load and V_{DD} is the supply voltage. Half of the total energy is dissipated during pre-charging and the remaining half is consumed during evaluation. The power consumption is almost constant despite the charge sharing between internal stack nodes.

On the other hand, ACSL uses sharing to replace pre-charging which effectively reduces the voltage charge by half. It is worth mentioning that ASCL is suitable to an array-based structure or other structures where capacitive load at each sharing node is close. The equation of a sharing event is given as:

$$Q = C_1V_1 = C_1V_2 + C_2V_2$$

$$\textit{Given that } C_1 = C_2\ ,\ C_1V_1 = C_1V_2 + C_2V_2 = 2C_1V_2$$

$$V_2 = \frac{1}{2}V_1 \tag{5.2}$$

As shown in Fig.5.1, C_1 is the capacitive load at *VPC1* and C_2 is the capacitive load at *VPC2*. V_1, which is the voltage of *VPC1*, is originally at V_{DD}. At the end of sharing process, V_2 (*VPC2* in Fig.5.1) approximately equals half of V_1, $V_{DD}/2$, assuming that C_1 is the same as C_2. For this condition, about 50% energy could be saved. The energy dissipation of ACSL, including one evaluation event by charging V_2 from $V_{DD}/2$ to full V_{DD} and one stand-by event by discharging V_2 from $V_{DD}/2$ to ground, is given by:

$$E_{ACSL} = C(\frac{1}{2}V_{DD})^2 + C(\frac{1}{2}V_{DD})^2 = C\frac{1}{2}V_{DD}{}^2 \tag{5.3}$$

where *C* is the capacitive load. It is obvious that compared to the energy dissipation of dual-rail dynamic logic, given enough time for charge sharing, theoretically half of the total energy is saved by the charge sharing technology.

But there is a small amount of energy which is dissipated during sharing by the transistors as heat. Nevertheless, it can be concluded that the dynamic power consumption of ACSL is nearly uniformly distributed, which indicates that the power consumption is (almost) independent of the change of inputs and thus nearly constant. This feature can be very promising for the power analysis of modular applications as well as predictable design [124].

In terms of leakage power, as VPCs are discharged to zero in stand-by mode and power gating is naturally incorporated, ultra-low leakage power can be achieved. Some simulation results will be discussed in Section 5.3.

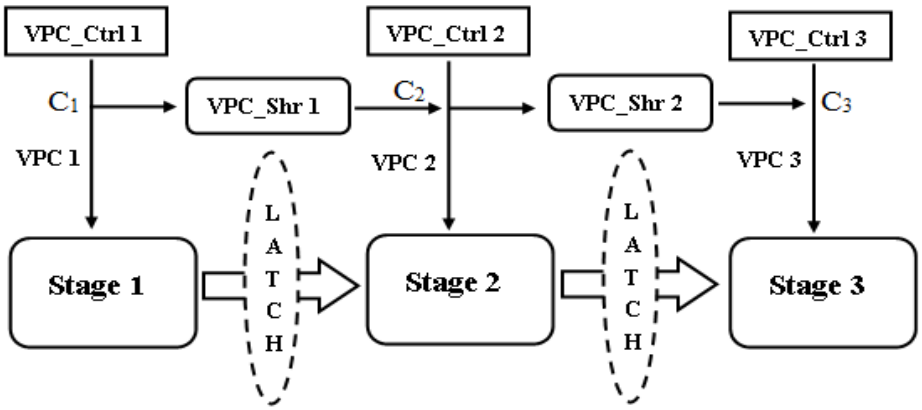


Figure 5.1 General Operation of ACSL

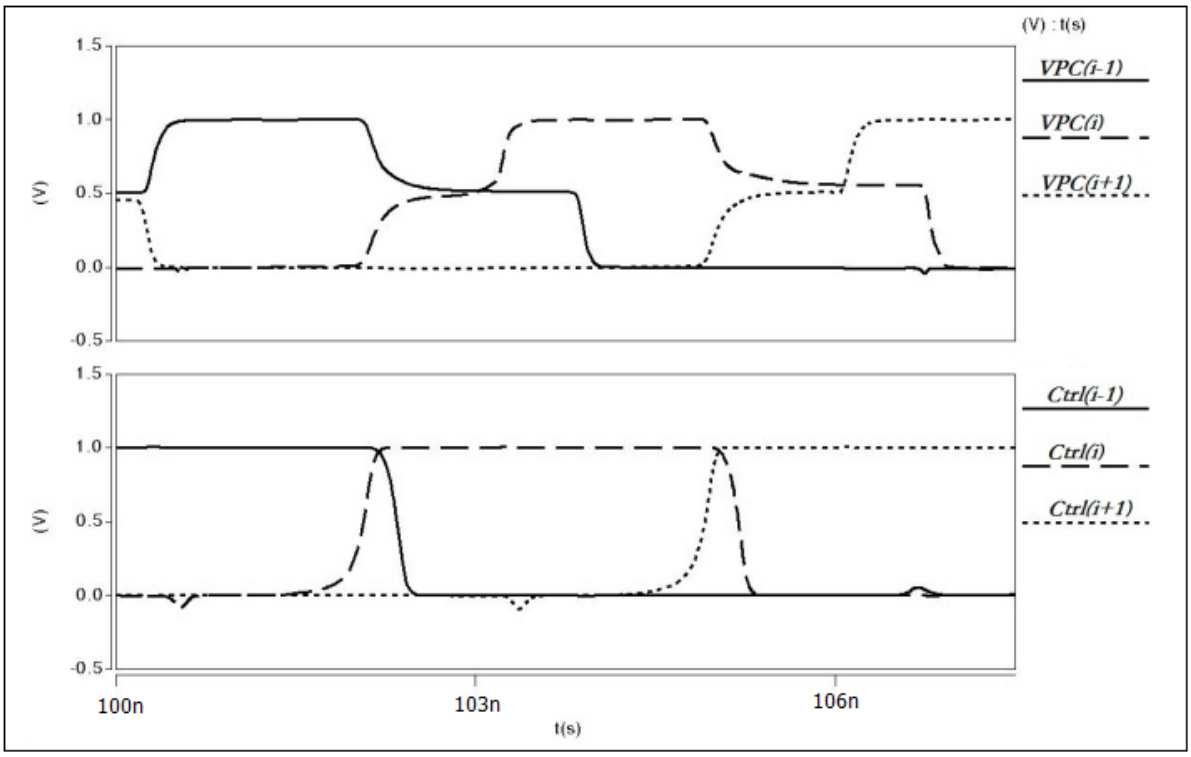


Figure 5.2 Simulation Waveforms for ACSL

5.2.2 Basic Block

The charging, discharging and sharing are performed by a power control block called *VPC_Ctrl* and a power sharing block called *VPC_Shr*. The *VPC_Ctrl* enables the evaluation

and discharging of the ACSL circuits while the *VPC_Shr* is used to share the energy between two neighboring stages. The length of sharing time is decided by Sharing Detector (SD). Besides these three new added blocks, other circuitry will be described at the end of this section.

5.2.3 General Operation of ACSL

The general operation of ACSL is illustrated by a three stage circuit in Fig.5.1. Firstly, *VPC_Ctrl1* evaluates the logic block of Stage 1 to generate the outputs to Stage 2. After the outputs are ready, *VPC_Shr1* placed between Stage 1 and Stage 2 is switched on. Then, energy is transferred from Stage 1 to Stage 2. Once *VPC1* and *VPC2* reach almost the same level, nearly $V_{DD}/2$, SD would then turn off *VPC_Shr1*. *VPC_Ctrl2* is then activated to charge *VPC2* from $V_{DD}/2$ to full V_{DD}. *VPC1* is discharged to zero through *VPC_Ctrl1* when the evaluation of Stage 2 is completed. Meanwhile, *VPC_Shr2* is turned on to pass energy from Stage2 to Stage 3. Then *VPC3* is charged up to V_{DD} by *VPC_Ctrl3*. *VPC2* becomes zero subsequently. To take most advantage from ACSL, the last *VPC* can be connected to the first *VPC* through *VPC_Shr*. All the operations are summarized by three steps as follows. The simulation waveforms are depicted in Fig.5.2.

(1) *VPC(i)* is charged up to V_{DD}. *VPC(i-1)* is discharged to ground.

(2) *VPC(i)* shares the energy with *VPC(i+1)* until both are at $V_{DD}/2$.

(3) *VPC(i+1)* is charged up to V_{DD}. *VPC(i)* is the discharged to ground.

Following the change of *VPC*, an important step in asynchronous logic, completion detection, could be accomplished by sensing *VPC* only rather than detecting every channel of the circuits using several AND and OR gates in dual-rail protocol. Nevertheless, it does not affect the robustness. So, *VPC* in ACSL serves not only as a power supply but also the symbol of completion status. By mixing these two functions, more power could be saved because completion detection of dual-rail protocol is usually expensive.

5.2.4 Circuit Design and ACSL Architecture

Fig.5.3 illustrates the detailed architecture of a two-stage ACSL circuit. Unlike the

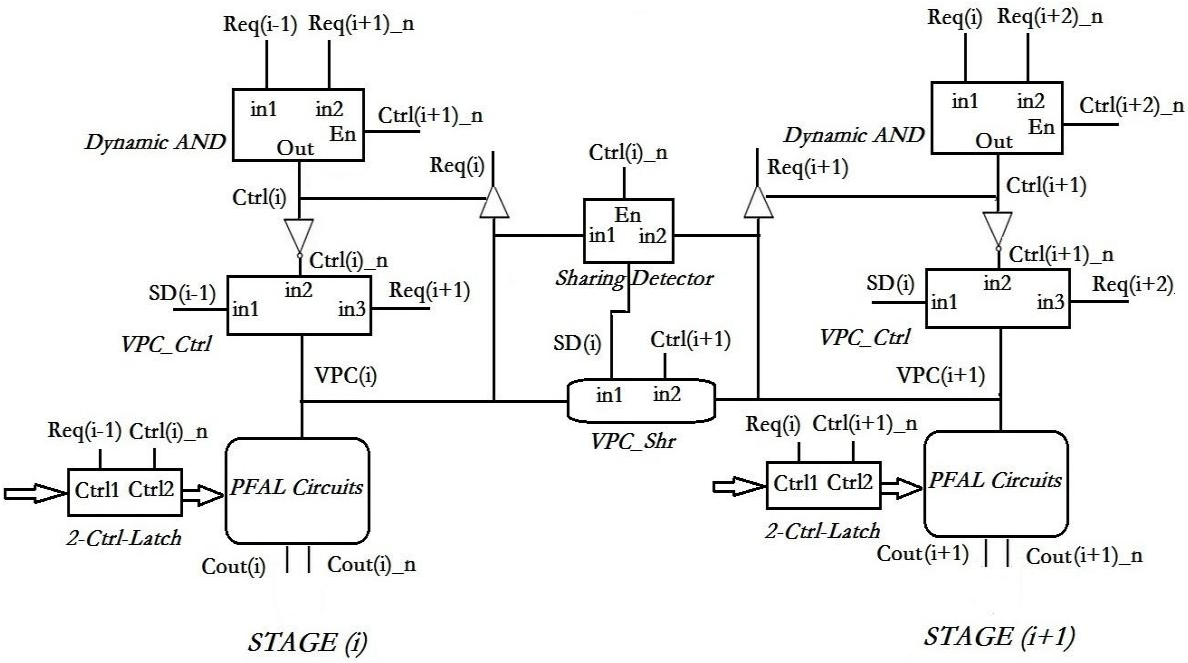


Figure 5.3 Two Stage Architecture of ACSL Circuits

conventional asynchronous circuits, which use the C-elements to manipulate registers and logic blocks, the *Dynamic AND* is accordingly used. The conventional two-input *C-element* produces zero output only if both in1 and in2 are low, shown in Fig.5.4 (a). In contrast, using *Dynamic AND*, shown in Fig.5.4 (b) leads to *Ctrl(i)* in Fig.5.3 switching to low voltage immediately once *Ctrl(i+1)_n* becomes low. In doing so, *Sharing Detector* (*SD*) exhibited in Fig.5.4 (c) is turned into evaluation mode, when both *VPC*s are higher than the threshold voltage of the NMOS transistor which leads the signal *SD* to become '0'. It indicates that the sharing operation can be stopped by switching off *VPC_Shr*, see Fig.5.4 (d). When sharing is finished, *VPC(i+1)* is then charged up to V_{DD} by *VPC_Ctrl* which is exhibited in Fig.5.4 (e). In PFCSL [125], which is the premier prototype of ACSL, delay elements are used in controlling the duration of charge sharing, which is set at about 0.5ns. However, by using *SD* instead, this duration can be reduced significantly which improves the speed of the ASCL circuits. Moreover, ACSL uses a dynamic buffer rather than multiple OR gates commonly used in conventional asynchronous circuits to accomplish the completion detection. The dynamic buffer which is controlled by the *Ctrl(i*) signal in Fig.5.3 is to sense the *VPC* signals and thus to generate the Request signal. In doing so, the design effort, area and power consumption for detection can be saved dramatically.

Furthermore, the signal transition diagrams of the four-phase handshaking model and the

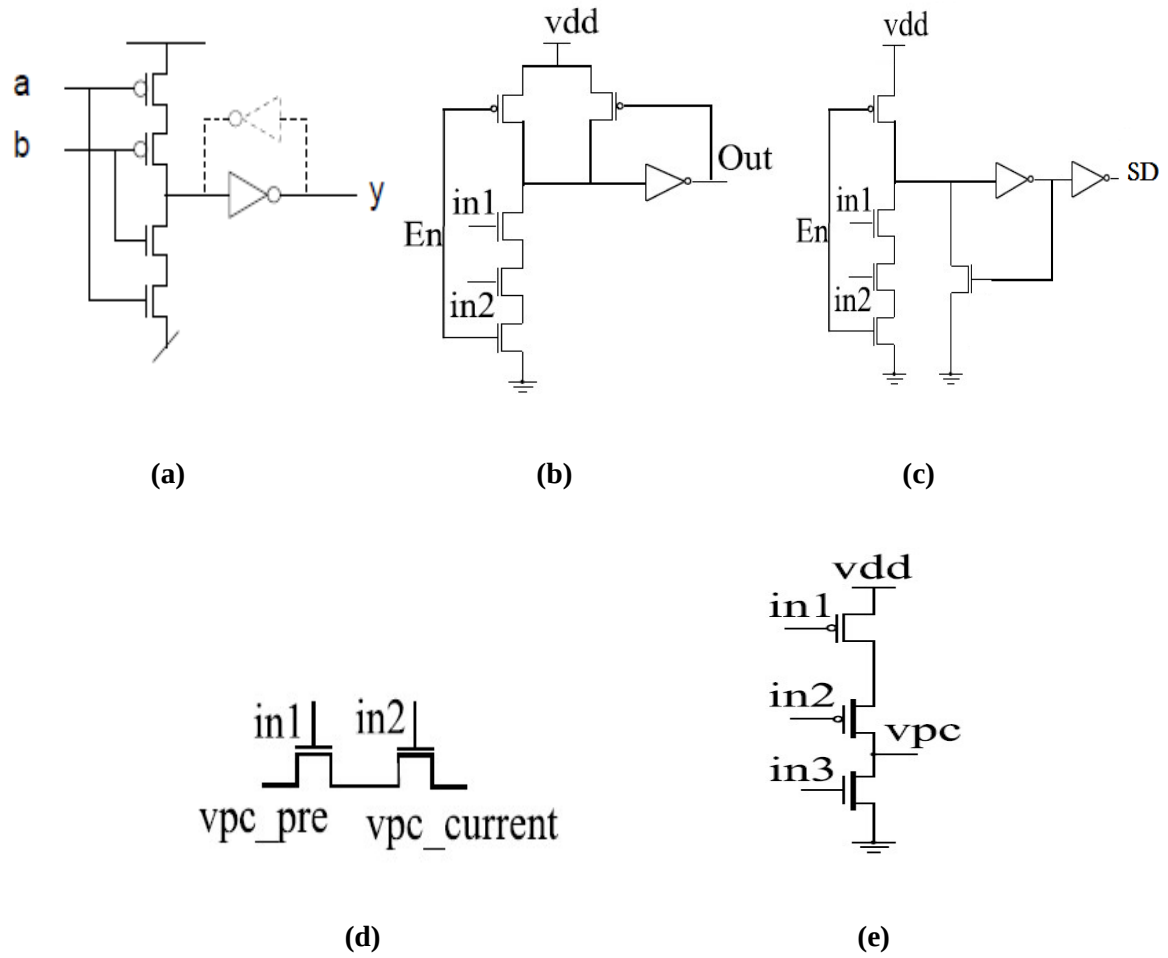


Figure 5.4 Schematics for (a) C-element (b) Dynamic AND gate (c) Sharing Detector (d) VPC_Shr (e) VPC_Ctrl

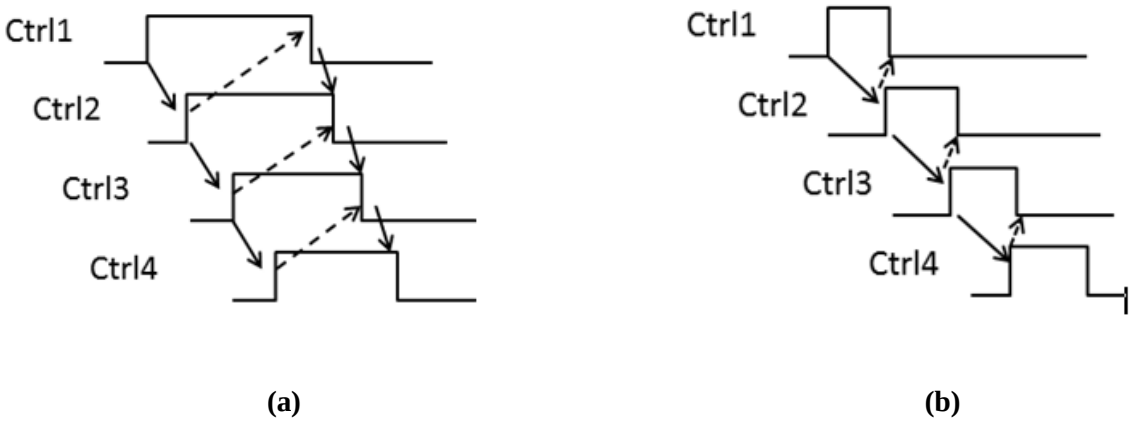
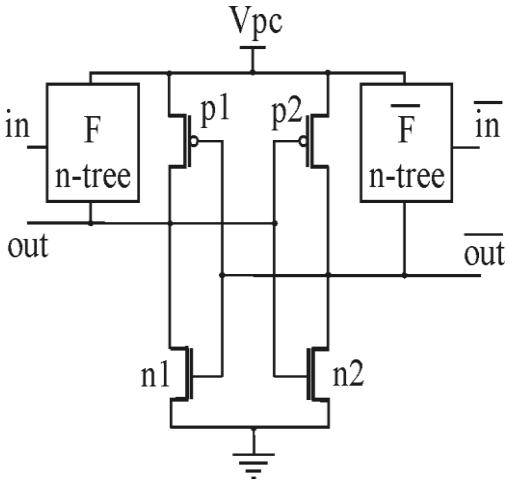
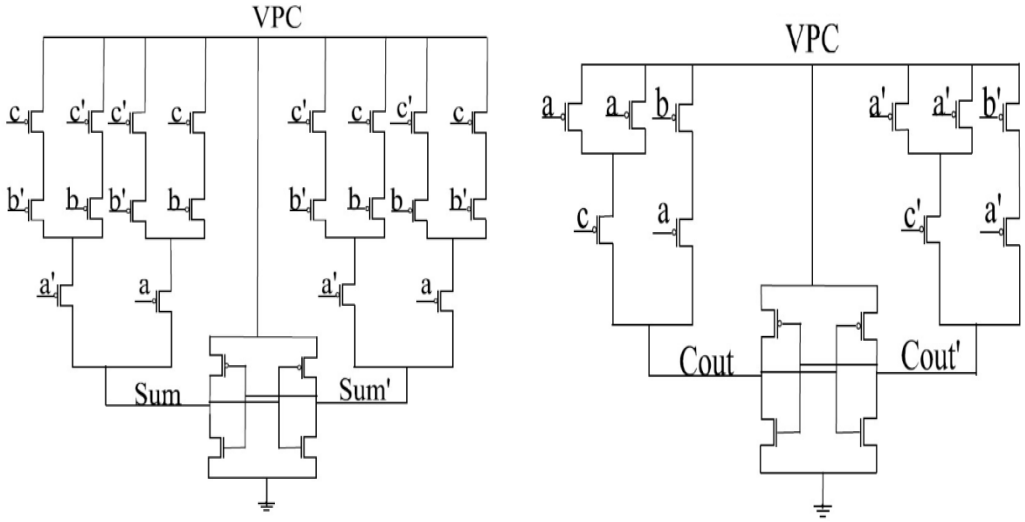


Figure 5.5 Signal Transition Diagram for (a) Four Phase Handshaking Protocol (b) ACSL Handshaking Protocol

ACSL handshaking model are exhibited in Fig.5.5. It can be concluded that it is not allowed that two neighboring stages are in active mode at the same time in ACSL. Also, the on-time of each stage in ACSL is much shorter than that in the conventional four-phase protocol which results in lower leakage power consumption.

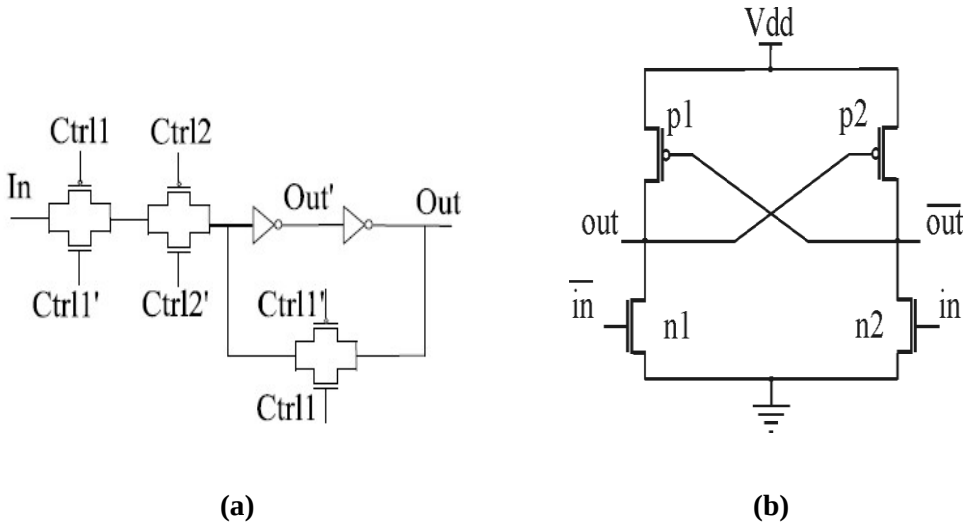


(a)



(b)

Figure 5.6 (a) Generic PFAL Function Block (b) PFAL Full Adder



(a) **(b)**

Figure 5.7 Schematic for (a) Dual-control Lines Latch (b) IACSL Memory

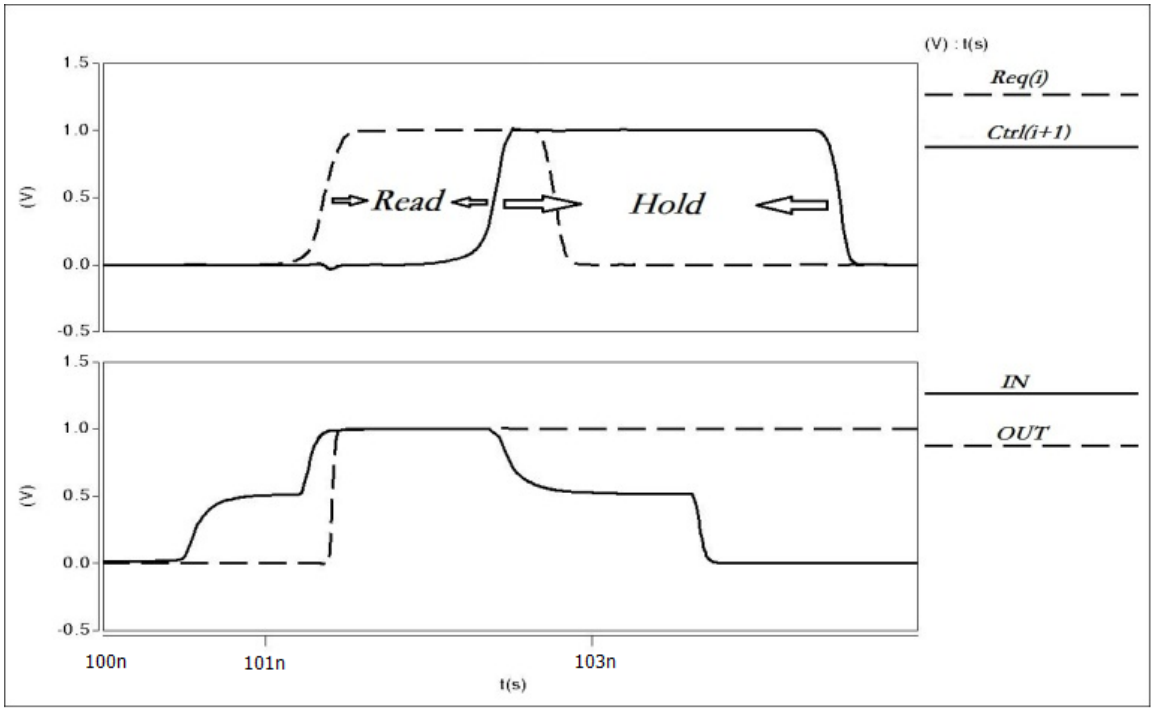
To build the ACSL circuits, the PFAL circuits, one-bit full adder as an example, is shown in Fig.5.6. The structure is well symmetric, and both N-MOS tree and cross-coupled inverter are powered by *VPC*, also the outputs are all followed by *VPC*. When the inputs are ready, *VPC* starts to evaluate the circuits by charging up to a certain value, usually V_{DD}. Meanwhile, the differential outputs are set at '1' or '0' depending on the function of the *n-tree*. In adiabatic logic, after the outputs are read *VPC* then recycles the energy stored in the circuits by discharging itself to zero. However, in ACSL, *VPC* transfers the energy stage by stage. Meanwhile, it also serves as the source of completion detection.

The other crucial part of ACSL components is the latch. It has to be assured that all data from the previous stage is loaded before the sharing happens. Two types of latch designs are illustrated in Fig.5.7. One is called dual-control lines latch and the other is named as IACS (Improved ACSL) memory.

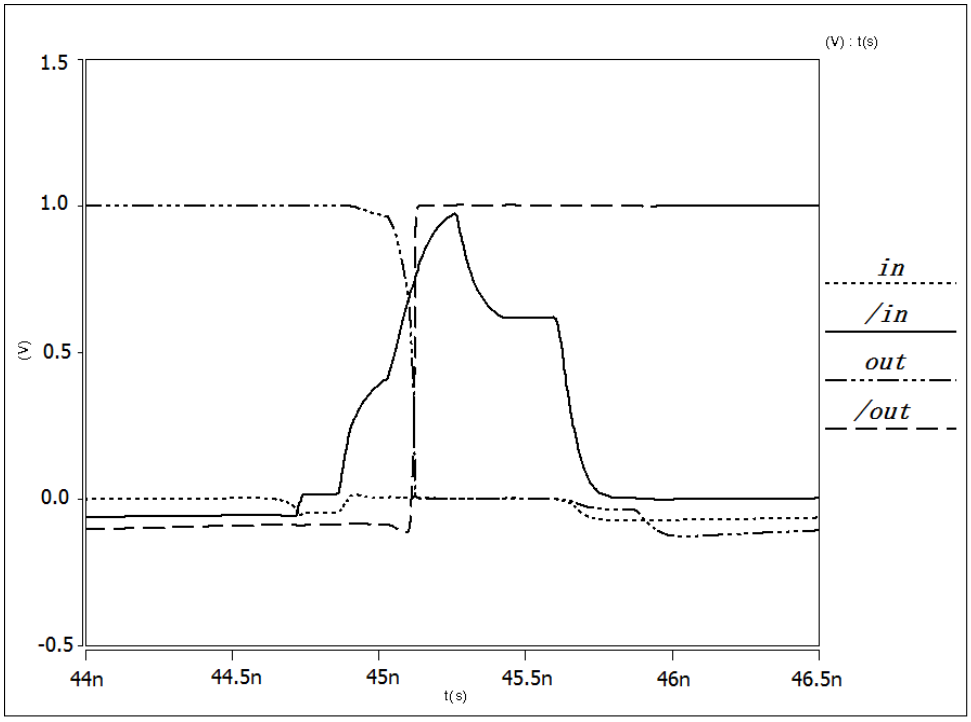
The dual-control lines latch, shown in Fig.5.7 (a), composed of 10 transistors, is manipulated by the controlling signals from two neighboring stages. The simulation waveforms of this latch are shown in Fig.5.8 (a). It can be seen that the latch is accessed only when the signal *Req(i)* is high and the signal *Ctrl(i+1)* is low. The latch enters into hold mode as soon as *Ctrl(i+1)* becomes high. One potential disadvantage of this latch is that two cross-coupled inverters are directly connected to the outputs of the evaluation circuits, which may increase the capacitive load and thus the power dissipation. Moreover, the sizing of the latch needs to be taken care of in order to avoid errors during overwriting.

Fig.5.7 (b) exhibits the circuit of the IACSL memory. The pair of inputs is connected to two outputs of the previous function block in ACSL while the two output nodes are attached to the *in* and */in* from function block of the next stage. Once the previous stage finishes evaluation which is controlled by the power supply *VPC*, one of the outputs is set high and the other is low. Then the outputs of the memory cell will then follow these values. In contrast with the dual-control lines latch, there is no timing control signal needed in IACSL memory which results in low area and power consumption (low capacitive load). The simulation waveforms of the IACSL memory are depicted in Fig.5.8 (b). It can be seen that

the pair of outputs follows the change of inputs. It is worth mentioning that once the previous stage finishes evaluation and data has been absorbed by the IACSL memory, as mentioned in Section 5.2.3, VPC of the previous stage will be discharged to zero and so do the outputs (connected to the inputs of the IACSL memory), and under this circumstance, the IACSL memory will enter the sleep mode which leads to ultra-low leakage power dissipation. In the next section, the performance of the same asynchronous arithmetic unit with these two different memory designs will be compared and discussed.



(a)



(b)

Figure 5.8 Simulation Waveforms for (a) Dual-control Lines latch (b) IACSL memory

5.2.5 Contributions of ACSL

Through all these modifications, ACSL inherits the fundamentals of asynchronous logic with some modifications in order to efficiently implement the charge sharing technology and maximize the benefits. Although it makes use of PFAL, a logic family in adiabatic logic, to build functional units, it is not using the overhead of power clock generator and manages to preserve energy significantly. Apart from this, two memory cell designs were introduced as well. In particular, the IACSL memory with only 4 transistors is integrated into the ACSL circuits seamlessly. It is known that high power consumption caused by dual-rail logic and comprehensive completion detectors sometimes restrict its popularity. ACSL addresses this by using charge sharing technology along with power gating technique implemented as a no cost bonus. Additionally, the realization of completion detection is simplified considerably. Moreover, other design efforts, such as wiring and layout, can also be saved significantly. Apart of the power reduction, ACSL brings other promising features in average power estimation such as input data-independency and scalability, which will be shown in the next section.

5.3 Arithmetic Units Based on ACSL

The proposed ACSL is employed to build arithmetic units, such as carry look-ahead adder[126], multiplier and Booth multiplier. A balanced structure is preferred so as to matching the duration of charge sharing between stages although unbalanced architecture could also be implemented in ACSL. In this Section, firstly, as a basic component in logic circuit design, several dual-rail one-bit full adders are compared. Secondly, the Kogge-Stone adder [127], one of the most popular parallel prefix formed carry look-ahead adders, is built based on ACSL with different memory designs, introduced in Section 5.2.4. Next, the performance of asynchronous array-based multipliers is investigated. Finally, a novel modified Booth multiplier design inspired by ACSL is proposed. All the circuits were simulated in commercially available 45nm CMOS process (GP model where V_{DD}=1V, V_{THN}=0.26V and V_{THP}=-0.34V). Input vectors in all following simulations are generated by Linear Feedback Shift Register with corresponding bit width. Synopsys HSIPICE tool is used to run the simulations and get the power consumption.

5.3.1 Ripple Carry Adder Comparison

As discussed in the last section, ACSL consumes about 50% of the total energy per calculation, while the other half is transferred to the next stage. If compared to other dual-rail dynamic logic, the total dynamic power consumption is half of the total dissipation plus the dissipation for the sharing operation. 4-bit ripple carry adder models based on four types of logic were built.

Fig.5.9 shows the effect of the supply voltage scaling from 1.2V to 0.8V on the power consumption for four 1-bit adders (from 4-bit ripple carry adders) all running at 100MHz. It can be seen that the proposed ACSL saves around 45% power compared to PFAL which runs in Non-Adiabatic mode. About 5% energy is lost during sharing. The power varies from 181nW to 405nW for ACSL. The power consumption of DDCVSL [84] is very close to PFAL (Non-Adiabatic) when the power supply is not greater than 1V. When V_{DD} is raised to 1.2 V, about 8% more power is dissipated by DDCVSL. In all cases, dual-rail Domino logic has the highest power dissipation, varying from 410nW to 1000nW.

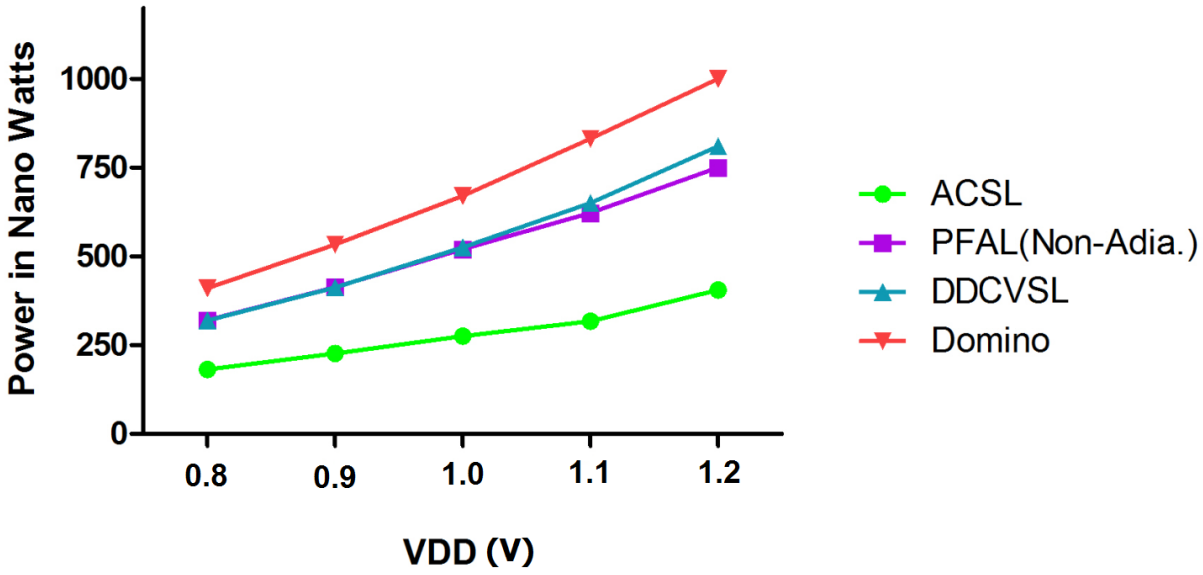


Figure 5.9 Dynamic Power Comparison of Adder Designs

Table 5.1 lists the Power-Delay Product, leakage power of four adder designs which all run at their maximum speed while V_{DD} is set to 1V. The power-delay product (PDP) of ACSL has 43% and 44% improvement compared to PFAL and DDCVSL, which is a similar saving to that of supply voltage scaling. The delay time includes charge sharing and evaluation for ASCL. For DDCVSL and Domino logic, it consists of the pre-charging and evaluation times. For PFAL (Non-Adiabatic), it is decided by the charging and discharging times. Leakage power is becoming significant when deep sub-micron process is used. For many embedded system applications including wireless sensor networks, the leakage power dominates in the total power consumption. Leakage power comparison of these four types of full adders in stand-by mode is shown in Table 5.1, where the ACSL adder has the lowest leakage power, 1.97nW while the PFAL adder consumes 10% more power. The leakage power of DDCVSL adder and Domino logic adder are 7.83nW and 8.24nW respectively, which are several times more than by using ACSL. This is due to all the internal nodes of ACSL circuits being discharged to zero in stand-by mode rather than being pre-charged to full V_{DD} for DDCVSL and Domino logic.

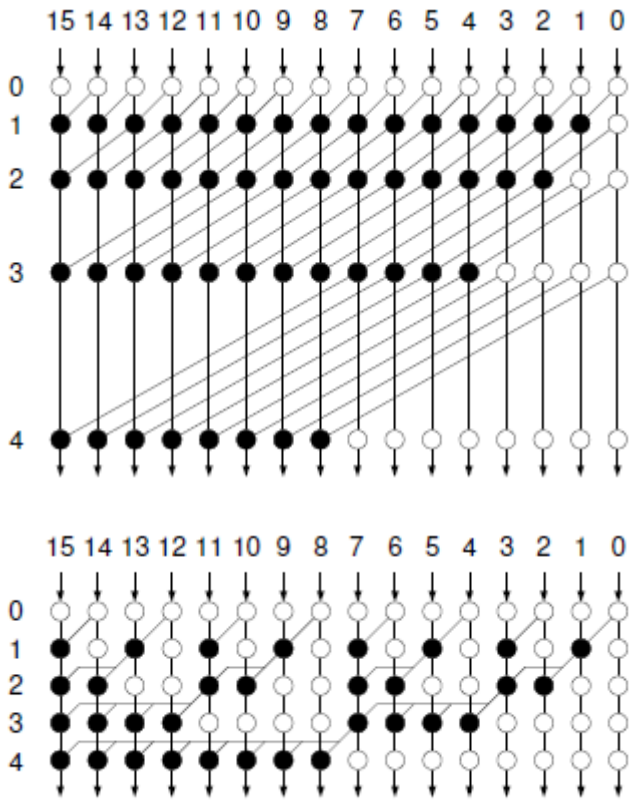
Table 5.1 Power, Speed Comparison of Four Types of Adders

V_{DD}=1V	ACSL	PFAL	DDCVSL	Domino
Power (μW)	23.3	51.4	62.3	73.4
Delay (nS)	0.124	0.099	0.083	0.09
PDP (fJ)	2.89	5.09	5.17	6.61
Power Ratio	/	+43%	+44%	+56%
Leakage (nW)	1.97	2.17	7.83	8.24

5.3.2 Kogge-Stone Adder

A parallel prefix formed carry look-ahead adder is widely used for its high speed by reducing the time needed to determine carry bits. There are several different structures for this type of adder. They are distinguished from each other by the fan-out and logic depth. The Kogge-Stone shown in Fig.5.10 (a) and Sklansky adder [128] Fig.5.10 (b) are famous models of this type of adder, which are also known as parallel-prefix adders. The former has low logic depth, high node count, minimal fan-out (only one at each node). Therefore, it sacrifices power and area consumption to achieve low propagation delay. While the latter is another situation, with higher logic depth and less nodes (high fan-out), longer calculation time and low power and area consumption.

Considering the whole structure of the circuits and their corresponding fan-out, the Kogge-Stone adder is favoured in my design due to its minimal fan-out. Moreover, the Kogge-Stone adder is a parallel prefix form carry look-ahead adder, where the execution of an operation is in parallel by small pieces, which produce the propagate bit, P_i, and the generate bit, G_i. Generally speaking, there are three main blocks of the parallel prefix adder, namely pre-calculation, carry calculation and post-calculation. All these blocks can be constructed by simple logic units, such as XOR gates, AND gates and OR gates. They have to be transformed into PFAL topology to match the ACSL operation.



(a) **(b)**

Figure 5.10 Parallel-prefix Adders (a) Kogge–Stone Adder (b) Sklansky Adder [128]

To demonstrate the efficiency of ACSL in power saving, two Kogge-Stone adders, 8-bit and 16-bit were built. For the 8-bit model, two proposed memory designs are both used. LFSR (Linear Feedback Shift Register) was used to generate pseudo-random input vectors for the adders for verification and average power estimation. Several synchronous and asynchronous carry look-ahead adder implementations [129, 130] were chosen as benchmarks. The energy, area and static power consumption of these different designs are summarized in Table 5.2 where the design combined with the proposed IACSL memory is called Improved Asynchronous Charge Sharing Logic (IACSL), while the topology using the dual-control line latch is still named as ACSL.

It is clear that the proposed IACSL adders have great energy saving compared with other designs. The energy is reduced by more than a factor of 4 in contrast with the Dual-V_{th} adder [130]. While compared to ACSL counterpart, about 10% energy reduction is achieved. In terms of static power, IACSL has superior performance, with more than 60% saving

Table 5.2 Comparison of Energy, Area and Static Power

Design	***Type/Bit***	***Input Rate (MHz)***	***Scaled Energy (fJ)***	***Static Power (nW)***	***Transistor Count***
ANT [129]	Asyn./8	100	2794	/	2250
Dual-V_t [130]	Syn./8	500	870	/	882
ACSL	Asyn./8	500	218	450	2036
IACSL	Asyn./8	500	200	280	1652
IACSL	Asyn./16	500	408	370	3587

*Scaling factor = Voltage Scaling * Energy Operation Scaling*

Voltage Scaling = (V_{DD} of the prior design/V_{DD} of the proposed design)^2

Energy Operation Scaling = Gate length of the prior design/Gate length of the proposed design

(Normally, the operand bit length is also considered in the scaling factor. As there is only one 16-bit model in the table, the effect of bit width is omitted.)

compared to the ACSL adder with same operand width, which has been demonstrated to possess ultra-low leakage property. It is interesting to see that even the 16-bit IACSL adder has lower static power dissipation than the 8-bit ACSL one. Moreover, the transistor count of the IACSL designs is also smaller than the other two asynchronous models.

Moreover, the great scalability is also demonstrated where the energy consumption of the proposed 16-bit ACSL Kogge-Stone adder is almost twice as much as that of the proposed 8-bit one. This feature could lead to easy-predictability in terms of power with the increase in input size.

5.3.3 Array-based Multiplier

The architecture of 8x8-bit array-based asynchronous integer multiplier is shown in Fig.5.11 where Stage 8 is composed of Nielsen adders [131]. Other adders in the multiplier are all based on ACSL, PFAL, DDCVSL and dual-rail Domino logic. Both IACSL and ACSL models were implemented. Each design runs at its maximum speed. The reason to choose an

Table 5.3 Power, Speed, Area Comparison of Four Integer Multipliers

V_{DD}=1V	IACSL	ACSL	PFAL	DDCVSL	Domino
Power (µW)	366	322	462	573	608
Delay (nS)	2.45	3	2.75	2.34	2.51
PDP (pJ)	0.897	0.965	1.27	1.34	1.525
Saving	/	8%	30%	34%	42%
Leakage (µW)	0.95	0.99	1.02	1.32	1.34
Transistors	4985	5546	5468	5342	5538

array-based multiplier rather than a tree-based style is due to the more balanced capacitive load contribution. Between each stage, latches are inserted and controlled following the handshake protocol. Standard static AND gates are used throughout the whole design, except at the first stage where dual-rail dynamic AND gates are deployed for the purpose of completion detection. It should be noted that the dual-control lines latches are used to connect AND gates with adders in IACSL while ACSL memories only link ACSL adders.

Table 5.3 lists the Power-Delay Product (PDP), the leakage power and the transistor count of all five designs including IACSL and ACSL. The delay time in Table 5.3 is defined as the time gap between the Request signal at the first stage and the Completion signal at the last stage. Itshould be noted that time consumed by pre-charging in DDCVSL and dual-rail Domino logic and by charge sharing in ACSL is included into the total delay time.

Regarding the dynamic power dissipation, the proposed ACSL saves 30% and 44% when compared to PFAL (Non-Adiabatic) and DDCVSL, respectively. While the IACSL consumes slightly more power due to its quicker speed, it is clear that IACSL is the second quickest among all, only slower than DDCVSL while ACSL design is a bit slower than the other four types of logic. It can be concluded that using the novel IACSL memory improves the performance substantially. The PDP of IACSL is the smallest, around 34% improvement against DDCVSL which is widely used in asynchronous designs and 30% to PFAL (Non-Adiabatic), compared to ACSL one, it is 8% better. It is interesting to see PFAL has smaller PDP than DDCVSL. To estimate the leakage power, all designs are turned into idle mode where no signal is switching. As mentioned above, due to the characteristic of ACSL, it has

the lowest leakage power. It can be seen from Table 5.3 that the IACSL and ACSL multipliers consume only 0.95uW and 0.99µW leakage power, which is more than 30% saving compared to DDCVSL and dual-rail Domino logic. It should be noted that IACSL memories are only chosen to connect with ACSL adders rather than AND gates. Thus, the leakage power reduction is not as significant as that in carry look-ahead adders, which only uses IACSL memories. Finally, the IACSL multiplier consumes the lowest area in terms of transistor numbers, more than 7% decrease in contrast with DDCVSL multiplier.

To sum up, IACSL is outstanding in almost every area; the second quickest, the most energy

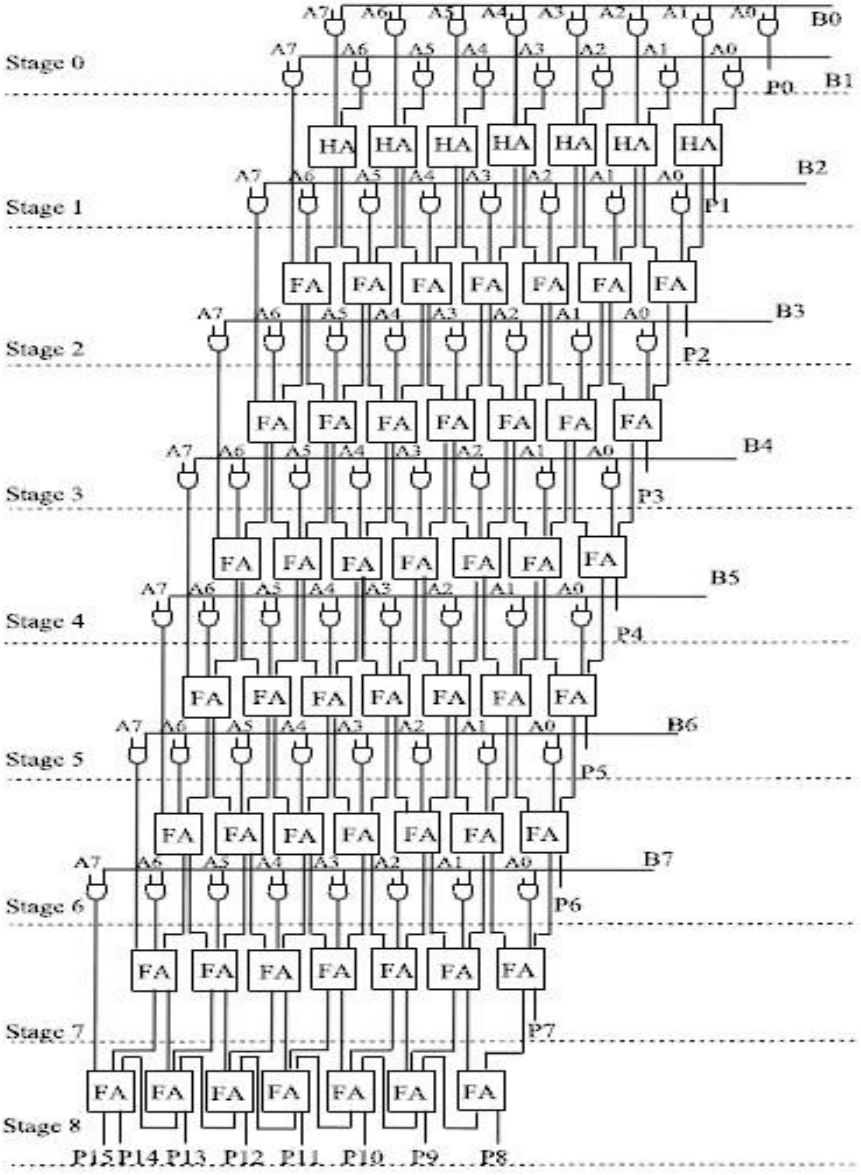


Figure 5.11 Block Diagram – 8x8-bit Multiplier

efficient and the lowest transistor consumption indicates it could serve as a quite promising alternative in asynchronous logic designs.

5.3.4 Booth Multiplier

The main architecture of a 16x16-bit Booth radix-4 array multiplier is shown in Fig.5.12. It is composed of the Partial Product Generator (PPG) and the adder block. The PPG generates Partial Products (PPs) based on the Multiplicand X and Multiplier Y as inputs. The PPs are then fed into the adder blocks. The modified Booth multiplication algorithm, which is used in this work, is explained in [126]. The main benefit of the Booth multiplier is that it uses a small number of PPs and reduces the number of rows in the adder blocks. Because the PPs are ready before the calculations in the adder block begin, there would be needless switching activity occurring during the summation by the full adders. Asynchronous logic is thus very useful in suppressing the spurious switching. At each stage the inputs are synchronized so that there will be no glitches. Therefore, the average number of switches is reduced significantly, hence consuming less power. In Fig.5.12, the first seven rows of the adder block are the *Carry Save Adders* (CSAs) [132] and the final row is the *Ripple Carry Adder* (RCA). The result of the multiplier P is represented on 32-bits.

5.3.4.1 Novel ACSL PPG Architecture

The PPG, which is the essential part to implement the Booth encoding algorithm, consists of N duplicate circuits, where N is the number of bits of the multiplier. The architecture of the proposed Booth PPG is shown in Fig.5.13(b) while the conventional PPG is exhibited in Fig.5.13 (a). I used the standard encoding part but did a small modification to the decoding circuits. A circuit called *Carry Decide* (CD) is introduced to decrease the number of unnecessary 0 to 1 switchings.

The other change to the PPG is that our new PPG only requires one set of encoding and decoding circuits instead of 8 duplicates in the conventional PPG architecture. This is due to the introduction of ACSL and a block called *Selector*. This block records a radix-4 group from the 16-bit Multiplier Y based on the current round [126]. Fig.5.14 depicts a schematic of the whole proposed PPG, including *Registers*, *Booth PPG* and the *Selector* which is composed of *AND* and *OR* gates and controlled by the signals namely *Ctrl(i)* from the ACSL

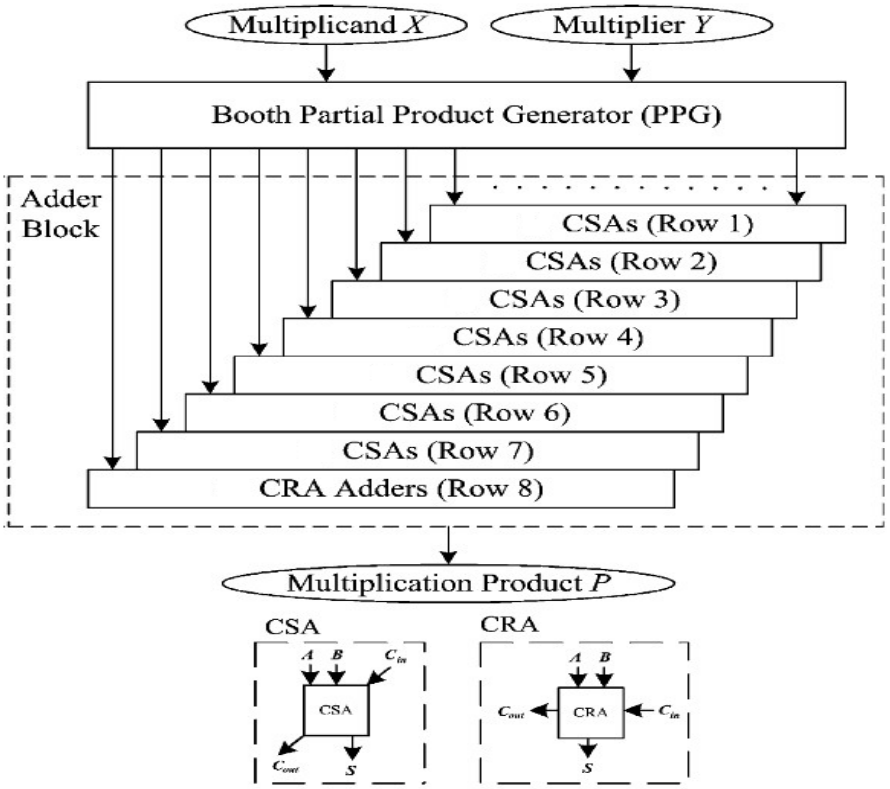
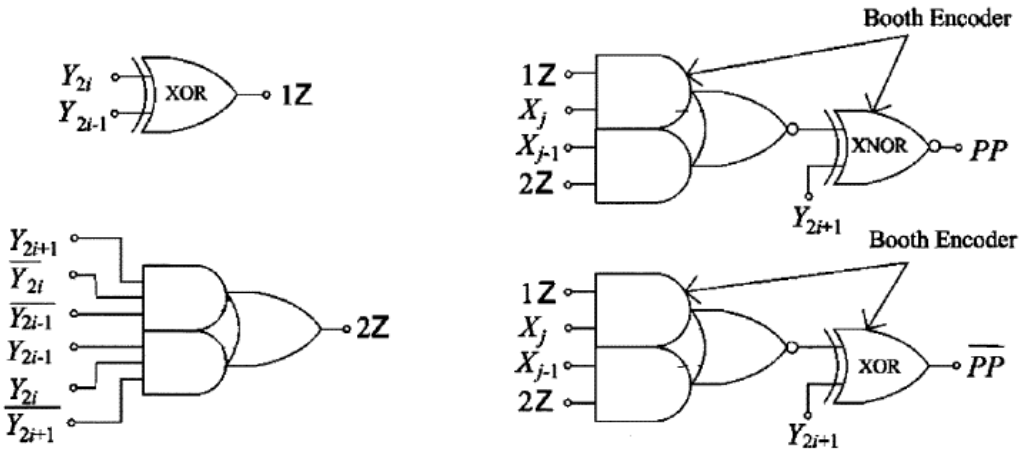
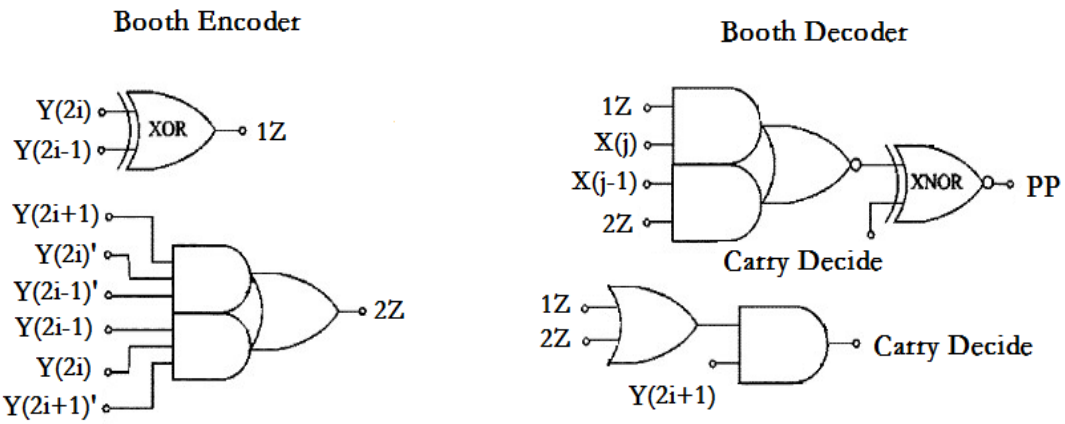


Figure 5.12 Block Diagram of a 16x16 Radix-4 Booth Multiplier



(a) Conventional Booth Partial Product Generator



(b) Proposed Booth Partial Product Generator

Figure 5.13 Two Booth Partial Product Generators

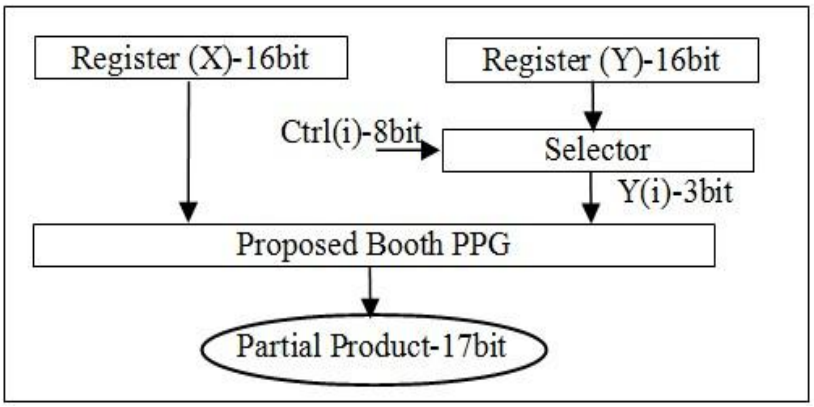


Figure 5.14 Block Diagram of Proposed PPG

Table 5.4 ACSL PPG versus Standard PPG

V_{DD}=1V	***ACSL PPG***	***Standard PPG***
Type	Radix-4	Radix-4
Technology (nm)	45	45
Dynamic Power @ 100MHz (µW)	78	83
Static Power (µW)	0.7	2.6
Transistor Count	1830	6544

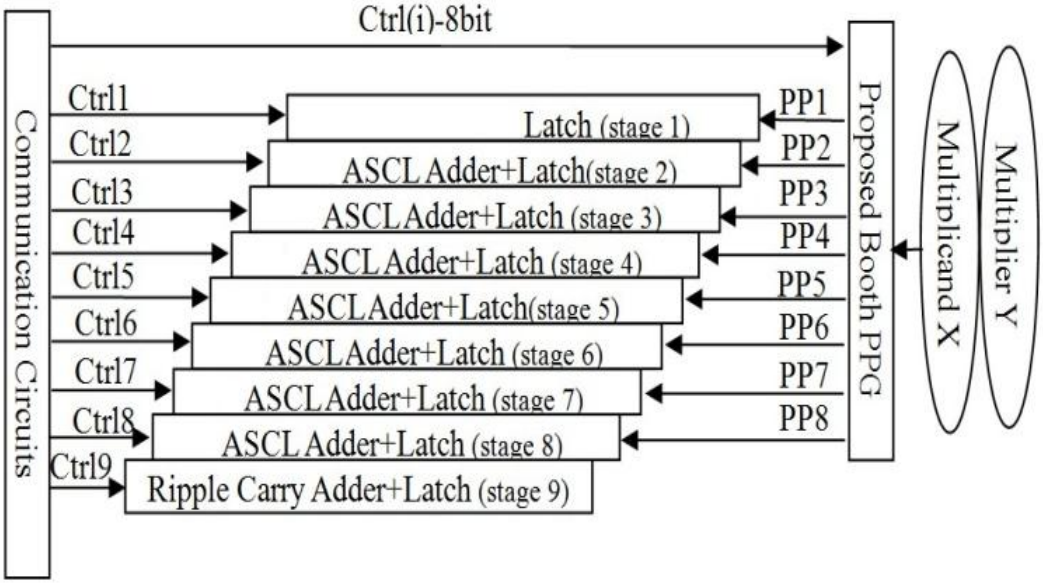


Figure 5.15 Block Diagram of ACSL Booth Multiplier

circuits (Fig.5.3). Slightly different to the protocol introduced in Section 5.2.4, there is only one control signal, *Ctrl(i)*, turned on at each time, which means there is no conflict among stages. In this way, there is only one set of *Y(2i-1)*, *Y(2i)* and *Y(2i+1)* selected by the *Selector* each time and thus only one group of Partial Products generated are fed to the corresponding ACSL full adders unlike the standard PPG which produces all PPs at once. For example, when Ctrl(2) is high and other Control signals are low correspondingly, a set of Multiplier Y, Y(3), Y(4) and Y(5) is chosen by the *Selector* and sent to the Booth Encoder.

Therefore, only one set of Booth encoders and Booth decoders is required in ACSL PPG, which could significantly reduce the area and thus static power consumption unlike the conventional PPG computes the all partial products simultaneously. However, this novel method introduces one more pipeline stage in ACSL which increase the latency a bit. Regarding the dynamic power consumption, the ACSL PPG may not have much improvement because it highly depends on the input vectors. In order to demonstrate the advantages of the proposed ACSL PPG, the standard 16-bit Radix-4 PPG is also built using 45nm CMOS process. Table5.4 summarizes the comparison between the ACSL PPG and the standard PPG in terms of dynamic power, static power and transistor count. There is little difference in dynamic power consumption. However, as expected, the ACSL PPG consumes only 0.7μW static power while the standard PPG dissipates about 3.7 times more. Moreover,

the proposed architecture only takes 1830 transistors while the conventional PPG uses 6544 transistors. More than 70% area is saved in this part particularly.

5.3.4.2 Proposed ACSL Booth Multiplier

The block diagram of the proposed 16x16-bit Booth multiplier is shown in Fig.5.15. The first *VPC* of ACSL circuits is connected with the last *VPC* of ACSL circuits through *VPC_Shr* to save as much power as possible. The ripple carry adders are used in the final stage of this multiplier. The operation of the multiplier is as follows. Firstly, the Multiplicand X and Multiplier Y are loaded into the proposed PPG, named ACSL PPG, by D-type Flip-flops. The first set of PP is then ready and fetched by Stage 1. Meanwhile, the ACSL PPG generates the second set of PP for stage 2. At stage 2, PP1 and PP2 are added by ACSL adders. The following operations can be referred to section 5.2. *VPC(i)* shares the energy with *VPC(i+1).* As mentioned above, PPs are generated one set by one set according to the value of Ctrl(i). It should be noted that the PP to be added in the current stage is generated when the previous stage is activated. For example, PP4, which is the input for stage 4, is calculated while stage 3 is computing.

A 16x16 Booth multiplier was designed using IACSL and cooperated with ACSL PPG. The multiplier was analyzed under the condition of a 1 volt supply and 27ºC temperature on 45nm CMOS technology. The input vectors were generated by 4 8-bit LFSRs. Compared to other multiplier designs in [133-135], where Hsu [134] and SPST [133] are synchronous designs, both using Booth encoding while ANS [135] works under asynchronous communication protocol and without Booth's algorithm. Also, in [135], it is claimed that the asynchronous multiplier consumes less than half of the synchronous counterpart and the Non-Booth multiplier saves 23 % of the power of the modified Booth’s algorithm.

Table 5.5 lists comprehensive comparisons among the four designs. Only ANS is with 32- bit operand width. It is worth mentioning that SPST could run up to 200 MHz while the power consumption in the table was given for 100MHz. The delay of the proposed IACSL multiplier is 6nS. The power-delay product is only 2.04pJ. Compared with the scaled PDP of other three designs, our IACSL multiplier achieves about 7 % and 24 % improvement against SPST and Hsu multipliers while 18% reduction is obtained in contrast with ANS. In terms of the area, the proposed IACSL design consumes almost the same number of transistors as that

Table 5.5 Performance Comparison with the Existing Multipliers

	IACSL	***Hsu[134]***	***SPST[133]***	***ANS[135]***
Type/Bit	Asyn. Booth/16	Syn. Booth/16	Syn. Booth/16	Asyn. Non-Booth/32
Technology (nm)	45	90	180	180
Power Supply (V)	1	1.3	1.8	1.8
Power (mW)	0.34	9	2.82	6.47
Delay (nS)	6	1	10	10
PDP (pJ)	2.04	9	28.2	64.7
Scaled PDP	2.04	2.66	2.18	2.49
Area	11820 (tr.)	0.03 (mm^2)	11028 (tr.)	/

*Scaling factor = Voltage Scaling * Energy Operation Scaling * Operand Length Ratio*

Operand Length Ratio = Bit length of the prior design/ Bit length of the proposed design

Voltage Scaling = (V_{DD} of the prior design/V_{DD} of the proposed design)^2

Energy Operation Scaling = Gate length of the prior design/Gate length of the proposed design

of SPST. Normally, asynchronous designs take more transistors than their synchronous counterparts. However, due to the ACSL PPG, a large number of transistors (and thus area) is saved.

5.4 ACSL Power Estimation

The input pattern dependence problem discussed in Chapter 3 poses significant difficulties in achieving efficient and accurate power estimation. Moreover, in synchronous circuits where the clock signal triggers the lathes, the corresponding latch power is drawn in synchrony with the clock edge. The situation is somewhat different for gates inside the combinational blocks. Even though the inputs to a combinational logic block are fed by latches/registers, the internal gates may still make several transitions before settling to the steady state. These so-called glitches lead to additional power consumption. These unwanted switches contribute typically around 20% of the total power and in some specific applications, like combinational

adders, the value could be as high as 70% [11]. Consequently, it brings a new challenge for accurate power estimation in synchronous designs. Because asynchronous circuits are speed (or delay) insensitive, especially for dual-rail handshaking protocol, hazards and glitches are thoroughly eliminated and thus high accuracy in power estimation could be expected. Previous work [136, 137] focus on a high-level behavioral description of asynchronous circuits such as Petri-Net model [138]. The problem is that the circuit behavior still depends on inputs. Otherwise, a pattern-independent approach could be an alternative, which captures the set of all possible input signal combinations by sacrificing the efficiency.

In Chapter 3, two concepts named random bag preservation and linear compositionality were introduced and demonstrated. However, ACSL is provided with easy-predictability regarding the power consumption. Recall the equation of energy consumption of ACSL:

$$Q = C_1V_1 = C_1V_2 + C_2V_2$$

$$\textit{Given that } C_1 = C_2\,,\ C_1V_1 = C_1V_2 + C_2V_2 = 2C_1V_2$$

$$V_2 = \frac{1}{2}V_1$$

$$E_{ACSL} = C(\frac{1}{2}V_{DD})^2 + C(\frac{1}{2}V_{DD})^2 = C\frac{1}{2}V_{DD}{}^2 \tag{5.3}$$

The total energy dissipation includes one evaluation event by charging V_2 from $V_{DD}/2$ to full V_{DD} and one stand-by event by discharging V_2 from $V_{DD}/2$ to ground. The dynamic energy consumption is constant, where half of the energy is transferred to the next stage. Independent of the input vectors, only one side of the function blocks get charged through *VPC_Ctrl*, then after evaluation finishes, the energy is shared through *VPC_Shr*. This is due to the dynamic dual-rail structure implemented with charge sharing technology. It should be noted that both dual-control lines latch and IACSL memory do not have the data-independent feature. For these sequential circuits, the assumption that all states are equally probable does not hold in practice.

Table 5.6 Power Summary of 8-bit ACSL Carry Look-ahead Adder

V_{DD}=1V	Test Vector 1	Test Vector 2	Test Vector 3	Average
VPC_Ctrl (µW)	3.218	3.213	3.200	3.211
Deviation from average	0.2%	0.06%	0.3%	/
VPC_Shr (µW)	1.637	1.640	1.649	1.642
Deviation from average	0.3%	0.1%	0.4%	/
Charge Sharing Efficiency	50.8%	51.0%	51.5%	51.1%
Carry_cell (nW)	19.5	20.1	19.3	19.6
Deviation from average	0.5%	2.5%	1.5%	/
8-bit Adder + LFSR (µW)	124.7	134.9	128.7	129.4
Deviation from average	3.6%	4.2%	0.5%	/

To demonstrate the predictability of ACSL, 8-bit Kogge-Stone adder introduced in last sub-section is used, which is provoked by an 8-bit LFSR. Three different sets of input vectors are thus generated, each contains 25 vectors.

Table 5.6 summarizes the dynamic power consumption of the three sections and their average values. The power of a *VPC_Ctrl*, a *VPC_Shr*, a *Carry_cell* the 8-bit adder (with LFSR), are listed specifically. It should be noted that one set of *VPC_Ctrl* and *VPC_Shr* takes charge of evaluation and charge sharing for one stage while it may supply power for several PFAL function blocks depending on the size of the circuit.

It can be seen that power consumption of *VPC_Ctrl* and *VPC_Shr* is almost unchanged where *VPC_Ctrl* consumes 3.211µW and *VPC_Shr* passes about 1.642 µW to the next stage while the maximum error of both blocks is less than 0.4%, which proves these two blocks are input-

vector independent. Meanwhile, the charge sharing efficiency is slightly over 50%, which proves the theory of ACSL. Regarding the power consumption of *Carry_cell*, it is also nearly constant despite it only consuming around 20nW per calculation. This is because most power is consumed through *VPC_Ctrl* or transferred by *VPC_Shr*. For the total power consumption of the 8-bit adder along with the LFSR, the maximum error is 4.2%, which is because of to the LFSR and the static property of memories in the circuit. Along with the scalability, ACSL is ideal for modular design and easily to allocate the corresponding power budget.

5.5 Conclusion

Due to the promising features of asynchronous logic, such as high stability, average-case performance, potential low power consumption, a novel logic style named Asynchronous Charge Sharing Logic is proposed. It has not only the advantage of asynchronous logic but also the principle of adiabatic logic. Dissimilar to adiabatic circuits, which require resonant circuits or capacitor tanks to generate a ramp-like voltage supply, introduced in Chapter 2, ACSL circuits directly use a DC voltage supply along with charge sharing technique. It successfully eliminates the overhead of those specially-designed power supplies. However, it inherits the main structure of PFAL function blocks, which is arguably the most energy-efficient logic family in adiabatic logic. In order to implement the charge sharing technology, in ACSL, there is only one stage that could be activated between two neighboring stages at each time. Hence, a modified dual-rail handshaking protocol is also introduced.

I present three prototypes in developing ACSL. The premier design is called asynchronous Positive Feedback Charge Sharing Logic (PFCSL), which employs delay elements between stages to ensure sufficient time margin for the charge sharing operation and thus is the slowest prototype. The second model is called ACSL, which has two improvements compared to the previous one. Firstly, the completion detection is determined by the value of power supply rather than dual-rail outputs. Secondly, circuits called Sharing Detectors are introduced to replace the delay elements in PFCSL to manipulate the charge sharing operation, which significantly reduces the sharing time and thus increases the performance greatly but sacrifice some efficiency in energy transferring. The final model is named Improved Asynchronous Charge Sharing Logic (IACSL). The biggest advance is to use the newly-designed IACSL memory to replace the dual-control lines latch, which could save up

to 6 transistors per latch. Compared to the dual-control lines latch, none of the handshaking signals are involved in the IACSL memory, the memory is directly controlled by outputs from PFAL function blocks. Area and power consumption, especially the static power consumption, gets improved substantially.

Several arithmetic units, such as the carry look-ahead adder, the array-based multiplier and the Booth multiplier, were designed and simulated. First, the 4-bit ACSL ripple carry adder was used to demonstrate the principle of charge sharing technology. Adders based on DDCVSL, PFAL and Domino logic were also built to compare with the ACSL adder. More than 40% improvement in PDP is gained. After that, both ACSL and IACSL were implemented in the Kogge-Stone adder and an array-based 8-bit multiplier. Considerable savings in area and power consumption are achieved. Then, the IACSL Booth multiplier was introduced with the novel ACSL PPG. Thanks to the protocol of ACSL, a large number of transistors is saved in PPG with significant leakage power consumption. Compared to other Booth multipliers, the proposed IACSL design has the lowest PDP with little increase in terms of transistor numbers. Finally, the easy-predictability of Asynchronous Charge Sharing Logic has been explored. According to the results, evaluation and charge sharing of function blocks are not affected by the input vectors. 50% charge sharing efficiency is also proved. This is a very useful feature for an efficient average-case estimation.

6 ASYNCHRONOUS SUB-THRESHOLD BIDIRECTIONAL ALU DESIGN

6.1 Introduction

An Arithmetic Logic Unit (ALU) forms one of the core parts of a processor design. From program counter update to the address calculation of a jump instruction ALU plays the major role. Deep sub-micron technology nodes raise several challenges in the digital circuit design. Several architectures of arithmetic circuits and ALU, have been proposed in the past decade. The main constraint of embedded systems is power consumption. With the goal of power optimization, various adder designs (a common component of an ALU) were proposed [139-141]. Different architectures and different logic styles were used to design these arithmetic circuits.

As introduced in the previous Chapter, asynchronous logic and corresponding circuit design is a promising design style and a number of asynchronous adder and ALU designs are reported in the literature [142, 143]. Other than adiabatic logic and asynchronous logic, Sub-threshold logic, mentioned in Chapter 1, is a methodology to lower the power consumption by trading off its performance. Moreover, an emerging logic family named reversible logic also draws attention due to its feasibility in quantum computing implementation which promises asymptotically zero-power dissipation.

A new asynchronous ALU with a ripple carry adder implemented using the logically reversible/bidirectional characteristic exhibiting ultra-low power dissipation with sub-threshold region operating point is presented in this Chapter. A brief background on sub-threshold logic and reversible logic is given in the next Section. The proposed ALU design shown in Section 6.3 presents the details of the basic components of the design from the constituent latch to the sense amplifier. Complete operation of the ALU based on a handshaking protocol is discussed in detail. Power and performance are the two main metrics used in this work. Simulation results of the ALU with size ranging from 4-bit to 32-bit are presented in Section 6.4 and the above mentioned metrics are analyzed against the conventional domino logic based adder and the reversible adder reported in [144].

6.2 Background

6.2.1 Sub-threshold Logic

Over the last decade, sub-threshold logic established itself among one of the efficient techniques for reducing the energy consumption per operation for digital circuits. The principle behind it is to save the power consumption by lowering the supply voltage, V_{DD}, to an extremely low level, below the threshold voltage, V_t, of a transistor. According to the equation of dynamic power consumption, given in Chapter 1, it directly leads to a quadratic reduction of power at the expense of increasing the gate delay significantly.

While the technology shrinks, it does not only result in smaller area but also decreases the parasitic capacitors and increases the sub-threshold current of the MOS transistors. These changes allow certain circuits to run at even lower V_{DD}; thereby, more power consumption can be saved without losing performance compared to the older process node.

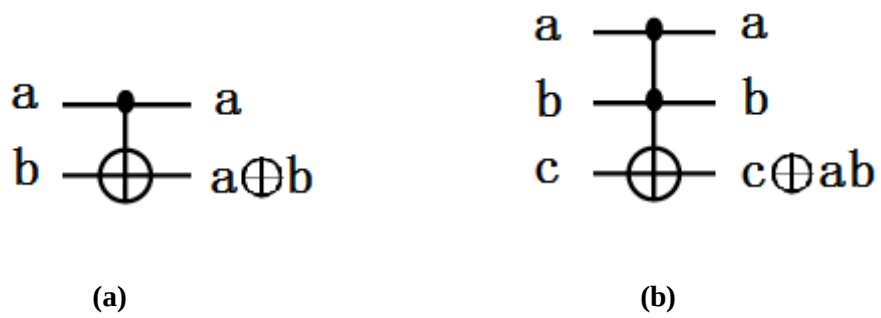
However, as mentioned before, the static power dominates the total power consumption in some deep sub-micron regions, especially for low throughput applications. Therefore, it is necessary to investigate the optimal power supply which is traded off by both dynamic and leakage power. The concept of minimum-energy point is an important characteristic of sub-threshold logic [145]. In [146], some preliminary work has been done in this area, and some methods have been proposed.

6.2.2 Reversible Logic

In the modern deep-submicron IC design, physical limits of scaling and power dissipation have become prime factors to be dealt with for efficient system design.

Landauer [147] has shown that for every bit of information lost in logic computations that are not reversible , kT *ln2 joules of heat energy is generated, where k is the Boltzmann's constant and T is the absolute temperature at which the computation is performed. Bennett [148] has shown that zero power dissipation in logic circuits is possible only if a circuit is composed of reversible logic gates. Several Reversible logic design approaches are explored lately to realize reversible designs [149, 150].

A gate is reversible, when the input and output of the circuits are bijective, i.e. if there is a



(a) **(b)**

Figure 6.1 Reversible Gates (a) Feynman Gate (b) Toffoli Gate

distinct output assignment for each distinct input. Thus, a reversible gate's inputs can be uniquely determined from its outputs. A reversible logic gate must have the same number of inputs and outputs [151]. Reversible gates are balanced, i.e. the outputs are 1 for exactly half of the number of inputs. Some of the major problems with reversible logic synthesis are [147]:

1) Fan-outs are not allowed;
2) Feedback from gate outputs to inputs are not permitted.

A logic synthesis technique using reversible gate should have the following features [147]

1) Use minimum number of garbage (unused) outputs;
2) Use minimum input constants;
3) Keep the length of cascading gates to a minimum;
4) Use minimum number of gates;

Several reversible combinational gates are widely used called the Toffolli gate, the Fredkin gate and the Feynman gate [147]. The circuit diagram of the Feynman and Toffolli gates are as shown in Fig.6.1.

6.3 Proposed ALU Design

Besides addition (or subtraction), there are other logic operations that should be performed within the ALU, such as AND, OR, etc. The proposed asynchronous ALU is able to run in three different modes namely Addition, AND and OR without using a separate AND array, OR array and the additional multiplexer. This leads to benefits in power and area reduction. To make the whole system more power efficient, the proposed ALU runs in the sub-threshold region. The core logic of the proposed adder is to take advantage of the bidirectional

characteristic of logic reversibility, and the asynchronous handshaking protocol. Thus we propose an asynchronous ALU with ADD, OR and AND operations built using the ripple carry adder. The propagation of the carry signal is the critical path of the ripple carry adder. The main principle of the proposed design is to make the carry signal available prior to the sum output and thereby increase the performance of the adder.

6.3.1 Proposed Full Adder Design

The proposed full adder uses some benefits of the pass gate based design introduced in [150] and the CMDK adder reported in [144]. The CMDK adder was built using the Majority (MAJ) gate and Un-majority and Sum (UMS) gate as shown in Fig.6.2, which contains six reversible gates in total (2 Feynman gates, 4 Toffoli gates). The symbol "⊕" in the figure represents the EXOR logic equivalent and the dots represent the logical AND. Also each level of the design represents one gate. For example, the first level of the MAJ gate is called Feynman gate (Controlled-gate), shown in Fig.6.1 (a), so is the second level gate and the third level is called Toffoli gate (Controlled-Controlled NOT gate) depicted in Fig.6.1 (b).

In [48], these two gates were realized using pass-transistor logic (PTL) based design as shown in Fig.6.3 (a) and (b). Each switch is composed of one NMOS transistor and one PMOS transistor using two complementary signals as control. It should be noted that the circuit is a dual-rail implementation with each signal represented by two lines (ex. A= (A,A'), B=(B,B')). The main reason for selecting these gates is that they provide the bidirectional/reversible operation of the adder: the inputs of the adder can act as outputs and vice versa as required.

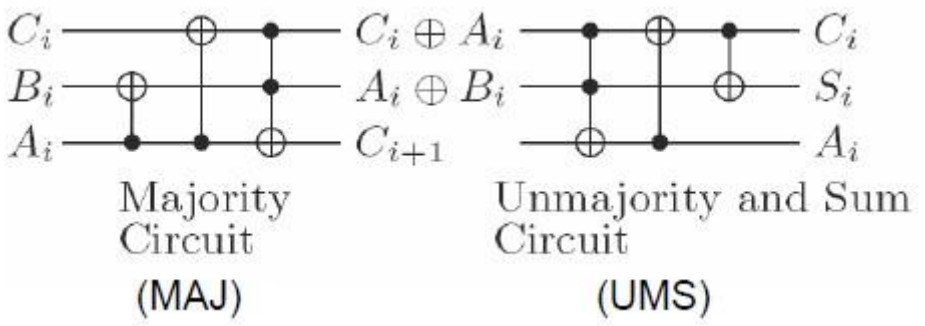
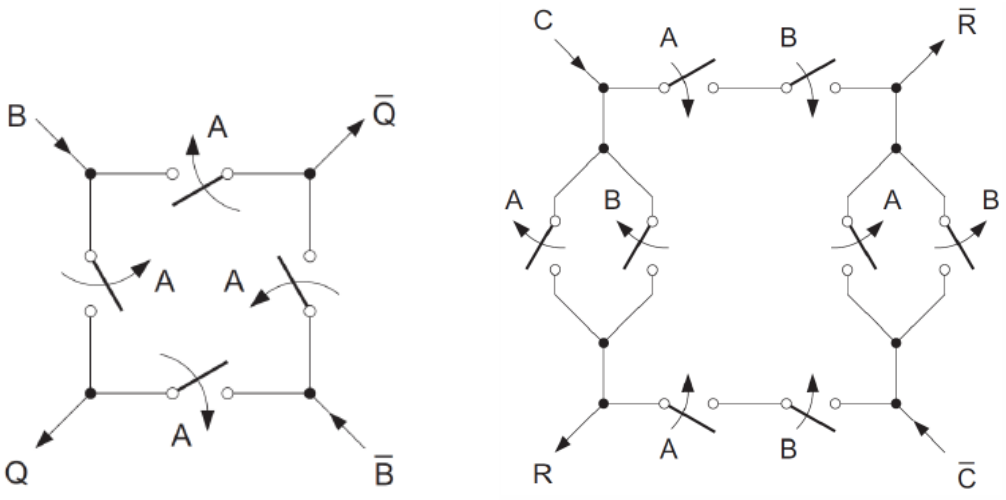


Figure 6.2 The CMDK Adder [144]



(a) **(b)**

Figure 6.3 PTL Implementation of (a) Feynman Gate (b) Toffoli Gate [48]

The architecture of the proposed adder is exhibited in Fig.6.4. It consists of 4 reversible gates (3 Toffoli gates and 1 Feynman gate). A signal called *Dir* is introduced in this design which controls the data flow direction of the adder. This full adder is designed in Asynchronous logic. The full adder works in forward and backward directions which is due to the usage of the transmission gates. The detail of the operation of the whole asynchronous ALU design will be given later. In this sub-section, I only focus on the description of the adder design.

When the full adder runs in normal mode, i.e. the forward direction (from left to right), the circuit works as the majority gate by setting the *Dir* signal at '0'. In this mode, the carry signal is generated by the circuit. In the backward mode (from right to left), *Dir* is switched to '1' and to change the functionality of the circuit. The original input A would be overwritten by the sum output. It should be noted that the reversibility of the circuit has not been affected even by the added control line which means the original inputs could still be re-generated by running the reverse operation.

An example is shown in Fig.6.5. First, in Fig.6.5 (a) inputs *A*=1 *B*=0 *Cin*=1 are being fed into the circuit while the *Dir* signal is set to 0. In this mode, the carry signal, *Cout*, is generated which is 1 in this case. Next, in Fig.6.5 (b), the control signal *Dir* is set to 1, which subsequently changes the value of *A* from 1 to 0, at this time, signal *A* could be regarded as *Sum.*

6.3.2 Proposed ALU Design

Figure 6.6 shows the architecture of a one stage ALU which consists of the proposed full adder, latch, sense amplifier, nor gate, inverter and C-element. There is no separate AND array and OR array in our ALU.

6.3.2.1 General Operation

In Fig.6.6, after the *Req* signal and *Ack* signal is low, and given that *Set* is kept low, C-element triggers *Dir* signal to high and thus enables the latches to load the data *A, B, Cin, Cin'* into the adder for the evaluation. Meanwhile, *Dir'* signal pre-charges the sense amplifier of *Sum* output to make *Complete_Sum* signal to high. Accordingly, it turns on the sense

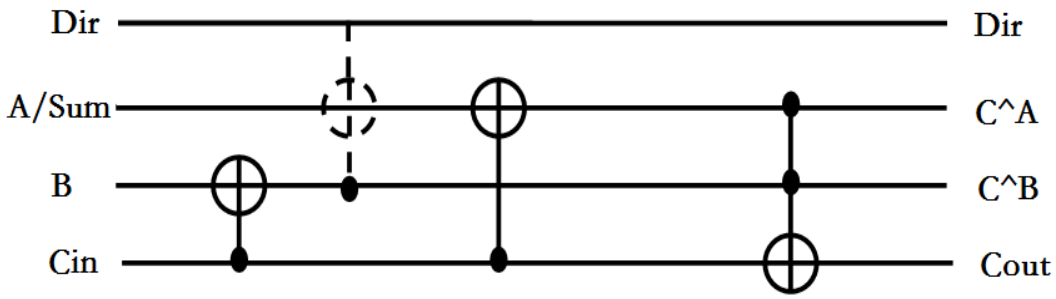
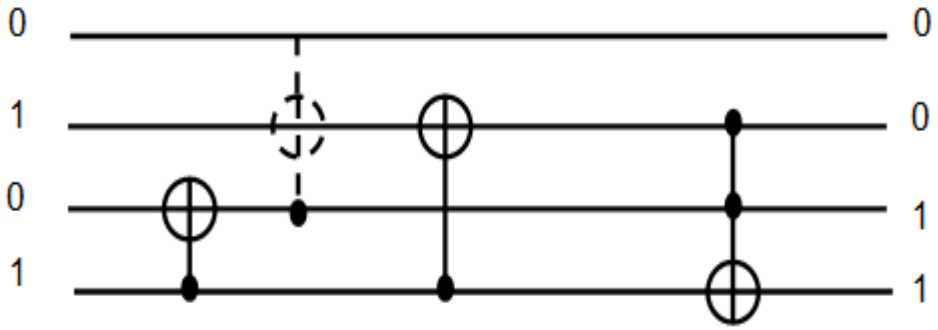
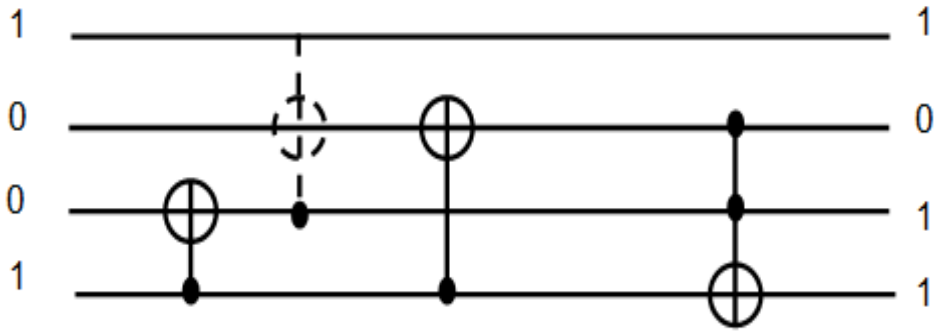


Figure 6.4 Proposed Bidirectional Full Adder



(a)



(b)

Figure 6.5 (a) Forward Computation (b) Reverse Computation

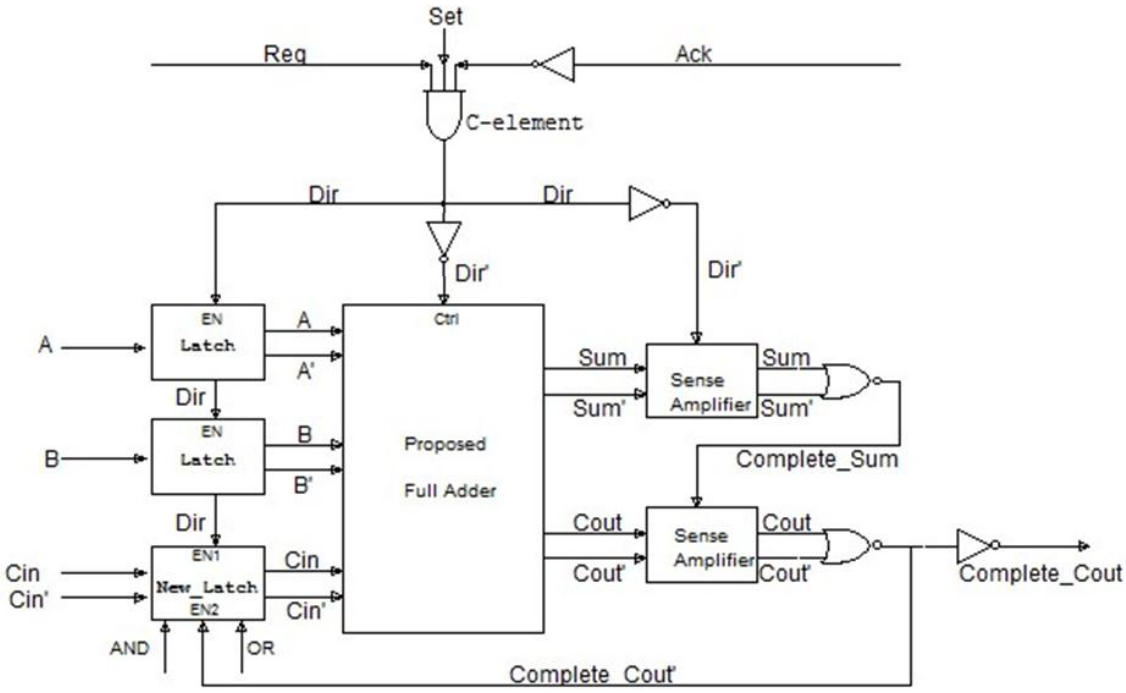


Figure 6.6 Proposed ALU Block Diagram

amplifier of *Cout* to generate the carry signals for next stage. The feedback-control signal *Complete_Cout'* becomes low to turn off the latch for *Cin* and *Cin'*. The *Complete_Cout* signal becomes high to inform the next stage that the carry signals are ready. The Sum operation takes place when *Req* is low and *Ack* is high. In this instance, *Dir'* signal turns high and thus enables the sense amplifier to sense the *Sum* and *Sum'* outputs.

6.3.2.2 Key Elements

Since the whole circuit operates in the sub-threshold region, the sense amplifier illustrated in Fig 6.7 is significant to improve the speed, and to enhance the robustness of the circuit. It should be noted that the sizing of the transistors in this amplifier should be considered [152]. Moreover, to make the full adder operate properly as OR and AND gates, the C-element and the latch for loading the pair of *Cin* signals needs to be re-designed. It is worth mentioning that the latches for loading inputs *A* and *B* are conventional. Fig.6.8 and Fig.6.9 show the schematic of the C-element and the new latch respectively. The signal *Set* is inserted into C-element to manipulate three different modes of ALU within the proposed bidirectional full adder. When *Set* is low, C-element works just like the normal one. When *Set* is switched to high value, the output of C-element, *Dir*, is forced to high value as well. With the specific combination of AND and OR signals which inserted into the Latch, the output of the latch is controllable.

Table 6.1 lists the conditions of three different modes. When Set, AND, OR signals are all 0, it behaves as an asynchronous ripple carry adder. Once *Set* becomes high, the ALU could be transferred into AND or OR mode depending on the value of AND and OR mode-controlled signals. The execution flow of three cascaded stages is illustrated in Fig. 6.10. The ALU works as the ripple carry adder in ADD mode. In contrast, the conventional architecture of ALU is shown in Fig. 6.11 where the 2-bit *Opcode* signal is used to select the data from the multiplexer. It should be noticed that, in the proposed design, the subtraction could be realized by controlling the input of B just as that depicted in Fig. 6.11.

6.4 Performance Evaluation

The proposed ALU and other benchmark designs were all implemented at transistor level using the commercially available 45nm technology. Simulation and design implementation were carried out using the Synopsys HSPICE. ALU designs of operand size ranging from 4, 8, 16 and 32 bits are designed. The input patterns were generated by LFSRs.

Table 6.2 gives the comparison of transistor count among three adder designs. The Domino full adder in the table is based on dual-rail logic, which consumes 36 transistors in total. The reversible CMDK full adder is composed of 64 transistors. The proposed adder takes 48

transistors, which saves 25% of transistors compared to the CMDK design. However, it still uses 12 more transistors than the domino adder due to the property of PTL.

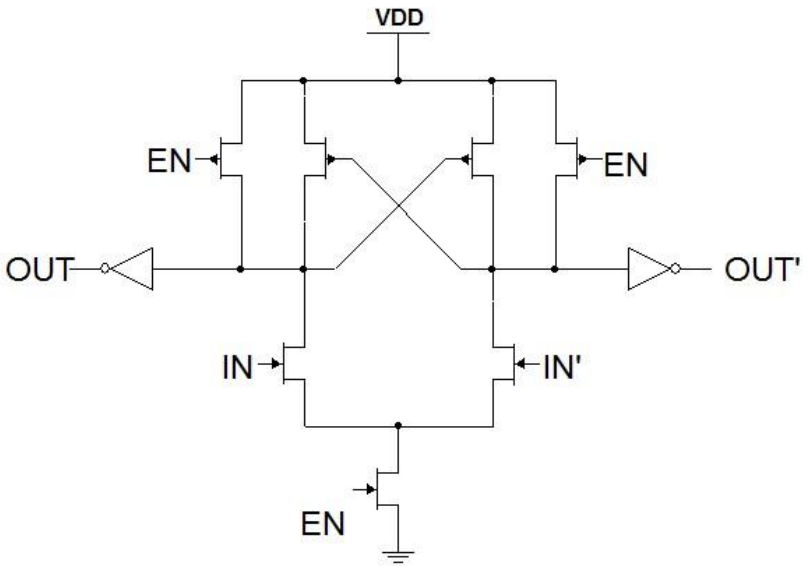


Figure 6.7 Schematic of Sense Amplifier

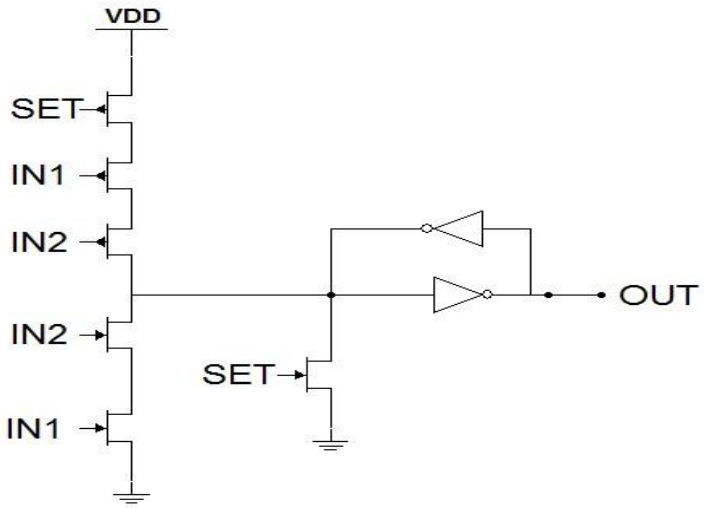


Figure 6.8 Schematic of C-element

Table 6.1 ALU Modes of Operation

SET	*AND*	*OR*	*MODE*
0	0	0	ADD
1	1	0	AND
1	0	1	OR

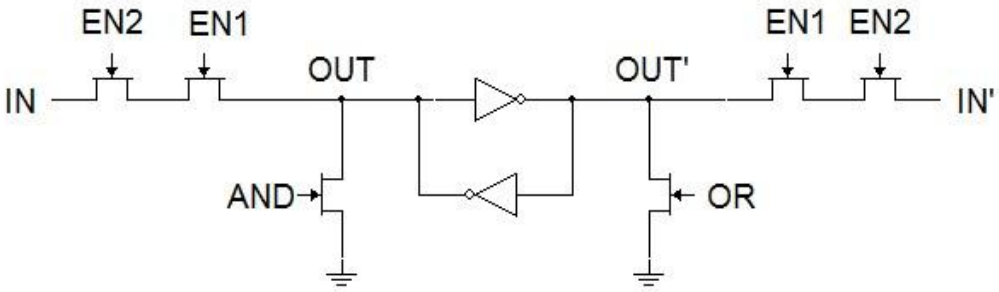


Figure 6.9 Schematic of Proposed Latch

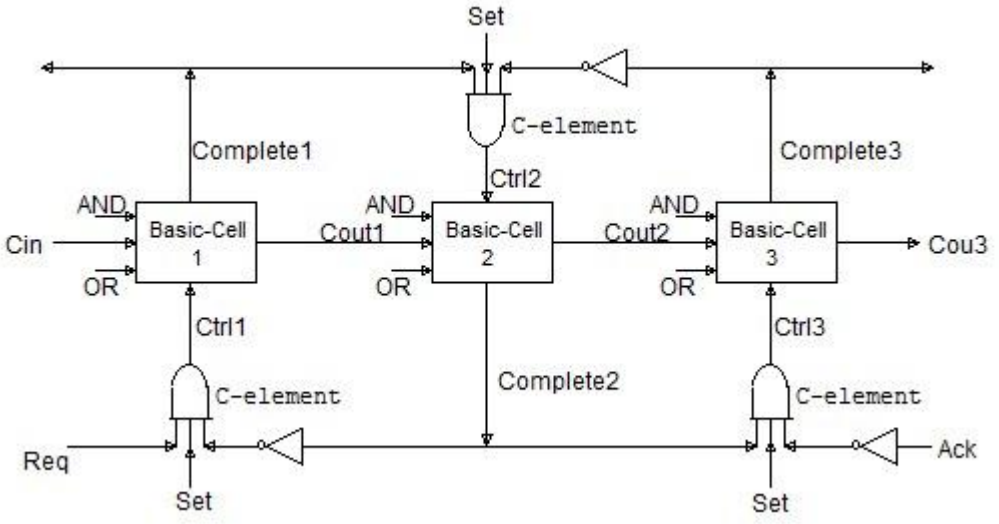


Figure 6.10 Two-stage Execution Flow

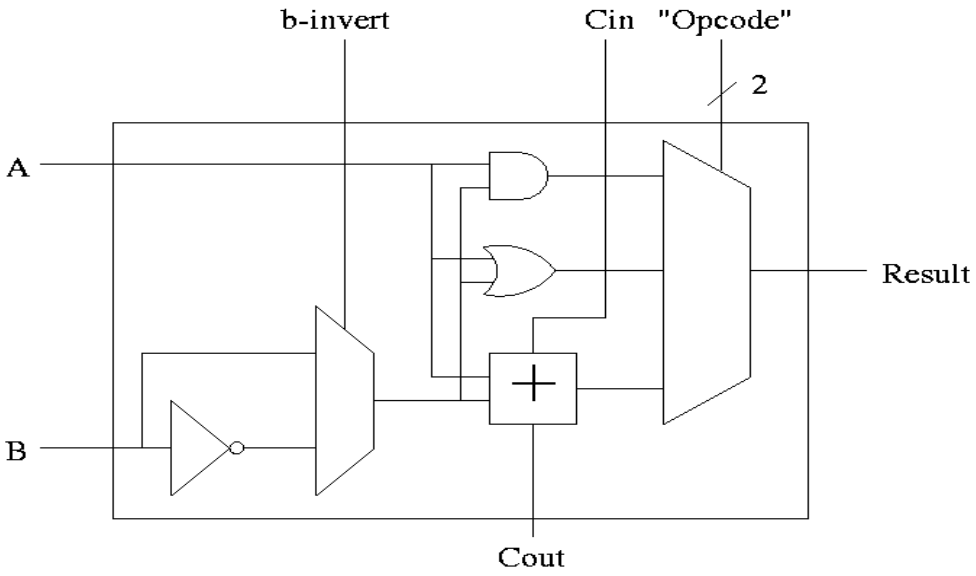


Figure 6.11 Architecture of a Conventional ALU

Table 6.2 Transistor Count Comparison

	Domino Full Adder Cell	***CMDK Full Adder Cell***	***Proposed Full Adder Cell***
Transistor Count	36	64	48

Table 6.3 gives the average power and leakage power dissipation results for the above mentioned three adders. The average and leakage power for the designs with the supply voltage sweeps from 0.2V to 0.3V, which is in the sub-threshold regime. Five scaling values of V_{DD} are selected which are 0.3V, 0.275V, 0.25V, 0.225V and 0.2 V. The average power of the domino full adder varies from 1.59nW to 3.49nW according to the change of V_{DD}. The reversible CMDK full adder dissipates average power from 2.03nW to 4.42nW for the same operating voltages. The proposed bidirectional full adder consumes the lowest power compared to other two designs, which is as low as 1.28nW, 20% and 37% less than the domino adder and the CMDK adder at 0.2V. It still 10% and 30% better than these two

Table 6.3 Dynamic and Leakage Power of Full Adders

Voltage (V)	***Power***	***Domino Full Adder Cell (nW)***	***CMDK Full Adder Cell (nW)***	***Proposed Full Adder Cell (nW)***
0.3	Average	3.49	4.42	3.12
	Leakage	1.82	2.67	1.63
0.275	Average	3.08	3.79	2.64
	Leakage	1.58	2.32	1.41
0.25	Average	2.60	3.21	2.19
	Leakage	1.35	1.98	1.20
0.225	Average	2.10	2.61	1.74
	Leakage	1.16	1.65	1.00
0.2	Average	1.59	2.03	1.28
	Leakage	1.01	1.35	0.81

adders when the V_{DD} is 0.3V. Moreover, the proposed adder also consumes the lowest leakage power throughout. The power consumption plot is shown in Fig.6.12. It is evident that the proposed full adder has constantly the lowest average and leakage power among the three designs. The front row of bars is associated with the proposed design while the middle row of bars represents the conventional domino adder and the back row represents the CDMK adder. It is interesting to see that the lower the voltage supply is, the closer the gap between leakage power and average power consumption is.

Table 6.4 summarizes the power results for the proposed ALU compared with the dynamic ALU implemented using Domino dynamic logic for the ripple carry adder and standard static CMOS gates are used for the AND operation, OR operation and multiplexer. Both designs are compared with asynchronous logic with size from 4 bits to 32 bits. The CMDK is not chosen in this comparison due to its low power efficiency. The same V_{DD} scaling values as

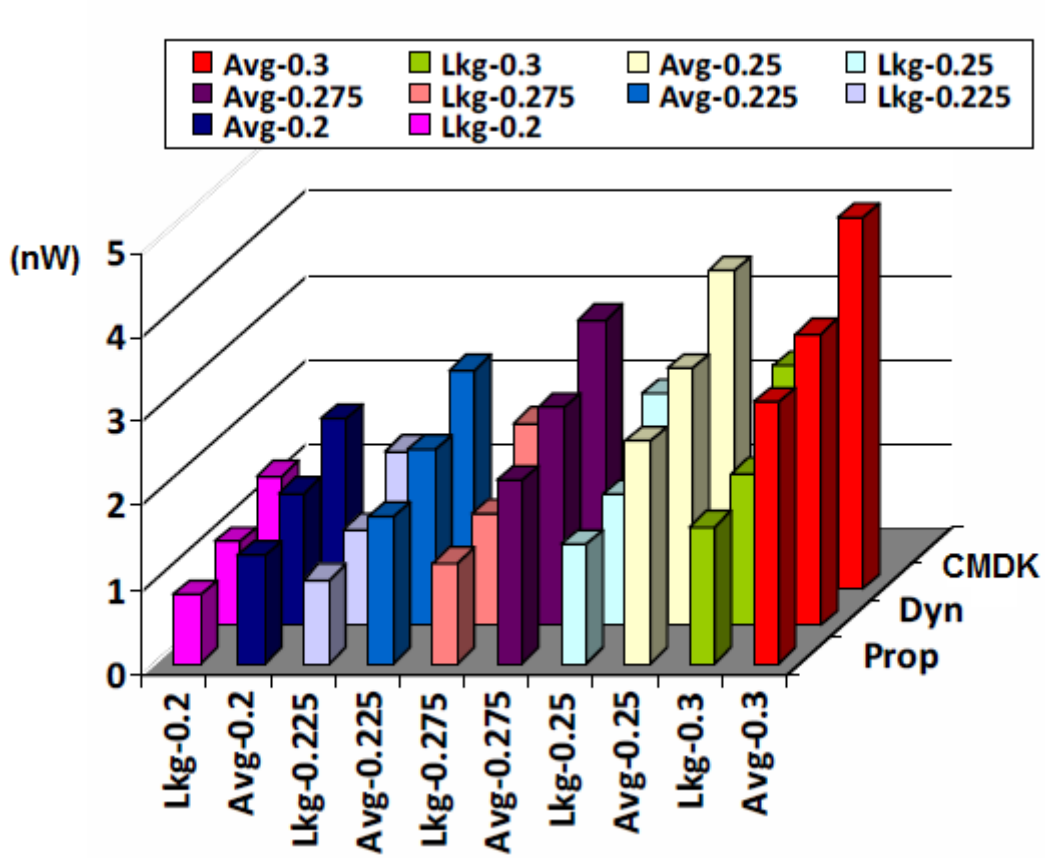


Figure 6.12 Full Adders Power Comparison

those for the adder simulation are used. The corresponding operating frequency is also reported. The power consumption for the addition operation and logic operation are given separately.

The power consumption for the addition operation in the dynamic ALU ranges from 17.6nW to 136nW for a 4-bit ALU while the proposed 4-bit ALU design consumes relatively lower power from 15.9nW to 104nW. For the ALU sizes of 8, 16 and 32 bits, the power dissipated by the dynamic ALU are in the ranges 35.1nW – 270nW, 61.3nW – 520nW and 122nW – 1029nW respectively. For the proposed corresponding ALU designs, they are 31.3nW – 201nW, 55.8nW – 379nW and 110.6nW – 761nW.

The power consumption for logic operations is just slightly higher than the addition operation for the dynamic ALU. It is because, in the dynamic ALU, three types of operation (ADD, AND, OR) are carried out in parallel, the output is determined by the multiplexer. However, the situation is different for the proposed ALU. As described in last Section, the proposed

adder could also work as an AND gate and an OR gate depending on specific control signals. Therefore, there are significant savings for the logic operations in most situations because the asynchronous communication between each adder-based stage is cut off and only forward execution is activated during logic operations rather than bidirectional execution. Nevertheless, it is noticeable that the power consumption for the logic operations is higher than the addition operations at the lowest power supply 0.2V. For 4, 8, 16 and 32 bits models, the logic operation in the dynamic ALUs consumes power in ranges of 22nW – 144nW, 44.4nW – 290nW, 88.8nW – 586nW and 165.7nW – 1151nW respectively for the V_{DD} scaling from 0.2V to 0.3V while the operating frequency varies from 7MHz to 50MHz. The proposed ALUs of the same sizes dissipate 17.5nW – 36nW, 35.1nW – 72nW, 70nW – 147nW and 133.3nW - 321nW.

The average power plot for the addition operation is depicted in Fig.6.13 and that of the logical operations is presented in Fig.6.14. The order of the rows in both figures is as follow: 4-bit, 8-bit, 16-bit and 32-bit from front to back. The label marked with P- "Voltage value" and D- "Voltage value" denotes the proposed ALU and the dynamic ALU with the particular voltage. For example, the first value P-0.2 represents the column of the power for the proposed ALU operated at 0.2V. Again, the proposed ALU shows lower power compared to the dynamic ALU.

For the addition operations, the proposed 4-bit ALU saves 8.6% - 24% average power compared to the corresponding dynamic ALU with V_{DD} scaling from 0.2V to 0.3V. From 10.8% to 26% power reduction is achieved by the proposed 8-bit ALU. For 16-bit and 32-bit model, they are 9% - 27% and 9% - 26% respectively. Regarding the logic operations, the proposed ALU consumes considerably less power than the dynamic ALU with conventional architecture. For 4-bit and 8-bit applications, the power savings are both in the range 20% - 75 %. The 16-bit ALU saves 21% to 75% power while the proposed 32-bit ALU is able to reduce the power consumption by 19% to 72%. It is obvious that the proposed ALU can save much more power when the system is switched to logical operation mode. Nevertheless, in both modes, the proposed design is power and area efficient.

Table 6.4 Power Comparison of ALUs

0.3V @ 50 MHz		***4 bit (nW)***	***8 bit (nW)***	***16 bit (nW)***	***32 bit(nW)***
Addition	Dynamic	136	270	520	1029
Operation	Proposed	104	201	379	761
Logic	Dynamic	144	290	586	1151
Operation	Proposed	36	72	147	321
0.275V @ 33 MHz		***4 bit (nW)***	***8 bit (nW)***	***16 bit (nW)***	***32 bit(nW)***
Addition	Dynamic	79	155	294	576
Operation	Proposed	64	129	246	475
Logic	Dynamic	86	172	340	681
Operation	Proposed	33	65	132	262
0.25V @ 20 MHz		***4 bit (nW)***	***8 bit (nW)***	***16 bit (nW)***	***32 bit(nW)***
Addition	Dynamic	46	91	167	318
Operation	Proposed	40	76	140	270
Logic	Dynamic	52	105	208	420
Operation	Proposed	25	50	110	226
0.225V @ 12.5 MHz		***4 bit (nW)***	***8 bit (nW)***	***16 bit (nW)***	***32 bit(nW)***
Addition	Dynamic	28	55	98	178
Operation	Proposed	26.5	51	88	163
Logic	Dynamic	33	65	130	261
Operation	Proposed	22	44	88	174
0.2V @ 7 MHz		***4 bit (nW)***	***8 bit (nW)***	***16 bit (nW)***	***32 bit(nW)***
Addition	Dynamic	17.4	35.1	61.3	122
Operation	Proposed	15.9	31.3	55.8	110.6
Logic	Dynamic	22	44.4	88.8	165.7
Operation	Proposed	17.5	35.1	70	133.3

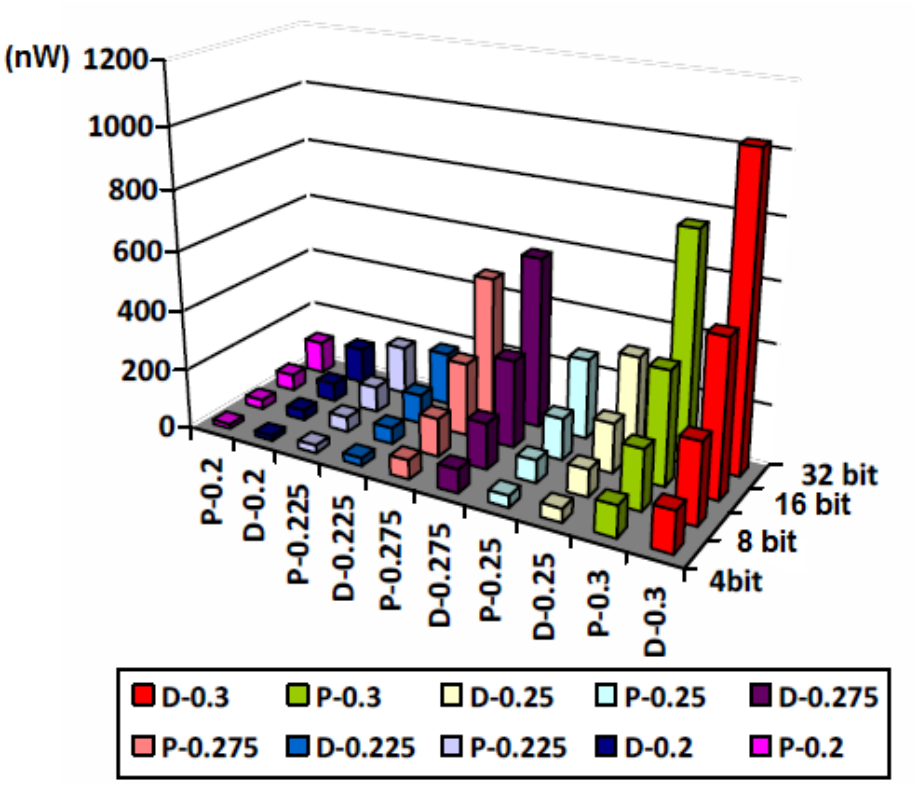


Figure 6.13 Power Comparison of Addition Operation

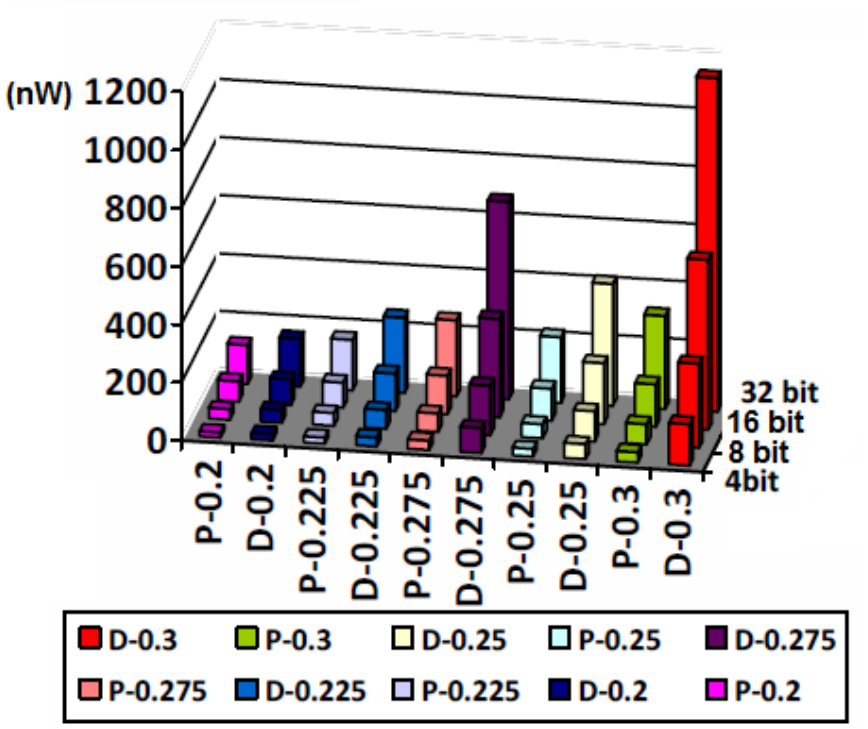
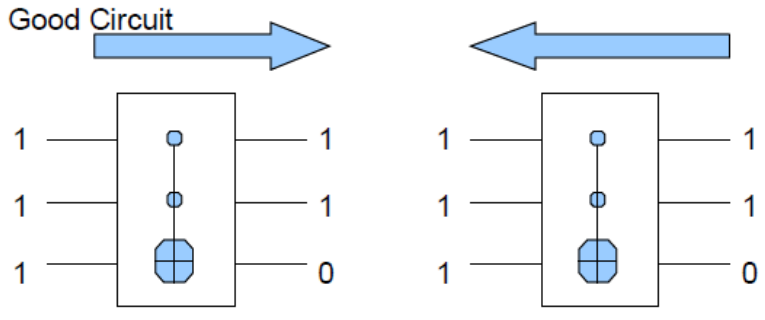


Figure 6.14 Power Comparison of Logic Operation

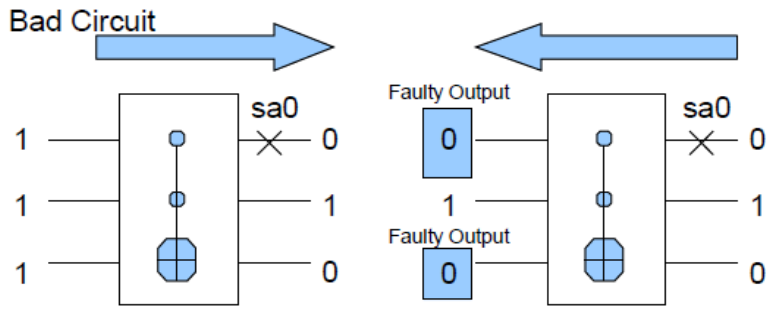
6.5 Testing For Reversible Circuits

Test generation for reversible designs is investigated in recent literature [153-155]. Several fault models, namely stuck-at fault and missing gate fault are proposed based on the technology implemented. Built-In-Self-Test (BIST) is a popular method not only for economical testing, but also for hierarchical testing and the reuse of the test logic [156]. Logic BIST uses a dedicated logic block called Built-In-Logic-Block-Observer (BILBO) for operating the BIST in different modes. BIST and reversible logic are very good candidates for an efficient, fully predictable power consumption during test as presented in Chapter 3. Each of the components of such a system satisfies the properties of randomness preservation and compositionality. A reversible BILBO architecture is proposed in [157] along with a bidirectional D latch and D flip flop, which take advantage of the bidirectional characteristic of reversible circuits. One-Hot encoded [158] test vectors are employed.

A simple stuck-at-0 fault simulation process is illustrated in Fig.6.15. The forward execution for the input pattern '111' produces an output pattern of '110' if there is no fault in the circuit, see Fig.6.15 (a) and (b). When the bidirectional D flip flop forces the circuit to run backwards, the vector '111' would be re-generated. For the faulty circuit, the forward execution with same input '111' will generate '010' and therefore the re-produced input is then '010', shown in Fig.6.15 (c) and (d). The general test procedure of reversible BILBO is exhibited in Fig.6.16. The proposed Reversible BILBO is used to build a BIST structure for the benchmarks used in [159]. The fault coverage for all the circuits is 100%, reported in [157]. The number of tests is only $n+1$ where n is the width of input bits.



(a) Forward Operation **(b) Backward Operation**



(c) Forward Operation **(d) Backward Operation**

Figure 6.15 Stuck-at-0 Fault Detection

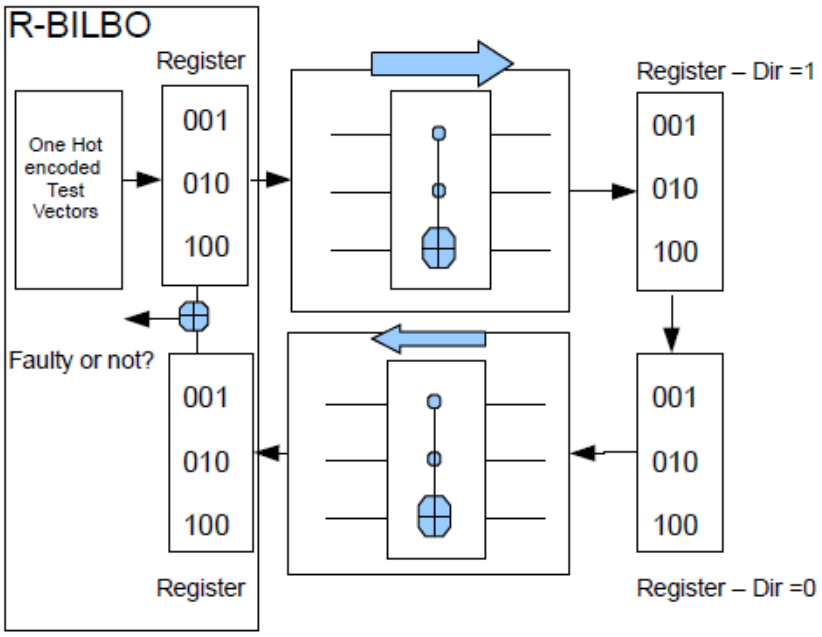


Figure 6.16 Reversible BILBO Test Procedure

6.6 Conclusion

As predicted to be the only method to avoid information loss and thus energy cost, reversible logic is getting more attention in last decade, particularly in the context of the emerging quantum computing. However, the overhead of garbage output seems inevitable in order to keep the balance of the reversible circuits. A modified reversible gate which takes the fully advantage of reversibility and bidirectional characteristic of reversible logic is proposed in this Chapter. It cooperates with the asynchronous communication protocol and works in the sub-threshold region so that the logic depth, or in other words, the latency of the circuits is

improved. Moreover, power consumption is also decreased. Besides the addition function, through manipulating the value of carryin signal for the gate, it can be switched to logic operation mode, such as AND and OR functions.

The proposed bidirectional full adder is demonstrated to be the most power efficient while compared to the CDMK reversible adder and domino adder with the voltage supply sweeping from 0.2V to 0.3V, below the threshold voltage of the transistors. 10% to 20% and 30% to 37% power savings are achieved in contrast to the domino adder and the CDMK adder, respectively. Furthermore, 4 different sizes of ALU were built using the proposed adders and the domino adders with other static logic units. Due to the multi-function capability of the novel adder, it manages to save about 10% to 26% average power for addition operations and 20% to 75% power for logical operations. Finally, an online testing methodology, reversible BILBO is introduced. For reversible circuits, 100% fault coverage is reported with high efficiency. It can also be concluded that reversible logic is promising in terms of testing because it simplifies the method of testing by duplicating in time rather than in space or something else. The testing circuits could be universal for all reversible circuits which means specialization is needed.

7 CONTRIBUTION AND FUTURE WORK

High demand of low power consumption in embedded systems, mobile devices and wireless sensor networks in particular, requires developed power management technologies in order to increase the battery life span. Meanwhile, CMOS process scaling brings new issues to be addressed, such as increased static power, reliability issues, mismatch, performance variation, etc. The vast majority of embedded systems operate at relatively low frequency (about 100MHz and below). In this region, adiabatic logic is quite efficient in terms of energy saving. In this thesis we aim to optimize the systems for the average case and we consider the asynchronous logic. Thereby, for power optimization, in this work we implemented adiabatic logic and asynchronous logic into our memory and processing unit designs. Moreover, we introduced a static approach for average power estimation for some classes of circuits. More specifically, the contributions of this thesis are as follows:

1. Static, average-case power estimation

Power estimation is significant for power budget allocation and power optimization during design process. With the increasing complexity of systems comes the challenging and time consuming process of estimating the complete power through extensive simulation. A novel power estimation technique was presented for a class of architectures, including block ciphers, reversible circuits and Modular Quantitative Analysis (MOQA) gates. It is a static approach, which requires significantly less timing and effort to predict the average dynamic power of the designs thanks to characteristics of random bag preserving and linearly (fully serial) compositionality. The simulation results showed great accuracy of the introduced theory and methodology. Additionally, an efficient algorithm called Loop Input Set was used to generate test vectors for the circuits. In contrast to the conventional complete IO-set, as many $(2^n \cdot 2^n!)–(2^n \cdot 2^n \cdot 2)$ patterns can be saved without losing accuracy.

2. Ultra low power memory cell design

Portable devices with limited battery-life require low standby power processors and memory. Often, embedded static random access memory (SRAM) arrays can be the dominant part of the whole static power consumption and also occupied chip area; thus, minimization of memory power is a crucial area of concern for today’s IC designers. Two new SRAM cells

were proposed which are composed of 8 transistors and 9 transistors respectively. Both of the proposed designs improved the write ability and read ability compared to the conventional 6T SRAM cell. Adiabatic logic was used in the proposed designs in order to achieve ultra-low dynamic power with 90% savings when compared to other popular models. Even in non-adiabatic mode, our SRAM cells could also save up to 50% energy for both write and read operations. With aggressive technology scaling, process variation was also taken into account during the simulation along with temperature variation. About 90% leakage power was saved.

3. Asynchronous Charge Sharing Logic

Global clock distribution schemes face big challenges in terms of power and timing issues, while asynchronous logic emerges as a potential alternative due to its localized communication. Also, due to the uncontrollable parameter variations across a chip, the asynchronous schemes are more robust than the fully synchronous solutions which are optimized for the worst case. A novel logic family named Asynchronous Charge Sharing Logic (ACSL) was proposed along with the design of all crucial components design. The robust 4-phase dual-rail protocol was preferred in ACSL with slight modification. To conquer the high power consumption brought by high switch activity and high overhead in terms of completion detection in conventional asynchronous dual-rail circuits, ACSL addressed it by implementing a charge sharing technology and also a simplified completion detection scheme. Apart from the power reduction, ACSL brought another promising feature in average power estimation called input data-independency where this characteristic could make power estimation effortless. Multiple architectures were built based on ACSL, including Kogge-Stone adder, array-based multiplier and Booth multiplier. Considerable savings in dynamic and static power consumption and area were achieved. Finally, the easy-predictability of Asynchronous Charge Sharing Logic had been explored. According to the results, evaluation and charge sharing of function blocks were not affected by the input vectors. 50% charge sharing efficiency was also proved.

4. Asynchronous sub-threshold bidirectional ALU design

Sub-threshold logic is a methodology to lower the power consumption by trading off its performance. Moreover, an emerging logic family named reversible logic draws attention due to its feasibility in quantum computing implementations which promises asymptotically zero-

power dissipation. However, the overhead of garbage output seems inevitable in order to keep the balance of the reversible circuits. A modified reversible full adder which could also work as logical operation units (i.e. AND and OR gate) was presented. A new asynchronous ALU built on this adder without individual logic array exhibiting ultra-low power dissipation with sub-threshold region operating point was proposed. 10% to 20% and 30% to 37% power savings were achieved in contrast to the domino adder and the CDMK adder, respectively, with the voltage supply sweeping from 0.2V to 0.3V. Furthermore, 4 different sizes of ALU were built using the proposed adders and the domino adders with other static logic units. Due to the multi-function capability of the novel adder, it managed to save about 10% to 26% average power for addition operations and 20% to 75% power for logical operations. Finally, an online testing methodology, reversible BILBO is introduced. For reversible circuits, 100% fault coverage is reported with high efficiency.

7.1 Future Work

With minor modifications to the existing work, and with extension and elaboration of the ideas presented, some possible future work can be summarized as follows:

- *Design of peripheral circuits of the memory cells.*

  To complete the memory design, peripheral circuits for write and read operation are required. As for the proposed memory cells, one additional operation called sharing is inserted. The timing of the control signals need to be taken great care of. A specific circuit which could generate these signals in a certain order is then demanded.

- *Further development of ACSL to expand the number of applications.*

  ACSL is currently only implemented in the design of arithmetic logic units while it could be potentially employed in other circuits. Furthermore, the idea can be realized globally throughout the entire processor design.

- *Integration of ACSL and memory into low power processor designs.*

  As the memory cells and ACSL have been developed, the integration of these two concepts into an ultra low power processor design, specifically with very low leakage,

is foreseeable. The memory cell designs could be used to replace the register file in the processor although the decoder and peripheral circuits require re-design. For ACSL, it is currently implemented in the data path. In the future, the idea may be expanded throughout the entire processor design so that each individual block can share the energy, such as data paths, decoders and other elements of the processor.

- *Full design of adiabatic WSN.*

  Adiabatic logic plays a significant role in this research. It will be worthwhile applying the logic into the designs of WSN whose primary constraint is power consumption. One issue regarding the power clock generator is that it operates periodically, while in practice, most WSNs are event-triggered, thereby some modification is needed. The possible combination of ACSL and adiabatic logic seems promising.

REFERENCES

1. Barr, M. and A. Massa, *Programming embedded systems: with C and GNU development tools*. 2006: O'Reilly Media, Incorporated.

2. Wolf, W.H., *Hardware-software co-design of embedded systems [and prolog].* Proceedings of the IEEE, 1994. **82**(7): p. 967-989.

3. Stankovic, J.A., *Wireless sensor networks.* Computer, 2008. **41**(10): p. 92-95.

4. Polastre, J., R. Szewczyk, A. Mainwaring, D. Culler, and J. Anderson, *Analysis of wireless sensor networks for habitat monitoring.* Wireless sensor networks, 2004: p. 399-423.

5. Rahimi, M., H. Shah, G.S. Sukhatme, J. Heideman, and D. Estrin. *Studying the feasibility of energy harvesting in a mobile sensor network*. in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*. 2003: IEEE.19-24

6. Horowitz, M., E. Alon, D. Patil, S. Naffziger, R. Kumar, and K. Bernstein. *Scaling, power, and the future of CMOS*. in *Electron Devices Meeting, 2005. IEDM Technical Digest. IEEE International*. 2005: IEEE.7 pp.-15

7. Benini, L., A. Bogliolo, and G. De Micheli, *A survey of design techniques for system-level dynamic power management.* Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2000. **8**(3): p. 299-316.

8. Roy, K., S. Mukhopadhyay, and H. Mahmoodi-Meimand, *Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits.* Proceedings of the IEEE, 2003. **91**(2): p. 305-327.

9. Royannez, P., H. Mair, F. Dahan, M. Wagner, M. Streeter, L. Bouetel, J. Blasquez, H. Clasen, G. Semino, and J. Dong. *90nm low leakage SoC design techniques for wireless applications*. in *Solid-State Circuits Conference, 2005. Digest of Technical Papers. ISSCC. 2005 IEEE International*. 2005: IEEE.138-589

10. Taur, Y., D.A. Buchanan, W. Chen, D.J. Frank, K.E. Ismail, S.H. Lo, G.A. Sai-Halasz, R.G. Viswanathan, H.J.C. Wann, and S.J. Wind, *CMOS scaling into the nanometer regime.* Proceedings of the IEEE, 1997. **85**(4): p. 486-504.

11. Najm, F.N., *A survey of power estimation techniques in VLSI circuits.* Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 1994. **2**(4): p. 446-455.

12. Szpankowski, W., *Average case analysis of algorithms*. 2010: Chapman & Hall/CRC.

13. Henzinger, T.A., *Two challenges in embedded systems design: predictability and robustness.* Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 2008. **366**(1881): p. 3727-3736.

14. Kim, N.S., T. Austin, D. Baauw, T. Mudge, K. Flautner, J.S. Hu, M.J. Irwin, M. Kandemir, and V. Narayanan, *Leakage current: Moore's law meets static power.* Computer, 2003. **36**(12): p. 68-75.

15. Yoo, C., *A CMOS buffer without short-circuit power consumption.* Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on, 2000. **47**(9): p. 935-937.

16. Liu, D. and C. Svensson, *Power consumption estimation in CMOS VLSI chips.* Solid-State Circuits, IEEE Journal of, 1994. **29**(6): p. 663-670.

17. Kaeslin, H., *Digital integrated circuit design: from VLSI architectures to CMOS fabrication*. 2008: Cambridge University Press.

18. Frank, D.J., *Power-constrained CMOS scaling limits.* IBM Journal of Research and Development, 2002. **46**(2.3): p. 235-244.

19. Rao, R., A. Srivastava, D. Blaauw, and D. Sylvester, *Statistical analysis of subthreshold leakage current for VLSI circuits.* Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2004. **12**(2): p. 131-139.

20. Pouwelse, J., K. Langendoen, and H. Sips. *Dynamic voltage scaling on a low-power microprocessor*. in *Proceedings of the 7th annual international conference on Mobile computing and networking*. 2001: ACM.251-259

21. Soeleman, H., K. Roy, and B.C. Paul, *Robust subthreshold logic for ultra-low power operation.* Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2001. **9**(1): p. 90-99.

22. Münch, M., B. Wurth, R. Mehra, J. Sproch, and N. Wehn. *Automating RT-level operand isolation to minimize power consumption in datapaths*. in *Proceedings of the conference on Design, automation and test in Europe*. 2000: ACM.624-633

23. Alidina, M., J. Monteiro, S. Devadas, A. Ghosh, and M. Papaefthymiou, *Precomputation-based sequential logic optimization for low power.* Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 1994. **2**(4): p. 426-436.

24. Devadas, S. and S. Malik. *A survey of optimization techniques targeting low power VLSI circuits*. in *Proceedings of the 32nd annual ACM/IEEE Design Automation Conference*. 1995: ACM.242-247

25. Sotiriadis, P.P. and A. Chandrakasan. *Low power bus coding techniques considering inter-wire capacitances*. in *Custom Integrated Circuits Conference, 2000. CICC. Proceedings of the IEEE 2000*. 2000: IEEE.507-510

26. Wu, Q., M. Pedram, and X. Wu, *Clock-gating and its application to low power design of sequential circuits.* Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on, 2000. **47**(3): p. 415-420.

27. Donno, M., A. Ivaldi, L. Benini, and E. Macii. *Clock-tree power optimization based on RTL clock-gating*. in *Design Automation Conference, 2003. Proceedings*. 2003: IEEE.622-627

28. Renaudin, M., *Asynchronous circuits and systems: a promising design alternative.* Microelectronic engineering, 2000. **54**(1): p. 133-149.

29. Athas, W.C., L.J. Svensson, J.G. Koller, N. Tzartzanis, and E. Ying-Chin Chou, *Low-power digital systems based on adiabatic-switching principles.* Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 1994. **2**(4): p. 398-407.

30. Martin, A.J., A. Lines, R. Manohar, M. Nystrom, P. Penzes, R. Southworth, U. Cummings, and T.K. Lee. *The design of an asynchronous MIPS R3000 microprocessor*. in *Advanced Research in VLSI, 1997. Proceedings., Seventeenth Conference on*. 1997: IEEE.164-181

31. Furber, S., D. Edwards, and J. Garside. *AMULET3: a 100 MIPS asynchronous embedded processor*. in *Computer Design, 2000. Proceedings. 2000 International Conference on*. 2000: IEEE.329-334

32. Kelly IV, C., V. Ekanayake, and R. Manohar. *SNAP: A sensor-network asynchronous processor*. in *Asynchronous Circuits and Systems, 2003. Proceedings. Ninth International Symposium on*. 2003: IEEE.24-33

33. Necchi, L., L. Lavagno, D. Pandini, and L. Vanzago. *An ultra-low energy asynchronous processor for Wireless Sensor Networks*. in *Asynchronous Circuits and Systems, 2006. 12th IEEE International Symposium on*. 2006: IEEE.8 pp.-85

34. Lines, A., *Asynchronous interconnect for synchronous SoC design.* Micro, IEEE, 2004. **24**(1): p. 32-41.

35. Moore, S., G. Taylor, R. Mullins, and P. Robinson. *Point to point GALS interconnect*. in *Asynchronous Circuits and Systems, 2002. Proceedings. Eighth International Symposium on*. 2002: IEEE.69-75

36. Kang, S. and K. Lee, *Emerging materials and devices in spintronic integrated circuits for energy-smart mobile computing and connectivity.* Acta Materialia, 2013. **61**(3): p. 952-973.

37. Keshavarzi, A., S. Ma, S. Narendra, B. Bloechel, K. Mistry, T. Ghani, S. Borkar, and V. De. *Effectiveness of reverse body bias for leakage control in scaled dual Vt CMOS ICs*. in *Low Power Electronics and Design, International Symposium on, 2001.* 2001: IEEE.207-212

38. Hu, Z., A. Buyuktosunoglu, V. Srinivasan, V. Zyuban, H. Jacobson, and P. Bose. *Microarchitectural techniques for power gating of execution units*. in *Proceedings of the 2004 international symposium on Low power electronics and design*. 2004: ACM.32-37

39. Gusev, E., D. Buchanan, E. Cartier, A. Kumar, D. DiMaria, S. Guha, A. Callegari, S. Zafar, P. Jamison, and D. Neumayer. *Ultrathin high-K gate stacks for advanced CMOS devices*. in *Electron Devices Meeting, 2001. IEDM'01. Technical Digest. International*. 2001: IEEE.20.21. 21-20.21. 24

40. Lisoni, J.G., L. Breuil, L. Nyns, P. Blomme, and J. Van Houdt, *High-k gadolinium and aluminum scandates for hybrid floating gate NAND Flash.* Microelectronic Engineering, 2013.

41. Schlom, D.G., S. Guha, and S. Datta, *Gate oxides beyond SiO2.* MRS bulletin, 2008. **33**(11): p. 1017-1025.

42. Buyuksahin, K.M. and F.N. Najm. *High-level power estimation with interconnect effects*. in *Low Power Electronics and Design, 2000. ISLPED'00. Proceedings of the 2000 International Symposium on*. 2000: IEEE.197-202

43. Carbone, P. and D. Petri, *Average power estimation under nonsinusoidal conditions.* Instrumentation and Measurement, IEEE Transactions on, 2000. **49**(2): p. 333-336.

44. Lee, D., D. Blaauw, and D. Sylvester, *Gate oxide leakage current analysis and reduction for VLSI circuits.* Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2004. **12**(2): p. 155-166.

45. Allan, A., D. Edenfeld, W.H. Joyner Jr, A.B. Kahng, M. Rodgers, and Y. Zorian, *2001 technology roadmap for semiconductors.* Computer, 2002. **35**(1): p. 42-53.

46. Burg, A.P., P. Meinerzhagen, A. Dogan, J. Constantin, M.S. Ali, G. Karakonstantis, D. Atienza, and L. Benini, *Near-and Sub-Threshold Design for Ultra-Low-Power Embedded Systems.* 2013.

47. Vieri, C.J., *Pendulum: A reversible computer architecture*. 1995, Citeseer.

48. De Vos, A., *Reversible computer hardware.* Electronic Notes in Theoretical Computer Science, 2010. **253**(6): p. 17-22.

49. Denker, J.S. *A review of adiabatic computing*. in *Low power electronics, 1994. Digest of technical papers., IEEE symposium*. 1994: IEEE.94-97

50. Bennett, C.H., *Logical reversibility of computation.* IBM Journal of Research and Development, 1973. **17**(6): p. 525-532.

51. Deutsch, D., *Quantum theory, the Church-Turing principle and the universal quantum computer.* Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences, 1985. **400**(1818): p. 97-117.

52. Shor, P.W., *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer.* SIAM Journal on Computing, 1997. **26**(5): p. 1484-1509.

53. Arsalan, M. and M. Shams. *Charge-recovery power clock generators for adiabatic logic circuits*. in *VLSI Design, 2005. 18th International Conference on*. 2005: IEEE.171-174

54. Nakata, S., *Adiabatic charging reversible logic using a switched capacitor regenerator.* IEICE transactions on electronics, 2004. **87**(11): p. 1837-1846.

55. Amirante, E., A. Bargagli-Stoffi, J. Fischer, G. Iannaccone, and D. Schmitt-Landsiedel. *Variations of the power dissipation in adiabatic logic gates*. in *Proceedings of the 11th International Workshop on Power And Timing Modeling, Optimization and Simulation, PATMOS*. 2001.9.1

56. Oklobdzija, V.G., D. Maksimovic, and F. Lin, *Pass-transistor adiabatic logic using single power-clock supply.* Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on, 1997. **44**(10): p. 842-846.

57. Yong, M. and D.K. Jeong, *An efficient charge recovery logic circuit.* IEICE transactions on electronics, 1996. **79**(7): p. 925-933.

58. Dickinson, A.G. and J.S. Denker, *Adiabatic dynamic logic.* Solid-State Circuits, IEEE Journal of, 1995. **30**(3): p. 311-315.

59. Vetuli, A., S. Pascoli, and L. Reyneri, *Positive feedback in adiabatic logic.* Electronics Letters, 1996. **32**(20): p. 1867-1869.

60. Teichmann, P., *Adiabatic logic: future trend and system level perspective*. Vol. 34. 2011: Springer.

61. Joonho, L., K. Dong-Gyu, and C. Soo-Ik, *Reversible energy recovery logic circuits and its 8-phase clocked power generator for ultra-low-power applications.* IEICE transactions on electronics, 1999. **82**(4): p. 646-653.

62. Mahmoodi-Meinnand, H., A. Afzali-Kusha, and M. Nourani. *Adiabatic carry look-ahead adder with efficient power clock generator*. in *Circuits, Devices and Systems, IEE Proceedings-*. 2001: IET.229-234

63. Gojman, B., *Adiabatic Logic.* CalTech University, CA, 2004. **8**.

64. Kramer, A., J. Denker, B. Flower, and J. Moroney. *2nd order adiabatic computation with 2N-2P and 2N-2N2P logic circuits*. in *Proceedings of the 1995 international symposium on Low power design*. 1995: ACM.191-196

65. Khatir, M., A. Moradi, A. Ejlali, M.T. Manzuri Shalmani, and M. Salmasizadeh. *A secure and low-energy logic style using charge recovery approach*. in *Low Power Electronics and Design (ISLPED), 2008 ACM/IEEE International Symposium on*. 2008: IEEE.259-264

66. Chanda, M., A. Dandapat, and H. Rahaman. *Ultra low-power sequential circuit implementation by a Quasi-Static Single phase Adiabatic Dynamic Logic (SPADL)*. in *TENCON 2009-2009 IEEE Region 10 Conference*. 2009: IEEE.1-5

67. Lau, K. and F. Liu, *Four-phase improved adiabatic pseudo-domino logic.* Electronics Letters, 1998. **34**(4): p. 343-344.

68. Maksimovic, D., V.G. Oklobdzija, B. Nikolic, and K.W. Current, *Clocked CMOS adiabatic logic with integrated single-phase power-clock supply.* Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2000. **8**(4): p. 460-463.

69. Teichmann, P., J. Fischer, S. Henzler, E. Amirante, and D. Schmitt-Landsiedel, *Power-clock gating in adiabatic logic circuits.* Integrated Circuit and System Design. Power and Timing Modeling, Optimization and Simulation, 2005: p. 909-909.

70. Ye, Y., K. Roy, and G.I. Stamoulis. *Quasi-static energy recovery logic and supply-clock generation circuits*. in *Proceedings of the 1997 international symposium on Low power electronics and design*. 1997: ACM.96-99

71. Kim, S., C.H. Ziesler, and M.C. Papaefthymiou. *A true single-phase 8-bit adiabatic multiplier*. in *Proceedings of the 38th annual Design Automation Conference*. 2001: ACM.758-763

72. Blotti, A. and R. Saletti, *Ultralow-power adiabatic circuit semi-custom design.* Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2004. **12**(11): p. 1248-1253.

73. Fant, K.M. *Front Matter*. in *Logically Determined Design: Clockless System Design with NULL Convention Logic™*. 2005: Wiley Online Library.i-xvi

74. Thonnart, Y., P. Vivet, and F. Clermidy. *A fully-asynchronous low-power framework for GALS NoC integration*. in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2010*. 2010: IEEE.33-38

75. Hemani, A., T. Meincke, S. Kumar, A. Postula, T. Olsson, P. Nilsson, J. Oberg, P. Ellervee, and D. Lundqvist. *Lowering power consumption in clock by using globally asynchronous locally synchronous design style*. in *Design Automation Conference, 1999. Proceedings. 36th*. 1999: IEEE.873-878

76. Tiwari, V., D. Singh, S. Rajgopal, G. Mehta, R. Patel, and F. Baez. *Reducing power in high-performance microprocessors*. in *Proceedings of the 35th annual Design Automation Conference*. 1998: ACM.732-737

77. Sparsø, J., *Asynchronous Circuit Design: A Tutorial.* Technical University of Denmark, 2006.

78. Martin, A.J., *The design of an asynchronous microprocessor.* 1989.

79. Day, P. and J.V. Woods, *Investigation into micropipeline latch design styles.* Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 1995. **3**(2): p. 264-272.

80. Ozdag, R.O. and P.A. Beerel. *High-speed QDI asynchronous pipelines*. in *Asynchronous Circuits and Systems, 2002. Proceedings. Eighth International Symposium on*. 2002: IEEE.13-22

81. Beigné, E. and P. Vivet. *Design of On-chip and Off-chip Interfaces for a GALS NoC Architecture*. in *Asynchronous Circuits and Systems, 2006. 12th IEEE International Symposium on*. 2006: IEEE.10 pp.-183

82. Muller, D.E. and W. Bartky. *A theory of asynchronous circuits*. in *Proceedings of an International Symposium on the Theory of Switching*. 1959.204-243

83. Bainbridge, J. and S. Furber, *CHAIN: A delay-insensitive chip area interconnect.* IEEE Micro, 2002. **22**(5): p. 16-23.

84. Ng, P., P.T. Balsara, and D. Steiss, *Performance of CMOS differential circuits.* Solid-State Circuits, IEEE Journal of, 1996. **31**(6): p. 841-846.

85. Jacobs, G.M. and R.W. Brodersen, *A fully asynchronous digital signal processor using self-timed circuits.* Solid-State Circuits, IEEE Journal of, 1990. **25**(6): p. 1526-1537.

86. Tang, T.Y., C.S. Choy, J. Butas, and C.F. Chan. *An ALU design using a novel asynchronous pipeline architecture*. in *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*. 2000: IEEE.361-364

87. Anis, M., M. Allam, and M. Elmasry, *Impact of technology scaling on CMOS logic styles.* Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on, 2002. **49**(8): p. 577-588.

88. Arsalan, M. and M. Shams. *Asynchronous adiabatic logic*. in *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*. 2007: IEEE.3720-3723

89. Macii, E., M. Pedram, and F. Somenzi, *High-level power modeling, estimation, and optimization.* Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 1998. **17**(11): p. 1061-1079.

90. Kahng, A.B., B. Li, L.-S. Peh, and K. Samadi. *Orion 2.0: A fast and accurate noc power and area model for early-stage design space exploration*. in *Proceedings of the conference on Design, Automation and Test in Europe*. 2009: European Design and Automation Association.423-428

91. Chen, D., J. Cong, Y. Fan, and Z. Zhang. *High-level power estimation and low-power design space exploration for FPGAs*. in *Proceedings of the 2007 Asia and South Pacific Design Automation Conference*. 2007: IEEE Computer Society.529-534

92. Najm, F.N. *Power estimation techniques for integrated circuits*. in *Proceedings of the 1995 IEEE/ACM international conference on Computer-aided design*. 1995: IEEE Computer Society.492-499

93. Najm, F.N., R. Burch, P. Yang, and I.N. Hajj, *Probabilistic simulation for reliability analysis of CMOS VLSI circuits.* Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 1990. **9**(4): p. 439-450.

94. Najm, F.N. *Improved estimation of the switching activity for reliability prediction in VLSI circuits*. in *Custom Integrated Circuits Conference, 1994., Proceedings of the IEEE 1994*. 1994: IEEE.429-432

95. Bryant, R.E., *Graph-based algorithms for boolean function manipulation.* Computers, IEEE Transactions on, 1986. **100**(8): p. 677-691.

96. Burch, R., F.N. Najm, P. Yang, and T.N. Trick, *A Monte Carlo approach for power estimation.* Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 1993. **1**(1): p. 63-71.

97. Xakellis, M.G. and F.N. Najm. *Statistical estimation of the switching activity in digital circuits*. in *Proceedings of the 31st annual Design Automation Conference*. 1994: ACM.728-733

98. Schellekens, M., *A Modular Calculus for the Average Cost of Data Structuring*. 2008: Springer.

99. Schellekens, M., *MOQA; unlocking the potential of compositional static average-case analysis.* Journal of Logic and Algebraic Programming, 2010. **79**(1): p. 61-83.

100. Agarwal, R., *Randomness Preservation, Implications to Algorithmic Timing and Energy Complexity*, in *MSc Thesis*. 2008.

101. M. Schellekens, D.E., D.Vasudevan, J.Chen, A.Gao and E.Popovici, *MOQA Logic Units For Modular Power Analysis.* Submitted to Journal Information and Computation, 2012.

102. Schellekens, M., *A Modular Calculus for the Average Cost of Data Structuring: Efficiency-Oriented Programming in MOQA*. 2008: Springer.

103. Ye, T., D. Vasudevan, J. Chen, E. Popovici, and M. Schellekens. *Static Average Case Power Estimation Technique for Block Ciphers*. in *Digital System Design: Architectures, Methods and Tools (DSD), 2010 13th Euromicro Conference on*. 2010: IEEE.689-696

104. Zeinolabedini, N., G. Qin, D. Vasudevan, M. Schellekens, and E. Popovici. *Static average-case power analysis for sorting applications*. in *Microelectronics (MIEL), 2012 28th International Conference on*. 2012: IEEE.397-400

105. Yang, G., W.N.N. Hung, X. Song, and M. Perkowski, *Majority-based reversible logic gates.* Theoretical computer science, 2005. **334**(1): p. 259-274.

106. Liu, Z. and V. Kursun, *Characterization of a novel nine-transistor SRAM cell.* Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2008. **16**(4): p. 488-492.

107. Bansal, A., S. Mukhopadhyay, and K. Roy, *Device-optimization technique for robust and low-power FinFET SRAM design in nanoscale era.* Electron Devices, IEEE Transactions on, 2007. **54**(6): p. 1409-1419.

108. Razavipour, G., A. Afzali-Kusha, and M. Pedram, *Design and analysis of two low-power SRAM cell structures.* Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2009. **17**(10): p. 1551-1555.

109. Sharifkhani, M. and M. Sachdev, *Segmented virtual ground architecture for low-power embedded SRAM.* Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2007. **15**(2): p. 196-205.

110. Kim, T.C., C. Kim, B.Y. Chung, and S.W. Kim, *A low-power cache with successive tag comparison algorithm.* Current Applied Physics, 2005. **5**(3): p. 227-230.

111. Grossar, E., M. Stucchi, K. Maex, and W. Dehaene, *Read stability and write-ability analysis of SRAM cells for nanometer technologies.* Solid-State Circuits, IEEE Journal of, 2006. **41**(11): p. 2577-2588.

112. Azizi, N., F.N. Najm, and A. Moshovos, *Low-leakage asymmetric-cell SRAM.* Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2003. **11**(4): p. 701-715.

113. Wang, Y., H. Ahn, U. Bhattacharya, T. Coan, F. Hamzaoglu, W. Hafez, C.H. Jan, R. Kolar, S. Kulkarni, and J. Lin. *A 1.1 GHz 12μA/Mb-Leakage SRAM Design in 65nm Ultra-Low-Power*

CMOS with Integrated Leakage Reduction For Mobile Applications. in *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*. 2007: IEEE.324-606

114. Chen, J., D. Vasudevan, E. Popovici, M. Schellekens, and P. Gillen. *Design and analysis of a novel 8T SRAM cell for adiabatic and non-adiabatic operations*. in *Electronics, Circuits, and Systems (ICECS), 2010 17th IEEE International Conference on*. 2010: IEEE.434-437

115. Morita, Y., H. Fujiwara, H. Noguchi, Y. Iguchi, K. Nii, H. Kawaguchi, and M. Yoshimoto. *An area-conscious low-voltage-oriented 8T-SRAM design under DVS environment*. in *VLSI Circuits, 2007 IEEE Symposium on*. 2007: IEEE.256-257

116. Hsu, L.L., R.V. Joshi, F. Assaderaghi, and M.J. Saccamango, *Method and system for improving the performance on SOI memory arrays in an SRAM architecture system*. 2003, Google Patents.

117. Qin, H., Y. Cao, D. Markovic, A. Vladimirescu, and J. Rabaey, *Standby supply voltage minimization for deep sub-micron SRAM.* Microelectronics journal, 2005. **36**(9): p. 789-800.

118. Kim, C.H., J.J. Kim, S. Mukhopadhyay, and K. Roy, *A forward body-biased low-leakage SRAM cache: device, circuit and architecture considerations.* Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2005. **13**(3): p. 349-357.

119. Kuhn, K.J. *Reducing variation in advanced logic technologies: Approaches to process and design for manufacturability of nanoscale CMOS*. in *Electron Devices Meeting, 2007. IEDM 2007. IEEE International*. 2007: IEEE.471-474

120. Agarwal, A., B.C. Paul, S. Mukhopadhyay, and K. Roy, *Process variation in embedded memories: failure analysis and variation aware architecture.* Solid-State Circuits, IEEE Journal of, 2005. **40**(9): p. 1804-1814.

121. Heald, R. and P. Wang. *Variability in sub-100nm SRAM designs*. in *Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*. 2004: IEEE Computer Society.347-352

122. Chang, L., R.K. Montoye, Y. Nakamura, K.A. Batson, R.J. Eickemeyer, R.H. Dennard, W. Haensch, and D. Jamsek, *An 8T-SRAM for variability tolerance and low-voltage operation in high-performance caches.* Solid-State Circuits, IEEE Journal of, 2008. **43**(4): p. 956-963.

123. Van Berkel, C., M.B. Josephs, and S.M. Nowick, *Applications of asynchronous circuits.* Proceedings of the IEEE, 1999. **87**(2): p. 223-233.

124. Talarico, C., J.W. Rozenblit, V. Malhotra, and A. Stritter, *A new framework for power estimation of embedded systems.* Computer, 2005. **38**(2): p. 71-78.

125. Chen, J., E. Popovici, D. Vasudevan, and M. Schellekens. *Ultra Low Power Booth Multiplier Using Asynchronous Logic*. in *Asynchronous Circuits and Systems (ASYNC), 2012 18th IEEE International Symposium on*. 2012: IEEE.81-88

126. Parhami, B., *Computer arithmetic*. 2000: Oxford university press.

127. Kogge, P.M. and H.S. Stone, *A parallel algorithm for the efficient solution of a general class of recurrence equations.* Computers, IEEE Transactions on, 1973. **100**(8): p. 786-793.

128. Zimmermann, R., *Binary adder architectures for cell-based VLSI and their synthesis*. 1998: Citeseer.

129. Wang, C.C., C.L. Lee, and P.L. Liu. *Power aware design of an 8-bit pipelining asynchronous ANT-based CLA using data transition detection*. in *Circuits and Systems, 2004. Proceedings. The 2004 IEEE Asia-Pacific Conference on*. 2004: IEEE.29-32

130. Wang, C.C., C.C. Huang, C.L. Lee, and T.W. Cheng, *A low power high-speed 8-bit pipelining CLA design using dual-threshold voltage domino logic.* Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2008. **16**(5): p. 594-598.

131. Nielsen, L.S. and J. Sparso. *A low-power asynchronous data-path for a FIR filter bank*. in *Advanced Research in Asynchronous Circuits and Systems, 1996. Proceedings., Second International Symposium on*. 1996: IEEE.197-207

132. Weinberger, A., *4: 2 carry-save adder module.* IBM Technical Disclosure Bulletin, 1981. **23**(8): p. 3811-3814.

133. Chen, K.H. and Y.S. Chu, *A low-power multiplier with the spurious power suppression technique.* Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2007. **15**(7): p. 846-850.

134. Hsu, S.K., S.K. Mathew, M.A. Anders, B.R. Zeydel, V.G. Oklobdzija, R.K. Krishnamurthy, and S.Y. Borkar, *A 110 GOPS/W 16-bit multiplier and reconfigurable PLA loop in 90-nm CMOS.* Solid-State Circuits, IEEE Journal of, 2006. **41**(1): p. 256-264.

135. Liu, Y. and S. Furber. *The design of a low power asynchronous multiplier*. in *Low Power Electronics and Design, 2004. ISLPED'04. Proceedings of the 2004 International Symposium on*. 2004: IEEE.301-306

136. Khouri, K.S., G. Lakshminarayana, and N.K. Jha, *High-level synthesis of low-power control-flow intensive circuits.* Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 1999. **18**(12): p. 1715-1729.

137. Penzes, P. and A.J. Martin. *An energy estimation method for asynchronous circuits with application to an asynchronous microprocessor*. in *Design, Automation and Test in Europe Conference and Exhibition, 2002. Proceedings*. 2002: IEEE.640-647

138. Kudva, P. and V. Akella. *A technique for estimating power in asynchronous circuits*. in *Advanced Research in Asynchronous Circuits and Systems, 1994., Proceedings of the International Symposium on*. 1994: IEEE.166-175

139. Shams, A.M., T.K. Darwish, and M.A. Bayoumi, *Performance analysis of low-power 1-bit CMOS full adder cells.* Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2002. **10**(1): p. 20-29.

140. Radhakrishnan, D. *Low-voltage low-power CMOS full adder*. in *Circuits, Devices and Systems, IEE Proceedings-*. 2001: IET.19-24

141. Zimmermann, R. and W. Fichtner, *Low-power logic styles: CMOS versus pass-transistor logic.* Solid-State Circuits, IEEE Journal of, 1997. **32**(7): p. 1079-1090.

142. Coleman, D. and J. Di, *Analysis and Improvement of Delay-Insensitive Asynchronous Circuits Operating in Subthreshold Regime.* Journal of Low Power Electronics, 2010. **6**(2): p. 320-324.

143. PAN, H. and S. JU, *Asynchronous Design of NULL Convention Arithmetic Logic Unit [J].* Microelectronics & Computer, 2008. **8**: p. 014.

144. Cuccaro, S.A., T.G. Draper, S.A. Kutin, and D.P. Moulton, *A new quantum ripple-carry addition circuit.* arXiv preprint quant-ph/0410184, 2004.

145. Calhoun, B.H., A. Wang, and A. Chandrakasan, *Modeling and sizing for minimum energy operation in subthreshold circuits.* Solid-State Circuits, IEEE Journal of, 2005. **40**(9): p. 1778-1786.

146. Calhoun, B.H. and A. Chandrakasan. *Characterizing and modeling minimum energy operation for subthreshold circuits*. in *Low Power Electronics and Design, 2004. ISLPED'04. Proceedings of the 2004 International Symposium on*. 2004: IEEE.90-95

147. Landauer, R., *Irreversibility and heat generation in the computing process.* IBM Journal of Research and Development, 1961. **5**(3): p. 183-191.

148. Bennett, C.H., *The thermodynamics of computation—a review.* International Journal of Theoretical Physics, 1982. **21**(12): p. 905-940.

149. Wille, R. and R. Drechsler. *BDD-based synthesis of reversible logic for large functions*. in *Design Automation Conference, 2009. DAC'09. 46th ACM/IEEE*. 2009: IEEE.270-275

150. De Vos, A., B. Desoete, A. Adamski, P. Pietrzak, M. Sibiński, and T. Widerski, *Design of reversible logic circuits by means of control gates.* Integrated Circuit Design, 2000: p. 255-264.

151. Fredkin, E. and T. Toffoli, *Conservative logic.* International Journal of Theoretical Physics, 1982. **21**(3): p. 219-253.

152. Ryan, J.F. and B.H. Calhoun. *Minimizing offset for latching voltage-mode sense amplifiers for sub-threshold operation*. in *Quality Electronic Design, 2008. ISQED 2008. 9th International Symposium on*. 2008: IEEE.127-132

153. Patel, K.N., J.P. Hayes, and I.L. Markov, *Fault testing for reversible circuits.* Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 2004. **23**(8): p. 1220-1230.

154. Perkowski, M., J. Biamonte, and M. Lukac. *Test generation and fault localization for quantum circuits*. in *Multiple-Valued Logic, 2005. Proceedings. 35th International Symposium on*. 2005: IEEE.62-68

155. Wille, R., D. Große, S. Frehse, G.W. Dueck, and R. Drechsler. *Debugging of Toffoli networks*. in *Design, Automation & Test in Europe Conference & Exhibition, 2009. DATE'09.* 2009: IEEE.1284-1289

156. Steininger, A., *Testing and built-in self-test–A survey.* Journal of Systems Architecture, 2000. **46**(9): p. 721-747.

157. Chen, J., D.P. Vasudevan, E. Popovici, and M. Schellekens. *Reversible online BIST using bidirectional BILBO*. in *Proceedings of the 7th ACM international conference on Computing frontiers*. 2010: ACM.257-266

158. Musoll, E., T. Lang, and J. Cortadella, *Working-zone encoding for reducing the energy in microprocessor address buses.* Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 1998. **6**(4): p. 568-572.

159. Shende, V.V., A.K. Prasad, I.L. Markov, and J.P. Hayes, *Synthesis of reversible logic circuits.* Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 2003. **22**(6): p. 710-722.