

9-13-2018

Techniques of Energy-Efficient VLSI Chip Design for High-Performance Computing

Zhou Zhao

Louisiana State University and Agricultural and Mechanical College, zhou.zhao.uestc@gmail.com

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_dissertations



Part of the [Electrical and Electronics Commons](#), and the [VLSI and Circuits, Embedded and Hardware Systems Commons](#)

Recommended Citation

Zhao, Zhou, "Techniques of Energy-Efficient VLSI Chip Design for High-Performance Computing" (2018). *LSU Doctoral Dissertations*. 4702.

https://digitalcommons.lsu.edu/gradschool_dissertations/4702

This Dissertation is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Doctoral Dissertations by an authorized graduate school editor of LSU Digital Commons. For more information, please contact gradetd@lsu.edu.

TECHNIQUES OF ENERGY-EFFICIENT VLSI CHIP DESIGN FOR HIGH-PERFORMANCE COMPUTING

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Division of Electrical and Computer Engineering

by

Zhou Zhao

B.S., University of Electronic Science and Technology of China, Chengdu, China, 2011

M.S., University of Electronic Science and Technology of China, Chengdu, China, 2014

December 2018

ACKNOWLEDGMENTS

Firstly, I sincerely dedicate my dissertation to my parents and my maternal grandmother for their great and endless support and education to my entire life.

I would like to thank my doctoral advisor, Dr. Ashok Srivastava. His solid understanding of VLSI circuit theory and kind encouragement gave me invaluable help during my graduate study. I sincerely appreciate Dr. Lu Peng for his help on the computer architecture with open minded vision. I would like to thank Dr. Lu Peng's former doctoral student, Dr. Shaoming Chen, for his help on the modeling of power delivery network. I also thank Dr. Saraju P. Mohanty of the University of North Texas for help and discussion. I would like to thank Dr. Ramachandran Vaidyanathan, Dr. David Koppelman and Dr. Scott Baldrige as my committee members with their precious suggestions on my dissertation. I deeply thank my friends I met in LSU, Xinlu, Xiaohan, Boyuan, William, Jojoe, Xiaochen and Wei. Part of the work is supported under NSF Grant No. 1422408.

Finally, I would like to thank everyone for blooming in my life like the fireworks beauty.

TABLE OF CONTENTS

ACKNOWLEDGMENTS	ii
LIST OF TABLES	v
LIST OF FIGURES	vi
LIST OF ABBREVIATIONS.....	x
ABSTRACT.....	xiii
CHAPTER 1. INTRODUCTION.....	1
1.1 Challenge of Computations in VLSI Chips.....	1
1.2 Power Regulation in Computational Chips.....	2
1.3 Overview of Multiple Logic Styles.....	3
1.4 Circuit Design for Logarithmic Computation.....	10
1.5 Overview of Optimization in ALU Design.....	13
1.6 Circuit Design for Neural Computation.....	15
1.7 Goals and Objectives.....	19
CHAPTER 2. SWITCHABLE PIN USED FOR POWER REGULATION IN CHIP- MULTIPROCESSOR.....	21
2.1 Analysis of IR Droop.....	21
2.2 Overview of Switchable Pin.....	24
2.3 Performance Evaluation using PDN.....	27
2.4 Customized Design of I/O Pad.....	30
2.5 Circuit Verification.....	35
2.6 Conclusion.....	44
CHAPTER 3. LOGARITHMIC CONVERTER WITH A NOVEL CALIBRATION SUPPLIED BY DC/CLOCKED POWER.....	46
3.1 Study of Logarithmic Conversion.....	46
3.2 Circuit Implementation.....	51
3.3 Error Analysis.....	59
3.4 Circuit Verification.....	61
3.5 Conclusion.....	71
CHAPTER 4. ALU DESIGN BASED ON NEURON-LIKE CELLS USING MULTIPLE INPUT FLOATING GATE MOSFETS.....	72
4.1 Design of Neuron-Like Cells.....	72
4.2 ALU Design.....	76
4.3 Result and Discussion.....	81
4.4 Conclusion.....	91

CHAPTER 5. LONG SHORT-TERM MEMORY NETWORK DESIGN FOR ANALOG COMPUTING.....	92
5.1 Long Short-Term Memory Network.....	92
5.2 MIFG MOSFETs for DAC Design.....	95
5.3 Long Short-Term Memory Network Design.....	96
5.4 Result and Discussion.....	112
5.5 Conclusion.....	129
CHAPTER 6. SUMMARY AND SCOPE OF FUTURE WORK.....	132
6.1 On-Chip Power Regulating Using Switchable Pins.....	132
6.2 Logarithmic Conversion with Calibration.....	133
6.3 Optimization and Calibration of ALU.....	134
6.4 Neural Accelerator based on ASP.....	135
BIBLIOGRAPHY.....	137
APPENDIX A. MOSIS SPICE LEVEL 7 CMOS MODEL PARAMETERS.....	148
A.1 AMI 500nm Technology.....	148
A.2 TSMC 180nm Technology.....	149
APPENDIX B. PROPOSED DESIGN FLOW OF AUTOMATIC PLACEMENT AND ROUTING USING MENTOR GRAPHICS EDA TOOLS.....	152
B.1 APR with Verilog Entry.....	152
B.2 APR with Customized Schematic Entry.....	152
APPENDIX C. ERROR TEST IN PSPICE SIMULATION.....	155
VITA.....	157

LIST OF TABLES

Table 2.1	Summary of dimension in via.....	31
Table 2.2	Summary of parameters in PDN.....	31
Table 2.3	Summary of testing cases.....	36
Table 2.4	Pin allocation of the chip in Figure 2.10.....	42
Table 3.1	Comparison with the reported conversion methods and the proposed work.....	60
Table 3.2	Comparison of hardware cost with the prior work and the proposed work.....	66
Table 3.3	Comparison of maximum length with the prior work and the proposed work.....	66
Table 3.4	Pin allocation of the chip in Figure 3.7.....	68
Table 4.1	Operation summary.....	80
Table 4.2	Comparison of static logic and proposed neuron-like cells using MIFG logic.....	84
Table 5.1	W/L ratio of transistors in the proposed analog multiplier of Figure 5.3.....	103
Table 5.2	W/L ratio of transistors in the circuits of Figures 5.5 and 5.6.....	108
Table 5.3	Performance comparison with prior work of LSTM.....	125
Table 5.4	Performance comparison with prior work of ASP based on neural network.....	125

LIST OF FIGURES

Figure 1.1	Logic diagram of (a) static logic, (b) three types of PT logic, (c) pseudo NMOS logic, (d) DCVSL under dc power or ECRL under clocked power, (e) PFAL, (f) domino logic. F and F' in circuits are pull-down and pull-up blocks built by NMOS and PMOS transistors, respectively.....6
Figure 1.2	Simulation of the inverter based on described logic styles, (a) transfer curve in dc sweep, (b) performance evaluation of power dissipation and delay.....8
Figure 1.3	FO4 simulation, (a) test circuit, (b) performance evaluation of power dissipation and delay.....9
Figure 1.4	Performance comparison of the multiplication in both binary and logarithmic domains using TSMC 180nm technology, (a) power dissipation, (b) computation latency.....12
Figure 1.5	Two digital calibration methods, (a) adaptive clock, (b) delay-aware feedback loop.....14
Figure 1.6	MIFG MOSFET, (a) device symbol, (b) equivalent circuit model.....16
Figure 2.1	Cross-section view of power supply interconnection using multiple layers.....22
Figure 2.2	IR droop simulations, (a) voltage loss ratio depending on various processes, (b) growth rate of fall/rise time depending on various processes at 1GHz.....25
Figure 2.3	Conceptual diagram of SoC with switchable pins.....28
Figure 2.4	Power delivery network, (a) 'two-chip' mode, (b) 'three-chip' mode, (c) normal mode without switchable pins.....29
Figure 2.5	Simulation results of signal attenuation using the proposed PDN, (a) READ, (b) WRITE.....32
Figure 2.6	Redesigned I/O pad used for both data transmission and power supply, (a) logic diagram, (b) layout view, (c) delay comparison with standard cells.....34
Figure 2.7	Layout view of two MSP430 cores, in which 12 of 16 I/O pads are modified as switchable pins.....36
Figure 2.8	Power simulation, (a) current simulation in both data mode and power mode, (b) power dissipation of extra circuits serving for switchable pins, (c) mean value of supplied voltage under power pad and switchable pin supply, (d) standard deviation of supplied voltage under power pad and switchable pin supply.....39
Figure 2.9	Signal integrity simulation, (a) WRITE in power mode, (b) WRITE in data mode, (c) READ in power mode, (d) READ in data mode.....40

Figure 2.10	Fabricated chip with switchable pins, (a) layout view, (b) die photo.....	42
Figure 2.11	Chip test, (a) transient simulation, N0 and N1 represent the outputs of the core supplied under power pad, SWP0 and SWP1 represent the outputs of the core supplied under the proposed switchable pins, (b) delay comparison of two cores, the scales of logic high and low are 5V and 0V, respectively, (c) current monitor in two work modes.....	43
Figure 3.1	Logarithmic conversion, (a) error curves of Mitchell’s algorithm and the one with the addition of C, (b) error curve using the proposed calibration, (c) calibration flow.....	50
Figure 3.2	Block diagram of the proposed logarithmic converter.....	52
Figure 3.3	Circuit diagram of the proposed <i>N</i> -bit logarithmic converter, (a) integer part, (b) fraction part, (c) calibration block, (d) clock generator.....	57
Figure 3.4	Error sweep of the proposed calibration and Mitchell’s algorithm.....	60
Figure 3.5	Simulations of clock generator, (a) waveform at 498MHz, (b) waveform at 0.981GHz, (c) oscillation frequency versus control voltage, (d) frequency deviation versus control voltage.....	63
Figure 3.6	Simulations of the entire logarithmic converter, (a) layout view, (b) DCVSL transient simulation, (c) ECRL transient simulation, (d) comparison of power dissipation under different logic.....	65
Figure 3.7	Fabricated chip of the logarithmic multiplier, (a) layout view, (b) die photo.....	68
Figure 3.8	Chip testing, (a) fraction results under clocked power, (b) integer results under clocked power, the scales of logic high and low are 5V and 0V, respectively, (c) power dissipation results including chip measurement and post-layout simulation, (d) current spikes depending on logic switch.....	70
Figure 4.1	Figure 4.1 Circuit diagram of the proposed MIFG logic gates, (a) INV and BUF, (b) NAND, AND, NOR and OR, (c) XOR and XNOR, (d) full adder.....	75
Figure 4.2	Proposed circuit block used for 32-bit sequence comparison.....	77
Figure 4.3	Proposed circuit block of digital calibration.....	77
Figure 4.4	PT-MIFG logic based TG with four floating gates.....	77
Figure 4.5	Proposed ALU, (a) block diagram, (b) feedback loop for error detection.....	79
Figure 4.6	Transient simulations of the MIFG logic based cells, (a) INV and BUF, (b) NAND and AND, (c) NOR and OR, (d) XOR and XNOR, (e) full adder, (f) transmission gate with 2, 3 and 4 floating gates.....	82

Figure 4.7	Full layout view of the proposed ALU.....	85
Figure 4.8	Simulation results, (a) each operation and average value, (b) power dissipation depending on frequency, (c) delay due to feedback loop, (d) delay depending load capacitor.....	87
Figure 4.9	Error rate with the variation of mismatch.....	90
Figure 5.1	Standard cell of LSTM network.....	93
Figure 5.2	MIFG MOSFET, (a) 3D view of overlapping-dual-ploy structure and (b) 3D view of adjacent n-diffusion capacitor structure. Note that both poly and active contacts are ignored in two 3D views.....	97
Figure 5.3	Circuit diagram of the proposed analog multiplier with the functions of both voltage scaling and DAC. Highlighted transistors are MIFG MOSFETs.....	102
Figure 5.4	Voltage attenuation versus unit capacitor in MIFG driven input transistor.....	102
Figure 5.5	The proposed analog multiplier with MIFGs, (a) transient simulation, two 4-bit inputs are from all high logic (1.8V) flipping to all low logic (0V) periodically under 1GHz and 10MHz and (b) dc transfer curve.....	103
Figure 5.6	Analog implementation of sigmoid function, (a) circuit diagram, (b) transfer characteristics.....	105
Figure 5.7	Analog implementation of hyperbolic tangent function, (a) circuit diagram, (b) transfer characteristics.....	106
Figure 5.8	Current comparator used as back-end stage, (a) circuit diagram, (b) delay dependence on current amplification.....	108
Figure 5.9	System level diagram of the proposed LSTM network.....	111
Figure 5.10	Layout view, (a) proposed analog multiplier, (b) sigmoid function, (c) hyperbolic tangent, (d) current comparator and (e) LSTM network.....	113
Figure 5.11	PDP comparison between the proposed analog multiplier and other digital multipliers.....	116
Figure 5.12	Break-down graph of power dissipation for each operation.....	116
Figure 5.13	Transient simulation based computing flow of LSTM network using the proposed ASP design.....	117
Figure 5.14	Overview of the proposed analog LSTM network with on-chip SRAM access used for extending the depth of network.....	120

Figure 5.15	Simulation results of the LSTM network with SRAM access, (a) power dissipation, (b) computing latency, (c) utilization ratio of SRAM and (d) throughput.....	121
Figure 5.16	Simulation results of error rate with PVT variation and introduced noise. All simulations are imported with mismatch of $\pm 20\%$ distribution of transistors, capacitors used for MIFGs and resistors.....	130
Figure 5.17	Simulation results using the proposed LSTM structure with PTB dataset, (a) training result and (b) predictive results of perplexity and required epochs for stabilizing perplexity with matrix expanding.....	130
Figure B.1	Design flow of APR using Mentor Graphics EDA tools, (a) HDL entry, (b) customized schematic entry.....	153
Figure C.1	Customized flow used for error testing.....	156

LIST OF ABBREVIATIONS

VLSI	Very Large Scale Integration
MIFG	Multiple Input Floating Gate
MOSFET	Metal-Oxide Semiconductor Field-Effect Transistor
ALU	Arithmetic Logic Unit
LSTM	Long Short-Term Memory
DVFS	Dynamic Voltage Frequency Scaling
LDO	Low Dropout
PCB	Printed Circuit Board
APR	Automatic Placement and Routing
PT	Pass Transistor
MUX	Multiplexer
CMOS	Complementary Metal Oxide Silicon
TG	Transmission Gate
CPU	Central Processing Unit
DCVSL	Differential Cascade Voltage Switch Logic
ECRL	Efficient Charge Recovery Logic
PFAL	Positive Feedback Adiabatic Logic
SRAM	Static Random Access Memory
WL	Word Line
FO4	Fan-Out of Four
HDL	Hardware Description Language
EDA	Electronic Design Automation
FA	Full Adder
LUT	Lookup Table

PLA	Programmable Logic Array
LOD	Leading-One Detector
GPU	Graphics Processing Unit
SOI	Silicon-on-Insulator
PLL	Phase Lock Loop
BDN	Binary Decision Neuron
CNN	Convolution Neural Network
RNN	Recurrent Neural Network
FPGA	Field-Programmable Gate Array
ASIC	Application Specific Integrated Circuit
DSP	Digital Signal Processing
ASP	Analog Signal Processing
DAC	Digital-to-Analog Converter
ADC	Analog-to-Digital Converter
ENOB	Effective Number of Bits
PDN	Power Delivery Network
IC	Integrated Circuit
PTM	Predictive Transistor Model
SoC	System-on-Chip
ESD	Electrostatic Discharge
CLA	Look Ahead Adder
RO	Ring Oscillator
CRA	Carry Ripple Adder
FPD	Floating Gate Potential Diagram
PTB	Penn TreeBank

THD	Total Harmonic Distortion
SAR	Successive Approximation Register
PDP	Power Delay Product
S/H	Sample and Hold
DPC	Digital-to-Pulse Converter
PVT	Process, Voltage and Temperature
BGA	Ball Grid Array

ABSTRACT

How to implement quality computing with the limited power budget is the key factor to move very large scale integration (VLSI) chip design forward. This work introduces various techniques of low power VLSI design used for state of art computing. From the viewpoint of power supply, conventional in-chip voltage regulators based on analog blocks bring the large overhead of both power and area to computational chips. Motivated by this, a digital based switchable pin method to dynamically regulate power at low circuit cost has been proposed to make computing to be executed with a stable voltage supply. For one of the widely used and time consuming arithmetic units, multiplier, its operation in logarithmic domain shows an advantageous performance compared to that in binary domain considering computation latency, power and area. However, the introduced conversion error reduces the reliability of the following computation (e.g. multiplication and division.). In this work, a fast calibration method suppressing the conversion error and its VLSI implementation are proposed. The proposed logarithmic converter can be supplied by dc power to achieve fast conversion and clocked power to reduce the power dissipated during conversion. Going out of traditional computation methods and widely used static logic, neuron-like cell is also studied in this work. Using multiple input floating gate (MIFG) metal-oxide semiconductor field-effect transistor (MOSFET) based logic, a 32-bit, 16-operation arithmetic logic unit (ALU) with zipped decoding and a feedback loop is designed. The proposed ALU can reduce the switching power and has a strong driven-in capability due to coupling capacitors compared to static logic based ALU. Besides, recent neural computations bring serious challenges to digital VLSI implementation due to overload matrix multiplications and non-linear functions. An analog VLSI design which is compatible to external digital environment is proposed

for the network of long short-term memory (LSTM). The entire analog based network computes much faster and has higher energy efficiency than the digital one.

CHAPTER 1

INTRODUCTION

1.1 Challenge of Computations in VLSI Chips

Research on the trading-off between performance and energy efficiency is always a hotspot in VLSI design. Especially, Moore's law continuously allows more and more computational blocks to be integrated into a single die [1], which largely increases area efficiency. However, overloaded computations obviously challenge the noise margin in the design of digital VLSI and so increase the switch current in each data path, which is the main part boosting both dynamic power dissipation and power density [2]. On the other hand, higher bit length required for current computations makes balancing design metrics such as area, latency, power dissipation and routing cost to be more difficult than early VLSI era. Recently, portable devices have been developed. The life time of batteries and cooling techniques largely constrain those portable devices to perform the same computation efficiency as in traditional personal computers [3, 4]. Even in devices with externally stable power supply, large heat dissipation violates the working characteristics of transistors reducing their carrier velocity in channels. Hereby, the allowable frequency of digital systems in electronic devices is restricted by the transistor's performance degradation due to the increase of temperature. Furthermore, the worst case is that a VLSI chip under high temperature has the risk of breakdown to the entire die and its package. Thus, to keep VLSI chips working safely under the highest frequency, it is anticipated that some blocks in a chip have to be shut down to confirm that the temperature cannot exceed the safe line. This possible phenomenon in future chips is called as 'dark silicon' [5, 6]. A very first strategy to avoid this potential phenomenon is to update cooling techniques. To directly cool down the chip inside, it is highly relying on the development of the novel materials embedded into chips [7, 8]. Effective off-chip methods such as liquid nitrogen cooling and fully covered fan are high cost and not suitable for portable devices.

With the rigorous restrictions of space and hardware cost, it is important to propose strategies regarding the optimization of computation algorithms and low cost circuit designs. This gives areas of both VLSI design and computer architecture more options than that of cooling technique.

1.2 Power Regulation in Computational Chips

In both digital and mixed signal VLSI chips, the block of power regulation is not optional but a necessity. The functions of power regulation are evenly supplying a stable in-chip voltage and achieving dynamic voltage and frequency scaling (DVFS) [9]. For stabilizing the supply voltage in all blocks of a chip, the simplest method is to increase the width of metal layer to suppress the voltage droop due to interconnection at the cost of increasing the routing difficulty [10]. Besides, current reduction can decrease this voltage droop. However, it is difficult to reduce switching current in complex logic blocks, especially in high-speed pipelined chains. With this background, a feasible method is to distribute feedback loop based voltage regulators all around inside a single die which can be seen a huge penalty to area and power dissipation [11]. For the implementation of DVFS, it is common to use a voltage regulator such as in DC/DC converter and a low dropout (LDO) regulator to manage in-chip voltage switching, both of which still bring large performance overhead [12]. All these regulators are designed in analog circuit which is sensitive to in-chip noise and the variations of temperature and dimension. Even though recent work [13, 14] have used digital-assisted techniques to reduce circuit cost and influence from the coupling noise, switch noise and variation, the implementation of voltage regulation cannot fully escape analog blocks. To reduce the cost and design difficulty from in-chip, the technique of off-chip power regulation has been developed [15]. It uses decoupling capacitors to reduce the noise due to chip package. The disadvantages of this techniques is increasing the design difficulty of printed circuit board (PCB) considering signal reflection on board and the noise from off-chip passive

devices. Specially, for the operation of DVFS, the response time of voltage switching using off-chip power regulation is larger than that using in-chip one which may result in incorrect data transmission during frequency scaling or boosting. Overall, it is desirable to propose other low cost power regulation methods which are different to current strategies.

1.3 Overview of Multiple Logic Styles

Static logic is the most classic logic style which has been widely used in both academia and industry since PMOS transistor is invented [16]. The logic diagram of static logic is shown in Figure 1.1 (a), which uses PMOS and NMOS transistors to build a complementary topology. PMOS transistors and NMOS transistors are responsible to enable logic high and low, respectively. Static logic is friendly to the algorithms of automatic placement and routing (APR) and circuit debugging due to its single signal path without differential signal. However, static logic has several disadvantages as follows:

- 1) For the multi-input NOR gate, the cascade PMOS transistor will largely reduce the switch speed due to the body effect increasing threshold voltage;
- 2) The single logic stage cannot output the functions such as AND and OR directly;
- 3) Logic high has to be evaluated by PMOS transistors which is slower than NMOS transistors;
- 4) The layout height of a pitch cell is sensitive to the complexity of combinational logic.

Given above disadvantages, other logic styles have been introduced with the consideration of specific design metrics. To save power dissipation and design area, pass transistor (PT) logic is proposed [17]. This logic style is highly suitable for the design of multiplexer (MUX) with long bit length and faster than static logic. For the implementation of switch function, NMOS transistor only, PMOS transistor only and complementary metal oxide silicon (CMOS) transmission gate

(TG) are three types of PT logic as shown in Figure 1.1 (b). Due to the threshold voltage in both NMOS and PMOS transistors, PT logic in PMOS transistor only and NMOS transistor only cannot output perfect logic low and logic high, respectively. This imperfect transmission can be calibrated by the inverter based restorer at the cost of power and latency. Thus, PT logic based on CMOS TG is the mainstream method to confirm correct signal flow. Note that in the illustrated logic diagrams, only AND operation is achieved. In PT logic, OR operation is simply achieved by interconnection. Using AND and OR, any Boolean logic can be achieved. The disadvantages of PT logic are weak capability of driven-in which restricts that PT logic is mostly used in the end of a logic chain, and leakage current corresponding to a TG in switch-off state which may charge the following circuits and lead to transmission error under high-speed operation if there is no discharge path available to it.

Pseudo NMOS logic is developed to reduce the layout height of a pitch cell and speed up logic switching [18]. The logic diagram is shown in Figure 1.1 (c). The logic is only evaluated by NMOS transistor array. The top PMOS transistor is always turn-on being responsible to output logic high. Due to the resistance from the turn-on PMOS transistor, this logic cannot achieve perfect logic low. In some specific layout placement with the restriction of block height, pseudo NMOS logic is a desired candidate.

In the design of state of art central processing unit (CPU), differential digital signal is widely used to suppress the transmission noise [19]. Differential cascade voltage switch logic (DCVSL) can output differential signal and speed up logic switching compared to static logic. The logic diagram of DCVSL is shown in Figure 1.1 (d). Same as pseudo NMOS logic, the logic is fully evaluated by NMOS transistor array. The cross coupled connection of two PMOS transistors mitigates the static power dissipated in those turn-off transistors. The disadvantage of DCVSL is

the cost of transistors. Another interesting application is that under four-phase clocked power, DCVSL is changed to efficient charge recovery logic (ECRL), which is one of adiabatic logic with the purpose of low power operation [20]. The problem of adiabatic logic is that its output signals are not standard pulse signals and thus not compatible to standard digital blocks.

Speaking of adiabatic logic, another logic style achieving slow charge and discharge is positive feedback adiabatic logic (PFAL) as shown in Figure 1.1 (e) [21]. It is built by an inverter based latch and differential logic evaluation block using NMOS transistors. Unlike the design of static random access memory (SRAM) with two transistors used for controlling the signal from the word line (WL), PFAL under dc power supply cannot work well due to the cycled data flow in the latch with the weak control in the output nodes. However, in four-phase operation of adiabatic logic, it provides both hold and wait stages to give the latch enough time for logic evaluation and avoiding unstable logic state. Once the correct signal is obtained in the output nodes after logic evaluation, the unwanted dynamic power dissipation can be reduced compared to ECRL. The disadvantage of PFAL is the same as that of ECRL, which is the cost of transistors.

Adiabatic logic is the strategy with the purpose of energy-efficient design. If the high-speed processing is the design priority, domino logic is a great choice [22]. The logic diagram of domino logic is shown in Figure 1.1 (f). Unlike the previously described logic styles which safely use the same topology stage by stage, a domino logic chain uses NMOS transistors and PMOS transistors as the evaluation block stage by stage. Same as adiabatic logic, domino logic is supplied under clocked power. However, the clocked power used for domino logic is the standard pulse signal to let a domino logic chain work for pre-charge and logic evaluation stage by stage. The disadvantage of this logic is the increase of noise sensitivity since the transistor's capacitor cannot be always perfectly charged or discharged for logic evaluation under high-speed working.

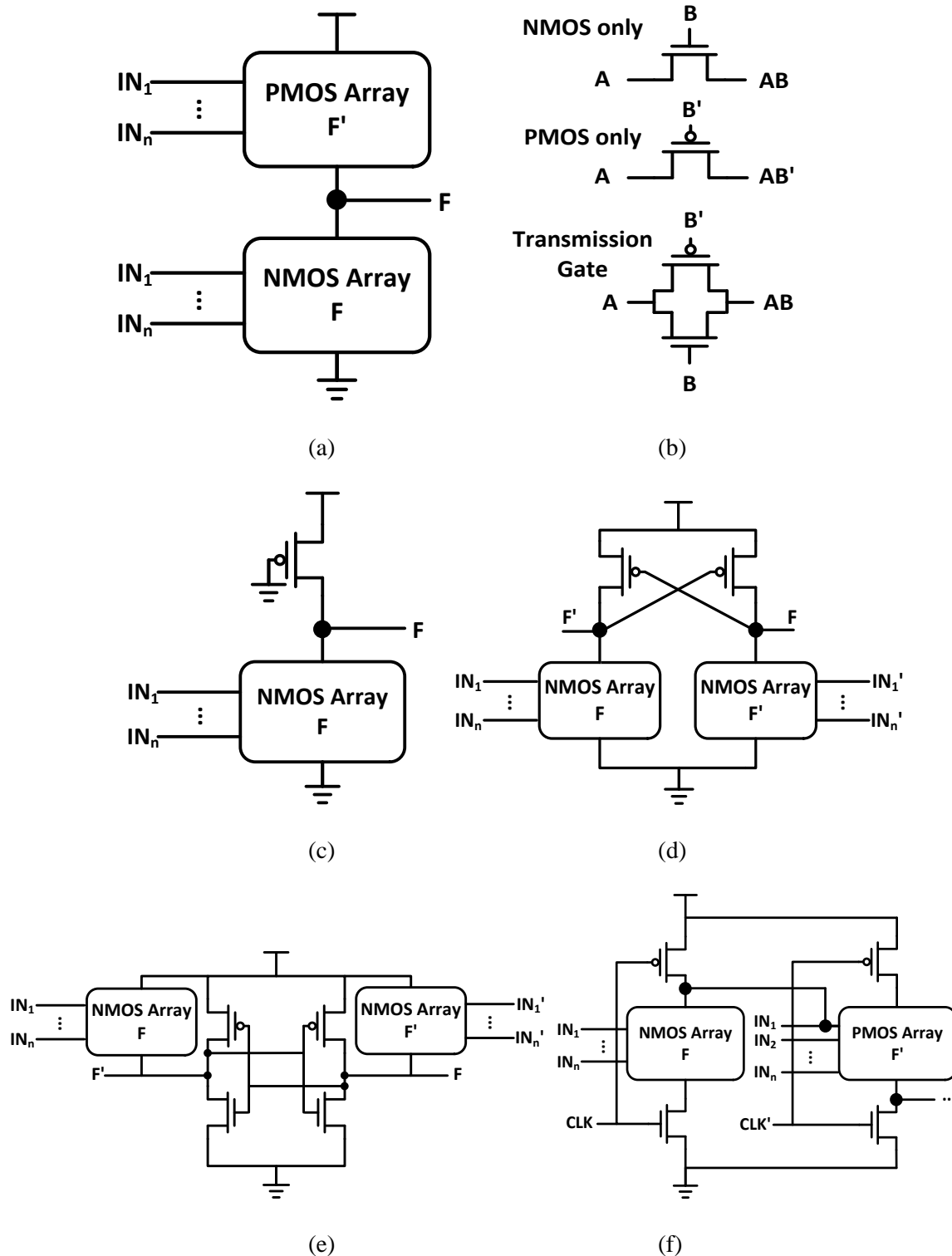
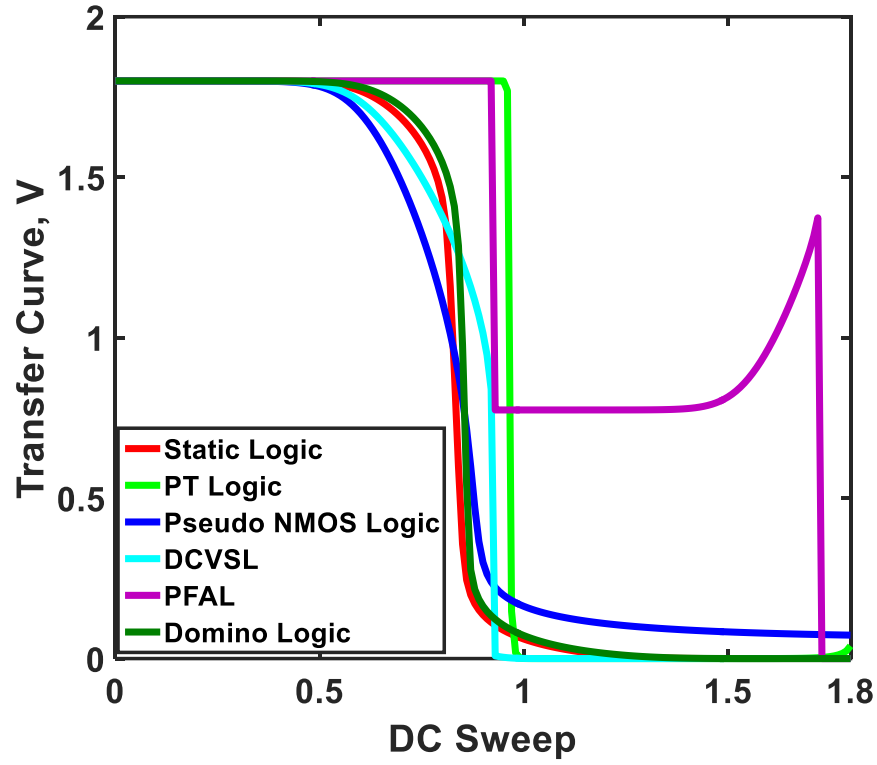


Figure 1.1 Logic diagram of (a) static logic, (b) three types of PT logic, (c) pseudo NMOS logic, (d) DCVSL under dc power or ECRL under clocked power, (e) PFAL, (f) domino logic. F and F' in circuits are pull-down and pull-up blocks built by NMOS and PMOS transistors, respectively.

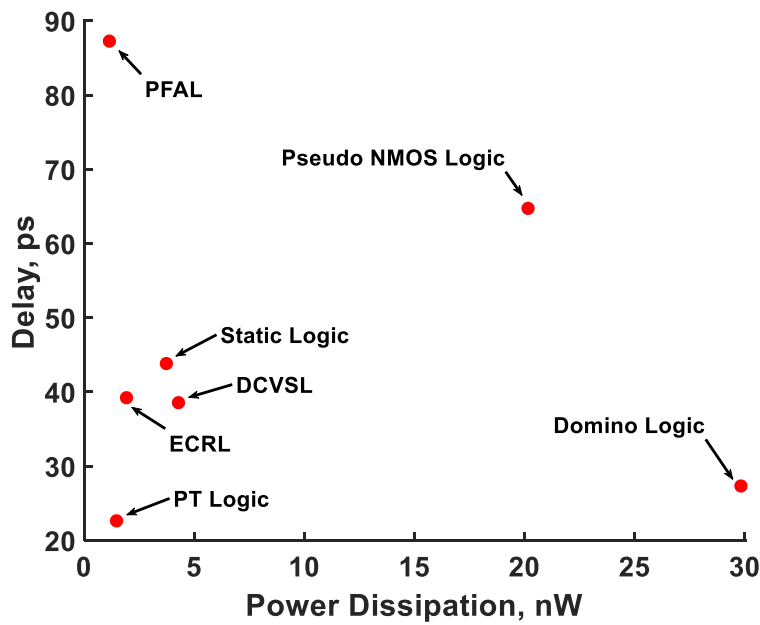
Besides above logic, other novel logic combining some of described logic styles have been reported [23, 24, 25] with the specific consideration of design metrics.

Figure 1.2 (a) shows the dc transfer curves of the inverter designed by the described logic styles using TSMC 180nm technology with 1fF capacitor load. We conclude that for static logic, PT logic based CMOS TG, DCVSL and domino logic, their transfer characteristics well support the logic operations and have large noise margins. Among these, PT logic shows the best transfer characteristic considering the switching current during logic switch. However, unlike other logic styles, PT logic cannot adjust the transfer curve through customizing transistors' dimensions. The transfer curve of pseudo NMOS logic matches the previous analysis that it cannot output perfect logic low. The feasible methods to mitigate this phenomenon are increasing the width of the top PMOS transistors at the cost of area and connecting a buffer at the cost of delay. For PFAL, it is obvious that it is not suitable to be worked under normal dc power supply as previous analysis. Figure 1.2 (b) shows the scattered plot of both power dissipation and gate delay of the inverter based on described logic styles. We observe both two adiabatic logic, ECRL and PFAL, can save more power than other logic except PT logic. ECRL works faster than PFAL. For normal digital logic, if both power and delay issues are considered, PT logic, static and DCVSL are desired candidates. If only the work speed is concerned, domino logic and PT logic should be used.

Another important design metrics is the capability of driven-in, which decides the allowable fan-out. Specially, in the VLSI design of parallel and distributed architectures, a large fan-out is necessary. A very common strategy to evaluate this metrics is the simulation of fan-out of four (FO4) [26]. For this strategy, two methods are widely used. The first one is sizing FO4, which is using four inverters in pipelined connection with the transistors' dimensions being doubled stage by stage. Another one is branching FO4 as shown in Figure 1.3 (a), which is using

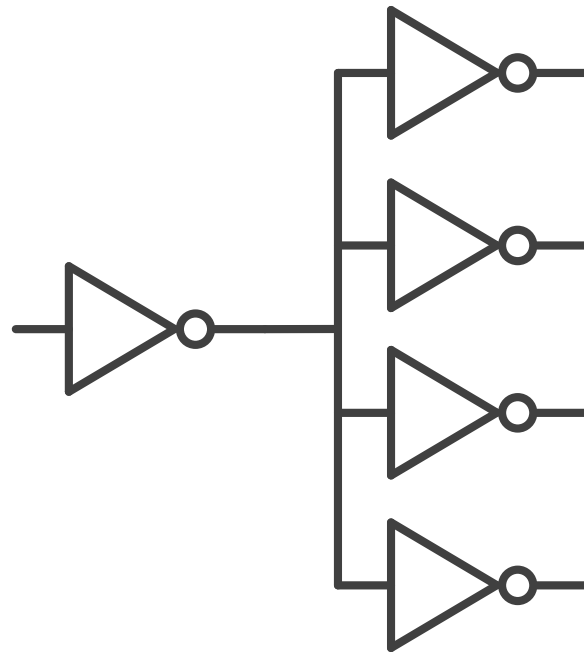


(a)

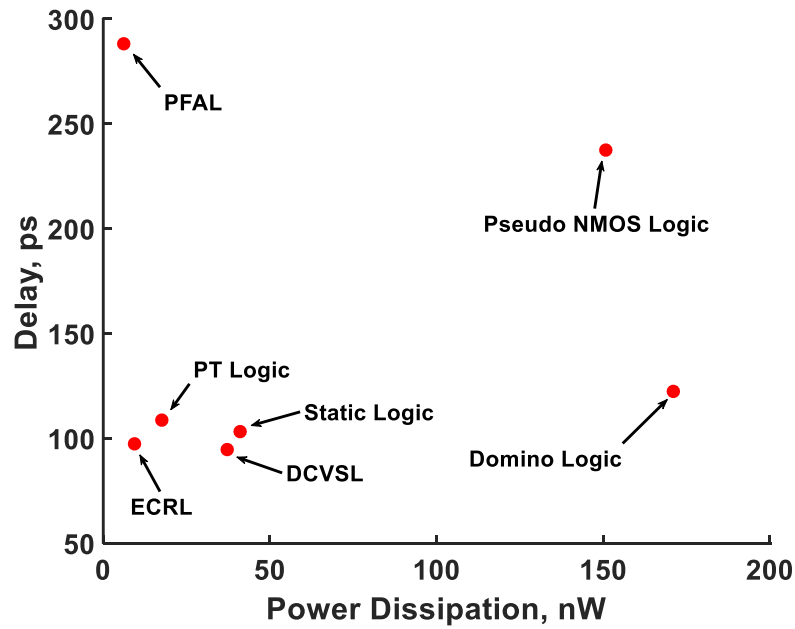


(b)

Figure 1.2 Simulation of the inverter based on described logic styles, (a) transfer curve in dc sweep, (b) performance evaluation of power dissipation and delay.



(a)



(b)

Figure 1.3 FO4 simulation, (a) test circuit, (b) performance evaluation of power dissipation and delay.

four parallel inverters as the output load of the inverter in the first stage. All transistors' dimensions are the same in this method. In our evaluation, we use branching FO4 method for the performance evaluation. Figure 1.3 (b) shows the scattered plot of both power dissipation and logic delay of FO4 simulation. Considering power issue, ECRL and PFAL are still more energy-efficient than other logic. An observation is that in the simulation of the single inverter shown in Figure 1.2 (b), PT logic shows an advantageous performance compared to other logic. However, in FO4 simulation, the performance degradation of PT logic is large due to the weak capability of drive-in. In addition, the signal integrity of PT logic is the worst in FO4 simulation which may lead to incorrect data transmission. The performance of speed in domino logic is also largely degraded in FO4 simulation since the capacitor existed for evaluation is not large enough to drive a large fan-out, which is the main reason explaining that dynamic logic is more suitable used for pipelined structure than the paralleled structure.

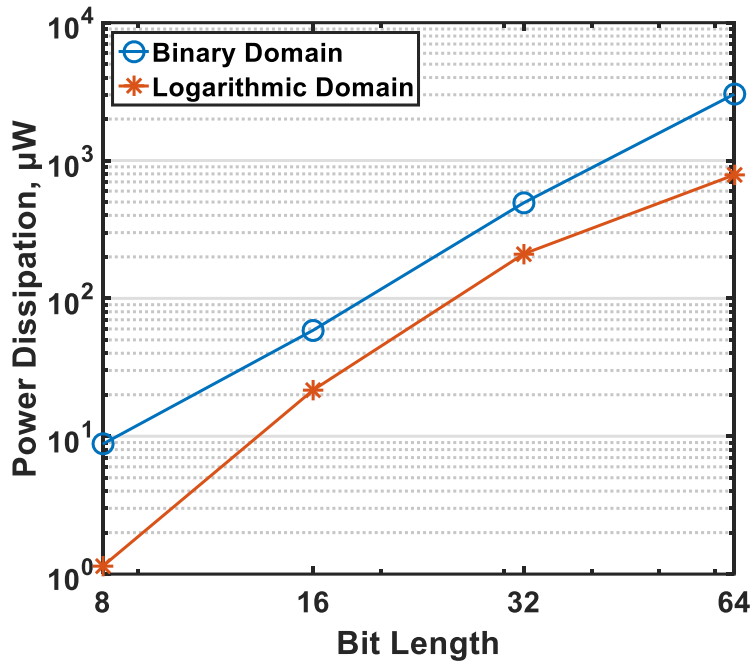
In summary, considering low power design, two adiabatic logic, PFAL and ECRL, are two desired strategies, in both of which the signal type, however, is not the standard digital signal. If pipelined structure is applied with the priority of high-speed, domino logic should be used. It is better to use static logic and DCVSL when the trading-off between power and latency is the design concern. When the circuit is located in the back end of the block with a small fan-out, PT logic is the best selection. For some layout placement with the limited height of the pitched cells, pseudo NMOS logic should be taken into the consideration.

1.4 Circuit Design for Logarithmic Computation

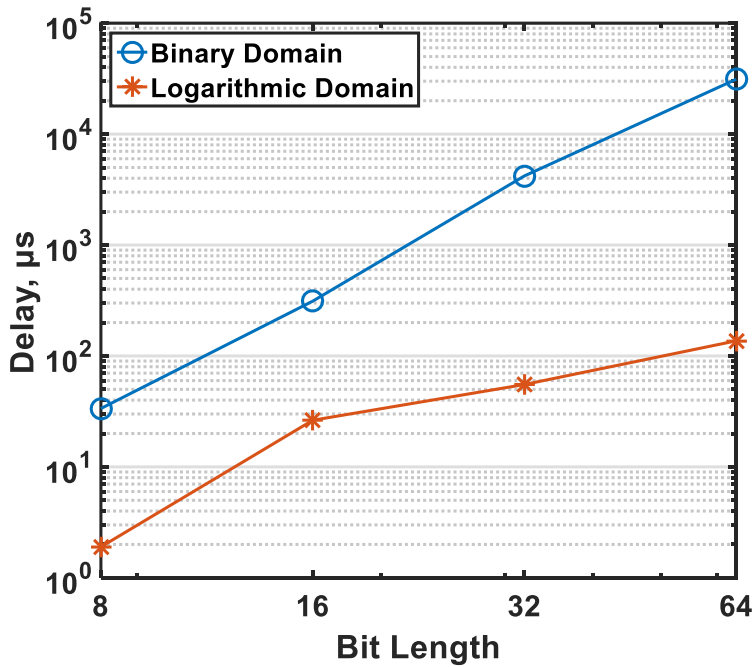
Signal process in binary domain is the industry standard of designing computing VLSI chips. The entire design flow, which is from the pre-silicon entry using hardware description language (HDL) or customized schematic to the final sign-off is maturely supported by advanced

electronic design automation (EDA) tools. However, with the overloaded computation introduced in recent applications, digital binary based computing has behaved low computing efficiency such as the multiplication with long bit length and non-linear computation.

In some specific applications, system can tolerate a little bit error that does not influence the performance visibly [27, 28]. Thus, some approximated computations can be applied in specific applications. A very interesting theory is that a binary number can be converted in the form of a logarithmic code which can be processed by the following task with conversion error. This theory has been used to simplify multiplication and division [29, 30]. Figure 1.4 shows the performance comparison of the full adder (FA) based multipliers in both binary and logarithmic domains under 100MHz operation. We observe that the multiplication in logarithmic domain can largely reduce power dissipation and computation latency when the bit length is increase. Several work have been done to reduce the conversion error, which is the main concern of the computation in logarithmic domain. One strategy is to use a multi-step function to further increase the conversion accuracy [31-34]. SanGregory et al. [35] used another curve to approximate the logarithmic curve instead of the linear function used for the generation of the fraction part. Iteration can mitigate the approximated error step-by-step at the cost of long computation time [36]. Lookup table (LUT) is also used to suppress the conversion error [37] but it introduces additional memory array. In addition, the search method for the logarithmic conversion has been studied with the purpose of effectively using LUT [38]. Programmable logic array (PLA) has also been used [39, 40]. This method can accelerate the design but it does not introduce the circuit optimization due to the restriction of already-defined logic gates in PLAs. For the VLSI circuit design, the leading-one detector (LOD) is widely used for determining both the integer part and fraction part [41, 42, 43]. The leading-one detector is the pipelined circuit using MUX and AND gates to search the first



(a)



(b)

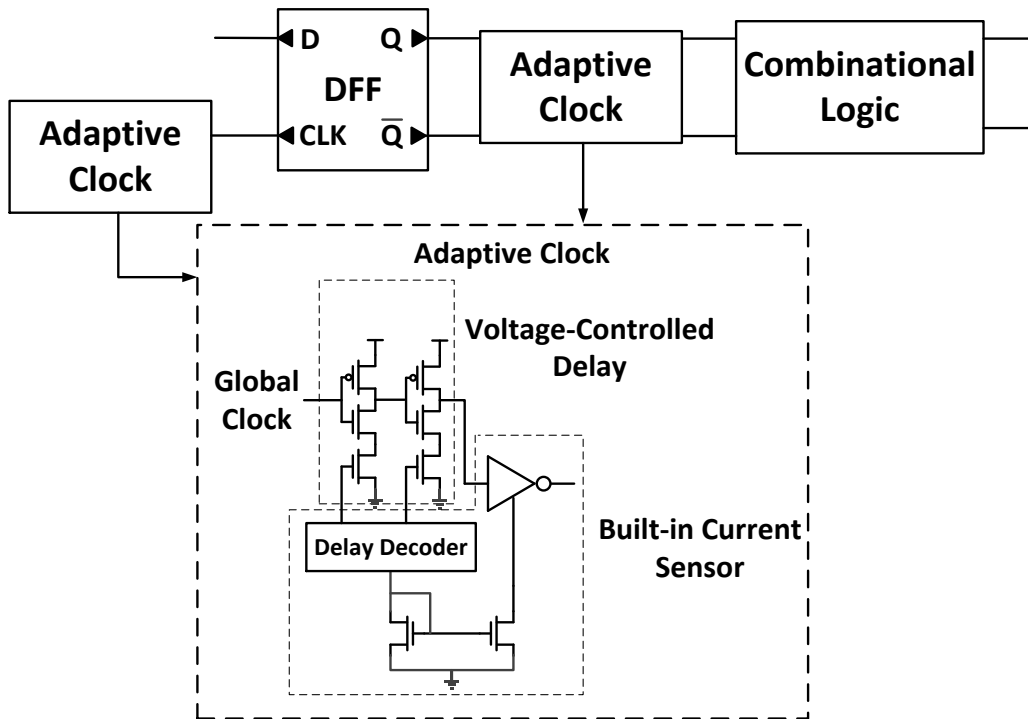
Figure 1.4 Performance comparison of the multiplication in both binary and logarithmic domains using TSMC 180nm technology, (a) power dissipation, (b) computation latency. Logarithmic conversion in this simulation is based on Mitchell's algorithm without any calibration.

high-bit for any given binary number. The computation time is highly dependent on both the word length and the location of the first high-bit.

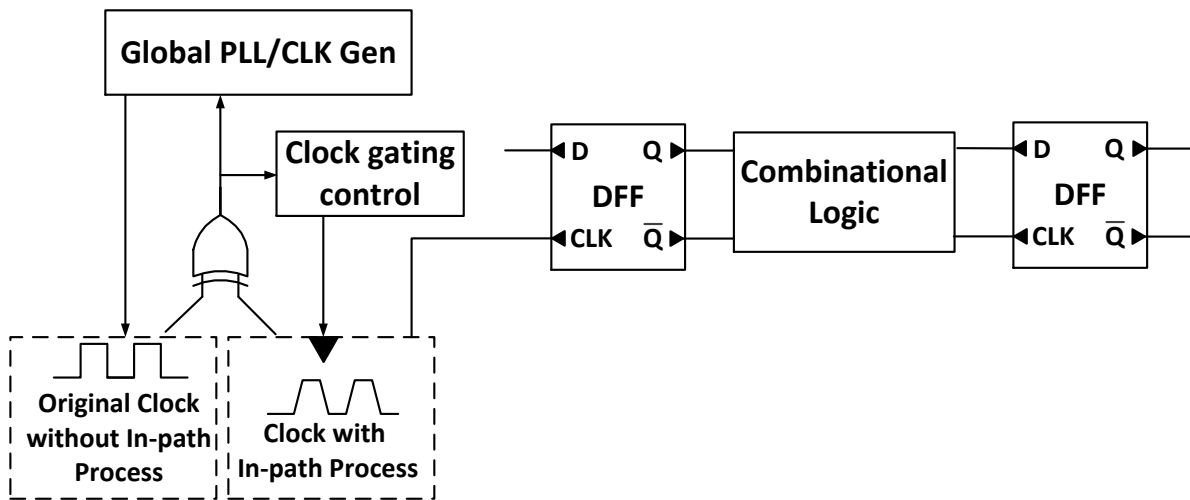
1.5 Overview of Optimization in ALU Design

ALU is one of the important blocks in present computer systems [44, 45]. Especially in graphics processing unit (GPU), products are embedded over 10 ALUs in order to strengthen computing ability [46, 47]. How to address the trade-off between several metrics such as power dissipation, computing latency and maximum fan-out is important. Based on this, ALU is designed also in silicon-on-insulator (SOI) process to effectively mitigate static power dissipation [48]. Furthermore, the application of dual-voltage has been reported to allow some non-critical blocks or paths to operate under lower power at the cost of computation speed [49]. Back gate forward biasing method has been used in ALU design [50, 51], which increases the speed by reducing transistors' threshold voltage. The method of bit-partition is used for reducing the pipeline stages to speed up the entire computing.

Another concern is how to suppress the possible occurrence of error during operation, adaptive clock has been utilized to avoid setup and hold violations [52]. The block diagram expressing this technique is shown in Figure 1.5 (a). It uses built-in current sensors applied in checkpoints to drive voltage-controlled delay block to prevent clock signal or logic signal reaching to the next flip flop early or lately. Besides, the ALU design using signal latency in internal computing flow as the detective signal for error calibration has also been proposed [53]. Figure 1.5 (b) shows this calibration method. It uses XOR gates to detect if the clock signal has deviation between two flip flops. Once clock error is detected, using clock gating, the internal combinational logic will be stopped to wait for new clock signal generated by phase lock loop (PLL). This techniques is highly suitable to be used for avoiding error due to clock skew and jitter. However,



(a)



(b)

Figure 1.5 Two digital calibration methods, (a) adaptive clock, (b) delay-aware feedback loop.

both of two described methods are available at checkpoint in critical paths considering circuit cost, which means errors occurring in other non-check paths or stages cannot be removed.

1.6 Circuit Design for Neural Computation

The hardware implementation of neural computation is widely studied. The main difficulties are design strategy of robust neural cells and how to address both matrix multiplication with large size and accurate activation functions in a given network at low circuit cost.

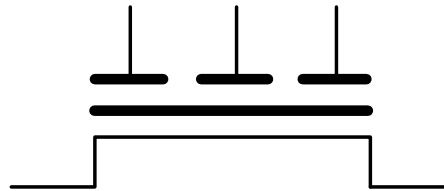
1.6.1 MIFG MOESFET based Neuron-Like Cell

The MIFG MOSFET has two gate layers [54]. The bottom one is the floating gate, which is over the transistor channel, and directly controls the conduction of carriers in the channel. The top gate layer is the input gate over the oxide. To obtain multiple inputs, the top layer is deposited as individual gates during fabrication so that each of them can serve as an input port of the device. The oxide layer isolating a floating gate and several input gates can be seen as an array of coupling capacitors to control the surface potential of the channel. Both device and equivalent circuit model of an MIFG MOSFET are shown in Figure 1.6. The on/off state of the transistor is decided by the surface potential on the floating gate similar to a traditional CMOS transistor. The surface potential itself is determined by both coupling capacitors and multiple inputs. For a single MIFG MOSFET, its surface potential, Φ_f , can be expressed as follows [55]:

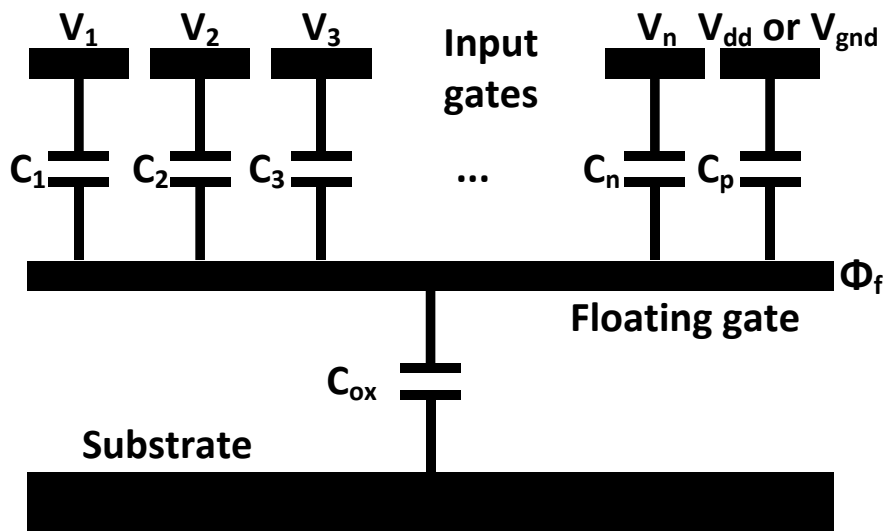
$$\Phi_f = \frac{C_1V_1 + C_2V_2 + \dots + C_nV_n}{C_{ox} + C_1 + C_2 + \dots + C_n + C_p} \quad (1.1)$$

where V_1, V_2, \dots, V_n represent multiple voltages which are inputs to top gate layer. Their corresponding coupling capacitors are C_1, C_2, \dots, C_n , all of which are contributed by the oxide layer isolating the floating gate and each top input gate. C_{ox} and C_p are the capacitors between the floating gate and the substrate, and parasitic capacitor, respectively. C_p can be safely removed

Multi-input



(a)



(b)

Figure 1.6 MIFG MOSFET, (a) device symbol, (b) equivalent circuit model.

since its value is much smaller than the rest of capacitors. For C_{ox} , when the dimension of an MIFG MOSFET is small, we can neglect C_{ox} . While in analog VLSI design using MIFG MOSFETs, some transistors' dimensions may be relatively large to assure that all transistors work in correct region of operation. In this case, C_{ox} should be seriously taken into consideration. Normally, one effective method to suppress the non-linearity from C_p and C_{ox} is to set C_1, C_2, \dots, C_n to large values.

For a binary decision neuron (BDN), its output can be expressed as follows:

$$Y = f\left(\sum_{i=1}^n W_i X_i - S\right) \quad (1.2)$$

where X_i is input signal and is weighted by W_i . S is the threshold value of BDN. $f(x)$ is the activation function, which outputs the final result. If we map W_i and X_i in Equation (1.2) to C_i and V_i in Equation (1.1), respectively and S in Equation (1.2) to the threshold voltage in a MOSFET, a circuit designed by MIFG MOSFETs can be built as a spike neuron. The designs of neuron-like cells have been reported [56, 57, 58], in which a transistor's state is decided by the multiple inputs so that the design diversity can be improved significantly, and thus the single block with multi-application can be achieved. The very obvious disadvantage of the MIFG MOSFET based circuit is the area cost due to coupling capacitors.

1.6.2 Implementation of Complex Neural Network

Convolution neural network (CNN), and LSTM network which is one of extensive recurrent neural network (RNN), are two typical algorithms with complex network, and overload matrix multiplications and activation functions. Current work in VLSI implementation of CNN are mainly based on digital binary domain [59-62]. The hardware implementations of LSTM mostly use field-programmable gate array (FPGA) platform [63, 64, 65]. The computation speed of LSTM in FPGAs is mainly restricted by both fixed digital gates operating at ~100MHz global

clock, and thus is lower than that in application specific integrated circuit (ASIC) design. Shin et al. [66] presented a neural processor that can implement LSTM in deep sub-micron CMOS technology. Especially, to avoid large latency introduced by matrix multiplication and activation functions in this work, both LUT and register array are used in advance to store all possible results of product and quantization codes. In conclusion, most of the practical work for neural computation use digital signal processing (DSP). Memristor based crossbar has been also introduced to speed up neural computation to transmit and process current signals with the aid of mixed signal CMOS circuits [67, 68]. However, to the best of our understanding, the compatibility of memristor and CMOS in fabrication is not yet established. Very little work [69, 70, 71] are reported on the fabrication of memristor based circuits with the verification of simple computation. It proves that current neural network with huge computation cannot be implemented by memristor if practical hardware implementation is to be considered.

Analog signal processing (ASP) is an alternative method to achieve neural computation. Neuron-like analog cells have been reported to implement neural computation [72-75]. To make ASP to work for complex neural networks reliably, Amant et al. [76] proposed an analog computation method. However, to match the external digital blocks, the signal interface implemented by digital-to-analog converter (DAC) and analog-to-digital converter (ADC), increases circuit area and power dissipation, and restricts the maximum frequency due to trading-off between effective number of bits (ENOB) and sample rate in ADC. In addition, it is very difficult to embed too many high-speed ADCs and DACs in a single VLSI chip due to the restriction of both power density and chip area. If we do not take signal interface into consideration and only compare between ASP and DSP addressing complex neural computation, ASP has several advantages over DSP:

1) A single analog signal can represent the digital signal with multiple bits after D/A conversion;

2) Current neural algorithms involve lots of non-linear activation functions, which are easier to be implemented in analog VLSI than that in digital VLSI;

3) Matrix multiplication, which is widely used in neural computations, implemented in analog design is much faster than that in digital design.

However, the use of DACs and ADCs as signal interface connecting to external digital blocks adds real estate to the chip, and computation latency.

1.7 Goals and Objectives

Several VLSI techniques used for high-performance computation are discussed in this dissertation, from the perspective of voltage regulation, adiabatic logic, computation beyond binary domain and MIFG MOSFET.

Chapter 2 develops a novel power regulation method called as ‘switchable pin’ to compensate IR droop resulted from the complex placement and routing at low circuit cost. Both simulation and chip test results are presented.

In Chapter 3, a novel calibration used for logarithmic conversion is proposed. A logarithmic converter under dual-mode working (ECRL and DCVSL) is designed. Both simulation and chip test results are presented.

Chapter 4 explains a novel design of the ALU using MIFG based neural-like cells. Both post-layout simulation results and sensitivity study are reported.

Chapter 5 uses ASP with MIFGs to design a LSTM network being compatible to digital blocks. A novel signal interface without traditional ADC or DAC is also introduced. Both post-layout simulation and sensitivity study are reported.

Chapter 6 provides a summary of the work presented and scope for future work.

Appendix A is the MOSIS level 7 CMOS model parameters of AMI 500nm and TSMC 180nm technologies for *PSpice* simulations. Appendix B summarizes the proposed design flow of APR with Verilog entry and customized schematic entry using *Mentor Graphics* EDA and *PSpice* tools under *Window* operation system. Appendix C describes the method of error test in *PSpice* simulations.

CHAPTER 2

SWITCHABLE PIN USED FOR POWER REGULATION IN CHIP-MULTIPROCESSOR

From the perspective of voltage regulation working for current VLSI chips with multi-core, complex placement and routing lead to irregular power supply, and thus obviously affect the reliable computation in chips. This work will present a low circuit cost method using the proposed switchable pins to dynamically compensate IR droop and transmit data inside die. The main contribution lies in the analysis of the IR droop in chip-multiprocessors, power delivery network (PDN) modeling, customized pad design, post-layout simulations and fabricated chip testing.

2.1 Analysis of IR Droop

Complex placement and routing, finite width of wire and contact resistance will cause voltage loss in chips. Previous theoretical study has proved that voltage droop can influence the performance of data transmission [77]. The current integrated circuit (IC) fabrication allows multi-layer process in silicon wafers to release the pressure of routing and placement. But complex interconnect/contact and inductor effect under high frequency degrade voltage distribution in chips. Figure 2.1 shows the cross-section view of power supply interconnection using multiple layers. The voltage loss can be described by the following equation:

$$V_{loss} = \sum_{i=1}^{m-1} R_{c_i} i_w + \rho_w \frac{l}{w} i_w + Z_L i_w \quad (2.1)$$

where m is the number of metal layers, R_{c_i} is the contact resistance, i_w is the current flowing

Part of work is reported in following publications:

1. Z. Zhao, A. Srivastava, L. Peng, S. Chen and S. P. Mohanty, "A novel switchable pin method for regulating power in chip-multiprocessor," *Integration, the VLSI Journal*, vol. 58, pp. 329-338, 2017.
2. Z. Zhao, A. Srivastava, L. Peng, S. Chen and S. P. Mohanty, "Circuit implementation of switchable pins in chip multiprocessor," *Proc. 2015 IEEE International Symposium on Nanoelectronic and Information Systems (iNIS)*, Indore, India, pp. 89-94, December 2015.

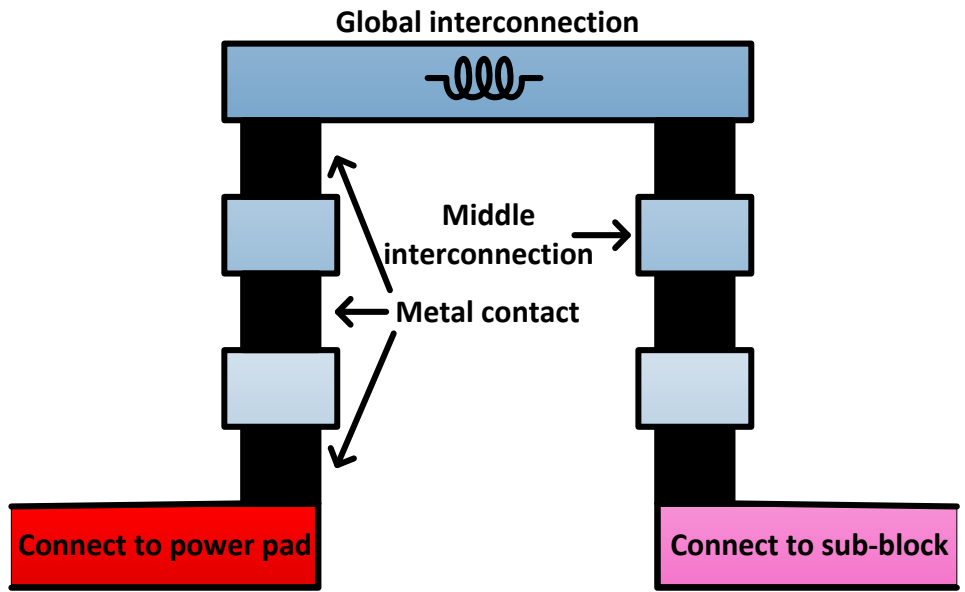


Figure 2.1 Cross-section view of power supply interconnection using multiple layers.

through the wire, l and w are the length and width of the globally top wire interconnected, respectively. Z_L is the impedance of inductor in globally top interconnection. The three terms in Equation (2.1) reflect the voltage loss contributed by contacts, wire resistance, inductor effect under high frequency, respectively.

To estimate the voltage loss in chips, we make following considerations:

1) The parameters of fabrication process we use for estimation is Predictive Transistor Model (PTM) [78];

2) The global wire serving for power supply is straight without complex rotation, which means we ignore mutual inductance effect between neighboring wires. The parameters of interconnect for the selected fabrication process are extracted as in [79];

3) We consider 10 multi-layers in a chip. The top and bottom layers are used as global layer and the layer supplying power to sub-block, respectively;

4) The voltage loss due to inductor at high frequencies includes dynamic loss and static loss as well [80]. Here we only focus on static loss, which means for the loss due to inductor effect, absolute value of impedance is taken into estimation without the consideration of phase;

5) We select a virtual core which has 100 million transistors. For each transistor, we use the minimum dimension in the selected process. Considering gaps between adjacent transistors/sub-blocks, the area of the virtual core is close to 1.2×100 million transistors. For a single transistor, DRC rules cannot be neglected, we explore currently mainstream layout of a single transistor with the strategy of saving area, the area of a single transistor is $(1.5 \times l) \times (3 \times w)$;

6) Largest voltage loss happens in the geometrical center of the virtual chip;

7) For a virtual chip-multiprocessor, we set that there are 2, 4, 6, 8 virtual cores corresponding to 32nm, 20nm, 10nm, and 7nm technologies, respectively.

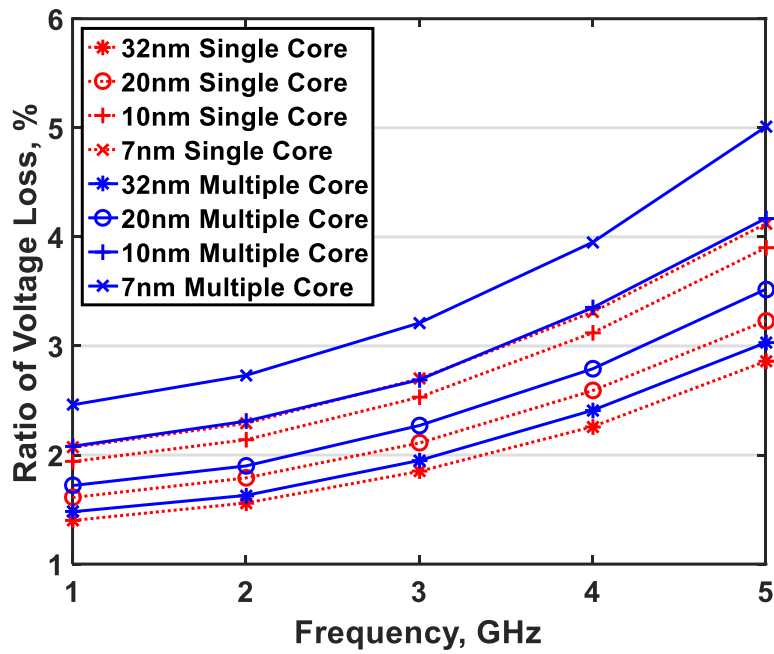
Guided by above considerations from (1) to (7), we can calculate the voltage droop in the virtual single core and chip-multiprocessor using various sub-nanometer technologies, as shown in Figure 2.2 (a). We observe that the voltage loss increases with the frequency due to inductance effect. It is anticipated that if emerging chips work under ultra-high frequency, or have too many layers, voltage loss will be continuously increased. Another issue which needs to be analyzed is the relationship between voltage loss and computation latency in chips. In this work, we mainly study how voltage loss influences rise and fall times, which are significant factors determining computation latency. The mathematical relationship can be shown as follows [81]:

$$t_f = t_r = \frac{2C_L}{\beta(V_{DD} - |V_{TH}|)} \left[\frac{|V_{TH}| - 0.1V_{DD}}{V_{DD} - |V_{TH}|} + 0.5 \ln \left(\frac{19V_{DD} - 20|V_{TH}|}{V_{DD}} \right) \right] \quad (2.2)$$

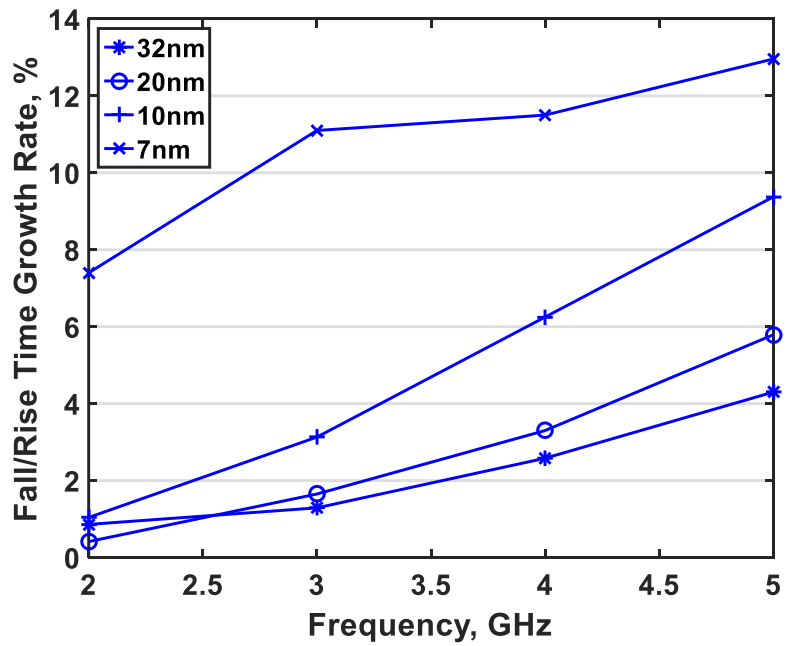
Current digital VLSI is based on static logic and the load capacitance is mostly contributed by the equivalent gate capacitance of the next stage. Figure 2.2 (b) shows the growth rate of computation latency calculated at 1GHz using PTM under the influence of voltage loss. It can be concluded that if no voltage compensation is used for VLSI chips, voltage droop will seriously influence their computation performance. The mainstream strategy to avoid this problem is embedding in-chip voltage regulators. However, this method is at the cost of power dissipation and real estate.

2.2 Overview of Switchable Pin

Normally, a complex function digital VLSI chip has one power pad, one ground pad, one clock pad, and several I/O pads used for data communication. This feature can maximize the bandwidth of data transmission for a specific package. However, only one pair of pads used for power supply cannot guarantee each sub-block to work under a perfect voltage as analyzed in Section 2.1. An interesting observation is that for a processor, it does not always need large bandwidth due to different states [82]. Under some specific instructions, some data pins will be in



(a)



(b)

Figure 2.2 IR droop simulations, (a) voltage loss ratio depending on various processes, (b) growth rate of fall/rise time depending on various processes at 1GHz.

idle state. That is where we can bring over the novel switchable pin concept. We can set several I/O pads dynamically switch between traditional data transmission and power supply. During the time of large bandwidth required, the I/O pad behaves like the normal one. While the required bandwidth is reduced, some of I/O pads will be changed to power pads to compensate the voltage loss in chips.

The verification of the correct data transmission between processors and memories is the key problem when applying switchable pins in system-on-chip (SoC). Our solution is that when some of I/O pads are used for power delivery, the rest of I/O pads not only still work for the original data lines, but also need to transmit the data from the switched lines. To achieve this method, two groups of paralleled switches, which are located in-chip and off-chip, are needed. Figure 2.3 shows the conceptual diagram of our design. We define two modes which are data mode and power mode. In data model, only S_{data} is enabled, the entire system works as usual, that the original core is supplied by one power pad and all I/O pads are working for data transmission, and the supported core is in idle state. In power mode, both S_{power} and S_{data_i} are enabled. Our switchable pins begin to work, some of I/O pads belonging to the original core will start to be used as virtual power pads for the supported core. In this mode, step-by-step data transmission in the original core is achieved by limited I/O pads with the aid of clock shifter. We set the off-chip switches thereby letting all of storage units receive data in power mode step by step. The signal from each data line has to pass three switches at the cost of delay. But this method avoids the modification of both decoder and controller in memory.

There are two feasible options including tri-state buffers, and CMOS TGs to achieve switchable function. A tri-state buffer can perform better signal integrity than the CMOS switch at the latency cost. However, we cannot ignore the delay issue. Our modification between in-chip

and off-chip has brought large delay to data transmission. Thus, we make a tradeoff that we choose a CMOS TG with a buffer instead of a tri-state buffer to balance delay and signal integrity.

For the right group with off-chip switches shown in Figure 2.3, we have two strategies. First method is adding an independent chip of paralleled switches between processor and memory. The advantage of this idea is to keep away from the modification of memory so as to reduce the difficulty encountered in memory design. We define this as a ‘three-chip’ model. The second one is embedding paralleled switches into memory chip. This can reduce delay between processor and memory intuitively since it eliminates an extra package. We define this as a ‘two-chip’ model.

2.3 Performance Evaluation using PDN

For the modeling of PDN, in [83], the core is disassembled to build a detailed network for in-chip power distribution, while in [84], a systematic model is proposed, which includes chip package and PCB. We mainly predict the signal integrity of the proposed idea, we assume that all signals that start from the output of in-chip and end in the I/O pins of memory are ideal. Based on this assumption, monitoring signals going through package and PCB are much more important than that going through in-chip circuits. Thus, the systematic model is suitable for our prediction.

We build two models corresponding to ‘three-chip’ model and ‘two-chip’ model, respectively. Figure 2.4 shows two proposed models and an ideal model without switchable pins. To determine each parameter in our models, we mainly use datasheet of TSMC 180nm fabrication process to set pad frame model. For PCB environment, we use classical C4 package, which is still the mainstream package. We use the work in [85] to set our C4 package including bump and its wire. For via, we use the industry standard model [86]. The major parasitic parameters in via are inductance and capacitance. These two parameters can be calculated as follows [87]:

$$C_{via} = \frac{1.41\epsilon_r TD_1}{D_2 - D_1} \quad (2.3)$$

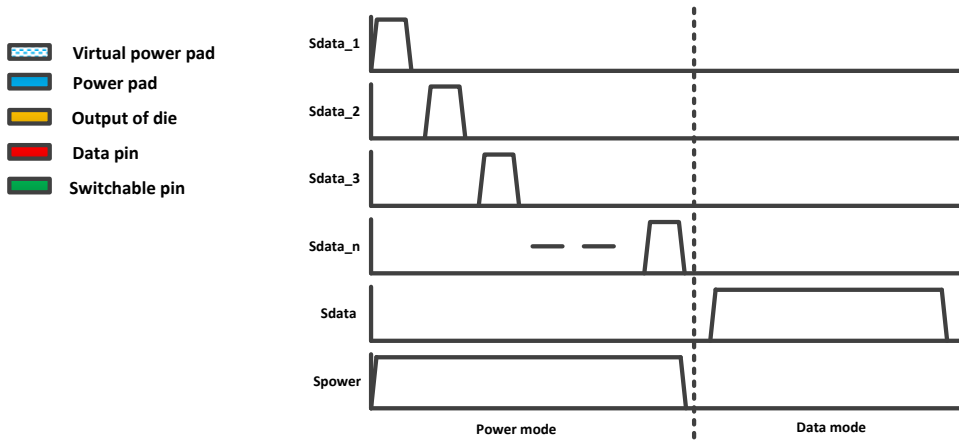
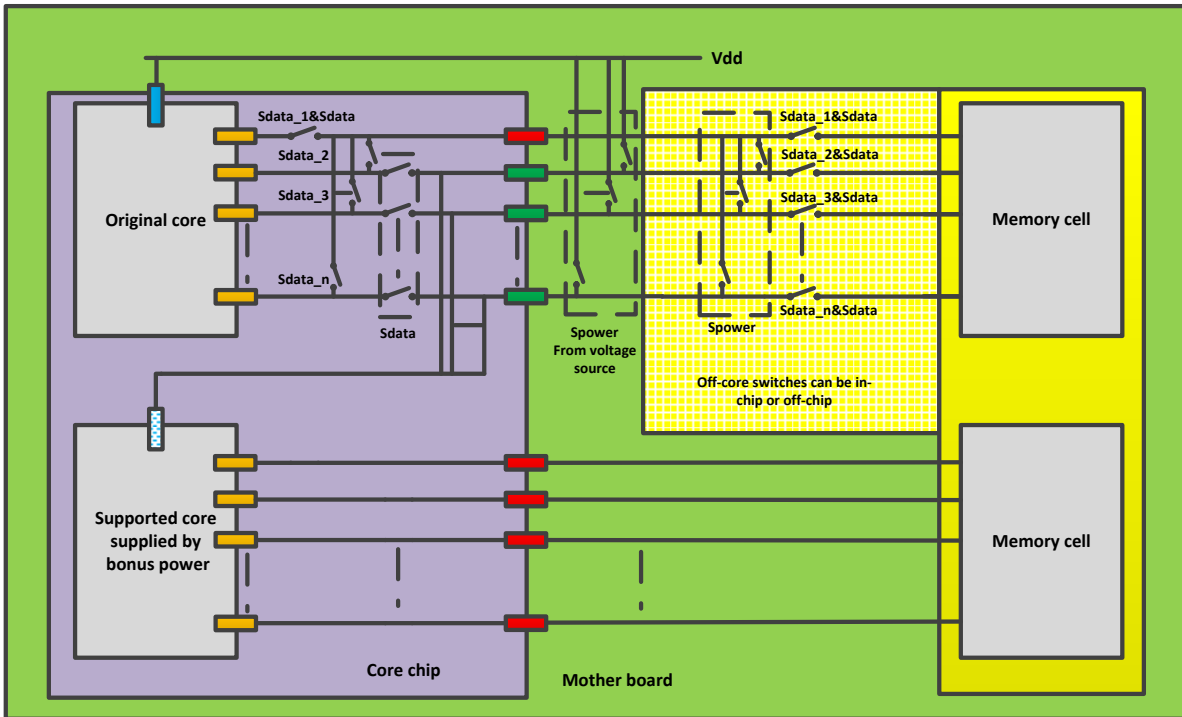
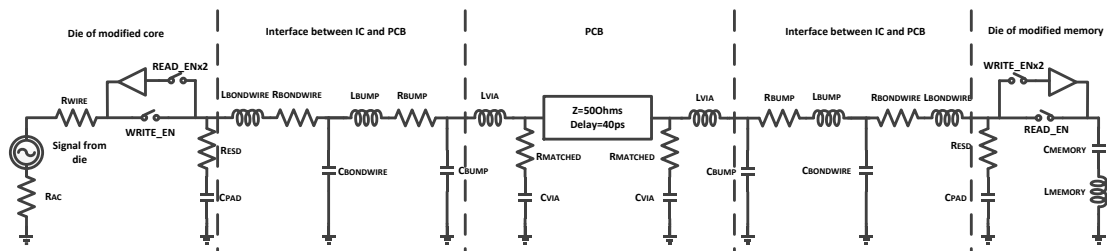
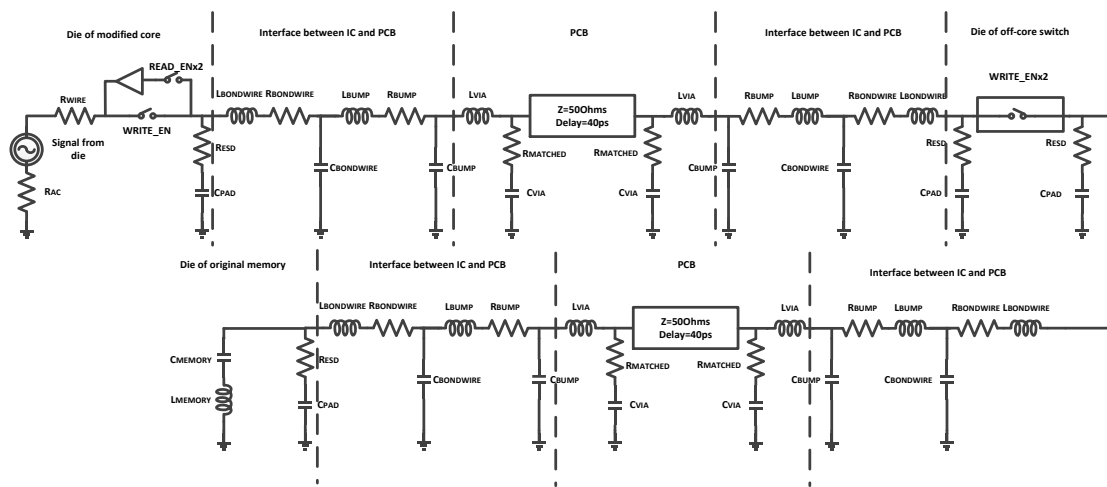


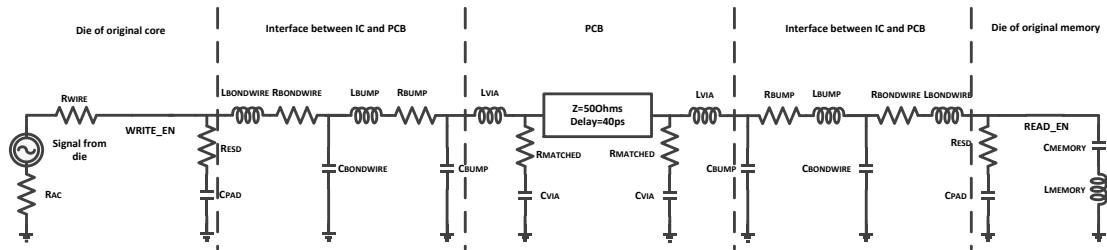
Figure 2.3 Conceptual diagram of SoC with switchable pins.



(a)



(b)



(c)

Figure 2.4 Power delivery network, (a) 'two-chip' mode, (b) 'three-chip' mode, (c) normal mode without switchable pins.

$$L_{via} = 5.08h \left[\ln \left(\frac{4h}{d} \right) + 1 \right] \quad (2.4)$$

where D_1 and D_2 are the diameters of via pad and anti-pad, respectively, h is the length of via, d is the diameter of via barrier and T is the thickness of PCB.

Properly using the dimensional restriction of PCB design, we define each dimensional parameter of via as shown in Table 2.1. Then we can obtain capacitance is 0.5pF and inductance is 1.02nH. The entire parameters in our models are shown in Table 2.2. Figure 2.5 shows the simulation results of signal attenuation. We notice that the performance of data transmission in WRITE mode is better than that in READ mode. For READ mode, signal attenuation drops quickly when frequency exceeds 1GHz. This is acceptable since that current mainstream DDR3 only runs under 900MHz [88]. From the view of signal integrity, ‘two-chip’ model is a good choice. The cost of this method needs to add paralleled switches into memory in order to confirm correct data transmission. But we do not need to modify decoder and controller in memory as mentioned before. We also plot the performance of the model without switchable pin. It can be seen that our switchable pin keeps the signal integrity in acceptable range, especially at low frequency operation.

2.4 Customized Design of I/O Pad

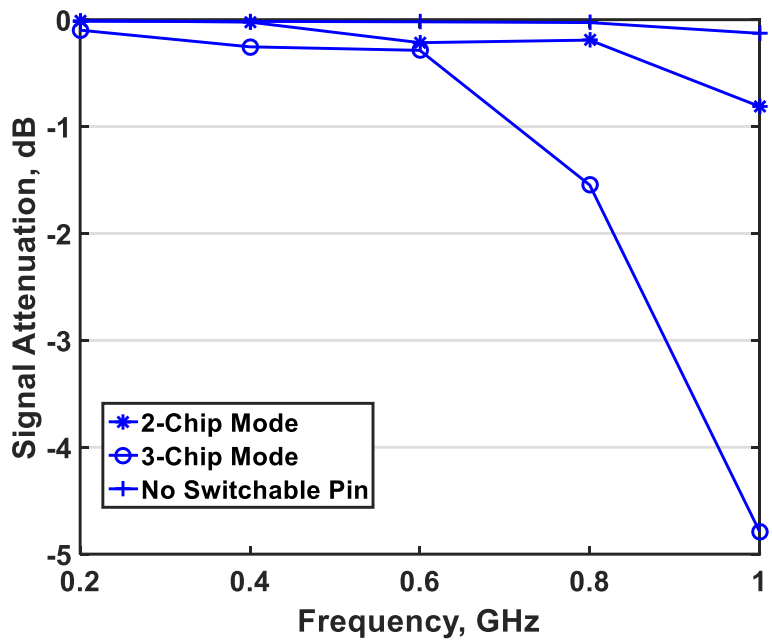
We have discussed how switchable pin works in SoC and modeled its PDN. One thing cannot be ignored is that, unlike the traditional function of I/O pads, which connect the gates of transistors to drive logic circuits, or the output stage in chip, in our work, some I/O pads will dynamically connect the power node of circuits. Since a CMOS transistor is a voltage-control device, in which the resistance of gate is extremely high, current through the gate is negligible. But the route of power supply will generate large dynamic current in complex logic processing. For achieving switchable function, the bi-directional pad seems to be a candidate. When power supplies to in-chip, the p-MOSEFT in buffer will tolerate huge current like power pad does.

Table 2.1 Summary of dimension in via

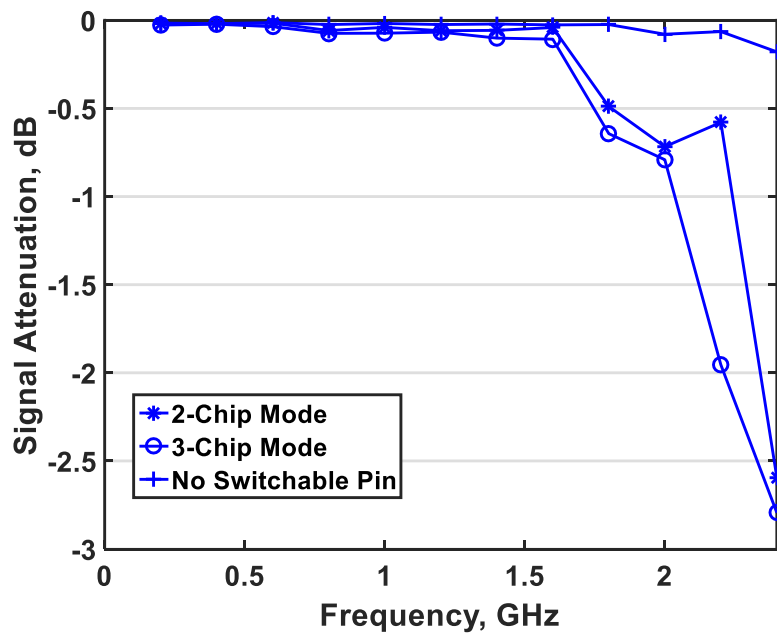
ϵ	4.4F/m
T	50mm
D ₁	20mm
D ₂	32mm
d	10mm

Table 2.2 Summary of parameters in PDN

PAD	R _{ESD}	50k Ω
	C _{PAD}	250fF
BUMP	L _{BUMP}	60pH
	R _{BUMP}	30m Ω
	C _{BUMP}	0.2pF
BOND WIRE	L _{BONDWIRE}	2.58nH
	R _{BONDWIRE}	90m Ω
	C _{BONDWIRE}	0.02pF
VIA	L _{VIA}	1.02nH
	C _{VIA}	0.5pF
TLINE	Delay	40ps
	Impedance	50 Ω
	R _{matched}	500k Ω
DRAM	L _{DRAM}	0.5nH
	C _{DRAM}	300fF
CORE	R _{AC}	50 Ω
	R _{WIRE}	100 Ω



(a)



(b)

Figure 2.5 Simulation results of signal attenuation using the proposed PDN, (a) READ, (b) WRITE.

However, a normal transistor cannot drive that large current. Based on this analysis, original bi-direction I/O pad is not suitable for our design.

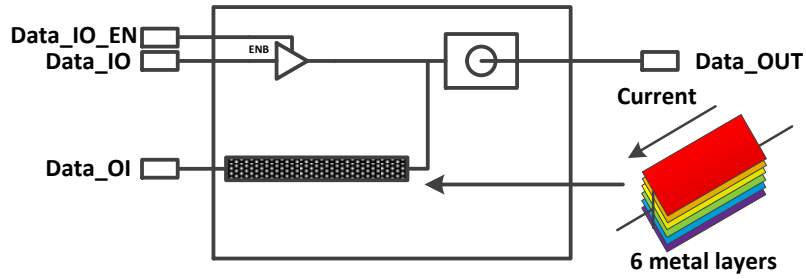
This can be addressed by modifying p-MOSEFT in buffer design as follows:

1) We can use numerous transistors in parallel to average a huge current going through each one at the cost of pad area;

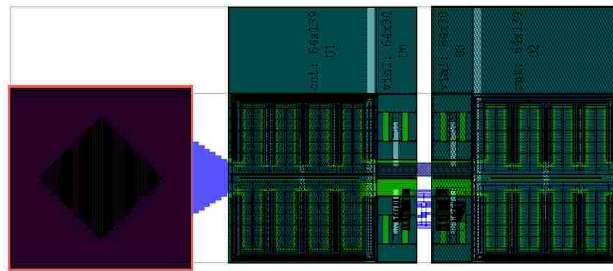
2) To tolerate huge current, the power MOSEFT is a good choice [89]. But the delay in power MOSFET is larger than that in traditional MOS transistor [90];

3) For the current technology of IC package, analog signal transmission in pad frame can be achieved by metal interconnection without logic gates. The allowable current in analog I/O pad is much larger than that in digital I/O pads.

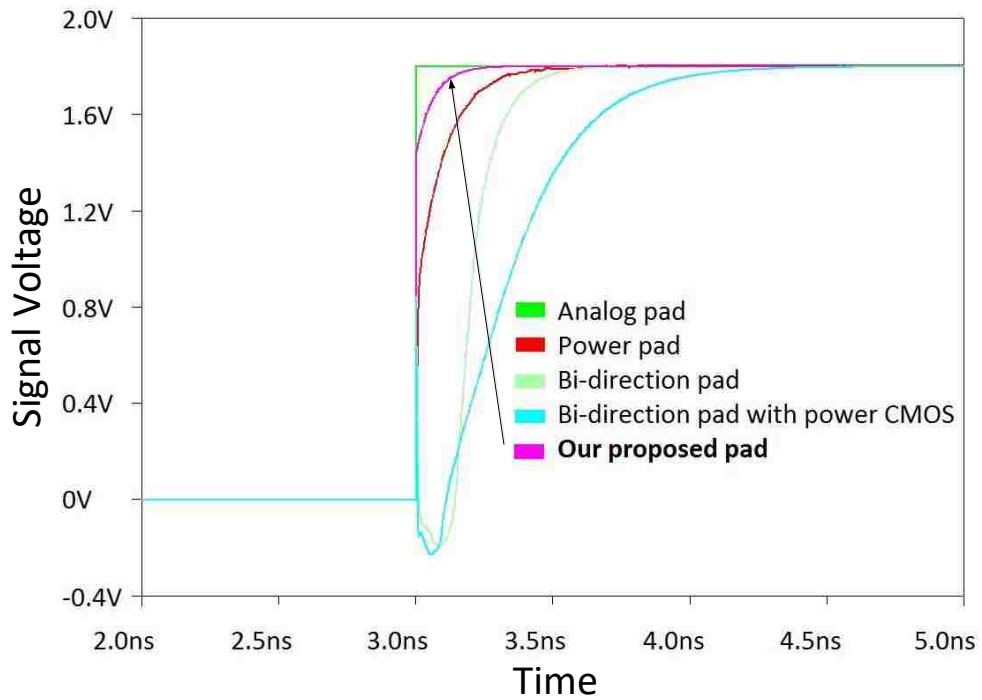
Deep looking into mainstream analog pad, typically only one layer metal is enough. To safely drive a chip reducing the risk of pad breakdown, we integrate six wide metal layers to let current go through. We also need to control signal direction by the proposed pad. The novelty of our following approach lies in combining a traditional bi-direction I/O pad and an analog pad with multiple metal layers. The circuit diagram and the layout view of the proposed pad are shown in Figure 2.6 (a) and (b), respectively. In the modified I/O pad, we set two routes, one is traditional output port using a tri-state buffer and the other one is six metal layers used for the power supply. When *Data_IO_EN* is high, the tri-state buffer is enabled to make signal going from in-chip to off-chip. During *Data_IO_EN* is low, the data route will be blocked, and off-chip power will drive the supported core. It can be concluded that this redesign can tolerate a large current due to in-chip gates through six metal layers overlapping at the cost of signal integrity. When data normally come from off-chip to in-chip, there is no buffer to buff the signal. To evaluate the response time of the proposed pad, we compare the delays of analog pad, traditional bi-direction pad, power pad, and



(a)



(b)



(c)

Figure 2.6 Redesigned I/O pad used for both data transmission and power supply, (a) logic diagram, (b) layout view, (c) delay comparison with standard cells.

power MOSFET based pad and our redesign as shown in Figure 2.6 (c). From the results, we can see that our design is faster than bi-direction pad, power MOSFET based pad and power pad. Even though analog pad is the fastest, our modification can confirm for the signal integrity from in-chip to off-chip with direction control. Since our modification only occurs in the logic block of I/O pad without the change in electrostatic discharge (ESD) part, there is no risk of transistor breakdown due to unwanted spike voltage.

2.5 Circuit Verification

2.5.1 Post-Layout Simulations

For the circuit design, we used the openMSP430 (a 16-bit microcontroller) [91] as the single core. The design technology is TSMC 180nm CMOS. The design flow is briefly explained in Appendix B.1.

The layout diagram is shown in Figure 2.7, in which 12 of 16 I/O pads are switchable pins. We design two single cores referring to an original core and a supported core. One real power pad only serves for the original core. The supported core will be powered under power mode using switchable pins.

The additional circuits to control switchable pins include paralleled switches and clock tree to shift enable signal achieving step-by-step data transmission during power mode. The resistance of the switch should be as small as possible. However, the dimension of the switch does not need to be very large with the consideration of delay. Clock block is designed by paralleled shift registers as in our previous work [92]. To precisely recognize and control two modes, we add a non-overlapping block [93] in the end of shift register to avoid two modes working at the same time causing data competition.

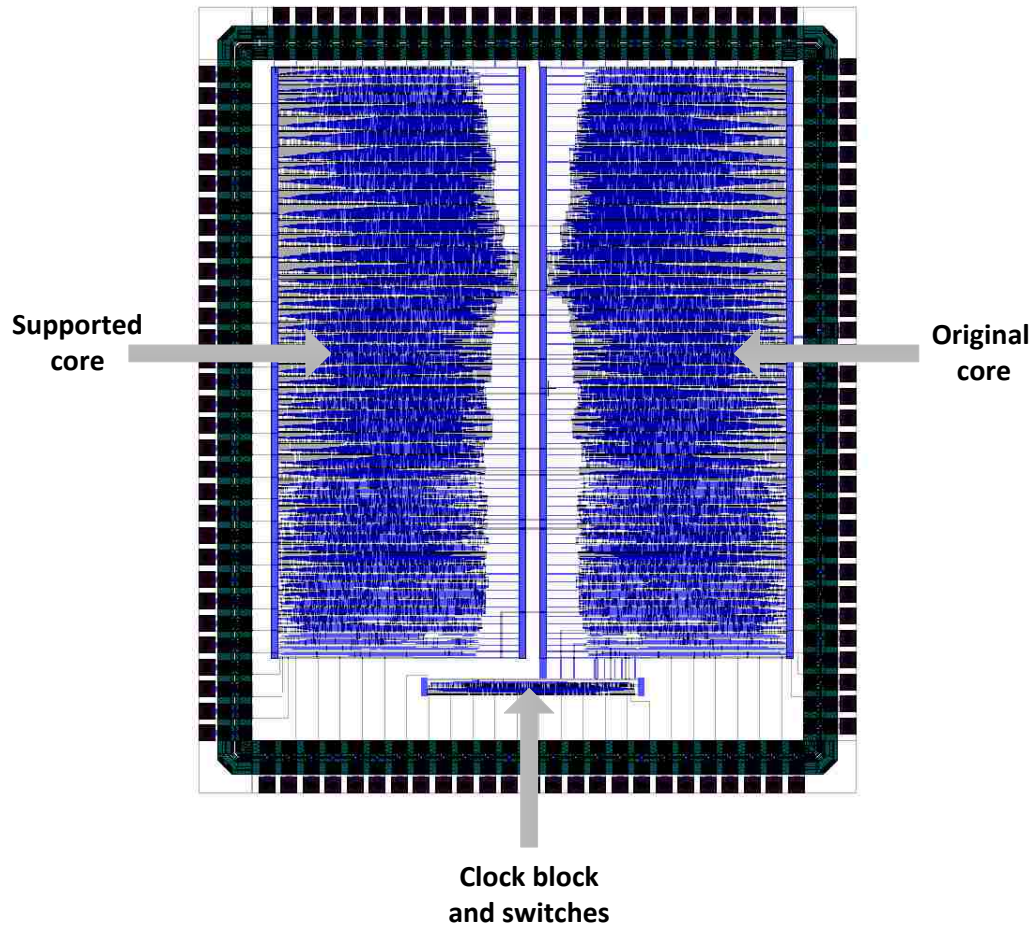


Figure 2.7 Layout view of two MSP430 cores, in which 12 of 16 I/O pads are modified as switchable pins.

Table 2.3 Summary of testing cases

	N_{swp}	$N_{\text{data_busy}}$	$N_{\text{data_normal}}$	$N_{\text{data_power}}$
SWP_8	8	8	0	2
SWP_10	10	5	1	3
SWP_12	12	4	0	4
SWP_14	14	2	0	8

How many switchable pins can work properly for the given chip deserves to be studied. We define the number of switchable pins as N_{swp} , the number of unchanged I/O pads helping data transmission of switched lines in power model as N_{data_busy} , the number of unchanged I/O pads still working normally in power mode as N_{data_normal} . For those pads serving for the data transmission of switched lines in power mode, each pad is responsible for transmitting N_{data_power} data sources (include the data belonging itself). We also define multiple cases corresponding to the number of switchable pins in the chip. The principle is that we fully use those unchanged I/O pads for data transmission in power mode as much as we can. Guided by this, several cases are described in Table 2.3.

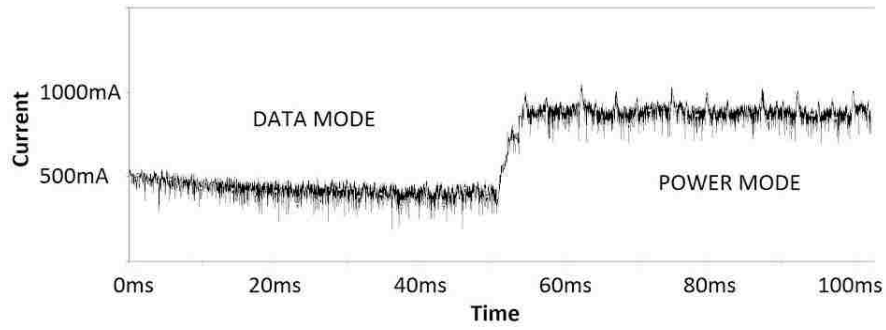
For area issue, the two cores occupy around 6.63mm^2 and the extra area brought by extra circuits corresponding to SWP_8, SWP_10, SWP_12, and SWP_14, are 0.1257mm^2 , 0.1263mm^2 , 0.1281mm^2 , and 0.1319mm^2 , respectively.

For the post-layout simulations of power issue, we input random signals to all input ports to obtain various dynamic current going through the single core. Figure 2.8 (a) shows the dynamic current simulation in the case of SWP_8. In power mode, the entire power can be roughly doubled compared to data model due to the fact that two cores are the same so that the current generated by each core is approximately identical. For different cases, current in additional blocks is always much smaller than that in two cores. Thus, no matter how many I/O pads are used as switchable pins, power is always seen to be doubled roughly when the chip is working under power mode. So this proves that our switchable pins can be seen as a dynamic power supply as the traditional power pad does. The response time for mode transmission is very small. Only if we do not need very fast transmission between two modes, this mode delay will be ignored. Figure 2.8 (b) shows extra power dissipation introduced by clock block and switches corresponding to four cases which are

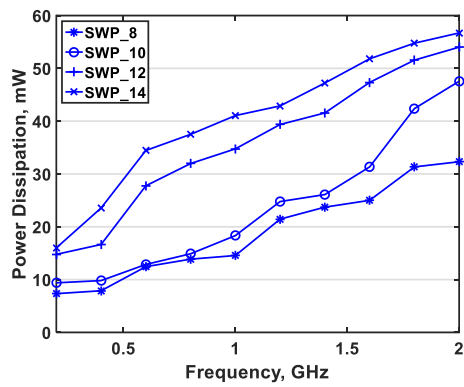
described in Table 2.3. From the results, we can see that extra circuit does not bring large power dissipation to the entire system, and overall power dissipation of extra circuits does not exceed 7% of the whole power dissipation of the entire chip.

For the issue of voltage compensation, we mainly monitor the voltage distribution in the supported core which is powered by the switchable pins and far from power pad. Thus, we let the supported core be supplied by both power pad and switchable pins to compare voltage distribution. APR in *Tanner Layout-Edit* is row-by-row style. Thus, we sample the supplied voltages of all rows under both power pad and switchable pins. Using these sampled supplied voltages, we compute mean value and standard deviation under several cases as shown in Figure 2.8 (c) and (d). We can see that the proposed switchable pin does compensate IR droop, and thus provides the supported core better power supply than the power pad does.

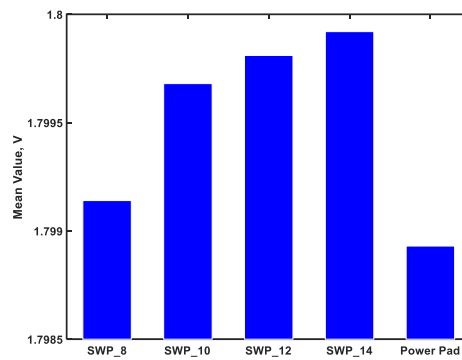
Signal integrity is another important issue evaluating the proposed switchable pin. Additionally we define another two cases, SWP_0_C1, and SWP_0_C2. SWP_0_C1 which represents only one core in chip without switchable pin. SWP_0_C2 refers that both cores supplied under only one power pad normally without switchable pin. For all cases, we need to compare WRITE and READ in both power and data modes. Figure 2.9 shows our test results. We see that the performance of WRITE stage is better than that of READ stage, which follows the previous PDN simulations. From the circuit point of view, this phenomenon is due to, in WRITE stage, data that goes from in-chip to off-chip, is buffered via tri-state buffer in pads. In READ stage, there is no buffer block in pads since we modify the original I/O pad. Thus, data going from off-chip to in-chip is not as clean as data with inverse flow. Comparing power mode and data mode, data integrity in data mode is better than in power mode. It is obvious that in data mode, the entire system works like a normal one without switchable pins, only some turned-on CMOS switches



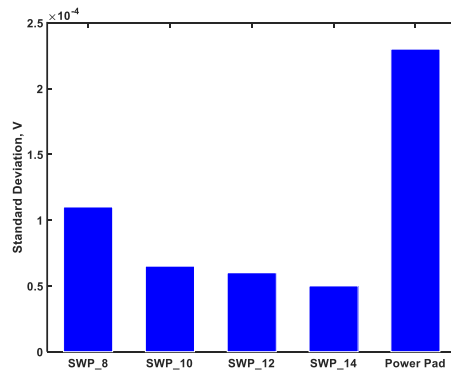
(a)



(b)

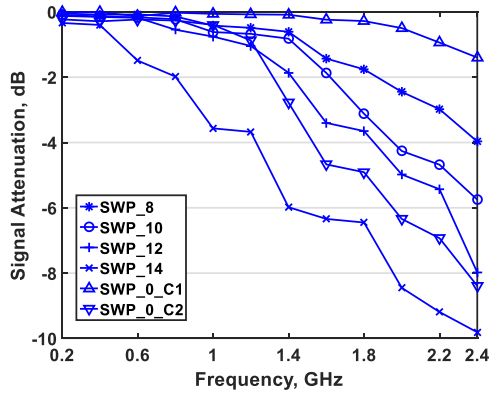


(c)

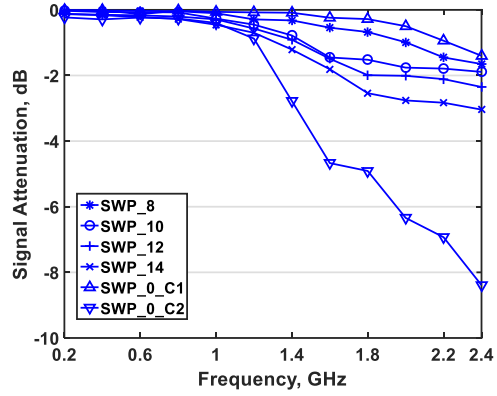


(d)

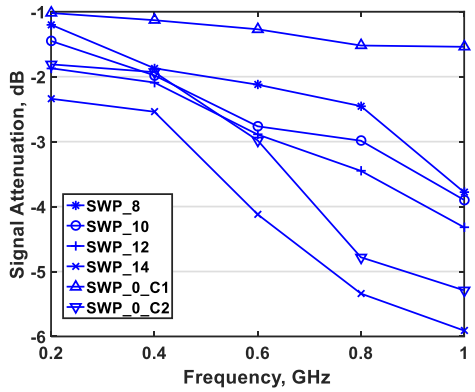
Figure 2.8 Power simulation, (a) current simulation in both data mode and power mode, (b) power dissipation of extra circuits serving for switchable pins, (c) mean value of supplied voltage under power pad and switchable pin supply, (d) standard deviation of supplied voltage under power pad and switchable pin supply.



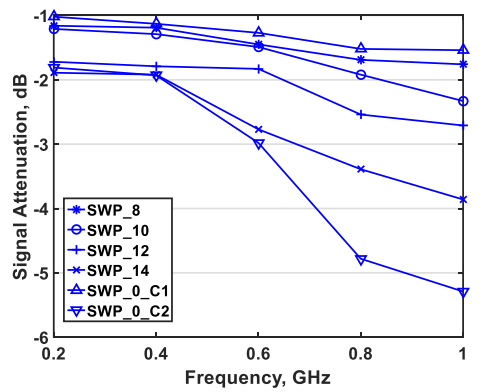
(a)



(b)



(c)



(d)

Figure 2.9 Signal integrity simulation, (a) WRITE in power mode, (b) WRITE in data mode, (c) READ in power mode, (d) READ in data mode.

contribute delay to data transmission. In power mode, only limited I/O pads are in charge of data transmission. One data line needs to transmit data from several data sources in one period. Thus, inadequate charging and discharging will happen resulting in imperfect signal. Analyzing all of cases in this work, single core chip without switchable pin performs the best. The chip with switchable pins gets an acceptable performance under low and medium clock frequency. Another observation is that, with the increase of switchable pins, the signal attenuation becomes larger due to the shorter time used for data transmission of a single line. Looking into our results, we conclude that SWP_8, SWP_10, and SWP_12 can perform with acceptable signal integrity in both WRITE stage and READ stage. However, in the case of SWP_14, signal attenuation is very large since one I/O route is needed to transfer 8 data routes in power mode.

2.5.2 Chip Testing

Besides post-layout simulations, we also fabricated a 4-bit ALU with the proposed switchable pin to verify the desired function in 500nm CMOS. Figure 2.10 (a) and (b) show its layout view and die photo, respectively. In the design, the 4-bit ALU is separated to two cores, one 2-bit ALU is powered by normal power pad and the other one can be supplied under normal power pad or the proposed switchable pin. Table 2.4 lists the pin location of the fabricated chip. VM, VS0 and VS1 are to decode logic/arithmetic operations. VSW is the switchable pin which can supply power. VDD_EN and VSW_EN are two control signals to decide how to supply the chip inside. Figure 2.11 (a) shows the transient simulations of two cores with the same inputs. We can observe that both two cores can perform correctly in the viewpoint of logic processing. The delay of the core supplied under power pad is larger than that under the proposed switchable pin. Figure 2.11 (b) shows the detailed plot of delay comparison. The delay difference is around 1.97ns. This is resulted from the effective IR compensation contributed by the switchable pin, which can be

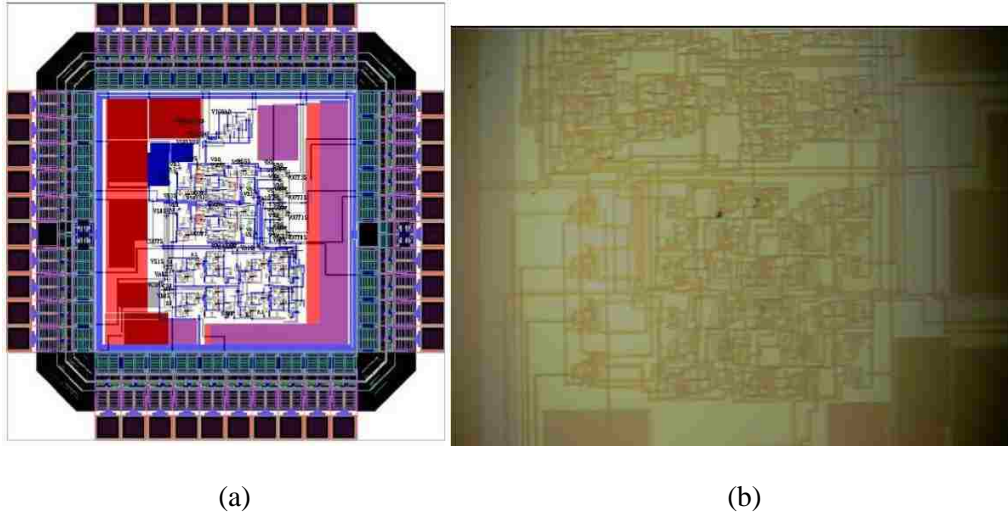
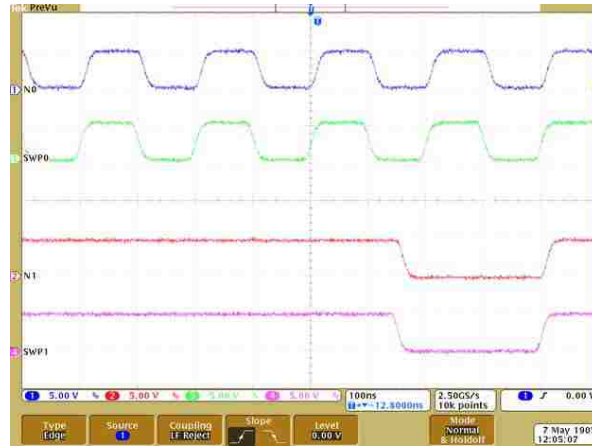


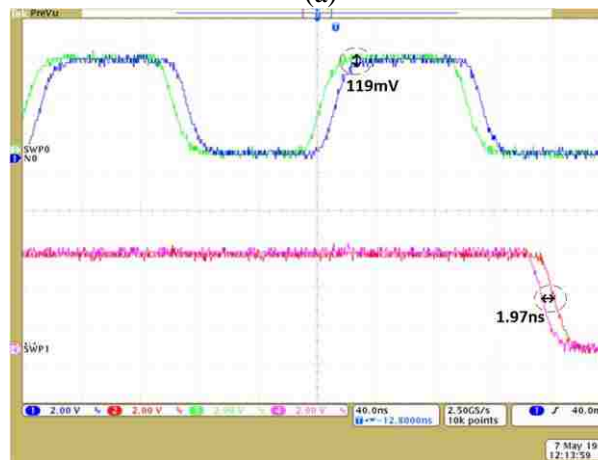
Figure 2.10 Fabricated chip with switchable pins, (a) layout view, (b) die photo.

Table 2.4 Pin allocation of the chip in Figure 2.10

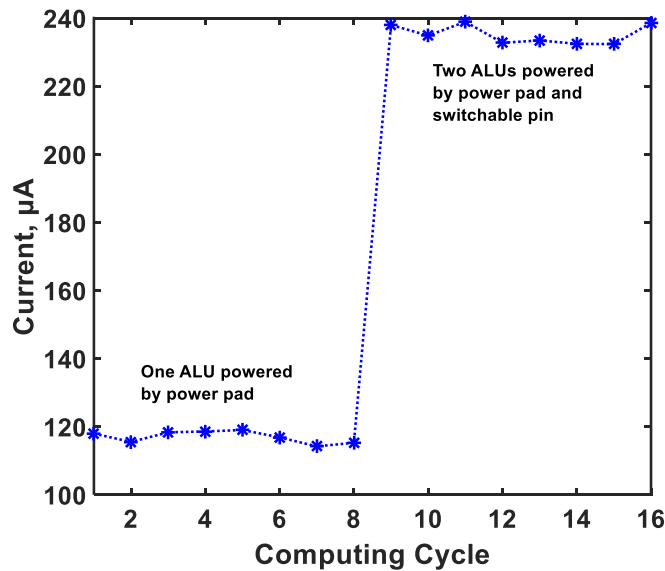
VA3	17	VO0	33
VA2	3	VO1	31
VA1	22	VO2	29
VA0	24	VO3	27
VB3	19	VCOOUT	25
VB2	38	VDD	21
VB1	1	GND	40
VB0	26	VSW	9
VM	7	VDD_EN	13
VS0	15	VSW_EN	11
VS1	30		
VCIN	28		



(a)



(b)



(c)

Figure 2.11 Chip test, (a) transient simulation, N0 and N1 represent the outputs of the core supplied under power pad, SWP0 and SWP1 represent the outputs of the core supplied under the proposed switchable pins, (b) delay comparison of two cores, the scales of logic high and low are 5V and 0V, respectively, (c) current monitor in two work modes.

seen also in Figure 2.11 (b). The average voltage compensation is around 119mV. Monitoring the total current going through the chip, when only one core is supplied under power pad, the average current is around 113.6 μ A, when two cores are supplied under power pad and the proposed switchable pin, respectively, the average current is around 231.2 μ A. Figure 2.11 (c) shows the result of current monitor in two modes. Thus, both voltage compensation and power boosting due to the proposed switchable pins are verified, which are same as the previous post-layout simulations.

2.6 Conclusion

This work presents a novel concept of switchable pin to regulate power distribution in chip multiprocessor at a low cost. We used several sub-nanometer CMOS technology to predict the serious performance degradation caused by voltage loss in complex function chips. With the inspiration of settling more power pads in chip, we proposed the switchable pin and described its fundamental principle. We proved the feasibility of our idea with studying the power delivery network suitable for SoC with switchable pins. Simulation shows that signal attenuation brought by switchable pin is acceptable. Furthermore, based on the problem in our work, I/O pads need to tolerate overlarge current with the purpose of supplying bonus power, we redesigned I/O pads that include output routes using tri-state buffers and power routes using multiple metal layers. Finally, we combined automatic layout flow and manual layout in EDA software to implement our idea at a circuit level. Final test results show that using switchable pins, the power in chip can be doubled without long response time and large voltage loss in pads. Through the simulation for signal integrity, we also found that switchable pins won't seriously degrade the performance of data transmission, especially under medium frequency (2GHz for WRITE behavior, and 1GHz for READ operation). The work is of first kind and so it has been limited to considerations such as the

PCB design from C4 package and use of 180nm CMOS process. Besides, we fabricated a 4-bit ALU based chip to verify the functions of the proposed switchable pin which is same as the simulation results of post-layout design.

CHAPTER 3

LOGARITHMIC CONVERTER WITH A NOVEL CALIBRATION SUPPLIED BY DC/CLOCKED POWER

Specific computation in logarithmic domain shows advantageous performance over in binary domain. In this work, a fixed-binary method to fast calibrate the conversion error based on Mitchell's logarithmic conversion is proposed. Based on this calibration, a generally paralleled circuit structure of the logarithmic converter which can be supplied by both dc power and clocked power to balance both power dissipation and computation latency is also developed. The proposed circuit structure is verified through post-layout simulations and fabricated chip testing.

3.1 Study of Logarithmic Conversion

3.1.1 Mitchell's Algorithm

The logarithmic multiplication and division are described below,

$$\log_2 f_{mul} = \log_2 (a \times b) = \log_2 a + \log_2 b \quad (3.1)$$

$$\log_2 f_{div} = \log_2 (a \div b) = \log_2 a - \log_2 b \quad (3.2)$$

Above two equations show that multiplication and division in binary domain can be converted to addition and subtraction in logarithmic domain, respectively. This logarithmic conversion can reduce the design complexity and the computation latency in the circuit implementation, which is a significant motivation of processing some specific operations in logarithmic domain.

Part of work is reported in following publications:

1. Z. Zhao, A. Srivastava, L. Peng and S. P. Mohanty, "Calibration method to reduce the error in logarithmic conversion with its circuit implementation," *IET Circuits, Devices, and Systems*, vol. 12, no. 4, pp. 301-308, 2018.
2. Z. Zhao, A. Srivastava, L. Peng and S. P. Mohanty, "A low-cost mixed clock generator for high speed adiabatic logic," *Proc. 2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, Pittsburgh, PA, USA, pp. 587-590, July 2016.

For any number, it can be factored as follows,

$$X = 2^{X_I} (1 + X_F) \quad (3.3)$$

Taking logarithmic conversion of Equation (3.3), we obtain:

$$\log_2 X = X_I + \log_2(1 + X_F) \quad (3.4)$$

where X_I is the integer part of the converted number and $\log_2(1+X_F)$ contributes the fraction part of the logarithm code. When X_F is between 0 and 1, $\log_2(1+X_F)$ can be approximated to X_F . Thus, the converted number can be shown as follows,

$$\log_2 X \approx X_I + X_F \quad (3.5)$$

This is the fundamental theory of Mitchell's logarithmic conversion [29]. The conversion process can be summarized as follows:

- 1) Convert the decimal number into a binary number and search for the MSB;
- 2) Use the digit of the MSB as the integer part of logarithmic code;
- 3) Copy all of numbers after MSB as the fraction part of logarithmic code.

As an example, take the decimal number 21. Its binary form is 10101. The MSB is in the fourth digital position. According to the described principle, the integer part is 100, and the bits after MSB, 0101 is the fraction part in logarithmic code. Therefore, the converted number is $(100.0101)_2$. It should be noticed that the converted numbers have an error, which will affect the accuracy of the following multiplication or division as analysed in [94]. The conversion error for a single number can be shown as follows,

$$\varepsilon = \log_2(1 + X_F) - X_F \quad (3.6)$$

If there is no calibration to reduce above conversion error, the average conversion error for a single binary number is 0.0572.

3.1.2 Proposed Calibration

From Equation (3.6), if the base in logarithmic form is 2, the error is always positive and occurs between $2k$ and $2k+1$, where k is any positive integer.

In Equation (3.6), if we add a positive binary code to X_F , thereby shifting the error curve down, the conversion error can be described as follows,

$$\varepsilon_{cal} = \log_2(1 + X_F) - (X_F + C) \quad (3.7)$$

where ε_{cal} is the new error and C is the calibration code. The average conversion can be computed as follows,

$$\varepsilon_{cal_avg} = \frac{1}{(X_{high} - X_{low})} \int_{X_{low}}^{X_{high}} [\log_2(1 + X_F) - (X_F + C)] dX_F \quad (3.8)$$

Equation (3.8) indicates that X_F is located in the interval $[X_{low}, X_{high}]$. Since the integral is applied to a pure fraction number, X_{low} and X_{high} are 0 and 1, respectively. Note that we choose an absolute value for the integral and in that the introduced binary number will make some part of the conversion error to be negative. If only original value is integrated, the calculated average value is not the real error. Absolute deviation only can reflect the real error.

Setting Equation (3.7) to 0, we obtain a transcendental equation. Using Newton-Raphson method to find the root step by step, two roots can be obtained which are:

$$\begin{cases} X_{F1} \approx \frac{-W(-2^{C-1} \ln 2)}{\ln 2} - 1 \\ X_{F2} \approx \frac{W(-2^{C-1} \ln 2)}{\ln 2} + 2 \end{cases} \quad (3.9)$$

where $W(x)$ is the product logarithm function. To make both roots to be the real numbers, the variable in the product logarithm function, $-2^{C-1} \ln 2$, should be larger than $-1/e$ [95]. Under this

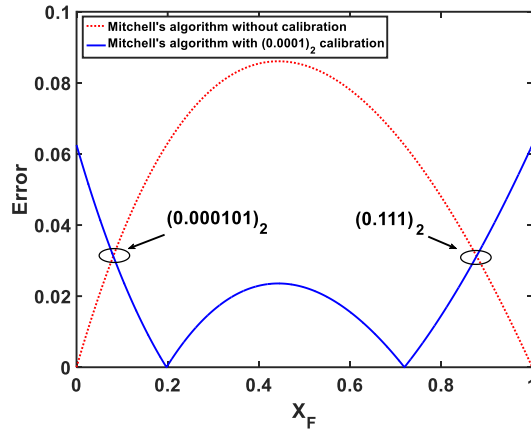
restriction, C should be smaller than 0.0861. Equation (3.9) gives two subsection points of the new error curve crossing the X-axis. Thus, Equation (3.8) can be rewritten as follows,

$$\begin{aligned}
\mathcal{E}_{cal_avg} &= \int_0^{\frac{-W(-2^{C-1}\ln 2)}{\ln 2}-1} [X_F + C - \log_2(1 + X_F)] dX_F \\
&+ \int_{\frac{W(-2^{C-1}\ln 2)}{\ln 2}-1}^{\frac{W(-2^{C-1}\ln 2)}{\ln 2}+2} [\log_2(1 + X_F) - (X_F + C)] dX_F \\
&+ \int_{\frac{W(-2^{C-1}\ln 2)}{\ln 2}+2}^1 [X_F + C - \log_2(1 + X_F)] dX_F \\
&= \frac{W(-2^{C-1}\ln 2) \left(-2 - 4C - \frac{4}{\ln 2} \right) - (5\ln 2)C - 4.5\ln 2 - 5}{\ln 2} \\
&+ \frac{\left[6 + \frac{2}{\ln 2} W(-2^{C-1}\ln 2) \right] \ln \left[3 + \frac{W(-2^{C-1}\ln 2)}{\ln 2} \right]}{\ln 2} \\
&+ \frac{\left[\frac{2}{\ln 2} W(-2^{C-1}\ln 2) \right] \ln \left[-\frac{W(-2^{C-1}\ln 2)}{\ln 2} \right]}{\ln 2}
\end{aligned} \tag{3.10}$$

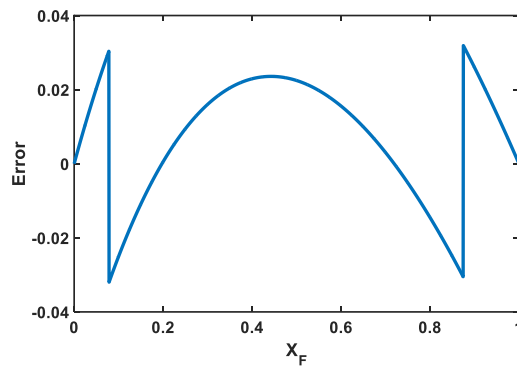
Taking derivative of above equation with respect of C , we obtain,

$$\begin{aligned}
\frac{\partial \mathcal{E}_{cal_avg}}{\partial C} &= \frac{-2^{C-1}(\ln 2)^2}{\ln 2} \left\{ -2 - 4C - \frac{4}{\ln 2} + \frac{2}{\ln 2} \ln \left[3 + \frac{W(-2^{C-1}\ln 2)}{\ln 2} \right] \right. \\
&+ \frac{2}{\ln 2} \ln \left[-\frac{W(-2^{C-1}\ln 2)}{\ln 2} \right] + \left[6 + \frac{2}{\ln 2} W(-2^{C-1}\ln 2) \right] \left[\frac{1}{3\ln 2 + W(-2^{C-1}\ln 2)} \right] + \frac{2}{\ln 2} \left. \right\} \\
&- \frac{4W(-2^{C-1}\ln 2) + 5\ln 2}{\ln 2}
\end{aligned} \tag{3.11}$$

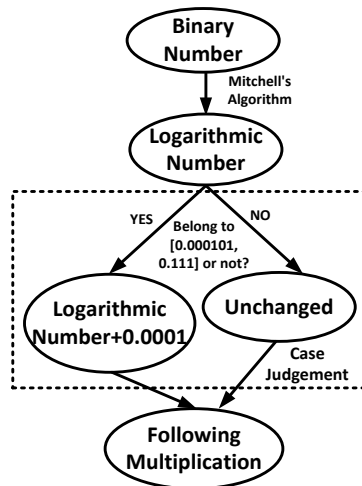
For the variable C , in the interval $[0, 0.063]$, Equation (3.11) is negative, while in the interval $[0.063, 0.0861]$, it is positive, so the minimum average error is around 0.0063. Converting this decimal number to a binary number, $(0.00010000001)_2$, is the highly approximated option. Considering that the longer bits contributing to the calibration lead to a more complex circuit block with higher power dissipation, and the LSB in $(0.00010000001)_2$ is a very small fraction number, we remove the LSB and set C to $(0.0001)_2$ as the fixed binary code to implement the calibration.



(a)



(b)



(c)

Figure 3.1 Logarithmic conversion, (a) error curves of Mitchell's algorithm and the one with the addition of C , (b) error curve using the proposed calibration, (c) calibration flow.

Figure 3.1 (a) shows error curves of Mitchell's algorithm and the one, which is based on Mitchell's algorithm with the addition of C . We notice that when the fraction part, X_F approaches to 0 or 1, the error with $(0.0001)_2$ calibration is larger than that without calibration. Thus, there should be no calibration for X_F in these two areas, which are marked by two subsection points of two curves, $(0.000101)_2$ and $(0.111)_2$, both of which are used to build a piecewise function for the final calibration. Figure 3.1 (b) and (c) show this error curve and its calibration flow, respectively. The proposed calibration only uses a fixed calibration code with a simple case judgement to suppress the conversion error. When X_F is between $(0.000101)_2$ and $(0.111)_2$, the calibration is enabled. Otherwise, the traditional conversion based on Mitchell's algorithm is still applied without the proposed calibration. Under this piecewise function, the average error computed by *Matlab* is 0.0155.

3.2 Circuit Implementation

We introduce a highly paralleled circuit structure for the logarithmic converter with the proposed calibration as shown in Figure 3.2. The entire system mainly consists of two parts: logic part and clock part. Logic part works for the logarithmic converter designed by DCVSL. It uses two bridge signals named as EN and R , both of which will be introduced later, to obtain non-calibration logarithmic code. The calibration uses case judgement and adder to obtain the final logarithmic code. The power part is designed to supply dc power or four-phase clocked power.

3.2.1 Adiabatic Logic

The way of designing digital logic using DCVSL under dc bias has been described in the first chapter. In addition, when supplying clocked power to the whole system, DCVSL is changed to ECRL behaving as an adiabatic logic.

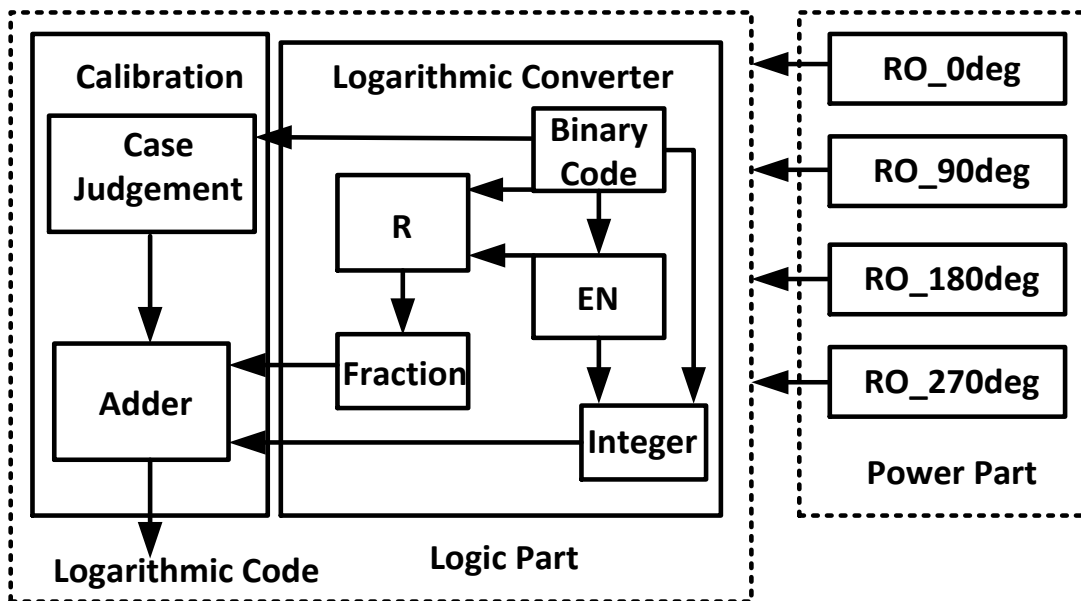


Figure 3.2 Block diagram of the proposed logarithmic converter.

Adiabatic logic has been introduced in VLSI design considering the power budget and the limitation of cooling technique. Under clocked power supply, the single work cycle in ECRL is built by four phases, which are evaluation phase, hold phase, recover phase and wait phase. The evaluation phase and recover phase correspond to slow charge and discharge, respectively. For the two connected logic stages, the second stage must work under the next phase compared to the first stage. Therefore, to ensure that the entire logic chain processes correct data transmission, both phases of slow charge and discharge should work one by one in cascade way. The highly pipelined clocked power must supply to the entire cascade chain with 90° phase difference. It means an adiabatic system needs to add dummy buffers to build a strict pipelined logic chain to confirm each path has the same number of logic stages from the initial node to the terminal node.

Another issue is regarding the highest fan-in for a single gate. We follow a paralleled structure to suppress computation delay. Thus, the drive-in ability should be seriously taken into consideration. Using a high fan-in structure, it can suppress the delay at the cost of signal integrity [81]. Making a trade-off between signal integrity and delay, we restrict the upper bound of fan-in to eight for a single gate.

3.2.2 Logic Part

The logic part is for the logarithmic conversion with a calibration block. The logarithmic converter can be split into two parts: integer conversion and fraction conversion. For the integer conversion, the leading-one detection and overflow detection [41, 42, 43] are widely used to find the MSB of a given binary number. Both of two strategies require a long logic chain with register, which brings long time to get the results and is sensitive to the external clock signal. Thus, it is required to find another way to achieve a fast integer conversion.

For a given N -bit binary number, $S_{N-1}S_{N-2}\dots S_1S_0$, if MSB is located in S_{MSB} bit, higher bits than S_{MSB} are all 0. This means,

$$\overline{S_{N-1}} \bullet \dots \overline{S_{MSB+1}} \bullet S_{MSB} = 1 \quad (3.12)$$

Using Equation (3.12), we create a series of enable signals to help both integer conversion and fraction conversion. The enable signal, EN_i , is expressed as follows,

$$EN_i = \begin{cases} \overline{S_{N-1} + S_{N-2} + \dots + S_1} & MSB = S_0 \\ \overline{S_{N-1} + \dots + S_{MSB+1} + S_{MSB}} & MSB \in [S_1, S_{N-2}] \\ S_{N-1} & MSB = S_{N-1} \end{cases} \quad (3.13)$$

Equation (3.13) indicates that for random numbers with a fixed MSB, the introduced enable signals are always unique that one is high logic and the rest of them are low logic. Thus, for a group of numbers with the same MSB, their integer parts in logarithmic codes are the same. For the circuit implementation, the enable signals can be obtained through NOR array guided by Equation (3.13). We can use enable signals to obtain the corresponding integer number as follows,

$$I_i = \left(\prod_{j=1}^{\frac{N}{2}} \overline{EN_{i0_j}} \right) \bullet \sum_{k=1}^{\frac{N}{8}} \begin{pmatrix} \overline{EN_{i1_a_k}} \bullet \overline{EN_{i1_b_k}} \bullet \overline{EN_{i1_c_k}} \bullet \overline{EN_{i1_d_k}} \\ + \overline{EN_{i1_a_k}} \bullet \overline{EN_{i1_b_k}} \bullet \overline{EN_{i1_c_k}} \bullet \overline{EN_{i1_d_k}} \\ + \overline{EN_{i1_a_k}} \bullet \overline{EN_{i1_b_k}} \bullet \overline{EN_{i1_c_k}} \bullet \overline{EN_{i1_d_k}} \\ + \overline{EN_{i1_a_k}} \bullet \overline{EN_{i1_b_k}} \bullet \overline{EN_{i1_c_k}} \bullet \overline{EN_{i1_d_k}} \end{pmatrix} \quad (3.14)$$

In Equation (3.14), I_i reflects a single bit in the integer part of a logarithmic code. If an N -bit binary number follows $N=2^{n+1}-1$, I_i ranges from I_0 to I_n . EN_{i0_j} is the group of enable signals that map I_i to be low logic in truth table. In this group, there are $N/2$ enable signals for any I_i to build the first term in Equation (3.14). $EN_{i1_a_k}$ is the group of enable signals that map I_i to be high logic in truth table. In this group, there are also $N/2$ enable signals for any I_i . Every 4 $EN_{i1_a_k}$ signals in Sigma function in Equation (3.14) are grouped again, that begins with the

first $EN_{i1_a_k}$ appeared and then sorted one by one in truth table, to form the second term of in Equation (3.14). Using the binary code, 00101101, as the case study of Equation (3.14). Only EN_5 corresponds to high logic and the rest of EN_i to low logic. From the truth table of EN_i and mapping to I_i and using Equation (3.14), we can determine the groups of both EN_{i0} and EN_{i1} as follows,

$$\begin{aligned}
I_2 &= \left(E_7 \cdot \overline{E_6} \cdot \overline{E_5} \cdot \overline{E_4} + \overline{E_7} \cdot E_6 \cdot \overline{E_5} \cdot \overline{E_4} + \overline{E_7} \cdot \overline{E_6} \cdot E_5 \cdot \overline{E_4} + \overline{E_7} \cdot \overline{E_6} \cdot \overline{E_5} \cdot E_4 \right) \\
&\quad \left(\overline{E_3} \cdot \overline{E_2} \cdot \overline{E_1} \cdot \overline{E_0} \right) \\
I_1 &= \left(E_7 \cdot \overline{E_6} \cdot \overline{E_3} \cdot \overline{E_2} + \overline{E_7} \cdot E_6 \cdot \overline{E_3} \cdot \overline{E_2} + \overline{E_7} \cdot \overline{E_6} \cdot E_3 \cdot \overline{E_2} + \overline{E_7} \cdot \overline{E_6} \cdot \overline{E_3} \cdot E_2 \right) \\
&\quad \left(\overline{E_5} \cdot \overline{E_4} \cdot \overline{E_1} \cdot \overline{E_0} \right) \\
I_0 &= \left(E_7 \cdot \overline{E_5} \cdot \overline{E_3} \cdot \overline{E_1} + \overline{E_7} \cdot E_5 \cdot \overline{E_3} \cdot \overline{E_1} + \overline{E_7} \cdot \overline{E_5} \cdot E_3 \cdot \overline{E_1} + \overline{E_7} \cdot \overline{E_5} \cdot \overline{E_3} \cdot E_1 \right) \\
&\quad \left(\overline{E_6} \cdot \overline{E_4} \cdot \overline{E_2} \cdot \overline{E_0} \right)
\end{aligned} \tag{3.15}$$

Using above three specific equations with all EN_i of the given number, the integer part can be obtained, which is 101 as desired. The circuit implementation of Equation (3.14) is shown in Figure 3.3 (a). The logic gates with three dotted symbols indicate that the single gate may extent to the logic chain if fan-in exceeds 8. The buffer shown by the dotted symbol is a dummy buffer or a chain of buffers, which makes the entire block to correctly work under ECRL.

To get the fraction part, we also try to find a paralleled topology. If we define an intermediate term $R_{m,n}$, which can be expressed by,

$$R_{m,n} = EN_m \cdot S_n \tag{3.16}$$

Using this intermediate signal, we can write,

$$F_i = \sum_{j=0}^i R_{N-1-j,i-j} \tag{3.17}$$

In Equation (3.17), F_i refers to any single bit in the fraction part of a logarithmic code, which is implemented by the circuit shown in Figure 3.3 (b). The final form of the logarithm code converted from the N -bit binary code can be expressed as $I_n I_{n-1} \dots I_1 I_0. F_{N-2} F_{N-3} \dots F_1 F_0$.

The calibration block is to reduce the error due to Mitchell's logarithmic conversion. As described in Section 3.1.2, whether the proposed calibration is enabled or not, is strictly based on the original converted code. Thus, we need to compare the converted code with the lower bound and upper bound of calibration region. Digital comparator is widely used for the binary code comparison [96]. Our calibration requires the fraction part to be compared only with the fixed reference numbers $(0.000101)_2$ and $(0.111)_2$. Thus, we give up the standard digital comparator but use a simple logic, which can compare with our fixed bounds at low circuit cost.

For a single converted number, first a signal, C_{EN} is defined, to judge if the number locates between upper bound and lower bound in which the number needs to be calibrated. For a given N -bit input, C_{EN} can be expressed as follows:

$$C_{EN} = \frac{\overline{F_{N-2} \cdot F_{N-3} \cdot F_{N-4} \cdot F_{N-5}} + \overline{F_{N-2} \cdot F_{N-3} \cdot F_{N-4} \cdot F_{N-6}}}{\overline{F_{N-2} \cdot F_{N-3} \cdot F_{N-4} \cdot F_{N-7}} + \overline{F_{N-2} \cdot F_{N-3} \cdot F_{N-4} \cdot F_{N-5}}} + \frac{\overline{F_{N-2} \cdot F_{N-3} \cdot F_{N-4} \cdot F_{N-5}} \cdot \overline{F_{N-6} \cdot F_{N-7}}}{\overline{F_{N-2} \cdot F_{N-3} \cdot F_{N-4} \cdot F_{N-5}} \cdot \overline{F_{N-6} \cdot F_{N-7}}} \quad (3.18)$$

Above equation shows high logic when the fraction part is located in the calibration region that enables the original logarithmic code to be added to $(0.0001)_2$. Otherwise, there is no calibration enabled. The entire calibration block is shown in Figure 3.3 (c) with the circuit implementation of Equation (3.18) and a simplified carry look ahead adder (CLA) to achieve the fixed binary calibration that C_{EN} is connected to the F_{N-5} bit. Under this connection, if there is no calibration enabled, the original converted number keeps the same value. Once the calibration is enabled, the original converted number is added with $(0.0001)_2$. Note that the circuit of the case judgement only uses two logic stages no matter how many bits will be converted. For the design

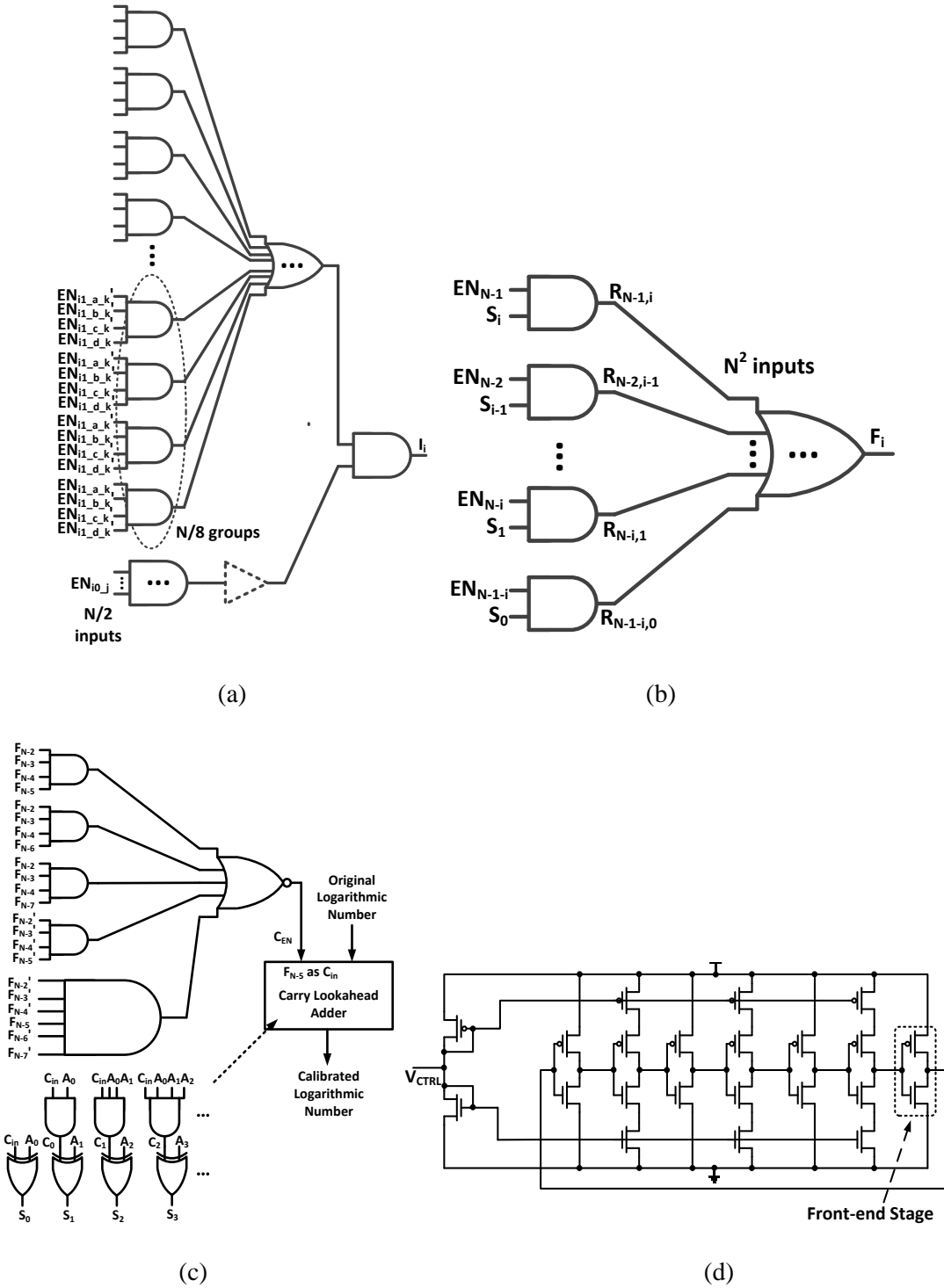


Figure 3.3 Circuit diagram of the proposed N -bit logarithmic converter, (a) integer part, (b) fraction part, (c) calibration block, (d) clock generator.

of simplified CLA, since F_{N-5} is the last bit in the adder, we can set F_{N-5} as carry-in bit so that the original logarithmic code adds a full zero sequence. Thus, the generate terms in a CLA are all low logic and the propagate terms are equal to the original logarithmic code. We cancel the AND array outputting the generate terms and XOR array outputting propagate terms. We leave only AND array to obtain the carry terms, and XOR array outputs the final result, which are also shown in Figure 3.3 (c). This simplified CLA can reduce two logic stages and the circuit cost due to the proposed calibration.

3.2.3 Clock Part

Another important issue is how to drive adiabatic logic using the clocked power. The types of clocked power include sine, trapezoid, and step by step waveforms. Sine clocked power can be obtained by oscillators with passive devices to boost oscillation frequency [97]. Trapezoid power can be obtained by the RL circuitry, which brings analog block to the entire system [98]. Step-by-step power can be obtained by tank capacitors [99]. To save the chip area by removing passive devices, the performance of clocked power built by purely logic gates has been studied [100]. It does reduce the occupied area with low power dissipation but signal integrity degrades when frequency boosts.

In this work, we have used a voltage controlled ring oscillator (RO) [101] to build our clock tree. To generate four-phase signals with 90° phase difference, we designed four ROs with modified dimensions of transistors since the phase difference is highly related to additive RC delay existed in transistors. We choose RO with seven stages to balance the oscillation frequency and signal integrity. The entire design of clock part is shown in Figure 3.3 (d). The first stage of this circuit is the control block with diode-connected transistors. The large current going through this control block can boost the oscillation frequency. With the decrease of V_{CTRL} , the oscillation

frequency will gradually reduce and the clocked power will finally change to the dc power. The W/L of both pMOS and nMOS transistors with diode connection are set to 800nm/200nm. The W/L of two transistors in the front-end stage of RO is set to 1000nm/200nm, 1600nm/200nm, 2000nm/200nm, and 2400nm/200nm, to build four sine waves with 90° phase difference. For the rest of transistors in this circuit, these are uniformly set to 200nm/200nm.

3.3 Error Analysis

The proposed calibration method is developed based on the Mitchell's logarithmic conversion. Figure 3.4 shows the error sweep of these two conversions with random fraction. It can be seen that the proposed calibration can obviously suppress the conversion error. To evaluate the performance of the proposed calibration in depth, we list a comparison of the proposed one and other reported conversion methods [29, 31, 32, 33, 35, 102, 103] in Table 3.1. The percentage error range can reflect the average error for a given method. The computed percentage error range of our work is 1.55% using *Matlab* simulation. Thus, using Mitchell's algorithm as the reference, the reduced rate of conversion error in our work is 71.1%, which is larger than reported in [31, 32, 33, 35]. On comparison with [103], the reduced rate of conversion error in the proposed work is larger than the case of four fraction regions. To obtain nearly the same error as in [103] for the proposed work, it requires eight fraction regions in [103]. The number of fraction regions is proportional to the circuit complexity and the computation latency. Our calibration method uses three fraction regions to be processed, which is larger than reported in [32, 33, 35] and one case in [102]. However, these three fraction regions are only generated by one case judgement. The calibration block only uses two logic stages for the case judgement as described in Section 3.2.2. Thus, three fraction regions in this work cannot lower the work speed significantly. Methods in [31, 35, 102] are very time consuming since counter, shifter, or leading-one detector are used,

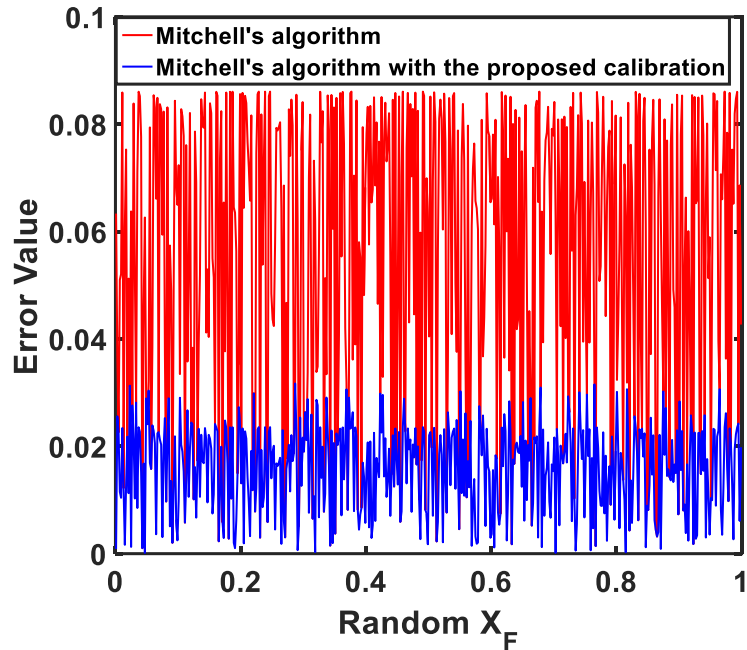


Figure 3.4 Error sweep of the proposed calibration and Mitchell's algorithm.

Table 3.1 Comparison with the reported conversion methods and the proposed work

	[29]	[31]	[35]	[102]		[32]	[33]	[103]		Proposed calibration
Number of fraction regions	1	4	2	2	3	2	2	4	11	3
Maximum positive error	0.086	0.0253	0.0292	0.0449	0.0293	0.036	0.0319	0.004	0.004	0.03189
Maximum negative error	0	-0.0062	-0.028	-	-0.021	-0.009	0	-	-	-0.03197
Error range	0.086	0.0315	0.0572	0.0632	0.0503	0.045	0.0319	0.008	0.008	0.06386
Maximum positive percentage error (%)	5.36	2.293	0.431	0.93	0.431	2.889	2.337	2.93	0.225	1.15
Maximum negative percentage error (%)	0	-0.468	-1.54	-0.554	-0.268	-0.45	0	-1.67	-	-0.4
Percentage error range (%)	5.36	2.761	1.971	1.484	0.699	3.339	2.337	4.6	0.406	1.55
Reduced rate compared to Mitchell's algorithm (%)	0	48.5	63.2	72.3	87	37.7	56.4	14.2	92.4	71.1
Circuit strategy	N/A	Counter; Register.	Shifter array; Adder.	ROM; LOD; Shifter.		Double-adder.	Logic array; Adder.	Segment encoder array.		Logic array; Adder.

which are not suitable to the fast computation. When the number of fraction regions increases, the computation in [102, 103] can reduce further conversion error than in our work. However, both of two work introduce a complex multi-step linear function, which is a penalty on computation speed since each linear function requires multi-bit multiplication and addition to be processed. In [103], it uses LUT to reduce the computation latency of multi-step function at the cost of chip area. The work in [32, 33] simply uses two fraction regions to speed up the conversion at the cost of conversion accuracy.

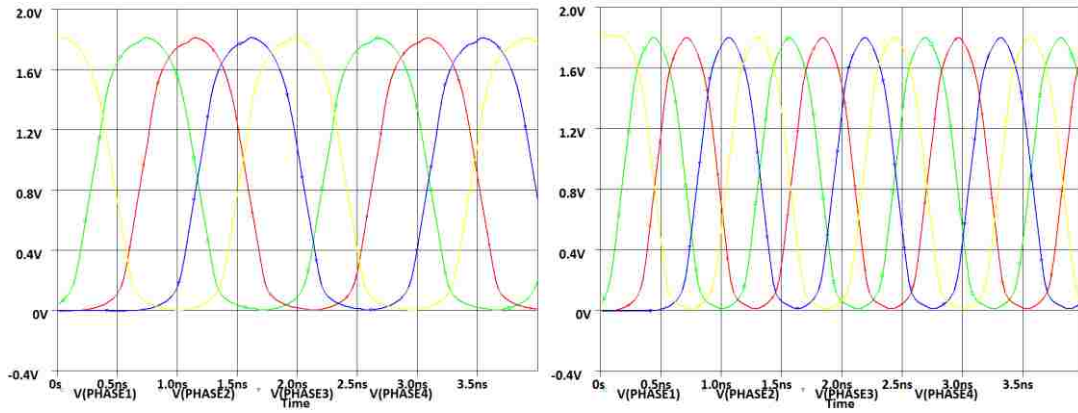
3.4 Circuit Verification

3.4.1 Post-Layout Simulations

We first test the performance of the clock block. Figure 3.5 (a) and (b) show the transient simulation results of 498MHz and 0.981GHz four-phase sine waves under two control voltages, 1.5V and 1.8V, respectively. We can see that the clocked power generated by the proposed circuit is roughly the same as the standard sine wave, which can be used for ECRL work. Figure 3.5 (c) shows the oscillation frequency with the variation of V_{CTRL} . We can see the oscillation frequency ranges from 79MHz to 0.981GHz when V_{CTRL} boosts from 1.3V to 1.8V. When the control voltage is lower than 1.2V, the clock block supplies dc power making the following logic block work in high speed DCVSL. Another important issue is the frequency deviation since each two adjacent logic stages must follow 90° phase difference to confirm the correct data transmission in ECRL. Comparing each two ROs outputting two adjacent clocked power, we use 90° phase difference as the standard value. The phase deviation is calculated from the difference between actual phase and 90° , and the difference divided by 90° . The phase deviation with the variation of oscillation frequency is shown in Figure 3.5 (d). It can be seen that the phase deviation increases with the

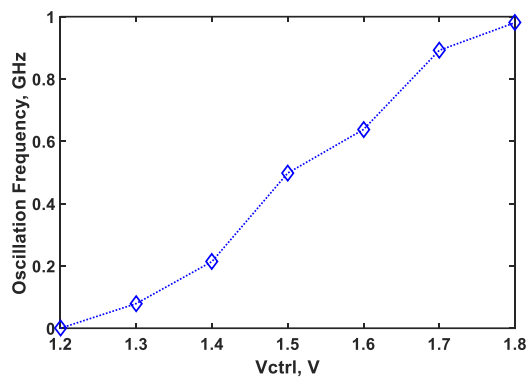
boost of the oscillation frequency and does not exceed 5% so that the clock part can reliably support the logic part working in ECRL.

Figure 3.6 (a) shows the layout of our design using TSMC 180nm CMOS. It includes 1162 transistors with the dimension of 0.063mm^2 . Besides some transistors in clock part, the dimensions of the rest of transistors are $200\text{nm}/200\text{nm}$. From the layout, we can see that both calibration block and clock generator do not occupy large chip area. The logarithmic converter is the largest block since we use highly paralleled structure to speed up the conversion instead of counter, LOD and any other blocks, which can process multiple-bit repeatedly using the single circuitry. Figure 3.6 (b) and (c) show the simulation results of the logarithmic codes under DCVSL and ECRL, respectively. The frequency of the clocked power in ECRL transient simulations is 498MHz . The input signal sequence in both DCVSL and ECRL transient simulations are $(11111111)_2$, $(11101000)_2$, $(00001111)_2$, and $(00000000)_2$ at 100MHz . Under these inputs, the outputs are $(111.1111111)_2$, $(111.1110000)_2$, $(011.1111000)_2$, and $(000.0000000)_2$, which are matched to the Mitchell's conversion with the proposed calibration method. The delay in DCVSL is 1.82ns with good signal integrity. In ECRL, the output in the first 5ns is unstable and incorrect since the adiabatic chain supplied by four-phase clocked power outputs the results stage by stage. After 5ns , the output port can correctly transmit the results. Through the sine waves, we can see the distinction of high logic and low logic in ECRL simulations. However, the signal integrity in ECRL is not as good as in DCVSL. This is due to the supplied clocked power, which is not a perfect sine signal as previously shown in Figure 3.5 (a) and (b). Incomplete charging and discharging are implemented during fast operation of ECRL. In addition, the limited driven-ability of the clock generator in this work degrades the clocked power supply. Figure 3.6 (d) shows the simulation results of the power dissipation under both ECRL and DCVSL. We set the frequency of clocked

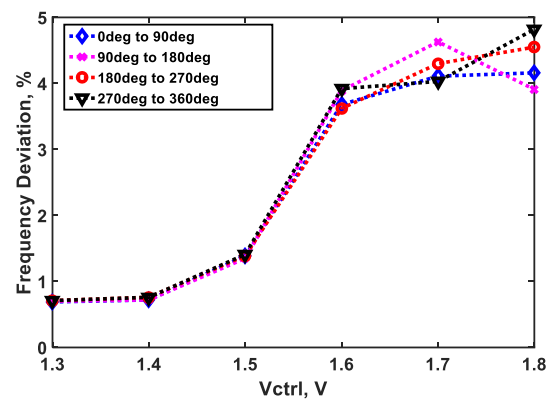


(a)

(b)



(c)

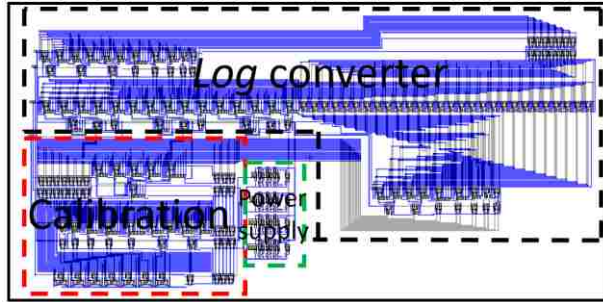


(d)

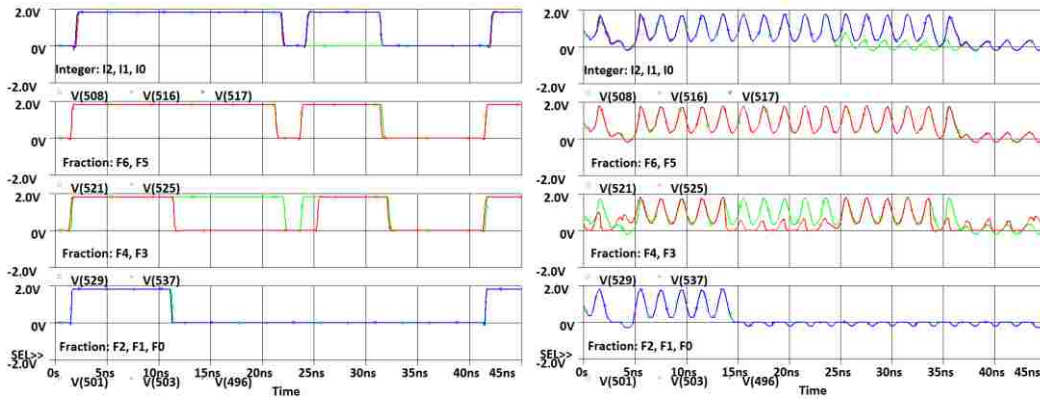
Figure 3.5 Simulations of clock generator, (a) waveform at 498MHz, (b) waveform at 0.981GHz, (c) oscillation frequency versus control voltage, (d) frequency deviation versus control voltage.

power at 498MHz and 0.981GHz used for our ECRL simulation. The power dissipation of the converter under dc power ranges from 2.4mW to 3.709mW with boost in input frequency. Under clocked power supply, when the frequency of the clocked power is 498MHz, the power dissipation varies from 1.12mW to 1.9352mW. The power dissipation under 0.981GHz clocked power varies from 1.6656mW to 2.1734mW. It can be concluded that ECRL does reduce power dissipation compared to DCVSL. The converter driven by the low frequency clocked power can save more energy than that driven by the high frequency clocked power.

To evaluate the performance of the proposed work depending on the increased bit length, a comparison is presented in Table 3.2. It compares our work with the prior work presented in [31, 32, 33, 35, 38, 102, 103] from the viewpoint of the number of all bottom cells. Three numbers identified by two slashes, from left to right, represent the number of basic logic cells (all Boolean logic, MUX and transmission gate), unit adders and unit memory cells, respectively. The method to estimate the number of all bottom cells is based on the diagrams of both circuit and system shown in the reported papers. For [102], we used the case of 3-regions to estimate the number of all bottom cells. While in [103], the case of 8-regions is used for the estimation. We observe that when the bit length is 8-bit or 16-bit, the number of all bottom cells in this work is smaller than that in [31, 38, 102, 103]. Over 16-bits, the proposed work and [103] consume much more bottom cells than the other work except [38], in which the conversion is fully implemented by memory array and the usage of memory cells is the largest since all converted results corresponding to the specific binary codes need to be stored. Hardware implementations in [32, 33, 35] is not as sensitive to bit width increasing as in other work. The reason that both [103] and the proposed one consume large number of bottom cells is that the two work use highly paralleled topology instead of the pipelined structure achieved by leading-one detectors or shifters. Thus, to process the input

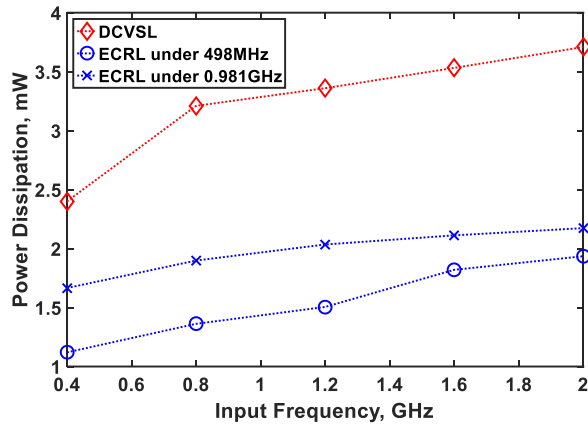


(a)



(b)

(c)



(d)

Figure 3.6 Simulations of the entire logarithmic converter, (a) layout view, (b) DCVSL transient simulation, (c) ECRL transient simulation, (d) comparison of power dissipation under different logic.

Table 3.2 Comparison of hardware cost with the prior work and the proposed work

	[31]	[35]	[102]	[32]	[33]	[103]	[38]	This work
8-Bit	200/0/0	72/8/0	202/16/40	8/18/0	10/8/0	64/0/64	0/0/192	120/0/0
16-Bit	400/0/0	144/16/0	404/32/80	16/34/0	20/16/0	128/0/256	0/0/640	343/0/0
32-Bit	800/0/0	288/32/0	808/64/160	32/66/0	40/32/0	256/0/1028	0/0/40k	1198/0/0
64-Bit	1600/0/0	576/64/0	1616/128/320	64/130/0	80/64/0	512/0/4112	N/A	4613/0/0

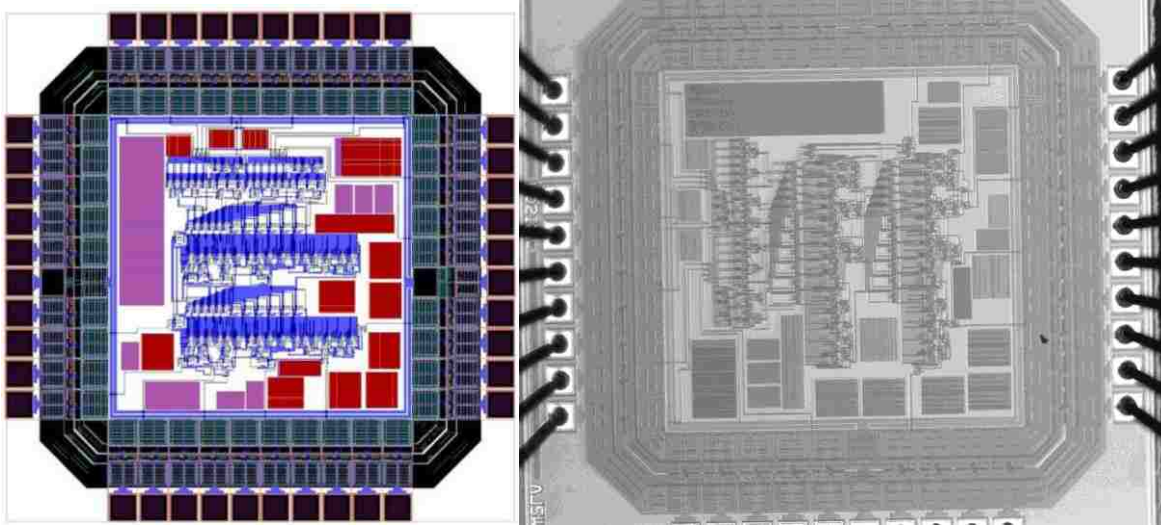
Table 3.3 Comparison of maximum length with the prior work and the proposed work

	[31]	[35]	[102]	[32]	[33]	[103]	[39]	This work
8-Bit	30	11	24	16	13	13	N/A	8
16-Bit	54	20	37	24	21	13	N/A	11
32-Bit	102	36	69	40	37	13	640	12
64-Bit	198	68	133	72	69	13	N/A	12

with large bit width with the consideration of the allowable fan-in, in a single stage, the requirement of bottom cells is larger than in the other work using step-by-step computation. Another issue is the conversion latency. A very significant metrics to evaluate it is the critical path, which determines the number of longest logic stages in entire conversion from the binary code to logarithmic code. Table 3.3 presents the comparison of maximum length between the proposed work and other prior work [31, 32, 33, 35, 39, 102, 103]. It is observed that the proposed one consumes the shortest path to obtain the converted result compared to other designs. The implementations with the computations of pipelined shifting [31, 35, 102] are sensitive to the increase of bit length. The work using PLA [39] is not suitable for the fast conversion. For work in [32, 33], the front-end circuit to achieve multi-step function without LUT contributes the most to the required path. The runner-up in conversion speed is the design in [103] since it also follows the path of highly paralleled topology. The reason that it is slightly slower than the proposed one is that its multi-step function is complex and requires more stages than in the proposed one.

In our design, the latency through the calibration block is not sensitive to the bit width. As mentioned in Section 3.2.2, the block to implement the case judgement of the calibration block requires only two stages without stage variation as bit width is increasing. Our simplified traditional CLA reduces the number of logic stages required by the calibration. The block outputting fraction number contributes mainly to both required stages and the number of bottom cells with the increase of bit width since one stage has N^2 inputs as shown in Figure 3.3 (b). Other blocks for the enable signals and the integer number do not introduce more stages and hardware cost when bit width increases.

3.4.2 Chip Testing



(a)

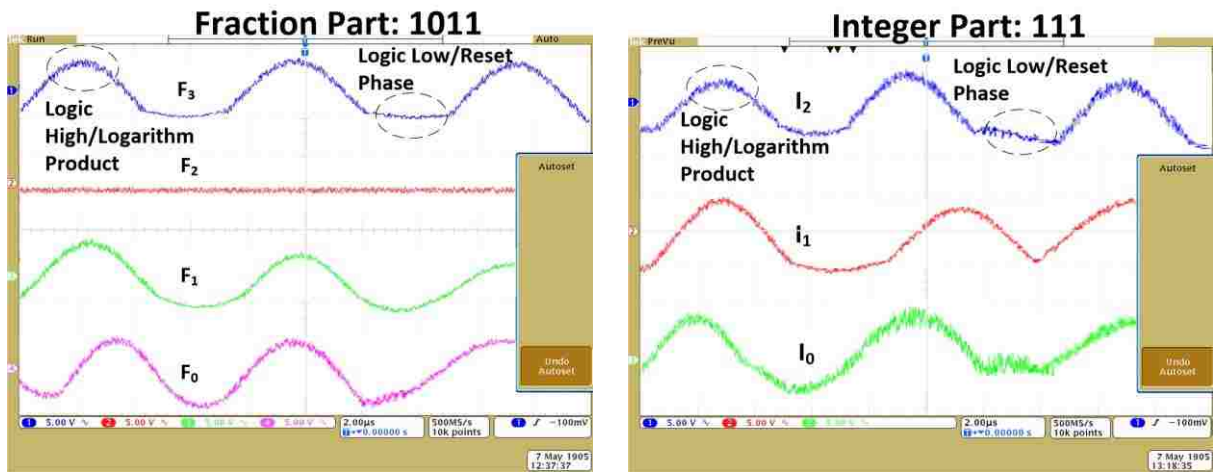
(b)

Figure 3.7 Fabricated chip of the logarithmic multiplier, (a) layout view, (b) die photo.

Table 3.4 Pin allocation of the chip in Figure 3.7

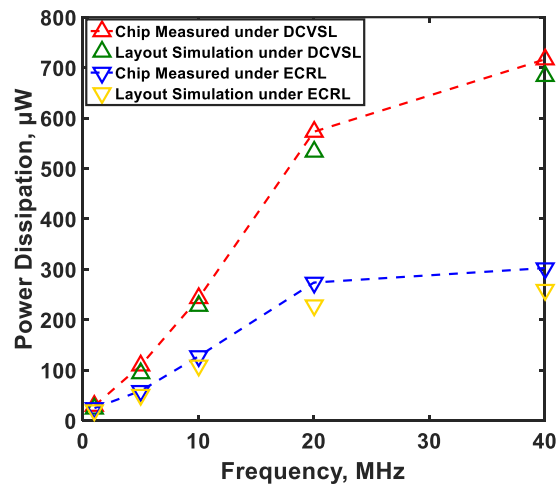
VA3	15	VO6	33
VA2	13	VO5	31
VA1	11	VO4	29
VA0	9	VO3	27
VB3	7	VO2	25
VB2	5	VO1	23
VB1	3	VO0	39
VB0	1	VDD	21
VCLOCK_0	17	GND	40
VCLOCK_90	19		
VCLOCK_180	22		
VCLOCK_270	24		

We have fabricated a 4-bit logarithmic multiplier using 500nm CMOS to verify the proposed work. Figure 3.7 (a) is the layout view of the fabricated chip, which includes logarithmic converter, calibration block using the proposed algorithm and carry ripple adder (CRA) used for multiplication in logarithmic domain. The die photo of the fabricated chip is shown in Figure 3.7 (b). The clocked power is externally supplied by signal generator. For output, the normal conversion of 4-bit input has 3-bit fraction number. However, with the proposed calibration, the fraction number can be extent to 4-bit to reduce conversion error. Further implemented by addition, the final output is a 7-bit signal. Table 3.4 lists the pin location of the fabricated chip. Figures 3.8 (a) and (b) show the computing results under clocked power, in which two inputs are 1111 and 1110. 1111 won't be calibrated so that the converted number is 11.111. While for 1110, calibration is enabled and the converted number is 11.1101. Final product in logarithmic domain is 111.1011 which matches the testing results. We also conclude the power dissipations obtained by post-layout simulation and chip testing as shown in Figure 3.8 (c). We observe that the system under ECRL operation consumes less power than that under DCVSL operation which is same as the previous simulation results in TSMC 180nm CMOS. Comparing results of simulations and chip testing, with the increase of frequency, the power difference between theoretical and experimental results under ECRL operation is larger than that under DCVSL operation. This can be explained by the noise from off-chip clocked power can shift the timing of the correct phase in adiabatic logic and so generates large short-circuit current. Also sampling the current spikes during two-phase switching in adiabatic operation, then averaging spikes from all output nodes, Figure 3.8 (d) shows this result, in which the only from logic low to high, DCVSL operation generates larger current than ECRL operation. For other switch activities, adiabatic logic always generates larger spikes than DCVSL operation due to phase mismatch in pipelined logic chains. The imperfect clocked

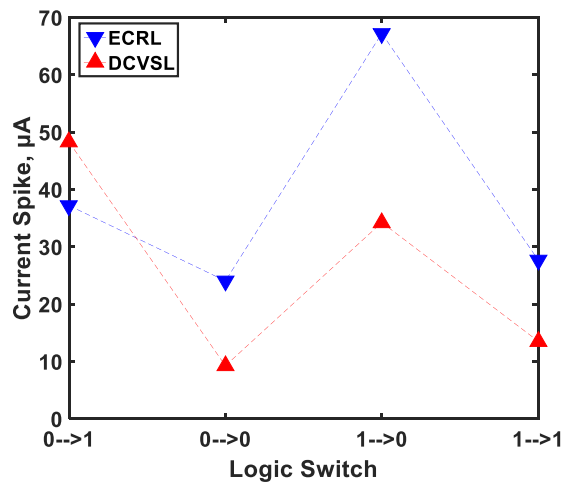


(a)

(b)



(c)



(d)

Figure 3.8 Chip testing, (a) fraction results under clocked power, (b) integer results under clocked power, the scales of logic high and low are 5V and 0V, respectively, (c) power dissipation results including chip measurement and post-layout simulation, (d) current spikes depending on logic switch.

signals shown in Figures 3.8 (a) and (b) also reflect this unwanted phase mismatch of adiabatic operation. However, only entirely considering power issue, adiabatic logic supplied by clocked power is more advantageous than normal logic styles supplied by dc power.

3.5 Conclusion

In this work, based on Mitchell's algorithm, we propose a novel calibration method to suppress the error during logarithmic conversion. The essence of our calibration is to use a fixed code to calibrate a single converted number. At the circuit level, we propose a highly paralleled structure to speed up the logarithmic conversion. The bottom gate is designed in DCVSL to let the system work under both adiabatic logic and fast logic. For the clocked power design, we used four voltage controlled ring oscillators without passive devices to generate four-phase sine signals to drive the converter. This clock block also can be controlled to supply dc power to support traditional DCVSL working. We designed an 8-bit logarithmic converter in layout level and fabricated a 4-bit logarithmic multiplier to successfully verify the proposed calibration method. The results prove that when the system work in ECRL, the power dissipation is smaller than that in DCVSL. The performance comparison proves that the conversion latency of the proposed design is not sensitive to the increase in bit width, since computations of shifting and iteration are not introduced into our work.

CHAPTER 4

ALU DESIGN BASED ON NEURON-LIKE CELLS USING MULTIPLE INPUT FLOATING GATE MOSFETS

Recent work have proposed several techniques to optimize ALU with the perspective of reducing leakage power, bit-partition and data-calibration. This chapter will introduce a new ALU design using MIFG MOSFETs. The most attractive feature of MIFG MOSFET based logic gates is that they can perform multiple operations using the same circuit with the control by multiple coupling capacitors [104, 105]. The cell-sharing is introduced to perform several operations, and a novel digital calibration method is proposed in this work. The verification is based on post-layout simulations and mismatch study.

4.1 Design of Neuron-Like Cells

For each operation, normal ALUs have independent blocks. In this work, we share the same neuron-like cell using MIFG logic to group some specific operations for saving the area.

For both inverter and buffer, we use the same cell as shown in Figure 4.1 (a). The first stage is an inverter realized from two floating gates and the second one is the normal static inverter. If we ignore both C_{ox} and C_p , the surface potential in the first stage can be expressed as follows:

$$\phi_f = \frac{C_{in}V_{in} + C_{via_th}V_{via_th}}{C_{in} + C_{via_th}} = \frac{C_{in}}{C_{total}}V_{in} + \frac{C_{via_th}}{C_{total}}V_{via_th} \quad (4.1)$$

where V_{via_th} is the variable and controlled by the instruction code of the proposed ALU, and C_{total} is the sum of C_{in} and C_{via_th} . If C_{in} is equal to C_{via_th} , the output after the first stage can be described as follows:

$$V_{out} = f_{INV} \left(\frac{1}{2}V_{in} + \frac{1}{2}V_{via_th} - V_{th} \right) \quad (4.2)$$

where $f_{INV}(x)$ is the inverter transfer function and V_{th} is the threshold voltage of the inverter. Through the adjustment of V_{via_th} , we can vary the threshold voltage of the inverter if V_{in} is seen as

the only input. Since discharging is always faster than the charging, to ensure signal integrity, V_{via_th} is set to 0V thus connecting to the ground. Compared to the MIFG inverter with the adjustable threshold voltage reported in [106], the advantage of our design is that it has less capacitors involved to save area and the variable to adjust the threshold voltage is another input signal but not the ratio of two capacitors. Then after the second stage, the buffer's output can be obtained. Thus, the proposed two-stage cell performs the functions of both inverter and buffer.

We map the design principle of both inverter and buffer for the design of a two-input gate performing NAND and NOR using MIFG logic where two data inputs plus a variable input are applied in the first stage. If three coupling capacitors are the same, the output after the first stage can be expressed as:

$$V_{out} = f_{INV} \left(\frac{1}{3}V_a + \frac{1}{3}V_b + \frac{1}{3}V_{via_th} - V_{th} \right) \quad (4.3)$$

where V_a and V_b are two inputs. If V_{via_th} is connected to the ground, Equation (4.3) can be expressed as follows:

$$V_{out} = f_{INV} \left(\frac{1}{3}V_a + \frac{1}{3}V_b - V_{th} \right) \quad (4.4)$$

According to above equation, when both two inputs are high, the output will be low. Otherwise, the surface potential does not exceed the threshold voltage so that the output stays in high, which matches the NAND operation.

If V_{via_th} is connected to supply voltage, Equation (4.3) can be expressed as follows:

$$V_{out} = f_{INV} \left(\frac{1}{3}V_a + \frac{1}{3}V_b + \frac{1}{3}V_{dd} - V_{th} \right) \quad (4.5)$$

In this case, only when both two inputs are low, the output will be high. Otherwise, the output is always low, which is NOR operation. Thus, as shown in Figure 4.1 (b), we propose a

two-stage circuit using MIFG logic to implement following four Boolean logic functions: NAND, AND, NOR, and OR.

For XNOR and XOR, it is obvious that the floating gate potential diagrams (FPDs) of both XNOR and XOR are not monotonically increasing functions [104]. Therefore, we cannot use an additional input to adjust the surface potential to program the desired function. One solution is to use a cascade structure with NAND, NOR and inverter gates to obtain both XNOR and XOR at the cost of delay and area. We propose a pass transistor based multiple input floating gate (PT-MIFG) logic to achieve XOR and XNOR as shown in Figure 4.1 (c). The first stage is in complementary topology, which uses four transistors with eight same coupling capacitors to execute XOR. The main principle of this circuit is that four pass transistors with MIFGs reflect four input cases to obtain the desired output of an XOR gate. For an n-type pass transistor with MIFGs, only if two inputs are high, the transistor will be switched-on. Likewise, only if two inputs of a p-type pass transistor with MIFGs are low, the transistor will be switched-on. XNOR is also obtained by connecting to a static inverter.

For addition and subtraction, the full adder using MIFG MOSFETs includes two circuit blocks to output sum bit and carry-out bit as shown in Figure 4.1 (d). The full adder has three inputs, which are two adding bits and a carry-in bit. For the block outputting the carry-out bit, it can be concluded that when more than one inputs are high, the output will be high. Thus, we can use three-input MIFG inverter to implement this function. For the design of sum block, it is the XOR function with all three inputs. We combine PT-MIFG logic and traditional pass transistor logic together to achieve the function. The circuit has four branches representing eight input cases. For a single branch, it includes a pair of paralleled MIFG MOSFETs and a CMOS transistor

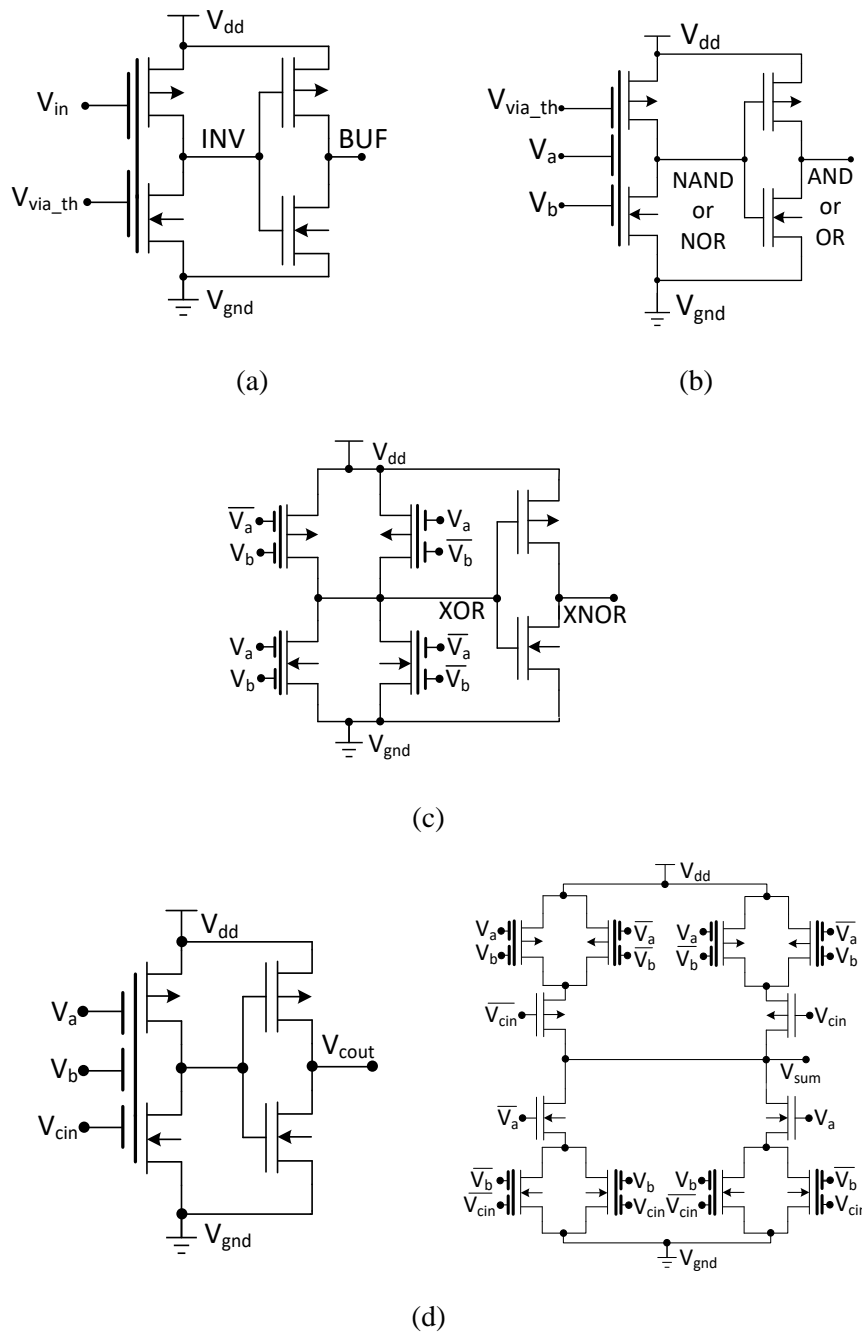


Figure 4.1 Circuit diagram of the proposed MIFG logic gates, (a) INV and BUF, (b) NAND, AND, NOR and OR, (c) XOR and XNOR, (d) full adder.

corresponding to two input cases, each input case can be enabled to output the sum bit corresponding to the unique input case.

Above designs are regarding eight standard Boolean operations and two arithmetic functions. We will introduce 32-bit sequence comparison, single-stage MUX and digital calibration to the following ALU design. For digital comparison, its function is to judge if two 32-bit sequence is the same. We use sequence partitioning to firstly compare each 4-bit sequence. The 4-bit digital comparator is shown in Figure 4.2. An XNOR gate compares two 4-bit signals from two vectors. The outputs of XNOR array are connected to a 4-input AND gate using MIFG logic. To obtain a correct output, the ratio of five coupling capacitance in the 4-input AND gate of Figure 4.2 is set to 1:1:1:1:3. After comparing each 4-bit sequence, the result of 32-bit sequence is determined by MIFG AND gate chain including 2-input MIFG AND gate and 4-input MIFG AND gate as also shown in Figure 4.2. The final output is decided by both the non-calibrated data and the correct data. Figure 4.3 shows the implementation of a single calibration block. It is similar to the circuit shown in Figure 4.1 (b). Data without calibration connects a single MIFG transistor. Data with calibration connects the rest of two MIFG transistors. This stage not only can output correct data but can improve also the signal integrity of the proposed ALU's output.

MUX is necessary in ALU and can be designed by the TG array or logic gates. In this work, we use PT-MIFG logic to design TG array. The proposed TG is shown in Figure 4.4. Each node on floating gates can be connected with an inverter to change the enable logic for ALU decoding. Since ALU design has 16 operations so that normally there are four CMOS TGs required for decoding each path. Using the proposed design, the only one TG implements decoding in each path to reduce process latency.

4.2 ALU Design

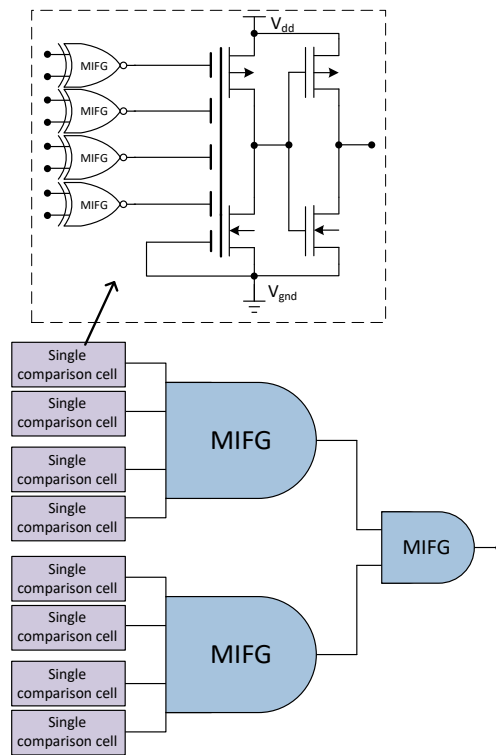


Figure 4.2 Proposed circuit block used for 32-bit sequence comparison.

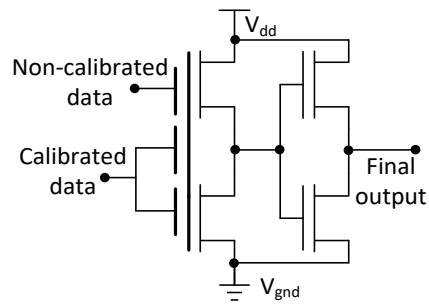


Figure 4.3 Proposed circuit block of digital calibration.

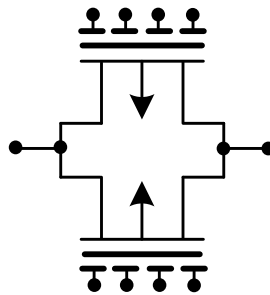


Figure 4.4 PT-MIFG logic based TG with four floating gates.

4.2.1 System Overview

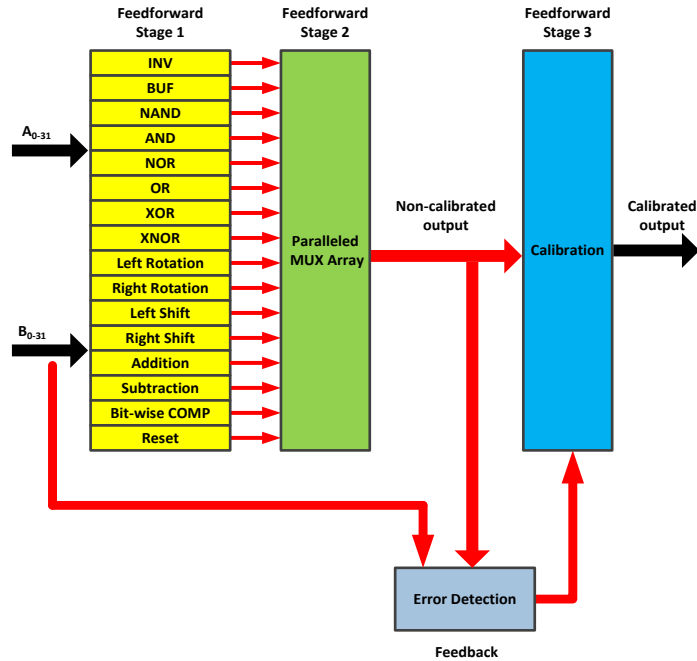
Figure 4.5 (a) shows the block diagram of the proposed ALU. In feedforward path, 32-bit input is processed in parallel operation blocks in feedforward stage 1. After this stage, the MUX array selects the operation in feedforward stage 2. The output from the feedforward stage 2 is then calibrated by the feedforward stage 3 to obtain final result. The unwanted error is detected through the feedback loop with the aid of both input data and instruction code. Table 4.1 summarizes operations with the corresponding instruction codes.

In feedforward stage 1, neuron-like cells shown in Figures 4.1 and 4.2 are used. Note that each cell for operating Boolean logic in this work performs at least one logic. MUX is implemented by PT-MIFG logic based TG as shown in Figure 4.4. As stated earlier, the proposed TG can decode faster than the conventional one with four cascading TGs. The calibration cell shown in Figure 4.3 builds the feedforward stage 3 to achieve error recovery with the support of the feedback loop.

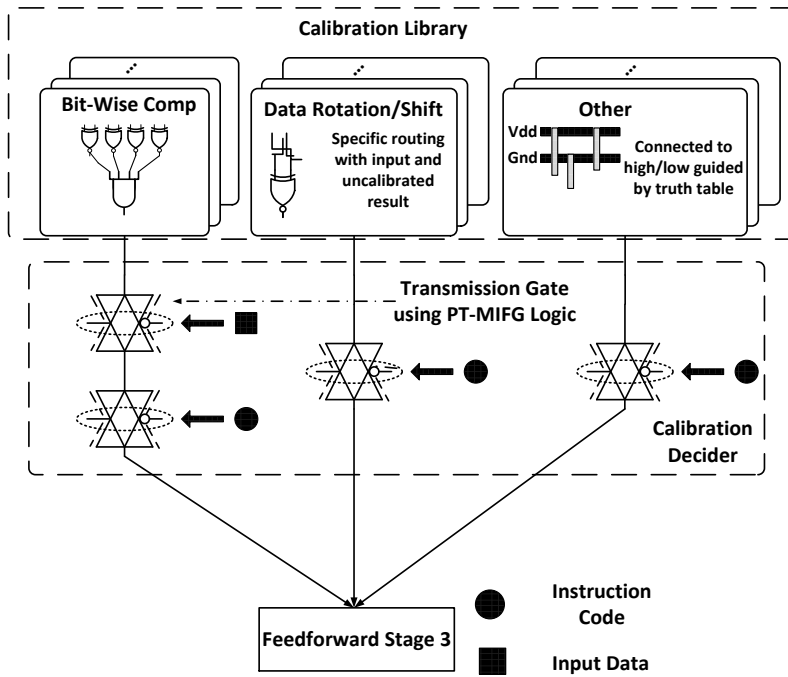
4.2.2 Feedback Loop

The feedback loop in the proposed design works for error detection. The current error detection mainly uses LUT, delay-aware counter and digital comparison in checkpoint [107, 108, 109]. The described methods can reliably calibrate error at the overhead of the chip area, computation latency and power dissipation.

Figure 4.5 (b) shows the block diagram of the feedback loop, which implements the error detection. This stage has two parts including calibration library and decider. We have classified the error detection as three methods explained as follows. For eight Boolean logic operations, two arithmetic operations and reset operation, we use truth table to guide the calibration. The correct data is obtained through the connection to either the power supply or ground. The calibration decider used for these operations is built by two TGs using PT-MIFG logic. The first TG is



(a)



(b)

Figure 4.5 Proposed ALU, (a) block diagram, (b) feedback loop for error detection.

Table 4.1 Operation summary

Operation	Instruction Code
INV	0000
BUF	0001
NAND	0010
AND	0011
NOR	0100
OR	0101
XOR	0110
XNOR	0111
Left Rotation	1000
Right Rotation	1001
Left Shift	1010
Right Shift	1011
Addition	1100
Subtraction	1101
Sequence Comparison	1110
Reset	1111

controlled by the input data to enable desired result going to the next TG. The second TG is controlled by the instruction code directly to judge which operation should be activated. Taking AND operation with the input of 01 as an example, the input of the first PT-MIFG logic based TG in the calibration decider connects to ground, and its controlled nodes connect to 01. The controlled nodes in the second TG in the calibration decider are connected to the instruction code, 0011. Thus, under this principle, the correct output, low logic, can be uniquely transferred to the feedforward stage 3 for calibration. For the operations of shift and rotation, the calibration library can be implemented by the specific routing between the original input data and input of a PT-MIFG logic based TG controlled by the instruction code, which is the calibration decider. As an example, considering the left rotation operation, in the calibration library of this operation, each bit in the original input will directly connect to the next left bit of the TG using the metal interconnection. The controlled nodes of this TG are connected to 1000. For sequence comparison, introduced XOR and AND chain used calibration library is to achieve bit-wise comparison to obtain the correct result. It is connected to the calibration decider, a PT-MIFG logic based TG controlled by the instruction code. Since the calibration decider for all operations are controlled by a unique instruction code, the desired calibration can be correctly enabled.

4.3 Result and Discussion

4.3.1 Logic Cell

Figure 4.6 shows the transient simulations of all proposed cells. We observe that all basic functions can be correctly implemented using MIFG logic. Table 4.2 lists the comparison of static logic and the proposed logic cells using MIFGs. We can see that for simple Boolean logic, the delay of MIFG logic is slightly larger than that of static logic. However, MIFG logic consumes less power than the static logic since the multiple coupling can mitigate the power consumed by

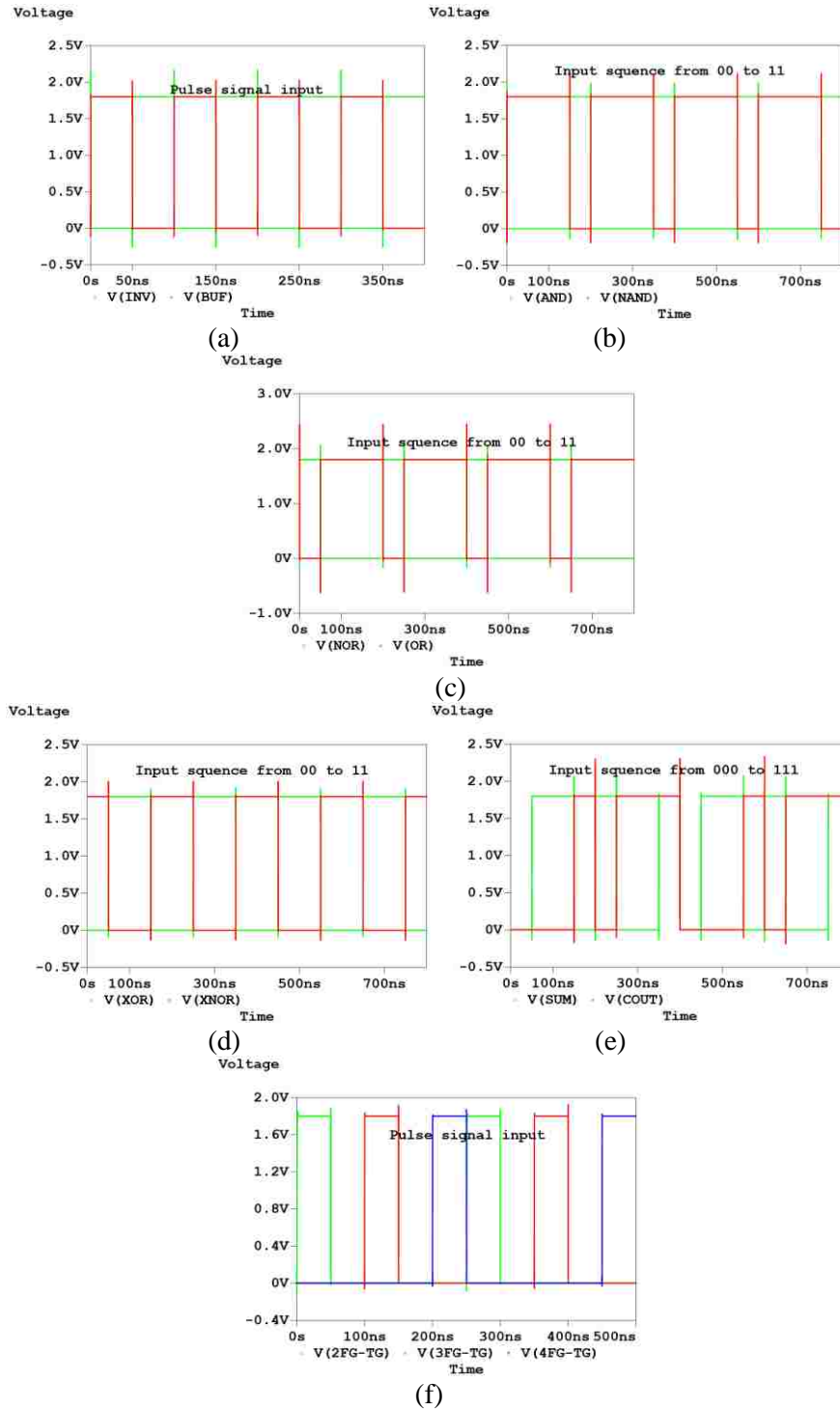


Figure 4.6 Transient simulations of the MIFG logic based cells, (a) INV and BUF, (b) NAND and AND, (c) NOR and OR, (d) XOR and XNOR, (e) full adder, (f) transmission gate with 2, 3 and 4 floating gates.

the dynamic current during logic switching, and coupling capacitors does not consume power. For complex logic including XOR/XNOR, and unit full adder, PT-MIFG logic are used without cascade topology. Thus, these blocks work faster than the gates built by static logic. For the transmission gate designed in PT-MIFG logic, the performance of the proposed one is slightly improved from the view of both power dissipation and delay. The static power dissipation due to leakage current in transistors is also listed as the data within brackets in Table 4.2. Simulation is done in TSMC 180nm technology with 1.8V supply and 0.1fF load. All results of static logic gates use TSMC 180nm standard library. The power dissipation contributed by the leakage current in static logic ranges from 0.64% to 2.13% of the total power dissipation. The power dissipation contributed by the leakage current in MIFG logic varies from 3.7% to 13.3% of the total power dissipation. Thus, the static power dissipation in MIFG logic is more dominant than that in static logic due to large leakage current in MIFG logic with various surface potentials. From the point of view of area, the ALU in MIFG logic consumes relatively large area than in static logic due to coupling capacitors. In essence, the proposed cells using MIFGs are more energy-efficient and behave better in complex logic considering both the speed and power dissipation at the cost of static power dissipation and area than the traditional static logic gates.

4.3.2 Entire System

Figure 4.7 shows layout of the proposed 32-bit ALU in TSMC 180nm CMOS. The chip area is 7.5mm^2 with 69 input ports, 33 output ports, a voltage supply port, and a ground port. Feedforward path and feedback loop occupy 3.3mm^2 and 2.2mm^2 , respectively. Four metal layers are used for interconnections. The most part of the chip is occupied by the coupling capacitors to implement MIFG MOSFETs.

Table 4.2 Comparison of static logic and proposed neuron-like cells using MIFG logic

Performance	Delay (ps)		Power Dissipation@1GHz (nW)*		Area (μm^2)	
	Static Logic	MIFG Logic	Static Logic	MIFG Logic	Static Logic	MIFG Logic
INV	15	21	2.002 (0.0161)	0.944 (0.0634)	7.9	105.5
BUFFER	63	64	3.896 (0.0335)	1.899 (0.0702)	11.8	105.5
NAND	23	28	3.881 (0.0283)	1.313 (0.1359)	11.8	167.8
AND	65	70	4.825 (0.0452)	1.845 (0.0813)	19.7	167.8
NOR	47	68	4.619 (0.0297)	1.148 (0.1523)	11.8	167.8
OR	88	92	5.808 (0.0462)	2.044 (0.0919)	15.7	167.8
XOR	94	46	6.833 (0.0738)	2.092 (0.2402)	27.6	1339.5
XNOR	124	99	7.487 (0.0681)	2.977 (0.1821)	23.6	1339.5
FA	658	216	11.393 (0.2425)	4.809 (0.4253)	339.1	2919.9
TG	37	36	0.723 (0.0092)	0.698 (0.0792)	13.6	537.4

*Data without brackets is the total power dissipation. Data within brackets is the static power dissipation.

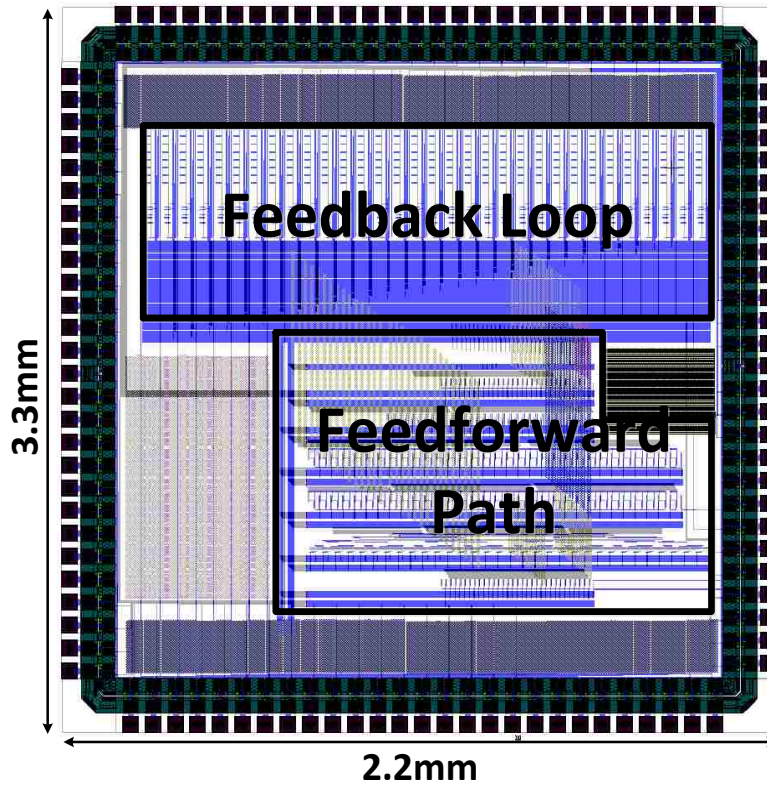
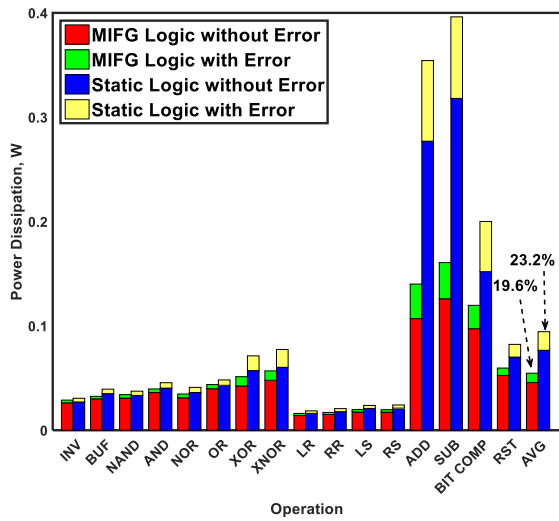
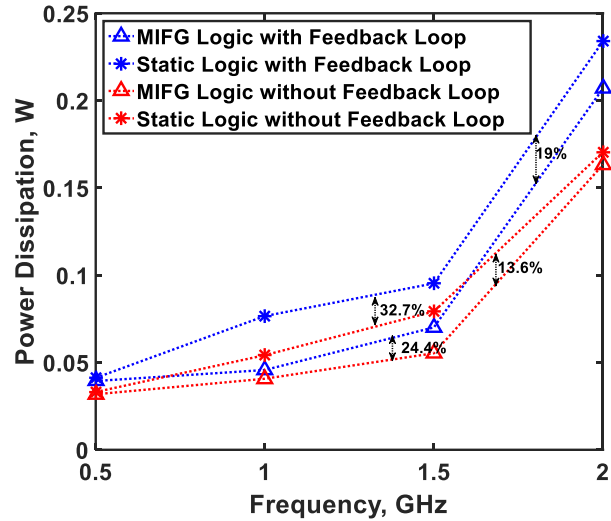


Figure 4.7 Full layout view of the proposed ALU.

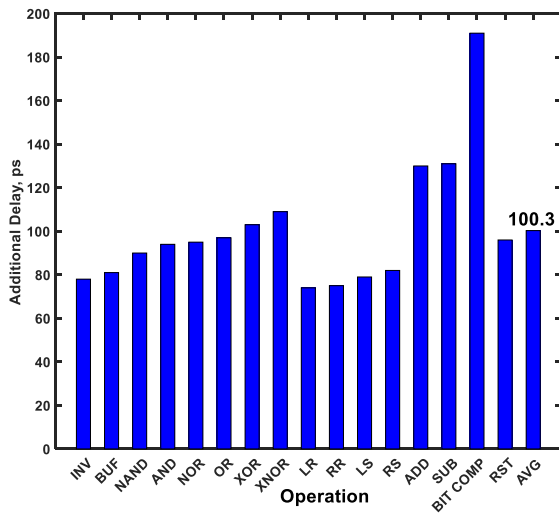
The first part of simulations focuses on power issue. Figure 4.8 (a) exhibits the power dissipation of each operation and its average value of the proposed ALU built by both MIFG logic and CMOS static logic for operation at 1GHz with random input data flow. For the static logic based simulation, we use the same schematic design, in which all MIFG logic are replaced by static logic. The proposed ALU built by MIFG logic dissipates less energy than that designed in static logic, especially for addition, subtraction and sequence comparison. This reduction in power dissipation is due to the short circuit current suppressed by the multiple coupling capacitances of MIFGs in the design. Figure 4.8 (a) also shows the penalty in power dissipation when the error occurs. It can be seen that when complex operations including addition, subtraction and sequence comparison have error, the power dissipation increases the most since the blocks of error detection used for these three operations are more complex than that used for the rest of operations. Averaging all operations, the power growth rate due to calibration in static logic and MIFG logic is 23.2% and 19.6%, respectively. Figure 4.8 (b) exhibits the power dissipation with the variation of frequency of four types of ALUs. It shows the overhead of power dissipation due to the proposed feedback loop and the power comparison of the proposed ALU designed in MIFG logic and static logic. The power dissipations in the proposed ALU built in MIFG logic are 0.0394W and 0.207W at 0.5GHz and 2GHz frequency, respectively. Compared to the proposed ALU designed in traditional static logic, the average reduced rate of power dissipation corresponding to MIFG logic is 19%. If the proposed feedback loop is cancelled, the average reduced rate of power dissipation due to different logic types is 13.6%. Considering the same logic type, the average reduced rate of power dissipation due to the cancellation of the feedback loop is 24.4% and 32.7% corresponding to MIFG logic and static logic, respectively. Thus, the ALU designed by MIFG logic is much more



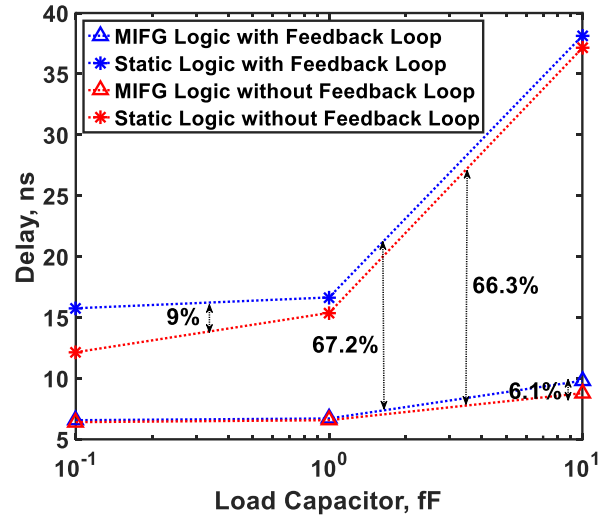
(a)



(b)



(c)



(d)

Figure 4.8 Simulation results, (a) each operation and average value, (b) power dissipation depending on frequency, (c) delay due to feedback loop, (d) delay depending load capacitor.

energy efficient than that by static logic and the proposed feedback loop does not bring large overhead to power dissipation.

Figure 4.8 (c) shows the overhead of computation delay due to the feedback loop. For most operations, additional delay due to error detection and calibration is below 100ps. The error contributes the largest delay for the operations of addition, subtraction and sequence comparison. Especially in sequence comparison, the additional delay almost reaches to 200ps. Since the detection block used for this operation requires logic chain as described in Section 4.1, the average delay is 100.3ps. We also looked into the relationship between load capacitor and computation delay. In most cases, an ALU is not fabricated into a single chip but connects to other large blocks to form a bigger chip, such as the CPU and GPU. Therefore, the drive-in capability of the output port in the ALU is important. We have simulated the computation delay with the variation of load capacitor of four types of ALUs as shown in Figure 4.8 (d). The most impressive feature from Figure 4.8 (d) is that the MIFG logic based ALU is insensitive to an external load. The computation delay in the proposed ALU built in MIFG logic is less than 10ns when the output is loaded by the 10fF capacitor. Compared to the proposed ALU designed by traditional static logic, the average reduced rate of delay corresponding to MIFG logic is 67.2%. If the proposed feedback loop is cancelled, this average reduced rate of delay due to different logic types is 66.3%. Considering the same logic type, the average reduced rate of latency due to the cancellation of the feedback loop is 6.1% and 9% corresponding to MIFG logic and static logic, respectively. In essence, the proposed design could work well if the ALU is a sub-module embedded in a VLSI chip with a large fan-out. The computation delay is not sensitive to the introduced feedback loop, especially in the design using MIFG logic.

The coupling capacitors are commonly used in our design. The dimension mismatch can be introduced due to variation in processing conditions. Coupling capacitors with mismatch influence the accuracy of surface potential, and so degrade the robustness of data processing in cascade logic chain. For the study of dimension mismatch, we set three cases in the following simulations, which are 10%, 15% and 20% mismatch to both capacitors and transistors. Under 500 computation cycles with mismatch at 1GHz with random input data flow in the proposed ALU, the case of 20% mismatch contributes the most additional power dissipation compared to the case without mismatch. The peak of additional power dissipation in this case is 11.7mW. For these three cases, the average additional power dissipations are 1.24mW, 3.11mW and 4.72mW, respectively. Compared to the power dissipation without dimension mismatch, the growth rates of power dissipation corresponding to three mismatches are 2.7%, 6.8% and 10.3%, respectively. Figure 4.9 shows the error rate with the variation of frequency of two types of ALUs. We observe that the occurrence of number of errors increases with the increase in frequency since high frequency operation cannot guarantee enough time given to each stage for complete processing. The error rates at 2GHz in MIFG logic based ALU are 1.7%, 2.2%, 6.2% and 7.3% corresponding to no mismatch, 10% mismatch, 15% mismatch and 20% mismatch, respectively. For the same ALU in static logic, these four error rates are 9.1%, 9.1%, 9% and 9%, respectively. From the view of point of mismatch, static logic is less sensitive to dimension mismatch than MIFG logic. In the design using MIFG logic, the error rate varies largely with the mismatch variation. The insensitivity to mismatch is an advantage of static logic compared to the proposed MIFG logic. However, the unwanted error can be detected and calibrated by the proposed feedback loop and feedforward stage 3.

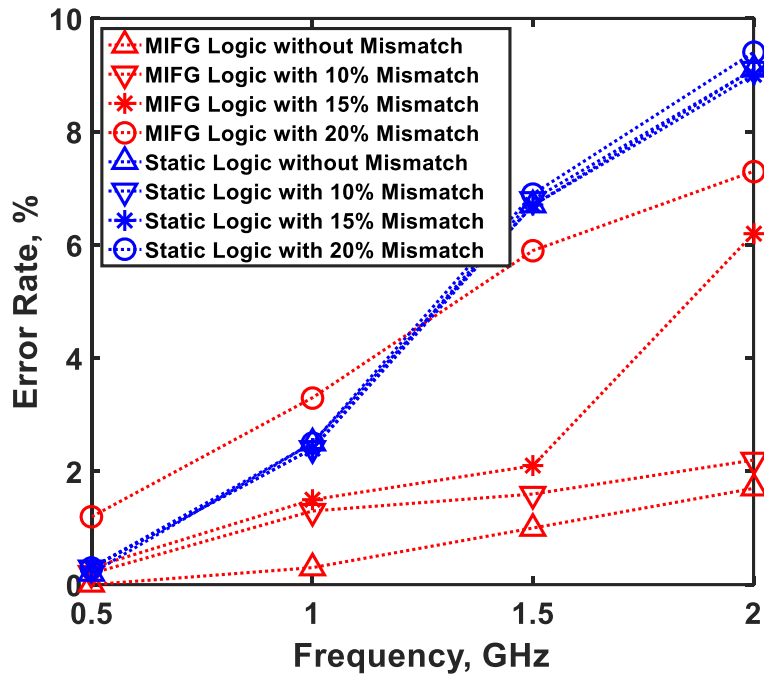


Figure 4.9 Error rate with the variation of mismatch.

Comparing to other work of MIFG MOSFET based ALU [110, 111], in our work, both the number of functions and bit length of our work are more than that in previous work. From the point of view of structure, the proposed design has introduced a feedback loop which can calibrate unwanted error. Besides, the design of logic cells in our work is simple in comparison to previous work. In [110], the design of full adder still used the cascade logic chain which is penalty to both latency and area. In [111], the grouped cell using MIFGs can achieve multi-function and so cancel the stage of MUX array. However, this design strategy is not suitable when the number of required functions is increasing since FPDs of all functions cannot be always the same.

4.4 Conclusion

We have proposed a 32-bit ALU. The essence of this design is to use MIFG logic to build neuron-like cells as the logic gates and use a feedback loop to detect error. The post-layout simulation shows that the proposed design is energy-efficient, fast-computing and insensitive to load capacitor compared to the one built in traditional static logic. The overhead due to the proposed feedback loop and the performance penalty of both power and error due to mismatch are acceptable.

CHAPTER 5

LONG SHORT-TERM MEMORY NETWORK DESIGN FOR ANALOG COMPUTING

In this chapter, we introduce a novel design method to implement a neural accelerator working for LSTM network. The proposed method not only fully uses ASP to process internal computing in LSTM network, but also has an energy-efficient signal interface to let the LSTM accelerator work with digital circuits. The case of the proposed computing core with memory access is studied to prove the feasibility of the computing extension of LSTM network layer. The verification methods include post-layout simulations, sensitivity study and LSTM network simulation using TensorFlow based on the dataset of Penn TreeBank (PTB).

5.1 Long Short-Term Memory Network

RNN is a cycle network that allows data processing with the dependency of past information and LSTM is one of the extensions of RNN. When the time span between two instances is large, the network will lose the ability of connecting two points used for processing and prediction. LSTM can solve the problem of long-term dependency and so it can be used in variety of applications [112]. LSTM uses the pipelined method using the same standard cells. The output of current state influences the next state. The standard cell of LSTM is shown in Figure 5.1, which consists of six independent steps to follow dotted ellipses.

The four steps in Figure 5.1 represent forget gate layer, input gate layer, cell gate layer and output gate layer, respectively. In these, the argument of each activation function is the sum of two matrix multiplications and a bias, $W_x x_t + W_h h_{t-1} + b$, where W_x , x_t , W_h , h_{t-1} and b are weighted matrix of input vector, input vector; weighted matrix of output vector, output vector; and bias, respectively. Considering the analog implementation of LSTM, there are two additions of column vectors in a single argument going through an activation function. To simplify the circuit design,

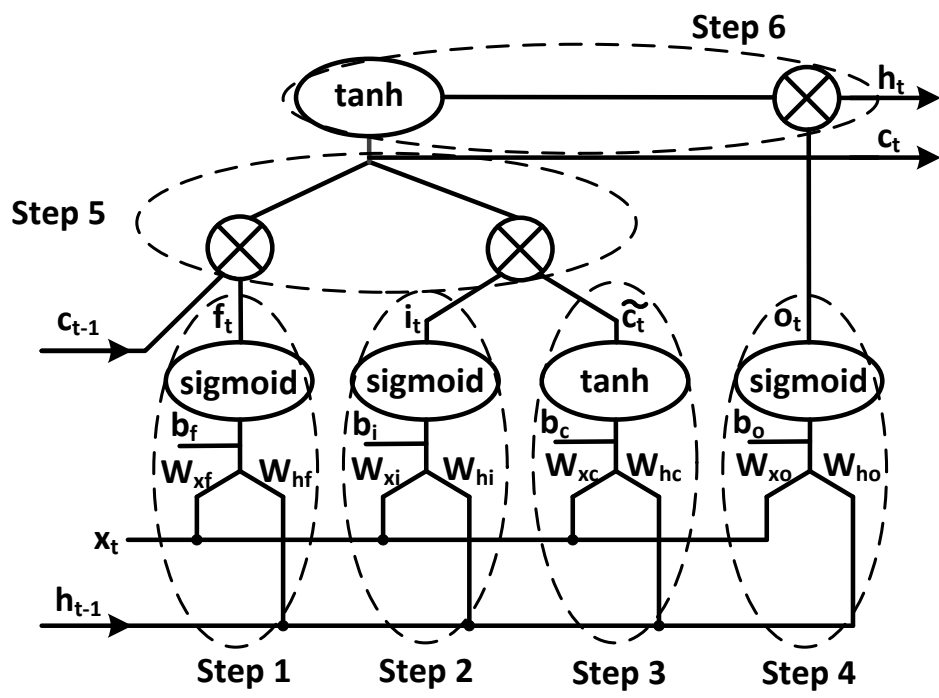


Figure 5.1 Standard cell of LSTM network.

we can convert these two additions into a single addition. If dimensions of all weighted matrixes are $N \times N$, and the dimensions of the rest of column vectors are $N \times 1$ in LSTM, we can embed the last bias term into the first term in the argument of the first four steps. Based on the Equations (3)-(8) in [64], the new form of LSTM can be then expressed as follows:

$$\begin{cases} f_t = \text{sigmoid}(\hat{W}_{xf}\hat{x}_t + \hat{W}_{hf}\hat{h}_{t-1}) \\ i_t = \text{sigmoid}(\hat{W}_{xi}\hat{x}_t + \hat{W}_{hi}\hat{h}_{t-1}) \\ \tilde{c}_t = \tanh(\hat{W}_{xc}\hat{x}_t + \hat{W}_{hc}\hat{h}_{t-1}) \\ o_t = \text{sigmoid}(\hat{W}_{xo}\hat{x}_t + \hat{W}_{ho}\hat{h}_{t-1}) \\ c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\ h_t = o_t \circ \tanh(c_t) \end{cases} \quad (5.1)$$

In Equation set (5.1), each new weighted matrix of the input vector, \hat{W}_x represents $[\mathbf{W}_x, \mathbf{b}]$, each new input vector, \hat{x}_t represents $[\mathbf{x}_t^T, \mathbf{1}]^T$, each new weighted matrix of the output vector, \hat{W}_h represents $[\mathbf{W}_h, \mathbf{0}]$, and each new output vector, \hat{h}_{t-1} represents $[\mathbf{h}_{t-1}^T, \mathbf{0}]^T$. From above set of equations, we can see that all bias terms are integrated in $\hat{W}_x\hat{x}_t$ and there is only one addition in the argument of the first four layers. The dimensions of all new weighted matrixes and column vectors in the arguments are $N \times (N+1)$ and $(N+1) \times 1$, respectively. It should be noted that in analog implementation, each element in both $\hat{W}_x\hat{x}_t$ and $\hat{W}_h\hat{h}_{t-1}$ is an analog signal. The matrix multiplication corresponding to $\hat{W}_x\hat{x}_t$ is $N \times (N+1)$ matrix multiplied by $(N+1) \times 1$ column vector in order to cover the addition with the bias term. The matrix multiplication corresponding to $\hat{W}_h\hat{h}_{t-1}$ can be simplified to $N \times N$ matrix multiplied by $N \times 1$ column vector without the null multiplication. Through this new form of LSTM shown in Equation set (5.1), we have removed one stage implementing the addition with the bias term to not only accelerate computation but also to reduce both the circuit cost and interconnection complexity. The output in every sub-equation is still an $N \times 1$ column vector as desired. All elements in matrixes still do the same calculation. Thus, there

is no error introduced by the proposed modification. The computation flow as shown in Equation set (5.1) includes matrix multiplication, bitwise multiplication to obtain Hadamard product, addition and two activation functions, which are sigmoid and hyperbolic tangent. The first four steps can be computed simultaneously.

5.2 MIFG MOSFETs for DAC Design

Section 1.6.2 has introduced the fundamental theory of MIFG MOSFET. The property of a MIFG MOSFET through which the output can be determined by multiple inputs, matches to the signal flow in DAC, which is the initial stage used for ASP. Therefore, we can use MIFG MOSFETs to achieve a DAC function. For N -bit digital signal applied in a MIFG MOSFET, $V_{D_{N-1}}$, $V_{D_{N-2}}$, ..., V_{D_1} , V_{D_0} , if we set the ratio of their corresponding coupling capacitors to $2^{N-1}:2^{N-2}: \dots :2^1:2^0$, the surface potential on the floating gate of a MIFG MOSFET can be programmed as follows:

$$\Phi_f = \frac{2^{N-1}C_{unit}V_{D_{N-1}} + 2^{N-2}C_{unit}V_{D_{N-2}} + \dots + 2^0C_{unit}V_{D_0}}{C_{ox} + 2^{N-1}C_{unit} + 2^{N-2}C_{unit} + \dots + 2^0C_{unit} + C_p} \quad (5.2)$$

where C_{unit} is the unit capacitor implemented in physical design to suppress the device mismatch. If we neglect both C_{ox} and C_p , the numerator in above equation shows a function of DAC in which the multiple inputs and the surface potential are digital and analog signals, respectively. If the denominator in above equation is C_{total} , the step voltage for the conversion is $V_{DD} \times C_{unit} / C_{total}$. This implementation does not require an independent circuit block to achieve a DAC function. Input transistors in analog multipliers of this work are directly connected to multiple digital inputs. Traditional DACs in ASP systems need several feedback loops with operational amplifiers to achieve stability, the required response time will largely increase the conversion latency. However, the proposed one, in which the multiple inputs directly couple to the floating gate of a transistor, does not need response time or the consideration of stability, and thus is much faster than

traditional DACs. The output of the proposed conversion is directly applied to the channel of the transistor to control its state to achieve multiplication. In addition, the power dissipation of the proposed conversion is relatively smaller than that in traditional DAC due to the cancellation of feedback loop. To increase the precision in computation, it is feasible to increase the N in Equation (5.2) at the cost of the area due to coupling capacitors.

For the physical layout design of MIFGs using the concept of Figure 1.6 (b), double-poly layer is widely used to form a floating gate and coupling capacitive gates shown in Figure 5.2 (a). Thus, we observe that this style of physical design can cause the values of coupling capacitors are restricted by the dimension of the transistor connected with MIFGs, which reduces the design freedom of analog IC. To avoid the mutual restriction between coupling capacitors and customized ratio of the transistor with MIFGs, in this work, for any transistor with MIFGs, we use another structure shown in Figure 5.2 (b) to achieve coupling capacitors, which are implemented by standard n-diffusion capacitors. This design style of MIFG MOSFETs is fully compatible with the fabrication flow of standard CMOS, and already proposed, fabricated and tested in the work of [54, 55, 110, 113]. The essence of this structure is that the coupling capacitors are designed in the area beside MIFG transistors but not over the floating gate. Through this structure, the dimension of a MIFG transistor and values of coupling capacitors are independent to each other. Another important issue is the method of simulating MIFG device and circuit. In this work, we use the resistor-capacitor pairs with a large grounding resistor to achieve signal synchronization and solve the ‘floating node’ error which is due to multiple inputs with the same electrical node in SPICE simulation [114].

5.3 Long Short-Term Memory Network Design

5.3.1 Analog Multiplier with Digital Signal Input

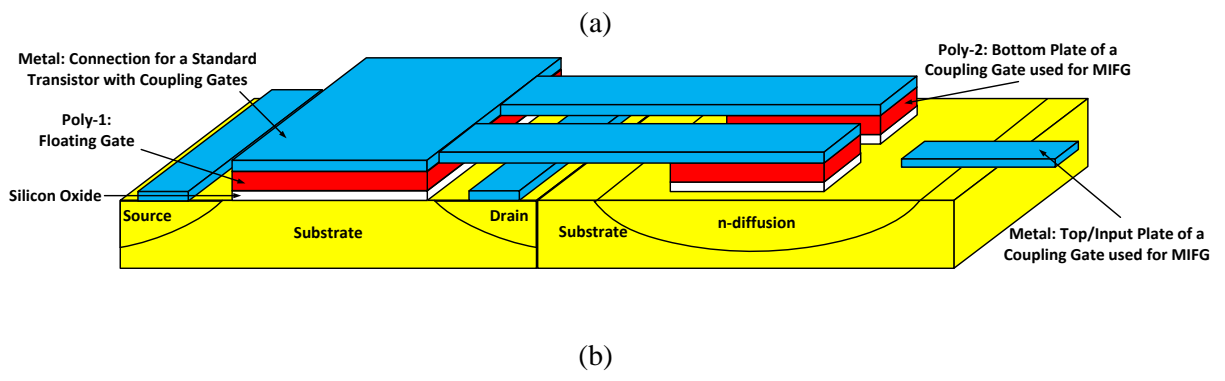
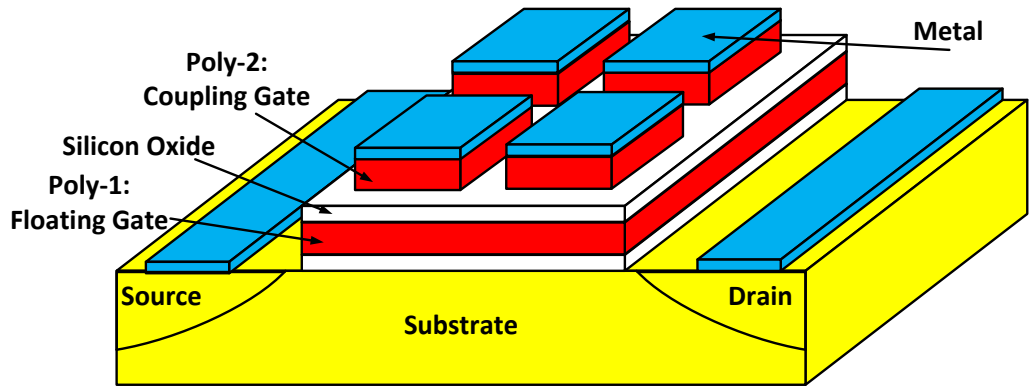


Figure 5.2 MIFG MOSFET, (a) 3D view of overlapping-dual-poly structure and (b) 3D view of adjacent n-diffusion capacitor structure. Note that both poly and active contacts are ignored in two 3D views.

Analog multipliers are more suitable to process data with increasing bit length than digital multipliers [115, 116]. However, it is very difficult to achieve rail-to-rail input and an output voltage swings. One feasible method to increase the signal swing of an analog multiplier is to increase supply voltage at the cost of power dissipation [117, 118]. Another method is to allow transistors in analog multipliers to work in the linear region [119]. In order to save power, we scaled down the swing of input signals. This is needed to make input digital signals and surface potential on the floating gate after data conversion to be compatible with the signal flow without increasing the supplied voltage. For the input stage of an analog multiplier using MIFG MOSFETs, we introduced another coupling capacitor, C_g connecting to the ground. Equation (5.2) can be expressed as follows:

$$\Phi_f = \frac{2^{N-1}C_{unit}V_{D_{-N-1}} + 2^{N-2}C_{unit}V_{D_{-N-2}} + \dots + 2^0C_{unit}V_{D_{-0}}}{C_{ox} + 2^{N-1}C_{unit} + 2^{N-2}C_{unit} + \dots + 2^0C_{unit} + C_p + C_g} \quad (5.3)$$

Setting the allowable input swing to $[0, \alpha V_{dd}]$, when all digital inputs are high logic, the surface potential reaches to the upper bound, αV_{dd} . This case can be expressed as follows:

$$\alpha V_{dd} = \frac{2^{N-1}C_{unit}V_{dd} + 2^{N-2}C_{unit}V_{dd} + \dots + 2^0C_{unit}V_{dd}}{C_{ox} + 2^{N-1}C_{unit} + 2^{N-2}C_{unit} + \dots + 2^0C_{unit} + C_p + C_g} \quad (5.4)$$

Since C_{ox} and C_p are small enough compared to other coupling capacitors, we can neglect these two terms. Then above equation can be rewritten as the function of C_g as follows:

$$C_g = \frac{(1-\alpha)}{\alpha} (2^{N-1} + 2^{N-2} + \dots + 2^1 + 2^0) C_{unit} \quad (5.5)$$

Depending on Equation (5.5), we can use C_g with other coupling capacitors shown in Equation (5.3) to execute the function of DAC with the scaling down of the input voltage.

Based on the design presented in [116], we propose an analog multiplier with programmable MIFG MOSFETs guided by Equations (5.3) and (5.5) as shown in Figure 5.3. It is a differential input pair mapping to differential output in four-quadrant. The two input pairs

generate the two square terms, $[(V_{I+}-V_{I-})+(V_{2+}-V_{2-})]^2$ and $[(V_{I+}-V_{I-})-(V_{2+}-V_{2-})]^2$. Then through the cross interconnection of source followers with inputs, these two terms are subtracted with each other so that the internal square terms such as $(V_{I+}-V_{I-})^2$ and $(V_{2+}-V_{2-})^2$ are cancelled and only the term with the function of $(V_{I+}-V_{I-})(V_{2+}-V_{2-})$ is left. The voltage output can be expressed as follows:

$$V_{o+} - V_{o-} = R_{load} (I_{o+} - I_{o-}) = \frac{1}{4} K_n \left(\frac{W}{L} \right)_{sf} R_{load} [(V_{I+} - V_{I-})(V_{2+} - V_{2-})] \quad (5.6)$$

where K_n is the transconductance parameter of n-channel transistors, R_{load} is the resistance of diode connected transistors (M_1 and M_4) in the output load. $(W/L)_{sf}$ represents the dimension ratio of the transistors (M_5 , M_6 , M_7 and M_8) in source followers. Looking into above equation, K_n and $(W/L)_{sf}$ are determined by the fabrication process and the unit gain of the source follower, respectively. R_{load} in the circuit is sensitive to temperature with positive temperature coefficient [120]. If we use the current signal as the output, R_{load} in above equation can be cancelled to mitigate the signal variation due to the temperature fluctuation. The most significant advantage of using current signal as the transmission signal is that the addition of two current signals can be achieved simply by the interconnection using the Kirchhoff's current law. However, the addition of two voltage signals cannot be achieved by direct series connection and results in the voltage sum with non-linear composition [121]. To obtain an accurate sum of two voltage signals, the mixer is required at the cost of delay and power dissipation. Therefore, considering the algorithm of LSTM and the hardware cost, it is better to use current signal as the transmission signal than the voltage signal.

The feasibility of quantization used in LSTM to reduce hardware cost has been studied [66, 122]. In our work, we set the single number in both matrixes and column vectors to be a 4-bit digital signal and α in Equation (5.4) to 0.5, which means the input voltage is scaled down from 0V to 0.9V that is the half of supplied voltage (1.8V) in TSMC 180nm CMOS process. Combining

the analysis from Equations (5.3), (5.5) and (5.6), the output current signal can be expressed as follows:

$$\begin{aligned}
I_{o+} - I_{o-} &= \frac{1}{4} K_N \left(\frac{W}{L} \right)_{sf} \gamma_{(W/L)} (V_1 V_2) \\
&= \frac{1}{4} K_N \left(\frac{W}{L} \right)_{sf} \gamma_{(W/L)} \left(\frac{8C_{unit} V_{A3} + 4C_{unit} V_{A2} + 2C_{unit} V_{A1} + C_{unit} V_{A0}}{8C_{unit} + 4C_{unit} + 2C_{unit} + C_{unit} + 15C_{unit}} \right) \\
&\quad \times \left(\frac{8C_{unit} V_{B3} + 4C_{unit} V_{B2} + 2C_{unit} V_{B1} + C_{unit} V_{B0}}{8C_{unit} + 4C_{unit} + 2C_{unit} + C_{unit} + 15C_{unit}} \right)
\end{aligned} \tag{5.7}$$

where $\gamma_{(W/L)}$ is the dimension ratio of two p-type MOSFETs in the current mirror used for the output stage. The last two terms in brackets are two 4-bit digital signals to be multiplied. The ratio of four coupling capacitors used in the function of DAC is 8:4:2:1 as shown in Equation (5.7). C_g is $15C_{unit}$ which is used for signal scaling. Using above equation, the external digital signals can be multiplied.

For determining the value of C_{unit} , we need to study the signal attenuation at the surface potential due to the influence of C_{ox} and C_p . Setting all digital inputs to high logic, Figure 5.4 shows the signal attenuation dependence on varying C_{unit} . We can see that when the dimension of the driven transistor is large ($W/L=2000\text{nm}/200\text{nm}$), it requires a large unit capacitor to mitigate the signal attenuation at the surface potential since C_{ox} is proportional to the dimension of a transistor. When the coupling capacitor is larger than that of C_p and C_{ox} , the surface potential can accurately reflect the programmable input using MIFG MOSFETs. We should consider circuit area also because capacitors in MIFG MOSFETs occupy larger area than in normal transistors. Therefore, we choose 50fF as the unit size capacitor as a trade-off between area and signal accuracy.

Table 5.1 shows W/L ratios of transistors shown in Figure 5.3. Setting the resistor used for biasing the tail current to $5\text{k}\Omega$ and common mode voltage to 0.9V , we obtain the transient

simulation of the multiplier as shown in Figure 5.5 (a). It can be seen that the proposed multiplier can process the signal from standard digital interface, 4-bit digital signal ($V_{pp}=1.8V$) to analog current signal with the swing of $\pm 180\mu A$. Figure 5.5 (b) is the dc transfer curve to observe the linearity of the proposed multiplier. We observe that a smaller input leads a better linearity of the output. Four curves can be roughly seen as linear curves which ensure the correct multiplication. Total harmonic distortion (THD) is another important metric to evaluate the performance of analog multipliers. THDs with input of 0.9V and 0.3V swings are 1.52% and 0.93%, respectively, both of which are acceptable and so cannot largely influence the multiplication due to harmonics. This analog multiplier will also be used for both matrix multiplication and bitwise multiplication, which will be introduced later. Compared to digital multiplier with multiple-bits, the proposed method can implement both multiplication and DAC function as the front-end of digital interface. The most attractive feature is that the analog multiplier is much faster than the digital multiplier since it only uses a single analog stage to obtain the result instead of the pipelined topology in digital multiplication.

5.3.2 Design of Other Functional Blocks

Besides analog multipliers used for matrix and bitwise multiplications, there are circuits implementing activation functions current comparator, and current mirror required for the entire computation flow of LSTM. The digital implementation of an activation function usually uses LUT to search results or a step linear function to approximate the non-linear curve [66, 123]. The accuracy mainly relies on table capacity and search principle. In analog design of an activation function, only one circuit block can execute the computation [124, 125]. In our design, the received signal from multiplication is a current signal and the output signal from an activation function is used for the following bitwise multiplication according to LSTM algorithm. Thus, we need to build

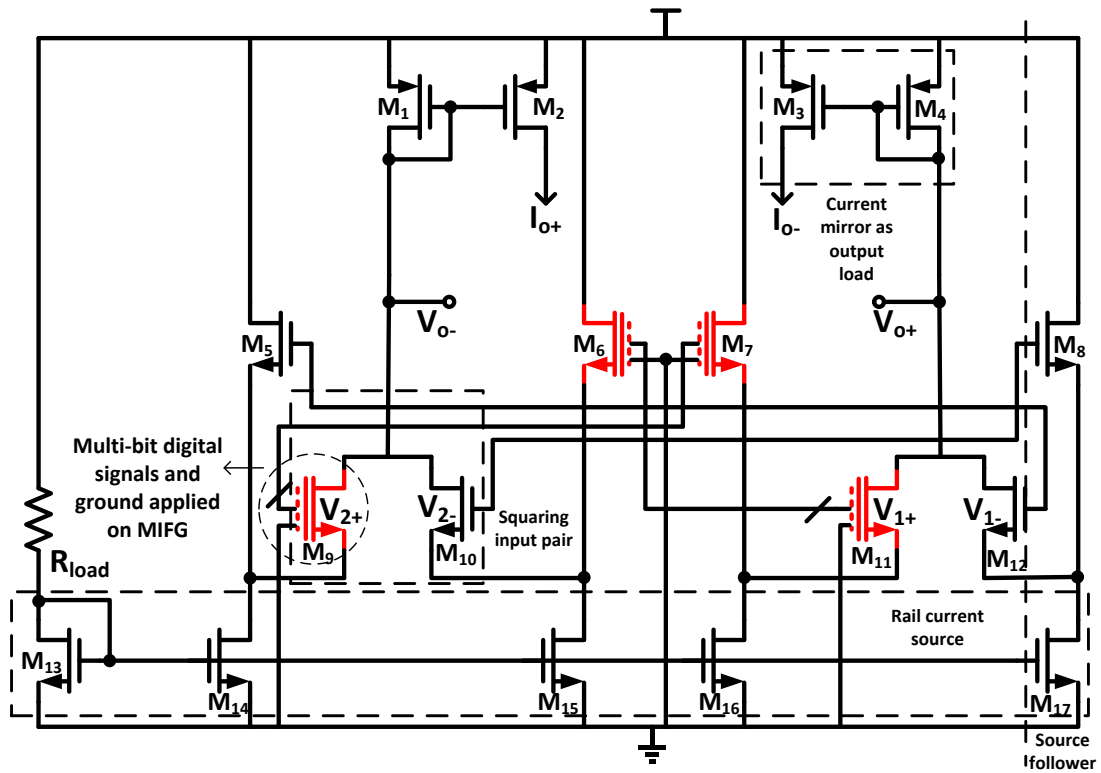


Figure 5.3 Circuit diagram of the proposed analog multiplier with the functions of both voltage scaling and DAC. Highlighted transistors are MIFG MOSFETs.

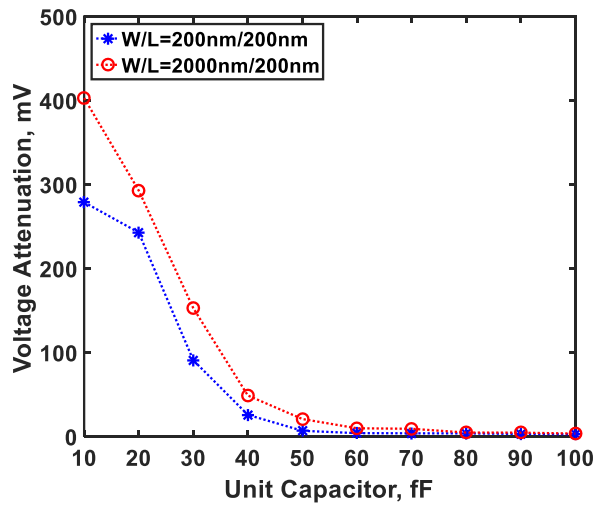
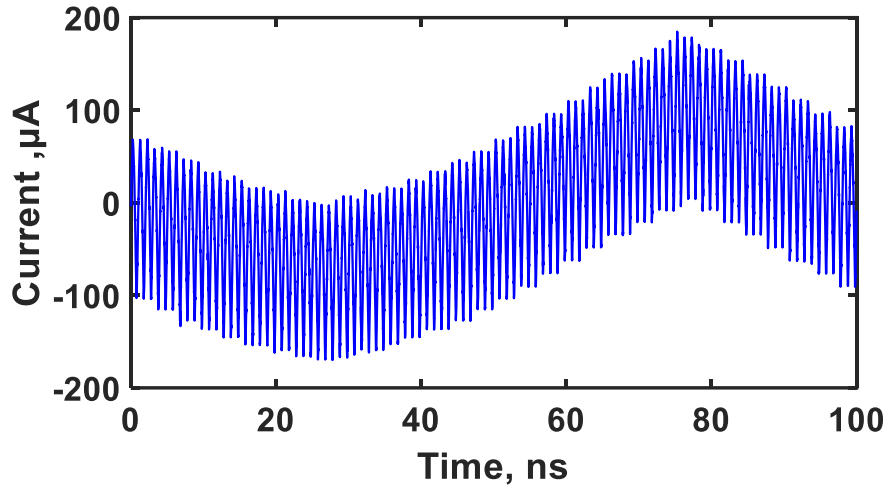
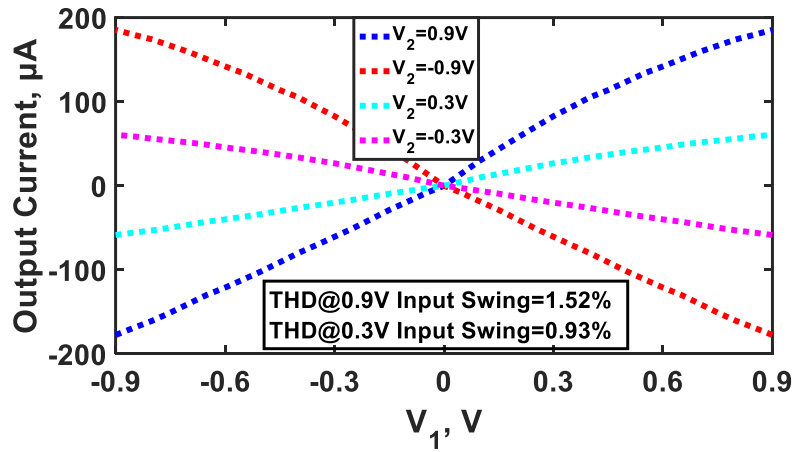


Figure 5.4 Voltage attenuation versus unit capacitor in MIFG driven input transistor.



(a)



(b)

Figure 5.5 The proposed analog multiplier with MIFGs, (a) transient simulation, two 4-bit inputs are from all high logic (1.8V) flipping to all low logic (0V) periodically under 1GHz and 10MHz and (b) dc transfer curve.

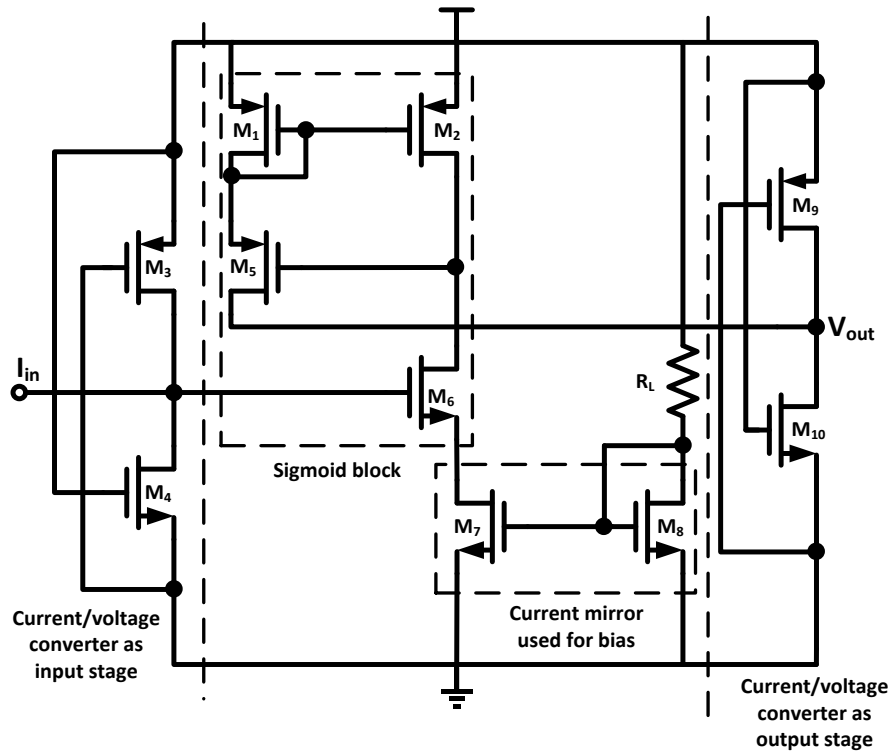
Table 5.1 W/L ratio of transistors in the proposed analog multiplier of Figure 5.3

Transistor	W/L (nm/nm)
M ₁₋₄	10000/400
M ₅₋₁₂	200/200
M ₁₃	400/400
M ₁₄₋₁₇	8000/400

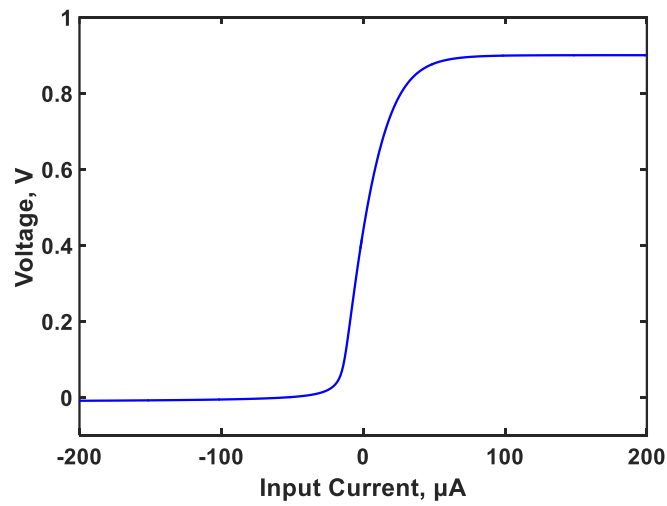
a block of the activation function with the current signal as an input and voltage signal as an output. The analog circuit design of sigmoid and hyperbolic tangent functions is shown in Figure 5.6 (a) and Figure 5.7 (a), respectively. The intermediate block computing the activation function uses voltage signal as the input and current signal as the output. It can be seen that there are two current/voltage converters to match to the correct signal flow. Since the hyperbolic tangent function has a negative value, a reverse voltage bias is used to allow the output drop to a negative value. Using the current signal as an input, the transfer functions corresponding to these two circuits are expressed as follows,

$$\left\{ \begin{array}{l} V_{sigmoid_out}(I_{in}) = \frac{1}{2} K_p \left(\frac{W}{L}\right)_5 (gm_6 r_{o2} r_{o4} I_{in} - \sqrt{\frac{gm_6^2}{K_n K_p \left(\frac{W}{L}\right)_2 \left(\frac{W}{L}\right)_6} - V_{dd} - 2V_{tp}})^2 r_{o10} \\ V_{tanh_out}(I_{in}) = \left\{ \frac{1}{2} K_p \left(\frac{W}{L}\right)_5 (gm_6 r_{o2} r_{o4} I_{in} - \sqrt{\frac{gm_6^2}{K_n K_p \left(\frac{W}{L}\right)_2 \left(\frac{W}{L}\right)_6} - V_{dd} - 2V_{tp}})^2 - I_{11/12} \right\} r_{o10} \end{array} \right. \quad (5.8)$$

In the transfer function of sigmoid, the input current shows a square relation with output voltage in effective input swing. Once the input current is out of effective input swing after front-end I/V converter (M₃ and M₄) due to M₅ as a feedback controlling the common source (CS) amplifier (M₂ and M₆), the output voltage will stop changing to go to cut-off region and form top and bottom signal-lines. This principle is also applied in the transfer curve of hyperbolic tangent function. However, we have introduced a reverse voltage bias (M₁₁ and M₁₂) to pull down the swing of output voltage in hyperbolic tangent function. The reason we let two transistors work in sub-threshold region to generate the negative voltage bias is that the characteristic of a transistor in sub-threshold region is less sensitive to external voltage variation than that in linear or saturation region. The W/L ratio of transistors in both two circuits are listed in Table 5.2. R_L in Figure 5.6 (a) is 5k Ω , R_{L1} in Figure 5.7 (a) is 20k Ω , and R_{L2} in Figure 5.7 (a) is 5k Ω . Figure 5.6 (b) and Figure

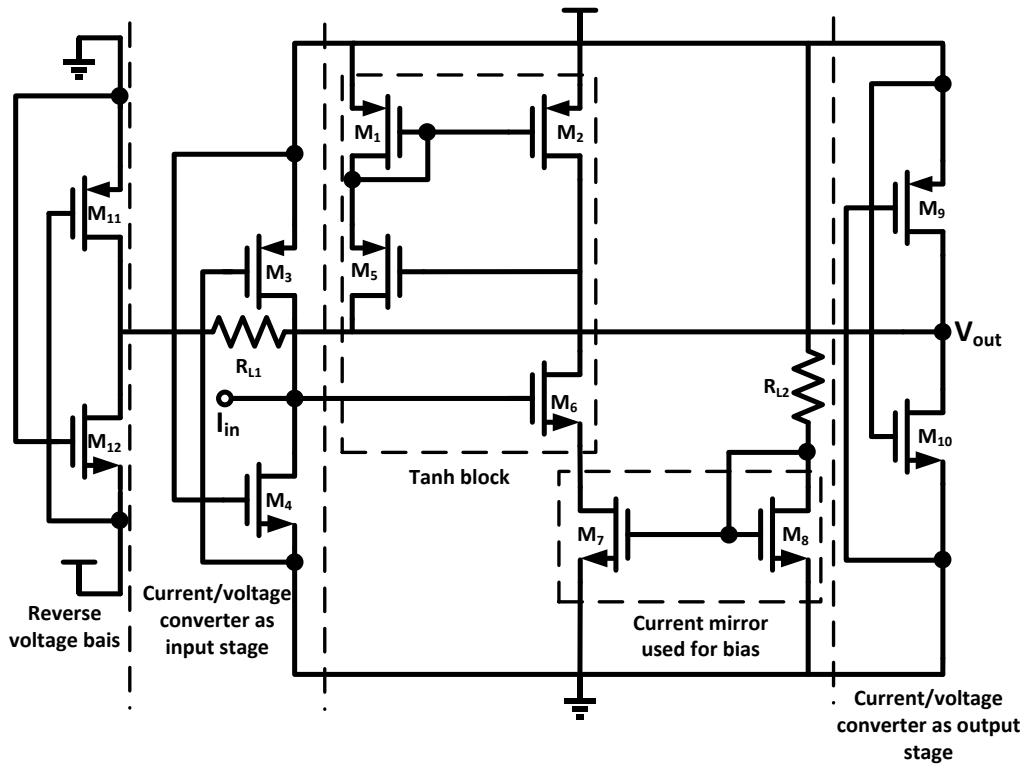


(a)

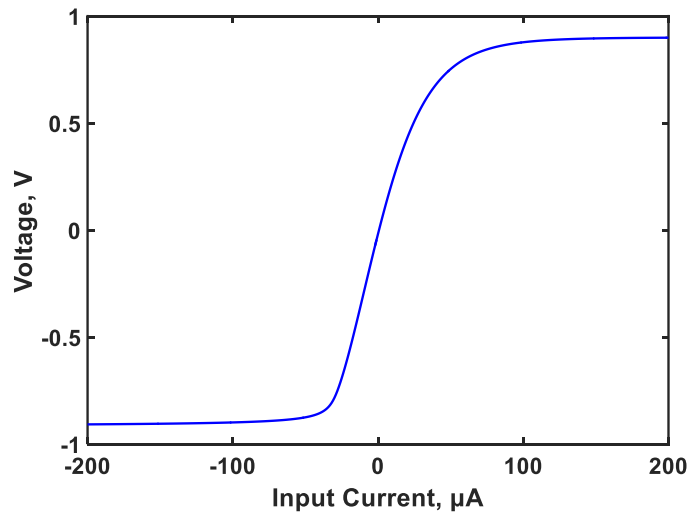


(b)

Figure 5.6 Analog implementation of sigmoid function, (a) circuit diagram, (b) transfer characteristics.



(a)



(b)

Figure 5.7 Analog implementation of hyperbolic tangent function, (a) circuit diagram, (b) transfer characteristics.

5.7 (b) show the simulated static characteristics of sigmoid and hyperbolic tangent functions, respectively.

For the back-end of our design in which the analog current signal is to be converted to a digital voltage signal, we use current comparators for the signal conversion. The delay of a current compactor is sensitive to the current resolution [126, 127, 128]. The design reported by [126] is an inverter chain which uses a passive device as a feedback loop to effectively boost the resolution to less than nano-ampere level. Since the resolution in our work only reaches to a micro-ampere level, considering design area and acceptable resolution, we use the inverter based circuit design without passive device as the current comparator shown in Figure 5.8 (a). The input current is the current difference between the output current of LSTM and reference current through the interconnection. The inverter chain in the bottom is used for signal integrity and leave enough time for the signal to reset. The inverter chain in the top feedback loop is to reset the voltage signal after a single comparison cycle. Setting the W/L of all n-channel MOSFETs and p-channel MOSFETs as 800nm/600nm and 1200nm/200nm, respectively, Figure 5.8 (b) shows the relationship between input current and delay. We observe maximum delay of 2.67ns when the current resolution is 1 μ A. For the design of current mirror biasing the reference current used for the function of DAC in the back-end, we used p-MOSFET current mirror to match the signal polarity in our design. An inevitable problem is the decrease of current accuracy due to the channel modulation effect in MOSFETs [120]. The reference current is the key signal to give correct digital signal output. Therefore, the channel lengths in all p-channel transistors of current mirrors are set to 800nm.

5.3.3 LSTM Operation

Figure 5.9 shows the system level design of LSTM. We have divided the entire signal flow into four operations shown in Figure 5.9, which are described as follows:

Table 5.2 W/L ratio of transistors in the circuits of Figures 5.5 and 5.6

Sigmoid		Hyperbolic tangent	
Transistor	W/L (nm/nm)	Transistor	W/L (nm/nm)
M ₁	10000/400	M ₁	12000/400
M ₂	2000/400	M ₂	2000/400
M ₃	3100/400	M ₃	3000/400
M ₄	1000/400	M ₄	1200/200
M ₅	1400/200	M ₅	5000/200
M ₆	200/200	M ₆	200/200
M ₇	800/400	M ₇	800/400
M ₈	400/400	M ₈	400/400
M ₉	800/200	M ₉	1000/200
M ₁₀	1600/200	M ₁₀	1800/200
		M ₁₁	400/400
		M ₁₂	400/400

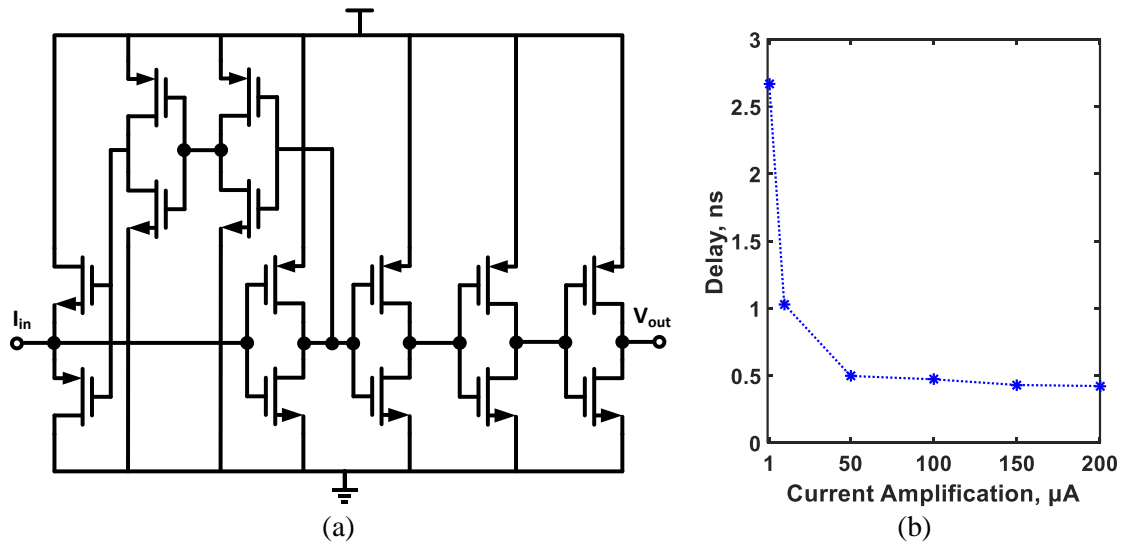


Figure 5.8 Current comparator used as back-end stage, (a) circuit diagram, (b) delay dependence on current amplification.

1) Operation 1 uses front-end analog multiplier based crossbars and interconnections to achieve DAC function and the term, $\hat{W}_x \hat{x}_t + \hat{W}_h \hat{h}_{t-1}$;

2) In operation 2, results from the operation 1 will go through the blocks implementing activation functions or I/V converters to obtain all internal gate layers in LSTM network;

3) Operation 3 uses analog multipliers without MIFG and the blocks of activation functions to obtain both c_t and h_t ;

4) Operation 4 uses current comparators with current mirrors to achieve ADC function.

In this flow, analog multipliers, sigmoid blocks and hyperbolic tangent blocks use 0.9V and -0.9V as power supply and ground levels, respectively. For the current comparators used as back-end interface, they use 1.8V and 0V as power supply and ground levels, respectively.

The signal flow with slash arrows represents both input and output in standard digital domain. Internal signal flow with wavy lines represents current signals. The signal flow with sigma function mean the sum of current signals use the wire interconnection in physical implementation. For both matrix multiplication and bitwise multiplication to obtain Hadamard product, we use the proposed analog multiplier to build a crossbar, in which current results in each row are obtained by the wire interconnection. Especially, the matrix multiplication uses MIFG MOSFETs as the front-end to achieve DAC function. The rest of blocks do not use MIFG MOSFETs since the signal flow has already been converted to an analog signal. This structure with only stage connecting to MIFG not only avoids large load for each analog block but also saves chip area. For the signal flow going through activation functions, it is current signal as the input and voltage signal as the output as described in the previous sub-section. Note that for the term c_{t-1} , it directly comes from the previous cell without going through the matrix multiplication or an activation function. Thus, to obtain the correct signal flow, we still allow c_{t-1} to go through matrix multiplication with MIFG

MOSFETs, which means c_{t-1} will be converted to an analog signal and multiplied by 1. Then current/voltage converter is applied to make c_{t-1} connect to the following crossbar of bitwise multiplication.

In our structure, the single analog current is converted to a 4-bit digital signal based on successive approximation. The method uses current signal to make a comparison with the reference current step by step to obtain multiple-bit as the final output. The proposed conversion flow is shown in Figure 5.9. First, we use the array of current mirrors as the current duplicator to generate four same current signals representing h_t . Then using wire interconnection, we obtain the current difference between the reference current and the current signal of h_t . The reference current in this step is set to $0\mu\text{A}$, $-90\mu\text{A}$, $-135\mu\text{A}$ and $-157.5\mu\text{A}$ corresponding to MSB to LSB in our design. The first reference current can be simply obtained via connecting to the ground. The rest of current signals can be obtained through current mirrors. The last step is to use current comparators to compute the final 4-bit output. When the current difference is a positive value, the digital bit is high. Otherwise, the digital bit is low. Compared to successive approximation register (SAR) ADC, we used an independent comparison block to each analog output to speed up data conversion instead of the single register with relative clock tree converting repeatedly. In addition, for the conversion of N -bit in flash ADC, it requires 2^{N-1} voltage comparators. While in our proposed conversion, it only requires N current comparators for each analog signal. In our design to implement LSTM network, we only use MIFG MOSFETs in the first stage that executes matrix multiplication to convert external digital signals to analog signals, which are processed in internal blocks. In the internal computation, all addition behaviors are implemented by the simple physical interconnection of current signals to speed up computation. Once the data computation is finished,

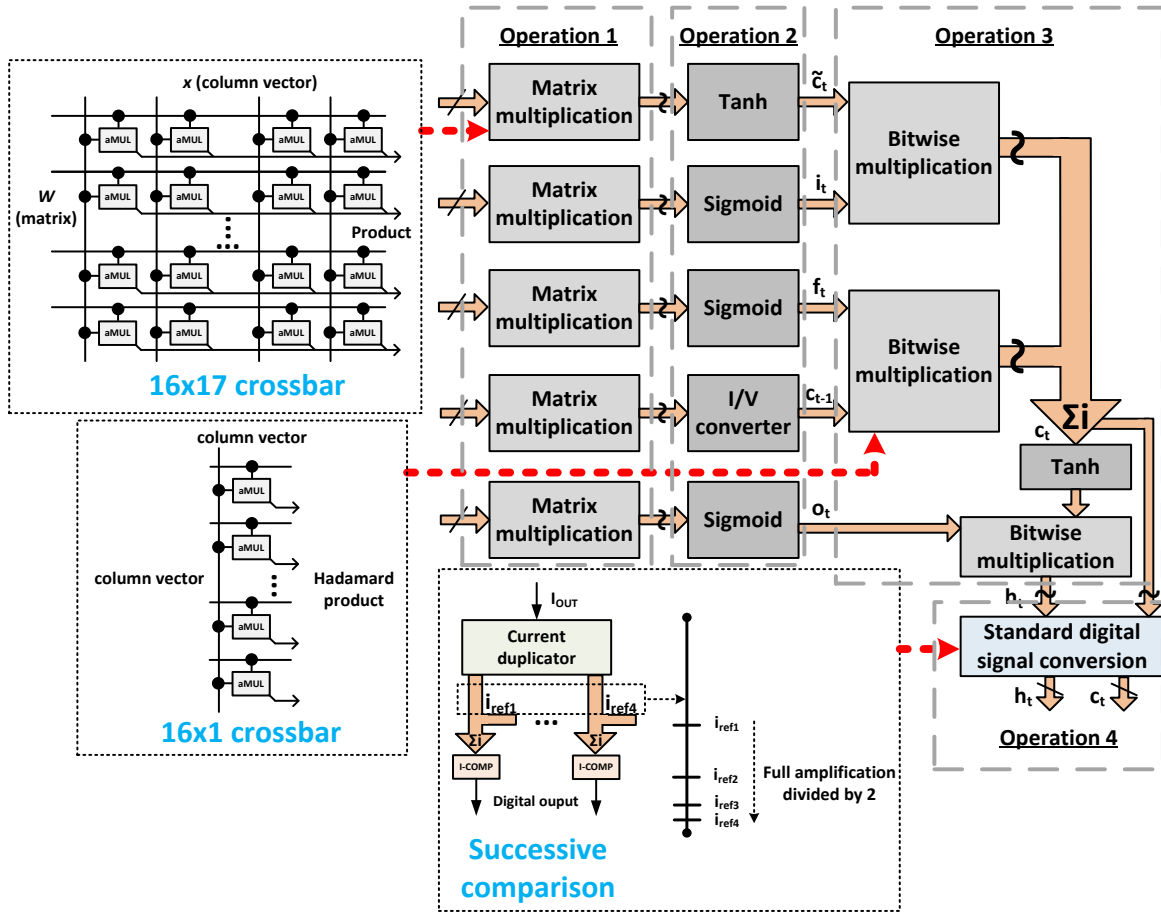


Figure 5.9 System level diagram of the proposed LSTM network.

we use current comparators at low circuit cost instead of traditional methods as the back-end to achieve ADC function.

5.4 Result and Discussion

This section will evaluate the performance of the proposed LSTM network based on the described analog circuit design. The simulations and discussions cover the proposed computing core of the analog LSTM network and the core with memory access.

5.4.1 Analog LSTM Core

The layout design of all required sub-blocks in TSMC 180nm CMOS are shown in Figures 5.10 (a)-(d). The entire design of the LSTM network with the matrix dimension of 16×16 is shown in Figure 5.10 (e). The dimension of the chip layout is $3.7\text{mm} \times 2.7\text{mm}$. The crossbar used for both DAC function and matrix multiplication as the first stage contributes 92% of total chip area. This is due to the proposed crossbar which uses multiple inputs at the cost of area of capacitors and the use of dummy devices in the layout design of differential pairs and current mirrors in multipliers, which is also seen in Figure 5.10 (a). The entire network is built by 16 independent computing paths to process analog signals of a 16×16 matrix in a parallel way. Thus, if some paths are set without input, the dimension of the matrix to be processed will be reduced, in which the only power due to dc bias will be dissipated.

We have used analog multipliers instead of conventional digital multipliers to process the specific matrix multiplication, $N \times N$ matrix times N -column vector, which is the dominant calculation in computing of LSTM network. It is necessary to compare two types of multipliers in term of performance. We illustrate the comparison of power delay product (PDP) of the matrix multiplication between the proposed analog multiplier and other digital multipliers including Booth encoding multiplier and Wallace tree multiplier [129, 130] as shown in Figure 5.11. We

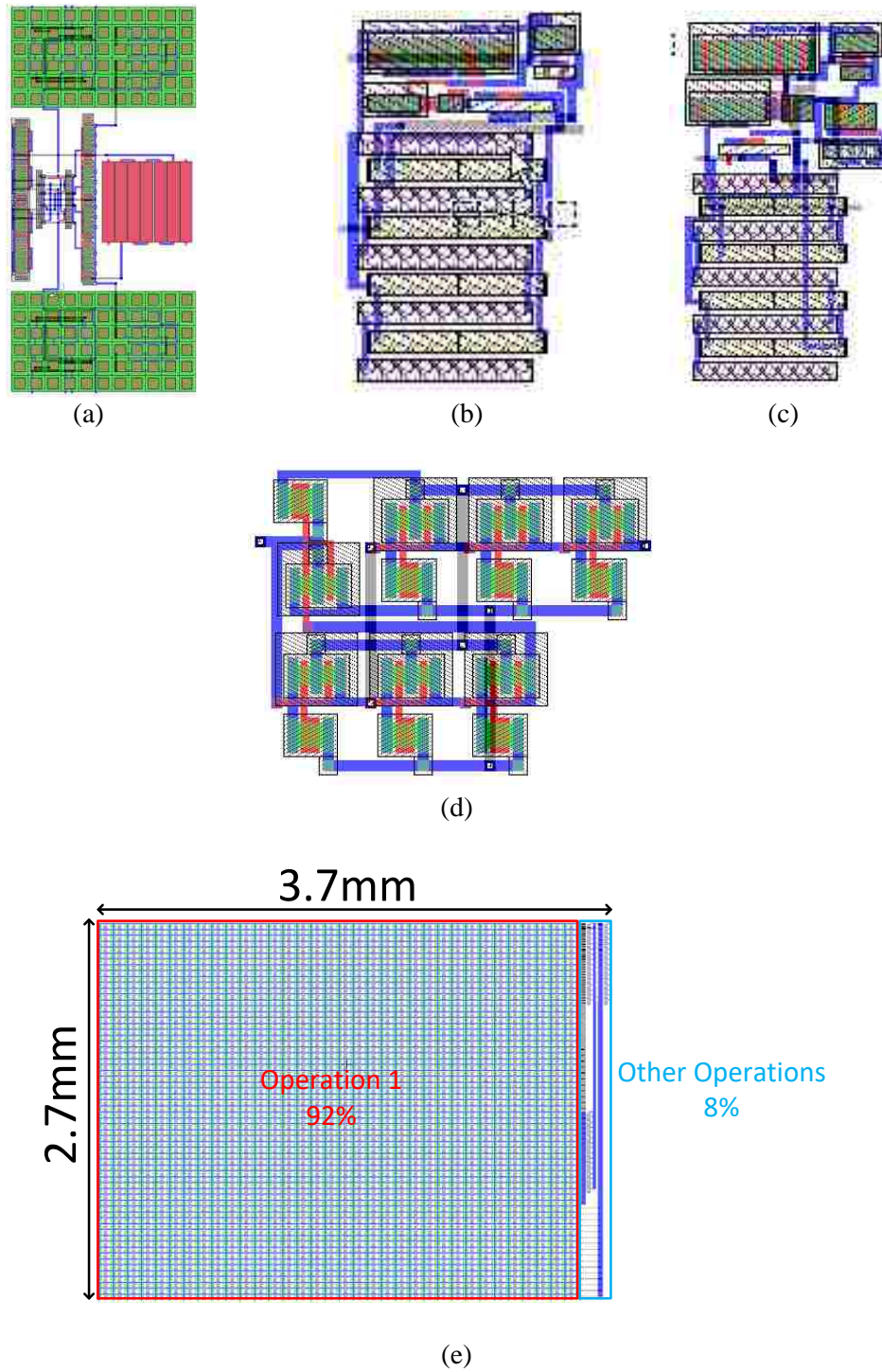


Figure 5.10 Layout view, (a) proposed analog multiplier, (b) sigmoid function, (c) hyperbolic tangent, (d) current comparator and (e) LSTM network.

observe that when the operation frequency is low, both Booth encoding and Wallace tree multipliers show advantageous performance compared to the proposed one. With the work frequency boosting, the PDP of the proposed multiplier will be less than that of two digital multipliers. When the dimension is increased, the PDPs of two digital multipliers are much more sensitive to frequency boosting than that of the proposed analog design. This is due to the switch current, which is a strong function of frequency in high speed digital circuits. For the proposed analog multiplier, it is always inputted by dc bias signal and the only term varied by frequency is the small signal as the input signal to be computed. In the case of 16×16 matrix multiplication under 800MHz, the growth rate of PDP of matrix multiplication using Booth encoding and Wallace tree multipliers compared to that using the proposed one is 12.3% and 37.6%, respectively. From 200MHz to 800MHz, in cases of both 8×8 and 16×16 matrixes, the growth rates of PDP of matrix multiplication using the proposed multiplier are all below 5%. In addition, the proposed design is capable of implementing DAC function as the front-end of the signal interface compared to previous work of analog multipliers. The proposed analog multiplier is suitable to work for large dimension matrix multiplication under high frequency. No matter what the bit length of input is, the analog multiplier only requires one stage to obtain the product after the implementation of DAC function using multiple inputs. For the addition in matrix multiplication, the current carry signal going through the metal interconnection is used to speed up the addition without additional circuit block.

The single computation cycle in analog LSTM network with no load is 0.87 to 1.19ns. The delay contributed by current comparators varies from 0.48 to 0.8ns depending on computed current amplitude before the back-end. The rest of blocks contribute 0.39ns delay. Our design is a purely computing core with a feedforward path and does not need an internal clock, on-chip memory or

internal sample and hold (S/H) circuit. Therefore, to avoid data competition, we restrict the maximum frequency of input signals to 800MHz. The total power dissipation of analog LSTM network with no load is 425.5mW. Based on defined four operations in Figure 5.9, Figure 5.12 shows the break-down graph of power dissipation for each operation. We see that the first operation to implement matrix multiplications consumes 86.3% of the total power dissipation, which is much larger than other operations. It is obvious since matrix multiplication is the main calculation in LSTM network. However, the proposed analog multiplier without feedback loop significantly reduces both power dissipation and latency considering the design of signal interface. The computing flow built by several important transient simulations including internal and final results are shown in Figure 5.13. We see that the internal signals can be shifted fast as the input flipping continuously due to absence of S/H circuit. The final results of c_t and h_t , in which both logic high and logic low can be recognized even though there are some unwanted signal components presented.

5.4.2 Memory Access

The proposed LSTM design is purely a computing core and can only deal with memory free computing of the 16×16 matrixes. It is necessary to apply memory access to achieve block multiplexing extending the network depth. To address memory access, how to efficiently utilize the memory in terms of hardware resource, computing latency and power dissipation is to be considered. For one weighted matrix in the proposed design, it has $(16 \times 17 \times 4) / 8 = 136$ bytes. Therefore, for a single computing cycle, the memory requires $136 \times 8 = 1088$ bytes for all weighted matrixes. Besides, for c_t and h_t , these are required to be stored temporarily in memory for the usage of input in the next computing cycle, and \hat{x}_t updated in each computing cycle also uses memory for data hosting. Therefore, each of these three vectors requires the memory of $16 \times 4 / 8 = 8$ bytes.

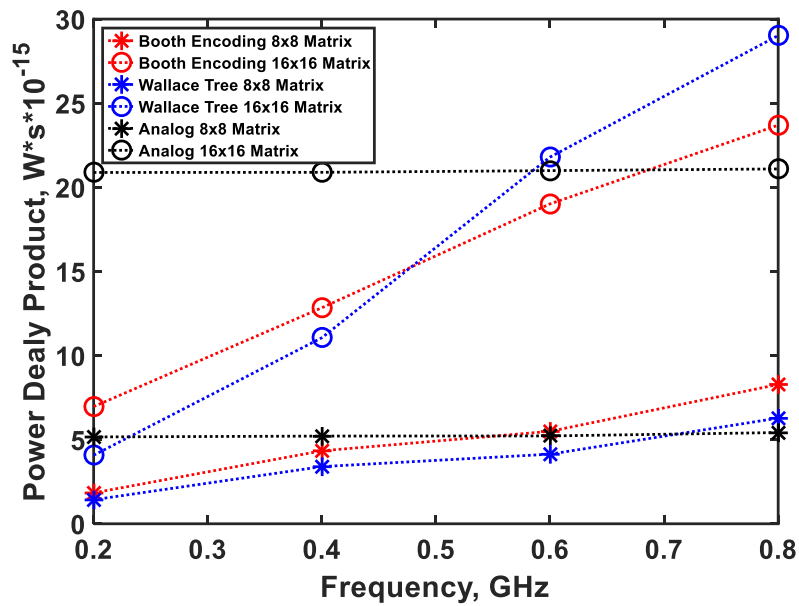


Figure 5.11 PDP comparison between the proposed analog multiplier and other digital multipliers.

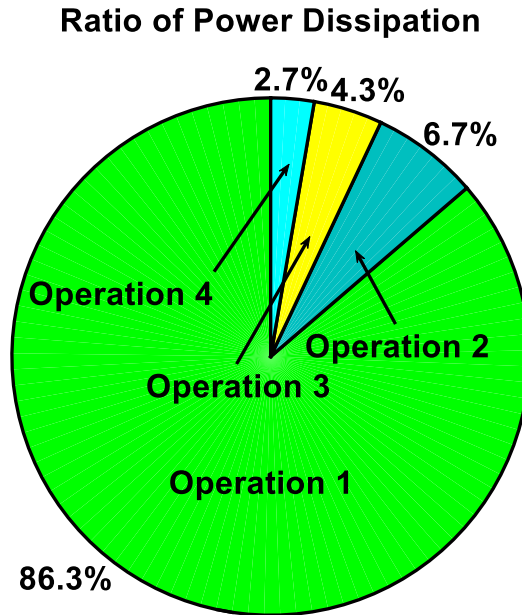


Figure 5.12 Break-down graph of power dissipation for each operation.

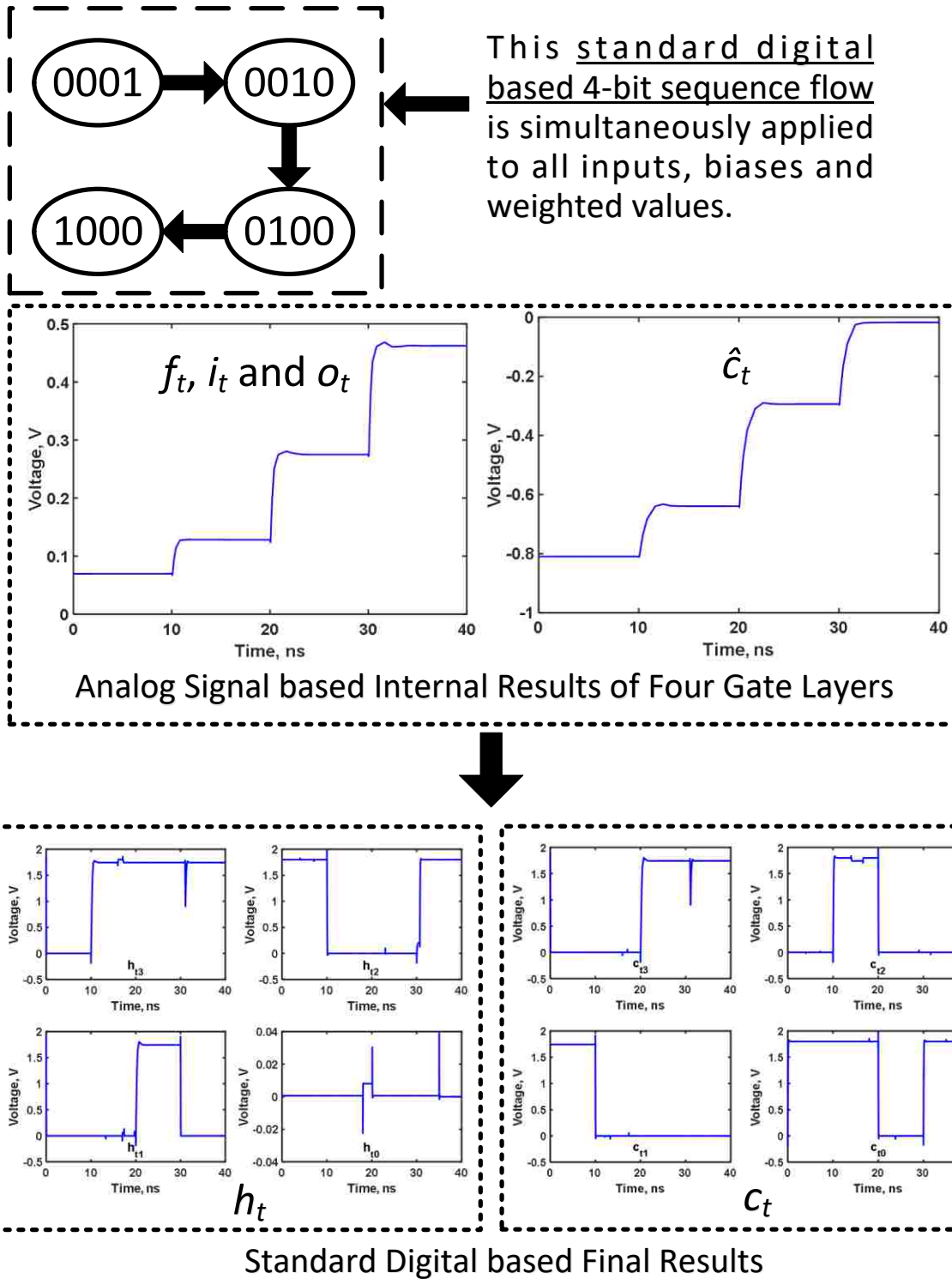


Figure 5.13 Transient simulation based computing flow of LSTM network using the proposed ASP design.

For the computing core of LSTM network in one computing cycle, the maximum memory size is $1088+8\times 3=1112$ bytes. For a trained LSTM network, if the pipelined topology is used for computing to extend the network depth, these weighted matrixes should be used to each computing cycle uniformly. \mathbf{h}_t obtained from each computing cycle not only is used as the input of the next computing cycle but it is also required to be stored for final prediction. If we define the network depth, which is corresponding to the number of computing cycle as D_{LSTM} , based on above description, the weighted matrixes after training occupy 1088 bytes constantly. The memory size of $\hat{\mathbf{x}}_t$, \mathbf{c}_t and \mathbf{h}_t is 8 bytes, 8 bytes and $(8+8D_{LSTM})$ bytes, respectively. Thus, the size of entire required memory is $(1112+8D_{LSTM})$ bytes. Since the utilized memory is small, the SRAM in chip can be used considering overall latency. Figure 5.14 shows the overview of the proposed analog LSTM network with on-chip SRAM. It is seen that the SRAM block is same as normal one to operate write-in and read-out. The communication cycle between computing and memory is described as follows:

- 1) All weighted matrixes, $\hat{\mathbf{x}}_t$, \mathbf{c}_t and \mathbf{h}_t are written into SRAM. \mathbf{h}_t is written doubly for both computing in the next layer of LSTM network and final prediction. The analog LSTM computing core is idle during the write step;
- 2) All data stored in SRAM in the last step is read out to analog LSTM core and the computing begins;
- 3) Above two clock cycles are seen as one computing cycle and repeated for obtaining deeper LSTM network.

Using above cycled flow, the network depth can be increased sequentially with SRAM access. Except \mathbf{h}_t being stored for final prediction, the rest of useful data is only stored temporarily in SRAM for the computing of the next cycle, which reduces the workload in SRAM. To control

the communication between analog LSTM core and SRAM, the design of memory controller should also be considered. The timing diagram of the controller is also shown in Figure 5.14, in which \overline{OE} , \overline{WE} , \overline{CE} , \overline{ME} and CLK are enable ports of read-out, write-in, analog LSTM core and SRAM, and clock, respectively. In these enable ports, only \overline{CE} is not the standard one. Figure 5.14 also shows the proposed circuit design of SRAM controller, in which \overline{CE} is outputted by CLK . \overline{OE} and \overline{WE} are generated by CLK and \overline{ME} . Once \overline{CE} is obtained, it can control the dc bias of the inputs of the proposed analog multipliers, which works for the analog LSTM network. During write-in, the analog LSTM core is idle and dc bias is disconnected through transmission gates controlled by \overline{CE} to save energy. While SRAM is in read-out, dc bias is applied to analog LSTM network via \overline{CE} to let the computing work.

Using the same structure of SRAM as shown in Figure 5.14, in which 6T-cell is used for the SRAM cell, and the size of SRAM array is $128 \times 128 = 2$ kilobytes, as the test configuration of memory access. Figure 5.15 shows the simulation results of the analog LSTM network with SRAM access depending on various network depths and frequencies. From Figure 5.15 (a), we compare the power dissipation when D_{LSTM} is 1. The overhead of power dissipation due to SRAM access is -6.5%, -1.7% and 8.3% corresponding to clock frequency of 100MHz, 150MHz and 200MHz, respectively. The reason that the overhead under 100MHz and 150MHz is negative during write-in is that the computing core is in idle state executed by the controller in SRAM and so only consumes static power. Therefore, from the view of average power dissipation given a full cycle of LSTM computing, the use of the analog LSTM network with SRAM can save power compared to analog LSTM network which works under non-clock without memory access. When the clock frequency increases to 200MHz, the SRAM access brings overhead of power dissipation to the

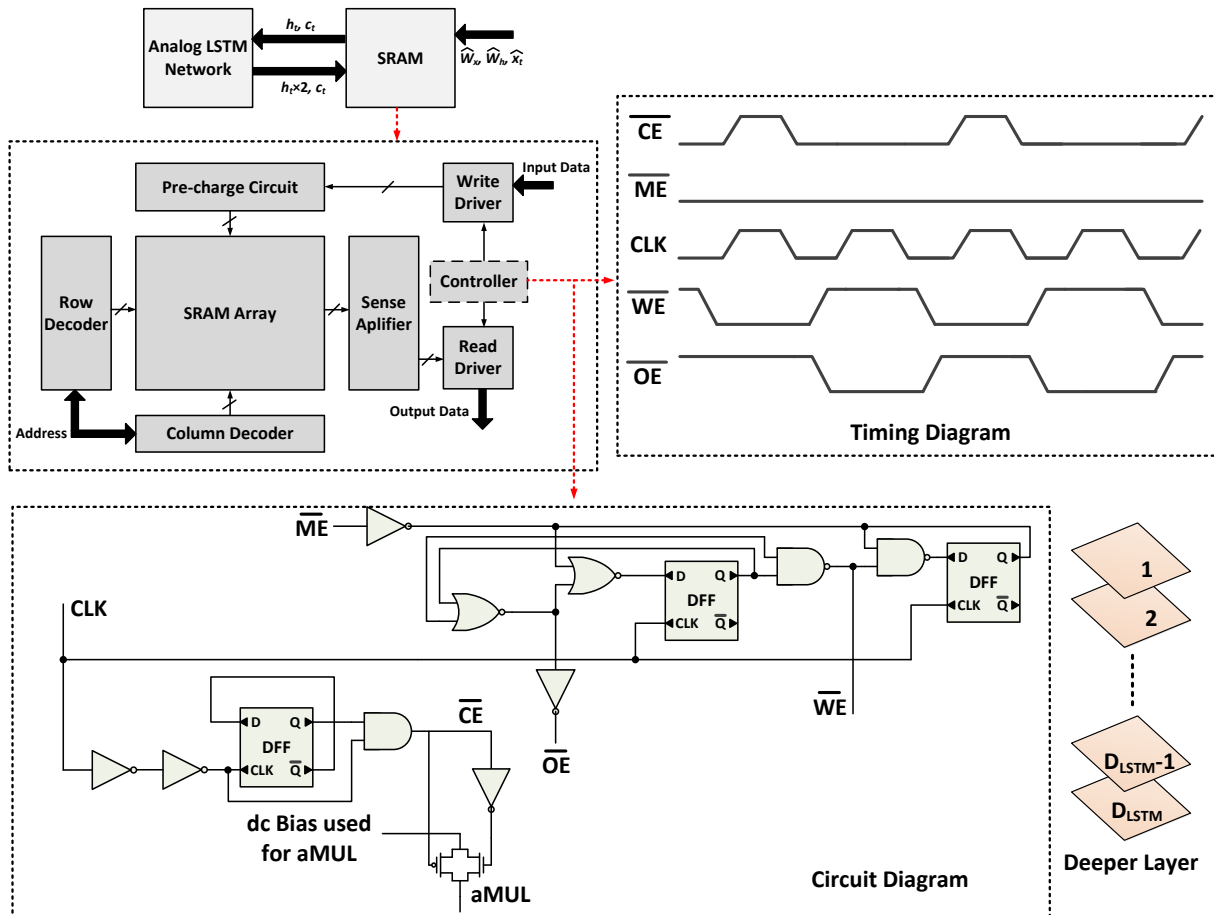


Figure 5.14 Overview of the proposed analog LSTM network with on-chip SRAM access used for extending the depth of network.

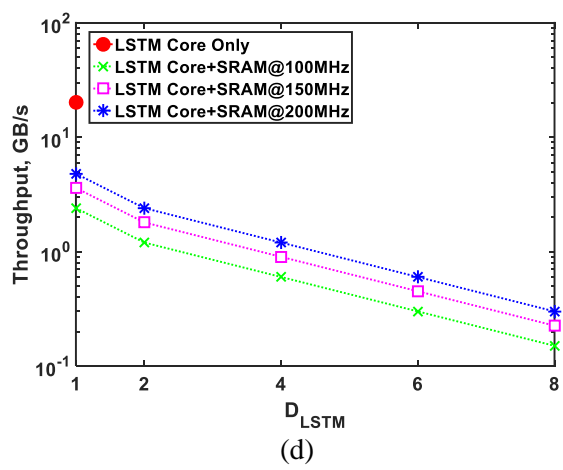
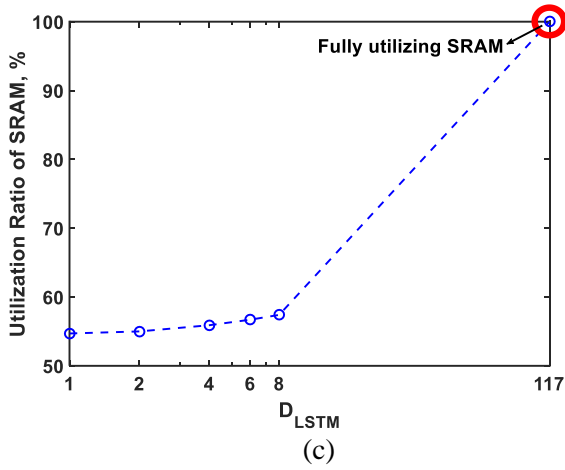
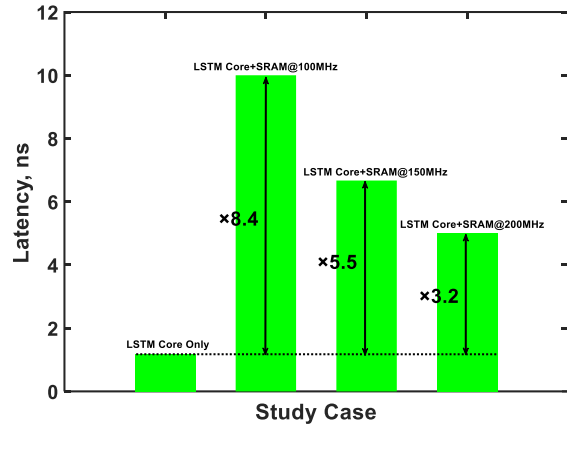
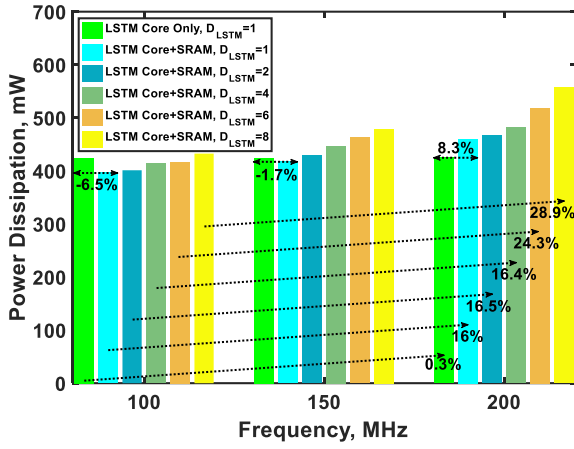


Figure 5.15 Simulation results of the LSTM network with SRAM access, (a) power dissipation, (b) computing latency, (c) utilization ratio of SRAM and (d) throughput.

proposed computing core due to the increase of dynamic power dissipation in SRAM. The overhead of power dissipation contributed by the proposed computing core is small. We can observe that from 100MHz to 200MHz, the increase of power dissipation in the proposed core is only 0.3% due to increase in frequency of small signal. In the cases that the proposed core works with SRAM access, the growth rates of power dissipation due to frequency boosting are all increased. In the case that D_{LSTM} is 8, this growth rate is up to 28.9%. Figure 5.15 (b) shows the overhead of latency due to SRAM access, in which the maximum overhead is $\times 8.4$ under 100MHz. Actually, the introduced SRAM restricts the computing speed of the proposed analog LSTM core due to clock constraint and the access time of SRAM. As reported earlier, the peak latency of the proposed analog LSTM core without memory access is only 1.19ns, which means the maximum frequency can be up to 800MHz. However, with the help of introduced SRAM, the LSTM core can be multiplexed to extend the depth of network. Since we have set the size of SRAM which is 2 kilobytes, the utilization ratio of SRAM should be studied. Figure 5.15 (c) shows the results of this metrics. The utilization ratio of SRAM is from 54.6% to 57.4% corresponding to D_{LSTM} is 1, 2, 4, 6, 8, respectively. Thus, the rest of unused SRAM cells suffers leakage current. Through the simulation of static power dissipation, this resource waste contributes 20mW to 43mW as the variations of network depth and frequency. As predicted, when D_{LSTM} is 117, SRAM will be fully used. Another important issue is throughput between the proposed LSTM core and SRAM. Figure 5.15 (d) shows this metrics. Here we only consider the computed data for the computing in the next cycle, h_t and c_t , and data stored in SRAM for the final prediction, h_t . When the proposed core works uniquely, throughput is up to 20.1GB/s. Accessed with SRAM, the throughput is reduced due to clock constraint. In addition, the depth of network constrains the throughput. In the fastest operation at 200MHz, when D_{LSTM} is 8, the throughput is 0.3GB/s. On the other hand, under

100MHz, when D_{LSTM} is 8, the throughput is 0.15GB/s. The proposed LSTM core with on-chip SRAM can extend the depth of network at the cost of lowering computing speed. For the power issue, the proposed work sequence and its controller can let the LSTM core be idle during write-in state to save power. The disadvantage of the proposed design is low utilization ratio of SRAM, which also contributes to static power dissipation. But the size of SRAM in this work is only 2 kilobytes and the utilization ratio of SRAM can be increased by increasing the depth of network.

Table 5.3 shows the performance comparison between our work and other reported work of LSTM. The listed data of our work is from the computing core accessed with SRAM under 200MHz and included D_{LSTM} is 1, 4 and 8. Considering the performance metrics of LSTM, quantization-bit, we observe that the LSTM implementation in FPGA platform is much more accurate than that in ASIC design since the single computation block is used repeatedly at the cost of computation latency in FPGAs. Work in [64] uses off-chip memory to confirm accuracy when the matrix dimension becomes large. However, this strategy largely increases computation latency and power dissipation. Another FPGA based work reported in [65] fully uses both logic units and memory cells in FPGA to speed up computation and suppress the power dissipation. Work in [131] balances the data communication that both on-chip LUT and off-chip DRAM are used for internal storage of matrix multiplication in order to reduce the latency due to off-chip memory access and workload of on-chip communication. Using digital signal to compute neural network, the longer bit length can cause larger power and latency. In the work of [132], the LSTM network with variable bit length is achieved to dynamically adjust both consumed energy and computing latency. Therefore, its peak energy efficiency is higher than other LSTM network designed in FPGA. Summarizing these work based on FPGA, their energy efficiencies are all lower at least 2 magnitudes than cited ASIC designs but the bit length in these designs are very high compared to

ASIC designs. For ASIC design, prior digital based work [66, 122] and our presented analog based work use 4-bit fixed-point number to balance computing error and circuit cost. Both work of [66, 122] can compute both CNN and LSTM, which is more functional than our design. [122] uses weight encoding and matrix sparsity to reduce both power and required computing cycles when large matrixes have to be processed. The error due to sparsity can be reduced by weight decoding and multi-index. Compared to [122], the boosting of energy efficiency of our design is $5.6\times$ corresponding to $D_{LSTM}=1$. As mentioned in introduction, the design in the work of [66] mainly uses LUTs and registers to store all possible results of multiplication and activation functions, which removes complex real-time computing and its switching power. Thus, the energy efficiency of this work is higher than in our work. However, the final results must be obtained through multiple cycles. In our design, analog implementation without traditional data converters or feedback loop largely speeds up computation, which gives on-chip SRAM enough time for data access. For computing of single-layer LSTM network ($D_{LSTM}=1$), it needs only one computing cycle in our work. A digital based LSTM implementation called ‘Chipmunk’ proposed in the work of [133] shows higher bit length and energy efficiency than [66] and our work. This is due to sparsity and matrix partitioning are widely used in this design to largely reduce the unnecessary energy consumed during logic switching at the cost of higher error and more epochs of training to increase accuracy or perplexity. If our ASP based design works for computing large matrixes, the strategy used in ‘Chipmunk’ is a potential one if slight error can be tolerated. Considering latency, the proposed work has the least computing latency than any other LSTM work. Once again, our ASP based work only uses one cycle to finalize one-layer computing which is different from other work.

Table 5.3 Performance comparison with prior work of LSTM

	[64]	[65]	[131]	[132]	[133]	[122]	[66]	Present work		
Platform	FPGA	FPGA	FPGA	FPGA	ASIC 65nm	ASIC 45nm	ASIC 65nm	ASIC 180nm		
Function	LSTM	LSTM	LSTM	LSTM	LSTM	CNN; LSTM	CNN; LSTM	LSTM		
Memory type	DDR3	SRAM	LUT; DDR3	LUT	SRAM	SRAM	LUT; REG	SRAM		
Clock frequency (MHz)	150	158.2	142	166	20	800	200	200		
Latency (ms)	390	4.6×10^{-4}	-	2.37×10^3	-	7.5×10^{-3}	-	10^{-5}		
Bit length	32-bit float	18-bit fixed	16-bit float	5~16-bit float	8~16-bit fixed	4-bit fixed	4-bit fixed	4-bit fixed		
Depth of network	-	-	2	-	1	1	1	1	4	8
Peak power dissipation (mW)	19.63×10^3	420	2.3×10^3	10^4	1.24	590	21	460.3	481.6	557.4
Energy efficiency (TOP/s/W)*	0.37×10^{-3}	-	1.46×10^{-4}	1.5×10^{-2}	3.08	0.17	1.1	0.95	0.9	0.78

*One multiplication is counted as two operations in neural computing. Due to different design platforms and implemented functions, the energy efficiency is the general metric to compare the performance of the proposed work and other reported work in the area of neural computing. This explanation is also used in Table 5.4.

Table 5.4 Performance comparison with prior work of ASP based on neural network

	[73]	[134]	Present work		
Platform	ASIC 65nm	ASIC 55nm	ASIC 180nm		
Function	CNN	Autonomous control	LSTM		
Memory type	SRAM	-	SRAM		
Data conversion type	ADC	DPC	ADC; DAC		
Clock frequency (MHz)	100	20	200		
Latency (ms)	263	-	10^{-5}		
Power supply (V)	0.46~0.8	0.4~1	1.8		
Bit length	16-bit fixed	6-bit	4-bit fixed		
Depth of network	5	-	1	4	8
Peak power dissipation (mW)	211	0.69	460.3	481.6	557.4
Energy efficiency (TOP/s/W)	0.47	3.12	0.95	0.9	0.78

In Table 5.4, we also compare the proposed design with other work of ASP based on neural network [73, 134]. In the work of [73], ADC array is designed in another chip considering limited power budget in a single chip. In the computing chip, it uses multi-core to increase the computing efficiency. Since off-chip communication is introduced increasing the communication delay, the allowable clock frequency cannot be very high to avoid data race in motherboard. Based on above, the latency of this work is much higher than that of our work. Compared to the work of [73], the boosting of energy efficiency of our design is $2\times$ corresponding to $D_{LSTM}=1$. Design in the work of [134] is suitable for applications in portable devices and its energy efficiency is higher than in our work. In this work, digital-to-pulse converter (DPC) is used as signal interface. Thus, counter array has to be embedded for signal processing. Based on this principle, the final results cannot be obtained in a single computing cycle. Thus, the latency of this work cannot be lower than that of our design. Compared to these work, the novelty of our work is that we have proposed a novel signal interface instead of conventional ADC and DAC. For DAC, the only power consumed in the physical connection between transistor's gate and MIFGs is due to resistances. For the back-end used as ADC, we use the array of current comparators to speed up the conversion and save chip area. Besides, the computing latency of the proposed LSTM core inside is fast without the restriction of clock or feedback loop, and we have used current signal to execute the row-addition for matrix multiplications instead of analog mixer.

5.4.3 Sensitivity Study

Another important issue is that analog implementation is more sensitive to the variations of process, voltage and temperature (PVT) and noise than digital one. The system used for the simulations of these non-ideal factors is the same as the one in the study of memory access. The work frequency is set at 200MHz. It is generally known that the real chip fabrication always has

device mismatch. Thus, in the following sensitivity study, based on Monte Carlo method, all simulations have mismatch within $\pm 20\%$ distribution covering all transistors and passive devices. For the evaluation of PVT variation, the corner cases in this work include TTTT, SSHL, SFHL, FSHL, SSHH, SFHH, FSHH and FFLH. Four letters from left to right in these cases represent the variations of n-type MOSFET, p-type MOSFET, temperature and voltage supply, respectively. S and F located in the first two letters are slow and fast MOSFETs, respectively. T located in the first two letter represents typical MOSFETs following the original SPICE model provided by the foundry. We use 3σ method to calculate the variable threshold voltage to define S and F for MOSFETs [135, 136, 137]. In this method, the standard deviation of threshold voltage can be represented by $\sigma_{th}=(qt_{ox}/\epsilon_{ox})\times(N_A W_d/4WL)^{1/2}$. H appeared in the last two letters is high temperature, 340K, or oversupply voltage, 1.85V. Accordingly, L is low temperature, 270K, or supply voltage with loss, 1.75V. T located in the last two letter represents default temperature in SPICE simulation, 300K, or standard supply power, 1.8V. We have introduced the fabrication mismatch and the corners representing variations of PVT. The last concern is noise introduced in our design. For CMOS technology, it includes two types of noise. The first one is device noise. For the analysis of a transistor or a resistor, it can be modeled using a paralleled current source. For the analysis of a capacitor, it can be modeled using a cascade voltage source. The detailed expressions of these noise sources have been deeply analyzed in [120]. The other one is the input noise which is imported from other blocks communicating with the proposed computing core. This one can be assumed to approximately follow Gaussian distribution which is represented by $V_{in_noise}=V_{in_ideal}\times(1+N(0,\sigma^2))$ according to the work of [138]. We set σ^2 as 0.15^2V and 0.3^2V for the following simulations. Based on above configuration, we use error rate to evaluate the performance influenced by the described corner cases and noises. Figure 5.16 shows the simulation

results. SSHL, FSHL, SSHH and FSHH are four corners of which the error rate is higher than other corners. This observation follows that SS and FS for n-type and p-type transistors are the worst corners for digital VLSI. The performance of digital based memory access in our work can be seriously influenced by these two corners. Thus, incorrect data communication can be occurred due to potential setup and hold violations. TTTT can be seen as an ideal case in our study and so its error rate is the lowest. Considering digital blocks, FFLH is the fastest corner in our work. However, it shows higher error rate than TTTT, which is due to the signal asynchronization between analog network without obvious acceleration and the faster memory access. Considering the variation of supplied voltage, the error rate under oversupply voltage is less than that under supply voltage with loss. Thus, it is necessary to avoid the loss of power supplying to the proposed design. For $D_{LSTM}=1$, the highest error rate is 3.4% corresponding to FSHL with the variance of input noise, $\sigma^2=0.3^2V$. For $D_{LSTM}=4$, the highest error rate is 10.3% also corresponding to FSHL with the variance of input noise, $\sigma^2=0.3^2V$.

5.4.4 Evaluation of Network Performance

We use word-level language prediction experiments to test the performance of LSTM network using our ASP architecture. For an input sequence, the task is to predict the next word given previous seen words. Our experiments are conducted on PTB dataset [139], which consists of 929k training words, 73k validation words and 82k test words. It has 10k words in its vocabulary. We stack our design to form a two-layer LSTM model, and train it for 30 epochs using stochastic gradient descent. The hidden units of each LSTM layer, as our proposed architecture, is 16. Learning rate starts at 1.0, keeps steady for 4 epochs, and then decays 0.5 after each epoch. The training results are shown in Figure 5.17 (a). From this figure, we can see that the perplexity keeps decreasing as epoch increases, and almost remains stable after 10th epoch. The best validation

perplexity is at the 25th epoch. Train, valid and test perplexities after being stable are 377.57, 408.94 and 383.87, respectively. The result shows that our LSTM architecture is capable of handling the tasks of real-world sequence prediction.

We also perform another predictive simulation experiment with different size of hidden units of LSTM layer. The model is the same with the last experiment but only changes the number of hidden units. As the Figure 5.17 (b) shows, the test perplexity becomes smaller as the size of hidden units grows. When the matrix size is 256, the perplexity can be lowered to 174.06. The result verifies that the bigger size of hidden units in LSTM is necessary for improving the performance of LSTM model. Considering the efficiency of training, we see that when the matrix size is 32, the proposed structure uses 24 epochs, which is smaller than any other case, to let the perplexity be stable. Mapping this prediction that larger matrixes are used for computing, we need to expand the proposed ASP network. For $N \times N$ matrixes used in the structure of our work, it requires N independent channels from front-end analog multipliers to back-end current comparators without additional power coupled by adjacent channels. Thus, the power overhead due to matrix expanding roughly shows a linearity to the number of channels. The computing latency will be mainly increased by current transportation in metals in the analog multiplier based crossbar, which is very short compared to signal going through digital blocks. Considering memory access, using our ASP structure connecting with on-chip SRAM can increase the overhead of power and latency. Summarizing this matrix expanding, the increase of power is much larger than that of computing latency.

5.5 Conclusion

We have proposed a strategy to implement LSTM network in analog circuit design. The entire analog based computing core is compatible to external digital blocks. To speed up

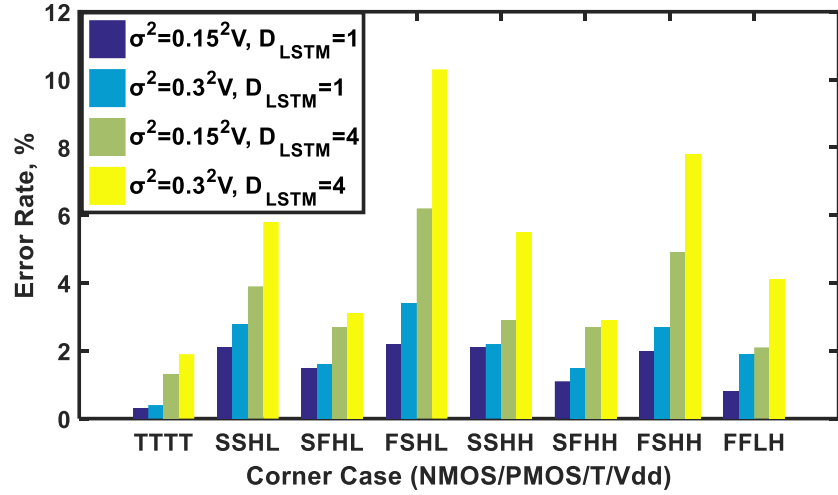
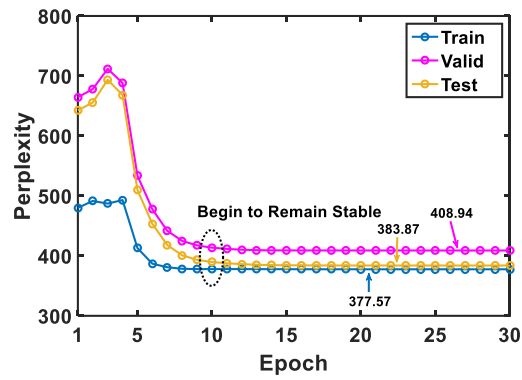
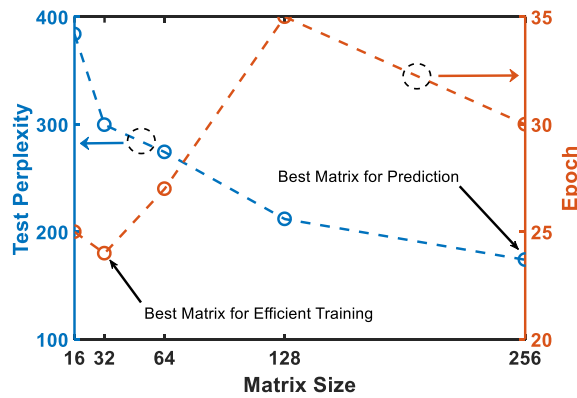


Figure 5.16 Simulation results of error rate with PVT variation and introduced noise. All simulations are imported with mismatch of $\pm 20\%$ distribution of transistors, capacitors used for MIFGs and resistors.



(a)



(b)

Figure 5.17 Simulation results using the proposed LSTM structure with PTB dataset, (a) training result and (b) predictive results of perplexity and required epochs for stabilizing perplexity with matrix expanding.

computing and reduce power dissipation, we have introduced a novel signal interface including MIFG MOSFETs and current comparators to achieve data conversion. In computing core inside, we have used current carry signal to reduce latency and remove analog mixer used for row-addition. The final post-layout simulation results show that the proposed analog LSTM network is suitable to work with on-chip standard memory. For the utilization of 2 kilobytes SRAM, LSTM network can be computed to 117 layers. The finalization of one-layer computing only requires one computing cycle (two clock cycles). The overhead of power dissipation and computing latency due to memory access at 200MHz are 8.3% and 3.2 \times , respectively. The energy efficiency can be up to 0.95TOP/s/W. If more advanced CMOS technology is used for our design, it is predicted that both clock frequency and energy efficiency can be obviously increased. The sensitivity simulations considering noise, device mismatch and PVT variation show that the worst case will lead to 10.3% error rate in the proposed design corresponding to 4-layer LSTM computing. Overall, the proposed design shows an acceptable performance to these non-ideal factors. Using the same network structure of the proposed LSTM design with PTB as dataset for performance evaluation, the training requires 10 epochs to let all perplexities be stable. Besides, using the proposed LSTM structure with matrix expanding, lower perplexity under the simulation using PTB dataset can be obtained without large overhead of latency.

CHAPTER 6

SUMMARY AND SCOPE OF FUTURE WORK

In this dissertation, several circuit designs used for optimizing on-chip computing are proposed. The summary and scope of future work are introduced as follows.

6.1 On-Chip Power Regulating Using Switchable Pins

We have verified the feasibility of the switchable pin used for delivering power to chips through post-layout simulations and chip testing. Final results show that using switchable pins, the power in chip can be doubled without long response time or large voltage loss in pads. We also found that switchable pins won't seriously degrade the performance of data transmission, especially under medium frequency. The most significant thing is that due to voltage compensated by the proposed switchable pin, the unwanted transmission delay of digital logic gates is reduced. Thus, besides art of place and route, and on-chip power regulator, the switchable pin is another option to suppress the IR droop in VLSI chips with complex interconnection. Especially compared to mainstream analog based on-chip power regulators, the proposed method is fully implemented by digital blocks, which is a huge improvement considering power and area efficiency. For the issue of noise, unlike analog based power regulators, which also contributes noise, the proposed strategy only contributes the noise from CMOS TG array, which is smaller than the one from analog circuits.

The work is of first kind and so it has been limited to considerations such as the PCB design based on C4 package and the core containing ~100K transistors. It is interesting that applying switchable pin to the SoC using ball grid array (BGA) package. In this platform, voltage can be easily distributed to the whole die through balls compared to C4 package. However, the technical difficulty lays on how to avoid large power density in case that a specific area has suffered a large IR droop. In this case, driven by a perfect voltage from switchable pins, a large current will go

from balls to die inside. Since the standard heat sink only works for cooling upon ICs, the locally heat transited by the balls under ICs is difficult to be dispersed. The worst case is that since BGA package is not mechanically compliant, high temperature will cause solder fracture. Thus, if switchable pins are used with BGA package, a threshold value of voltage compensation should be set through thermal modelling before going to VLSI design, which is deserved to be studied.

6.2 Logarithmic Conversion with Calibration

The second contribution of this dissertation is that a novel calibration based on mathematical analysis working for logarithmic conversion is developed. Unlike previous work using several linear functions for iteration and directly searching results from LUT, the proposed algorithm is to use a single linear curve to approximate the real conversion curve. In addition, a case judgement is used to further minimize the conversion error. Compared to Mitchell's algorithm, the reduced rate of error using our work is 71.1%. Mapping this calibration to circuit design, it only requires two logic stages no matter what the bit length is. The entire structure is paralleled to speed up conversion. To reduce the power dissipated in conversion, we have introduced a dual-mode to the logarithmic converter. Under dc supply, the system normally operates to output pulse signals. Under clocked power, the system operates adiabatic logic to reduce dynamic power dissipation. Both post-layout simulations and chip testing functionally verify the proposed work.

This work has illustrated a low-error conversion from binary number to logarithmic number. If the computing core is integrated in a standard digital chip processing binary numbers, the reverse conversion from logarithmic number to binary number is required. Obviously, this reverse conversion also generates error. So the first desired improvement based on this work is how to design a low-error reverse conversion. Another thing we have to address is that under adiabatic logic, the results are all clocked signals which are not compatible to standard digital

environment. A straightforward solution is to apply digital buffer chains to convert clocked signals to pulse signals at the cost of large short-circuit current. Thus, the second work needs to be studied is to find an energy-efficient method for signal buffering under adiabatic operation.

6.3 Optimization and Calibration of ALU

Motivated by the fact that ALU is the soul of computing in both CPU and GPU, we have used MIFG MOSFETs to build the neuron-like cells to achieve cell-sharing. Another challenge is how to implement digital calibration used for ALU since under high frequency operation, setup violation, hold violation, clock inaccuracy, CMOS TGs used for MUX, other errors out of ALU block and switching state in DVFS can cause incorrect results. Current techniques for calibrating computing error cannot monitor all signal nodes but are limited to set checkpoints in critical paths considering power budget. In these checkpoints, clock recovery and delay based detection are two common strategies. In this work, we give up the calibration only in checkpoints and propose a LUT-like method to calibrate all operations with all possible inputs. We used metal layers connecting power ring or ground ring in pad frame but not real LUT array to obtain the correct result, which is decoded by floating gate based TGs. This metal interconnection based method consumes less power than using LUT. Generally speaking, each path in the block of error detection represents one group of input signals and its corresponding result for a given function except sequence comparison. For sequence comparison, we use static logic to recalculate the operation since using former metal interconnection based method will largely increase metal routings. For 32-bit operation, there are 2^{32} routings required for covering all possibilities. But the method we used based on static logic processing also introduces large power to calibration. Post-layout simulations show that the overhead of power dissipation and computing latency due to the proposed error detection and calibration are ~24% and ~100ps, respectively. Thus, exploring an

energy-efficient strategy for calibration should be further studied. Another observation is that since coupling capacitors are applied to logic gates, the ALU designed by MIFG MOSFETs can tolerate larger fan-out than that by static logic. In mismatch simulations, the error rate of the proposed ALU under 20% mismatch is less than 8%. Further theoretical study on how to properly ratio the sizes of transistors to make the transfer characteristic not to be sensitive to variation of surface potential due to mismatch is valuable.

6.4 Neural Accelerator based on ASP

The last work in this dissertation develops an ASP based accelerator exclusively computing LSTM network. We have introduced a novel signal interface to make this analog core be compatible with digital chips, which consumes less power than conventional data converters. For internal processing, we used current carry signal to accelerate the addition in matrix multiplication instead of conventional voltage signal. The latency of the proposed design without memory access is only 1.19ns. Using on-chip SRAM under 200MHz as memory access, the maximum energy efficiency can reach to 0.95TOP/s/W.

The bit length of this work is 4-bit since we have used ratioed MIFGs, which occupy large chip area, applied on analog multipliers for DAC function. With the support of memory access, the proposed design can periodically extend network depth, which can increase computing accuracy from the perspective of RNN algorithm. For the design strategy of the proposed analog accelerator, it is difficult to increase bit length since capacitor mismatch will seriously influence the DAC accuracy and the accuracy of multiple reference currents will be reduced due to PVT variation and so decrease the ADC accuracy. Thus, the design strategy in the proposed accelerator mainly serves for the LSTM network with low resolution and deep layer.

Besides resolution and network depth, the dimension of matrix is another metrics to evaluate the performance of LSTM network since a larger matrix can capture more information in one cycle which can improve the accuracy of real-time prediction and reduce the dependency of past information. This work used analog multiplier based crossbar with metal connection to deliver current signal. The signals obtained by matrix multiplication are going through activation functions. Thus, this design cannot extend the dimension of matrix due to non-linear process. To solve this restriction, it is suggested that on-chip memory is also introduced after matrix multiplication to extend the dimension of matrix for further work. Besides, each partitioning result is current signal which cannot be directly stored in SRAM, it is necessary to introduce additional interface for I/V conversion to ensure correct read and write between the accelerator and on-chip memory. Here, another challenge of future work is introduced that how to design a highly linear I/V converter with a large signal swing which can be written into SRAM. The last issue should be concerned is the design of analog blocks which are not sensitive to power variation since the PVT simulations show that the proposed work has the least tolerance of voltage variation.

BIBLIOGRAPHY

- [1] W. M. Holt, "Moore's law: A path going forward," *Proc. 2016 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA, pp. 8-13, February 2016.
- [2] V. De and S. Borkar, "Technology and design challenges for low power and high performance [microprocessors]," *Proc. 1999 International Symposium on Low Power Electronics and Design (ISLPED)*, San Diego, CA, USA, pp. 163-168, August 1999.
- [3] M. Ouchi et al., "Thermal management systems for data centers with liquid cooling technique of CPU," *Proc. 13th InterSociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*, San Diego, CA, USA, pp. 790-798, May 2012.
- [4] J. Donald and M. Martonosi, "Techniques for multicore thermal management: Classification and new exploration," *Proc. 33rd International Symposium on Computer Architecture (ISCA)*, Boston, MA, USA, pp. 78-88, June 2006.
- [5] L. Yang, W. Liu, W. Jiang, M. Li, P. Chen and E. H. M. Sha, "FoToNoC: A folded torus-like network-on-chip based many-core systems-on-chip in the dark silicon era," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 7, pp. 1905-1918, 2017.
- [6] H. Bokhari, H. Javaid, M. Shafique, J. Henkel and S. Parameswaran, "DarkNoC: Designing energy-efficient network-on-chip with multi-Vt cells for dark silicon," *Proc. 2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, San Francisco, CA, USA, pp. 1-6, June 2014.
- [7] C. Rhebergen, B. Bilgin, A. Emadi, E. Rowan and J. Lo, "Enhancement of electric motor thermal management through axial cooling methods: A materials approach," *Proc. 2015 IEEE Energy Conversion Congress and Exposition (ECCE)*, Montreal, QC, Canada, pp. 5682-5688, September 2015.
- [8] T. Caroff, R. Mitova, B. Wunderle and J. Simon, "Transient cooling of power electronic devices using thermoelectric coolers coupled with phase change materials," *Proc. 19th International Workshop on Thermal Investigations of ICs and Systems (THERMINIC)*, Berlin, Germany, pp. 262-267, September, 2013.
- [9] X. Chen et al., "Dynamic voltage and frequency scaling for shared resources in multicore processor designs," *Proc. 2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC)*, Austin, TX, USA, pp. 1-7, June 2013.
- [10] N. A. Sherwani, *Algorithms for VLSI Physical Design Automation*, Springer Science & Business Media, Berlin, Germany, 2012.
- [11] P. Hazucha et al., "Area-efficient linear regulator with ultra-fast load regulation," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 4, pp. 933-940, 2005.
- [12] L. Benini and G. D. Micheli, *Dynamic Power Management: Design Techniques and CAD Tools*, Springer Science & Business Media, Norwell, MA, USA, 1997.

- [13] C. Zhang, D. Ma and A. Srivastava, "Integrated adaptive DC/DC conversion with adaptive pulse-train technique for low-ripple fast-response regulation," *Proc. 2004 International Symposium on Low Power Electronics and Design (ISLPED)*, Newport Beach, CA, USA, pp. 257-262, August 2004.
- [14] F. Sluijs, K. Hart, W. Groeneveld and S. Haag, "Integrated DC/DC converter with digital controller," *Proc. 1998 International Symposium on Low Power Electronics and Design (ISLPED)*, Monterey, CA, USA, pp. 88-90, May 1998.
- [15] X. Wang et al., "An analytical study of power delivery systems for many-core processors using on-chip and off-chip voltage regulators," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 9, pp. 1401-1414, 2015.
- [16] C. Mead and L. Conway, *Introduction to VLSI Systems*, Addison-Wesley, Boston, MA, USA, 1980.
- [17] W. Wolf, *Modern VLSI Design*, Prentice Hall, Boston, MA, USA, 2008.
- [18] F. Lu and H. Samueli, "A high-speed CMOS full-adder cell using a new circuit design technique-adaptively-biased pseudo-NMOS logic," *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, New Orleans, LA, USA, pp. 562-565, May 1990.
- [19] G. K. Yeap, *Practical Low Power Digital VLSI Design*, Springer Science & Business Media, New York, NY, USA, 2012.
- [20] Y. Moon and D. k. Jeong, "Efficient charge recovery logic," *Proc. 1995 IEEE Symposium on VLSI Circuits, Kyoto, Japan*, pp. 129-130, June 1995.
- [21] A. Vetuli, S. D. Pascoli and L. M. Reyneri, "Positive feedback in adiabatic logic," *Electronics Letters*, vol. 32, no. 20, pp. 1867-1869, 1996.
- [22] V. G. Oklobdzija and R. K. Montoye, "Design-performance trade-offs in CMOS-domino logic," *IEEE Journal of Solid-State Circuits*, vol. 21, no. 2, pp. 304-306, 1986.
- [23] N. Lindert, T. Sugii, S. Tang and Chenming Hu, "Dynamic threshold pass-transistor logic for improved delay at lower power supply voltages," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 1, pp. 85-89, 1999.
- [24] J. A. Pretorius, A. S. Shubat and C. A. T. Salama, "Latched domino CMOS logic," *IEEE Journal of Solid-State Circuits*, vol. 21, no. 4, pp. 514-522, 1986.
- [25] Y. Xu and A. Srivastava, "New energy recovery CMOS XNOR/XOR gates," *Proc. 2007 50th Midwest Symposium on Circuits and Systems (MWSCAS)*, Montreal, Canada, pp. 948-951, August 2007.
- [26] I. E. Sutherland, R. F. Sproull and D. F. Harris, *Logical Effort: Designing Fast CMOS Circuits*, Morgan Kaufmann, San Francisco, CA, USA, 1999.
- [27] Y. J. Chen et al., "A 130.3 mW 16-core mobile GPU with power-aware pixel approximation techniques," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 9, pp. 2212-2223, 2015.

- [28] B. Chrétien, A. Escande and A. Kheddar, "GPU robot motion planning using semi-infinite nonlinear programming," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 10, pp. 2926-2939, 2016.
- [29] J. N. Mitchell, "Computer multiplication and division using binary logarithms," *IRE Transactions on Electronic Computers*, vol. 11, no. 4, pp. 512-517, 1962.
- [30] V. Mahalingam and N. Ranganathan, "Improving accuracy in Mitchell's logarithmic multiplication using operand decomposition," *IEEE Transactions on Computers*, vol. 55, no. 12, pp. 1523-1535, 2006.
- [31] M. Combet, H. Van Zonneveld and L. Verbeek, "Computation of the base two logarithm of binary numbers," *IEEE Transactions on Electronic Computers*, vol. 14, no. 6, pp. 863-867, 1965.
- [32] T. B. Juang, S. H. Chen and H. J. Cheng, "A lower error and ROM-free logarithmic converter for digital signal processing applications," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 56, no. 12, pp. 931-935, 2009.
- [33] T. B. Juang, P. K. Meher and K. S. Jan, "High-performance logarithmic converters using novel two-region bit-level manipulation schemes," *Proc. 2011 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, Hsinchu, Taiwan, pp. 1-4, April 2011.
- [34] D. K. Kostopoulos, "An algorithm for the computation of binary logarithms," *IEEE Transactions on Computers*, vol. 40, no. 11, pp. 1267-1270, 1991.
- [35] S. L. SanGregory, C. Brothers, D. Gallagher and R. Siferd, "A fast, low-power logarithm approximation with CMOS VLSI implementation," *Proc. 42nd Midwest Symposium on Circuits and Systems (MWSCAS)*, Las Cruces, NM, USA, pp. 388-391, August 1999.
- [36] S. C. Huang, L. G. Chen and T. H. Chen, "The chip design of a 32-b logarithmic number system," *Proc. 1994 IEEE International Symposium on Circuits and Systems (ISCAS)*, London, UK, pp. 167-170, June 1994.
- [37] A. F. Florea, L. Li, M. A. Thornton and D. W. Matula, "A discrete logarithm number system for integer arithmetic modulo 2^k : algorithms and lookup structures," *IEEE Transactions on Computers*, vol. 58, no. 2, pp. 163-174, 2009.
- [38] Y. Wan and C. L. Wey, "Efficient algorithms for binary logarithmic conversion and addition" *Proc. 1998 IEEE International Symposium on Circuit and Systems (ISCAS)*, Monterey, CA, USA, pp. 233-236, June 1998.
- [39] F. S. Lai, "The architecture and analysis of a hybrid number system processor," *Proc. 1992 IEEE International Symposium on Circuits and Systems (ISCAS)*, San Diego, CA, USA, pp. 803-806, May 1992.
- [40] H. Y. Lo and Y. Aoki, "Generation of a precise binary logarithm with difference grouping programmable logic array," *IEEE Transactions on Computers*, vol. 34, no. 8, pp. 681-691, 1985.
- [41] K. H. Abed and R. E. Siferd, "VLSI implementations of low-power leading-one detector circuits," *Proc. of 2006 IEEE SoutheastCon*, Memphis, TN, USA, pp. 279-284, April 2006.

- [42] M. Gok, M. J. Schulte and M. G. Arnold, "Integer multipliers with overflow detection," *IEEE Transactions on Computers*, vol. 55, no. 8, pp. 1062-1066, 2006.
- [43] K. H. Abed and R. E. Siferd, "CMOS VLSI implementation of 16-bit logarithm and anti-logarithm converters," *Proc. 43rd IEEE Midwest Symposium on Circuits and Systems (MWSCAS)*, Lansing, MI, USA, pp. 776-779, August 2000.
- [44] What is Arithmetic Logic Unit, Available: <http://ecomputernotes.com/fundamental/introduction-to-computer/what-is-arithmetic-and-logic-unit-alu>
- [45] M. Dolle et al., "A 32-b RISC/DSP microprocessor with reduced complexity," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 7, pp. 1056-1066, 1997.
- [46] NVIDIA's Fermi: The First Complete GPU Computing Architecture, Available: [http://www.nvidia.com/content/PDF/fermi_white_papers/P.Glaskowsky NVIDIA's Fer mi-The_First_Complete_GPU_Architecture.pdf](http://www.nvidia.com/content/PDF/fermi_white_papers/P.Glaskowsky_NVIDIA's_Fer mi-The_First_Complete_GPU_Architecture.pdf)
- [47] The Shader Core – Caches, ALUs and All Things Compute, Available: <https://www.beyond3d.com/content/reviews/55/7>
- [48] S. K. Mathew, R. K. Krishnamurthy, M. A. Anders, R. Rios, K. R. Mistry and K. Soumyanath, "Sub-500-ps 64-b ALUs in 0.18- μ m SOI/bulk CMOS: design and scaling trends," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 11, pp. 1636-1646, 2001.
- [49] Y. Shimazaki, R. Zlatanovici and B. Nikolic, "A shared-well dual-supply-voltage 64-bit ALU," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 3, pp. 494-500, 2004.
- [50] A. Srivastava and D. Govindarajan, "A fast ALU design in CMOS for low voltage operation," *VLSI Design*, vol. 14, no. 4, pp. 315-327, 2002.
- [51] J. T. Kao and A. P. Chandrakasan, "Dual-threshold voltage techniques for low-power digital circuits," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 7, pp. 1009-1018, 2000.
- [52] S. Ghosh, P. Ndai and K. Roy, "A novel low overhead fault tolerant kogge-stone adder using adaptive clocking," *Proc. 2008 Design, Automation and Test in Europe (DATE)*, Munich, Germany, pp. 366-371, March 2008.
- [53] B. Chatterjee and M. Sachdev, "Design of a 1.7-GHz low-power delay-fault-testable 32-b ALU in 180-nm CMOS technology," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 11, pp. 1296-1304, 2005.
- [54] T. Shibata and T. Ohmi, "A functional MOS transistor featuring gate-level weighted sum and threshold operations," *IEEE Transactions on Electron Devices*, vol. 39, no. 6, pp. 1444-1455, 1992.
- [55] A. Srivastava and H. N. Venkata, "Quaternary to binary bit conversion CMOS integrated circuit design using multiple-input floating gate MOSFETS," *Integration, the VLSI journal*, vol. 36, no. 3 pp. 87-101, 2003.

- [56] T. Morie, T. Matsuura, M. Nagata and A. Iwata, "A multinanodot floating-gate MOSFET circuit for spiking neuron models," *IEEE Transactions on Nanotechnology*, vol. 2, no. 3, pp. 158-164, 2003.
- [57] A. Basu, S. Ramakrishnan, C. Petre, S. Koziol, S. Brink and P. E. Hasler, "Neural dynamics in reconfigurable silicon," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 4, no. 5, pp. 311-319, 2010.
- [58] S. C. Liu and B. A. Minch, "Silicon synaptic adaptation mechanisms for homeostasis and contrast gain control," *IEEE Transactions on Neural Networks*, vol. 13, no. 6, pp. 1497-1503, 2002.
- [59] L. Ni et al., "An energy-efficient and high-throughput bitwise CNN on sneak-path-free digital ReRAM crossbar," *Proc. 2017 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, Taipei, Taiwan, pp. 1-6, July 2017.
- [60] B. Moons, R. Uytterhoeven, W. Dehaene and M. Verhelst, "Envision: A 0.26-to-10TOPS/W subword-parallel dynamic-voltage-accuracy-frequency-scalable convolutional neural network processor in 28nm FDSOI," *Proc. 2017 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA, pp. 246-247, February 2017.
- [61] J. Sim, J. S. Park, M. Kim, D. Bae, Y. Choi and L. S. Kim, "A 1.42TOPS/W deep convolutional neural network recognition processor for intelligent IoE systems," *Proc. 2016 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA, pp. 264-265, February 2016.
- [62] Y. H. Chen, T. Krishna, J. Emer and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *Proc. 2016 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA, pp. 262-263, February 2016.
- [63] Y. Guan et al., "FP-DNN: An automated framework for mapping deep neural networks onto FPGAs with RTL-HLS hybrid templates," *Proc. 2017 IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, Napa, CA, USA, pp. 152-159, May 2017.
- [64] Y. Guan, Z. Yuan, G. Sun and J. Cong, "FPGA-based accelerator for long short-term memory recurrent neural networks," *Proc. 2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*, Chiba, Japan, pp. 629-634, January 2017.
- [65] J. C. Ferreira and J. Fonseca, "An FPGA implementation of a long short-term memory neural network," *Proc. 2016 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*, Cancun, Mexico, pp. 1-8, December 2016.
- [66] D. Shin, J. Lee, J. Lee and H. J. Yoo, "DNPU: An 8.1TOPS/W reconfigurable CNN-RNN processor for general-purpose deep neural networks," *Proc. 2017 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA, pp. 240-241, January 2017.
- [67] Y. Zhang, X. Wang and E. G. Friedman, "Memristor-based circuit design for multilayer neural networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 2, pp. 677-686, 2018.

- [68] M. Hu, Y. Chen, J. J. Yang, Y. Wang and H. H. Li, "A compact memristor-based dynamic synapse for spiking neural networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 8, pp. 1353-1366, 2017.
- [69] S. Jo et al. "Nanoscale memristor device as synapse in neuromorphic systems," *Nano letters*, vol. 10, no. 4, pp. 1297-1301, 2010.
- [70] O. Kavehei et al., "Fabrication and modeling of Ag/TiO₂/ITO memristor," *Proc. 2011 IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS)*, Seoul, South Korea, pp. 1-4, September 2011.
- [71] A. Mazady, "Modeling, fabrication, and characterization of memristors." Ph.D. Dissertation, University of Connecticut, Storrs, CT, USA, 2014.
- [72] F. Corradi, C. Eliasmith and G. Indiveri, "Mapping arbitrary mathematical functions and dynamical systems to neuromorphic VLSI circuits for spike-based neural computation," *Proc. 2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, Melbourne VIC, Australia, pp. 269-272, June 2014.
- [73] K. Bong, S. Choi, C. Kim, S. Kang, Y. Kim and H. J. Yoo, "A 0.62mW ultra-low-power convolutional-neural-network face-recognition processor and a CIS integrated with always-on haar-like face detector," *Proc. 2017 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA, pp. 248-249, January 2017.
- [74] D. Maliuk and Y. Makris, "An experimentation platform for on-chip integration of analog neural networks: a pathway to trusted and robust analog/RF ICs," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 8, pp. 1721-1734, 2015.
- [75] D. Maliuk and Y. Makris, "A dual-mode weight storage analog neural network platform for on-chip applications," *Proc. 2012 IEEE International Symposium on Circuits and Systems (ISCAS)*, Seoul, South Korea, pp. 2889-2892, May 2012.
- [76] R. S. Amant et al., "General-purpose code acceleration with limited-precision analog computation," *Proc. 2014 ACM/IEEE 41st International Symposium on Computer Architecture (ISCA)*, Minneapolis, MN, USA, pp. 505-516, June 2014.
- [77] S. P. Mohanty, *Nanoelectronic Mixed-Signal System Design*, McGraw-Hill Education, New York, NY, USA, 2015.
- [78] Latest Predictive Transistor Model, Available: <http://ptm.asu.edu/latest.html>
- [79] Interconnect Estimation, Available: <http://ptm.asu.edu/>
- [80] M. Shevgoor, J. S. Kim, N. Chatterjee, R. Balasubramonian, A. Davis and A. N. Udipi, "Quantifying the relationship between the power delivery network and architectural policies in a 3D-stacked memory device," *Proc. 2013 46th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Davis, CA, USA, pp. 198-209, December 2013.
- [81] J. M. Rabaey, A. P. Chandrakasan and B. Nikoli, *Digital Integrated Circuits*, Prentice Hall, Upper Saddle River, NJ, USA, 2002.

- [82] H. Akkary and M. A. Driscoll, "A dynamic multithreading processor," *Proc. 31st Annual ACM/IEEE International Symposium on Microarchitecture (MICRO)*, Dallas, TX, USA, pp. 226-236, December 1998.
- [83] R. Zhang, K. Wang, B. H. Meyer, M. R. Stan and K. Skadron, "Architecture implications of pads as a scarce resource," *Proc. 2014 ACM/IEEE 41st International Symposium on Computer Architecture (ISCA)*, Minneapolis, MN, USA, pp. 373-384, June 2014.
- [84] M. Popovich, A. V. Mezhiba, E. G. Friedman, *Power Distribution Networks with on-Chip Decoupling Capacitors*, Springer Science, Business Media, New York, NY, USA, 2007.
- [85] K. DeHaven and J. Dietz, "Controlled collapse chip connection (C4)-an enabling technology," *Proc. 44th Electronic Components and Technology Conference*, Washington, DC, USA, pp. 1-6, May 1994.
- [86] Via Optimization Techniques for High-Speed Channel Designs, Available: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/an/an529.pdf
- [87] High Speed PCB Layout Techniques, Available: <http://www.ti.com/lit/ml/slyp173/slyp173.pdf>
- [88] MICRON DDR3 SDRAM, Available: <https://www.micron.com/products/dram/ddr3-sdram>
- [89] R. R. Boudreaux and R. M. Nelms, "A comparison of MOSFETs, IGBTs, and MCTs for solid state circuit breakers," *Proc. 1996 Applied Power Electronics Conference and Exposition (APEC)*, San Jose, CA, USA, pp. 227-233, March 1996.
- [90] D. A. Grant and J. Gowar, *Power MOSFETs: Theory and Applications*, Wiley-Interscience, Hoboken, NJ, USA, 1989.
- [91] OpenMSP430, Available: <http://opencores.org/project,openmsp430>
- [92] Z. Zhao, A. Srivastava, L. Peng, S. Chen and S. Mohanty, "Circuit implementation of switchable pins in chip multiprocessor," *Proc. 2015 IEEE International Symposium on NanoElectronic and Information Systems (iNIS)*, Indore, India, pp. 89-94, December 2015.
- [93] A. M. Abo, "Design for reliability of low-voltage, switched-capacitor circuits," Ph.D. Dissertation, University of California, Berkeley, Berkeley, CA, USA, 1999.
- [94] R. Gutierrez and J. Valls, "Low Cost Hardware Implementation of Logarithm Approximation," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 12, pp. 2326-2330, 2011.
- [95] Properties of Logarithmic Functions, Available: http://www.montereyinstitute.org/courses/DevelopmentalMath/TEXTGROUP-1-19_RESOURCE/U18_L2_T2_text_final.html
- [96] J. Y. Kim and H. J. Yoo, "Bitwise competition logic for compact digital comparator," *Proc. 2007 IEEE Asian Solid-State Circuits Conference (ASSCC)*, Jeju, South Korea, pp. 59-62, November 2007.

- [97] P. Teichmann, *Adiabatic Logic: Future Trend and System Level Perspective*, Springer Science, Business Media, Berlin, Germany, 2011.
- [98] A. B. Stoffi, E. Amirante, J. Fischer, G. Innaccone and D. Landsiedel, "Resonant 90 degree shifter generator for 4-phase trapezoidal adiabatic logic," *Advanced Radio Science*, vol. 1, no. 9, pp. 243–246, 2003.
- [99] S. Nakata, H. Makino and Y. Matsuda, "A new stepwise adiabatic charging circuit with a smaller capacitance in a regenerator than a load capacitance," *Proc. 2014 IEEE 57th International Midwest Symposium on Circuits and Systems (MWSCAS)*, College Station, TX, USA, pp. 439-442, August 2014.
- [100] Z. Zhao, A. Srivastava, L. Peng and S. P. Mohanty, "A low-cost mixed clock generator for high speed adiabatic logic," *Proc. 2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, Pittsburgh, PA, USA, pp. 587-590, July 2016.
- [101] C. Zhang, "A study of phase noise and jitter in submicron CMOS phase-locked loop circuits," Ph.D. Dissertation, Louisiana State University, Baton Rouge, LA, USA, 2006.
- [102] K. H. Abed and R. E. Siferd, "CMOS VLSI implementation of a low-power logarithmic converter," *IEEE Transactions on Computers*, vol. 52, no. 11, pp. 1421-1433, 2003.
- [103] D. D. Caro, M. Genovese, E. Napoli, N. Petra and A. G. M. Strollo, "Accurate fixed-point logarithmic converter," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 7, pp. 526-530, 2014.
- [104] T. Shibata and T. Ohmi, "Neuron MOS binary-logic integrated circuits. I. Design fundamentals and soft-hardware-logic circuit implementation," *IEEE Transactions on Electron Devices*, vol. 40, no. 3, pp. 570-576, 1993.
- [105] T. Shibata and T. Ohmi, "Neuron MOS binary-logic integrated circuits. II. Simplifying techniques of circuit configuration and their practical applications," *IEEE Transactions on Electron Devices*, vol. 40, no. 5, pp. 974-979, 1993.
- [106] H. N. Venkata, "Ternary and quaternary logic to binary bit conversion CMOS integrated circuit design using multiple input floating gate MOSFETs," M.S.E.E. Thesis, Louisiana State University, Baton Rouge, LA, USA, 2002.
- [107] P. Nigh and W. Maly, "A self-testing ALU using built-in current sensing," *Proc. 1989 IEEE Custom Integrated Circuits Conference (CICC)*, San Diego, CA, USA, pp. 22.1/1-22.1/4, May 1989.
- [108] M. Anders et al., "A 6.5GHz 130nm single-ended dynamic ALU and instruction scheduler loop," *Proc. 2002 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA, pp. 332-534, February 2002.
- [109] M. Fojtik et al., "Bubble Razor: Eliminating timing margins in an ARM Cortex-M3 processor in 45nm CMOS using architecturally independent error detection and correction," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 1, pp. 66-81, 2013.

- [110] A. Srivastava and C. Srinivasan, "ALU design using reconfigurable CMOS logic," *Proc. 2002 Midwest Symposium on Circuits and Systems (MWSCAS)*, Tulsa, OK, USA, pp. 663-666, August 2002.
- [111] E. A. Cortés-Barrón, M. A. Reyes-Barranca, L. M. Flores-Nava and A. Medina-Santiago, "4-bit arithmetic logic unit (ALU) based on neuron MOS transistors," *Proc. 2012 International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, Mexico City, Mexico, pp. 1-6, September 2012.
- [112] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.
- [113] S. Subramanian, "Ternary logic to binary bit conversion using multiple input floating gate MOSFETS in 0.5-micron n-well CMOS technology," M.S.E.E. Thesis, Louisiana State University, Baton Rouge, LA, USA, 2005.
- [114] Floating Gate Techniques and Applications, Available: <http://ece.tamu.edu/~s-sanchez/607-2010-Floating%20Gate%20Circuits.pdf/>
- [115] C. W. Kim and S. B. Park, "New four-quadrant CMOS analogue multiplier," *Electronics Letters*, vol. 23, no. 24, pp. 1268-1269, 1987.
- [116] H. J. Song and C. K. Kim, "An MOS four-quadrant analog multiplier using simple two-input squaring circuits with source followers," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 3, pp. 841-848, 1990.
- [117] J. N. Babanezhad and G. C. Temes, "A 20-V four-quadrant CMOS analog multiplier," *IEEE Journal of Solid-State Circuits*, vol. 20, no. 6, pp. 1158-1168, 1985.
- [118] D. C. Soo and R. G. Meyer, "A four-quadrant NMOS analog multiplier," *IEEE Journal of Solid-State Circuits*, vol. 17, no. 6, pp. 1174-1178, 1982.
- [119] G. Han and E. S. Sinencio, "CMOS transconductance multipliers: a tutorial," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 45, no. 12, pp. 1550-1563, 1998.
- [120] P. E. Allen and D. R. Holberg, *CMOS Analog Circuit Design*, Oxford, New York, NY, USA, 2002.
- [121] C. Mead and I. Mohammed, *Analog VLSI Implementation of Neural Systems*, Springer Science & Business Media, New York, NY, USA, 2012.
- [122] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz and W. J. Dally, "EIE: Efficient inference engine on compressed deep neural network," *Proc. ACM/IEEE Annual International Symposium on Computer Architecture (ISCA)*, Seoul, South Korea, pp.243-254, June 2016.
- [123] F. Piazza, A. Uncini and M. Zenobi, "Neural networks with digital LUT activation functions," *Proc.1993 International Conference on Neural Networks (IJCNN)*, Nagoya, Japan, pp. 1401-1404, October 1993.

- [124] K. M. Al-Ruwaihi, "CMOS analogue neurone circuit with programmable activation functions utilising MOS transistors with optimised process/device parameters," *IEE Proceedings-Circuits, Devices and Systems*, vol. 144, no. 6, pp. 318-322, 1997.
- [125] S. Tabarce, V. G. Tavares and P. G. de Oliveira, "Programmable analogue VLSI implementation for asymmetric sigmoid neural activation function and its derivative," *Electronics Letters*, vol. 41, no. 15, pp. 863-864, 2005.
- [126] X. Tang and K. P. Pun, "High-performance CMOS current comparator," *Electronics Letters*, vol. 45, no. 20, pp. 1007-1009, 2009.
- [127] H. Traff, "Novel approach to high speed CMOS current comparators," *Electronics Letters*, vol. 28, no. 3, pp. 310-312, 1992.
- [128] D. Banks and C. Toumazou, "Low-power high-speed current comparator design," *Electronics Letters*, vol. 44, no. 3, pp. 171-172, 2008.
- [129] J. Fadavi-Ardekani, "M×N Booth encoded multiplier generator using optimized Wallace trees," *Proc. 1992 IEEE International Conference on Computer Design: VLSI in Computers & Processors*, Cambridge, MA, USA, pp. 114-117, October 1992.
- [130] C. S. Wallace, "A suggestion for a fast multiplier," *IEEE Transactions on Electronic Computers*, vol. EC-13, no. 1, pp. 14-17, 1964.
- [131] A. X. M. Chang and E. Culurciello, "Hardware accelerators for recurrent neural networks on FPGA," *Proc. 2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, Baltimore, MD, USA, pp. 1-4, May 2017.
- [132] V. Rybalkin, N. Wehn, M. R. Yousefi and D. Stricker, "Hardware architecture of Bidirectional Long Short-Term Memory Neural Network for Optical Character Recognition," *Proc. Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Lausanne, Switzerland, pp. 1390-1395, March 2017.
- [133] F. Conti, L. Cavigelli, G. Paulin, I. Susmelj and L. Benini, "Chipmunk: A systolically scalable 0.9 mm², 3.08Gop/s/mW @ 1.2 mW accelerator for near-sensor recurrent neural network inference," *Proc. 2018 IEEE Custom Integrated Circuits Conference (CICC)*, San Diego, CA, USA, pp. 1-4, April 2018.
- [134] A. Amravati, S. B. Nasir, S. Thangadurai, I. Yoon and A. Raychowdhury, "A 55nm time-domain mixed-signal neuromorphic accelerator with stochastic synapses and embedded reinforcement learning for autonomous micro-robots," *Proc. 2018 IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA, pp. 124-126, February 2018.
- [135] T. McConaghy, K. Breen, J. Dyck and A. Gupta, *Variation-Aware Design of Custom Integrated Circuits: a Hands-on Field Guide*, Springer Science & Business Media, New York, NY, USA, 2012.
- [136] M. J. M. Pelgrom, A. C. J. Duinmaijer and A. P. G. Welbers, "Matching properties of MOS transistors," *IEEE Journal of Solid-State Circuits*, vol. 24, no. 5, pp. 1433-1439, 1989.

- [137] A. Asenov, "Random dopant induced threshold voltage lowering and fluctuations in sub-0.1 μ m MOSFET's: A 3-D "atomistic" simulation study," *IEEE Transactions on Electron Devices*, vol. 45, no. 12, pp. 2505-2513, 1998.
- [138] Y. Long, T. Na and S. Mukhopadhyay, "ReRAM-based processing-in-memory architecture for recurrent neural network acceleration," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. pp, no. 99, pp. 1-14, 2018.
- [139] M. P. Marcus, B. Santorini and M. A. Marcinkiewicz, "Building a large annotated corpus of English: The Penn Treebank," *Computational linguistics*, vol. 19, no. 2, pp. 313-330, 1993.

APPENDIX A

MOSIS SPICE LEVEL 7 CMOS MODEL PARAMETERS

The following level 7 CMOS model parameters have been obtained from www.mosis.org.

A.1 AMI 500nm Technology

NMOS Model

```
.MODEL NMOS NMOS                LEVEL = 7
+TNOM = 27          TOX = 1.39E-8
+XJ = 1.5E-7       NCH = 1.7E17    VTH0 = 0.5916823
+K1 = 0.9970315   K2 = -0.1275822  K3 = 32.4012543
+K3B = -7.5008412  W0 = 1.064896E-8  NLX = 1E-9
+DVT0W = 0        DVT1W = 0        DVT2W = 0
+DVT0 = 0.7187527  DVT1 = 0.3412308  DVT2 = -0.5
+U0 = 449.0974306  UA = 1.085166E-13  UB = 1.425387E-18
+UC = -1.74612E-14  VSAT = 1.669807E5  A0 = 0.5933847
+AGS = 0.1035164   B0 = 1.668729E-6  B1 = 5E-6
+KETA = -3.86833E-3  A1 = 2.152036E-4  A2 = 0.3536877
+RDSW = 1.061307E3  PRWG = 0.0822544  PRWB = 1.740605E-5
+WR = 1           WINT = 2.482453E-7  LINT = 7.591645E-8
+XL = 1E-7        XW = 0            DWG = -1.279102E-8
+DWB = 4.853492E-8  VOFF = 0          NFACTOR = 0.9088018
+CIT = 0          CDSC = 2.4E-4      CDSCD = 0
+CDSCB = 0        ETA0 = 2.907125E-3  ETAB = 0.9951793
+DSUB = 0.0492559  PCLM = 2.2082829  PDIBLC1 = 3.324146E-4
+PDIBLC2 = 1.488616E-3  PDIBLCB = -0.1429457  DROUT = 8.319308E-5
+PSCBE1 = 2.516822E9  PSCBE2 = 5E-10    PVAG = 0
+DELTA = 0.01      RSH = 84          MOBMOD = 1
+PRT = 0           UTE = -1.5        KT1 = -0.11
+KT1L = 0          KT2 = 0.022       UA1 = 4.31E-9
+UB1 = -7.61E-18   UC1 = -5.6E-11    AT = 3.3E4
+WL = 0           WLN = 1          WW = 0
+WWN = 1          WWL = 0          LL = 0
+LLN = 1          LW = 0          LWN = 1
+LWL = 0          CAPMOD = 2       XPART = 0.5
+CGDO = 1.82E-10   CGSO = 1.82E-10   CGBO = 1E-9
+CJ = 4.29253E-4   PB = 0.8858652    MJ = 0.4353384
+CJSW = 3.622458E-10  PBSW = 0.8        MJSW = 0.2339166
+CJSWG = 1.64E-10  PBSWG = 0.8       MJSWG = 0.2339166
+CF = 0           PVTH0 = -0.0504111  PRDSW = 277.8034293
+PK2 = -0.0683031  WKETA = 5.494298E-3  LKETA = -6.64089E-3
*
```

PMOS Model

```
.MODEL PMOS PMOS                LEVEL = 7
```

```

+TNOM = 27      TOX = 1.39E-8
+XJ  = 1.5E-7   NCH = 1.7E17   VTH0 = -0.9152268
+K1  = 0.553472 K2  = 7.871921E-3 K3  = 0.3045116
+K3B = 0.4207623 W0  = 5.814395E-7 NLX = 6.71009E-8
+DVT0W = 0      DVT1W = 0      DVT2W = 0
+DVT0 = 0.6597447 DVT1 = 0.3261802 DVT2 = -0.3
+U0  = 201.3603195 UA  = 2.408572E-9 UB  = 1E-21
+UC  = -1E-10    VSAT = 9.345943E4  A0  = 0.8172376
+AGS = 0.097476  B0  = 5.101025E-7 B1  = 1.701779E-8
+KETA = -4.865785E-3 A1  = 0      A2  = 0.6429159
+RDSW = 3E3      PRWG = -0.0280817 PRWB = -0.0480265
+WR   = 1        WINT = 2.908993E-7 LINT = 1.225719E-7
+XL   = 1E-7     XW   = 0      DWG  = 1.554343E-9
+DWB  = -1.9281E-8 VOFF = -0.0705167 NFACTOR = 1.016159
+CIT  = 0        CDSC = 2.4E-4   CDSCD = 0
+CDSCB = 0      ETA0 = 6.97667E-4 ETAB = -0.2
+DSUB = 1       PCLM = 2.4034299 PDIBLC1 = 0.0525071
+PDIBLC2 = 3.609644E-3 PDIBLCB = -0.0206211 DROUT = 0.2542493
+PSCBE1 = 1E8    PSCBE2 = 3.365142E-9 PVAG = 0.0150027
+DELTA = 0.01    RSH  = 107.2   MOBMOD = 1
+PRT  = 0        UTE  = -1.5    KT1  = -0.11
+KT1L = 0        KT2  = 0.022   UA1  = 4.31E-9
+UB1  = -7.61E-18 UC1  = -5.6E-11 AT  = 3.3E4
+WL   = 0        WLN  = 1     WW   = 0
+WWN  = 1        WWL  = 0     LL   = 0
+LLN  = 1        LW   = 0     LWN  = 1
+LWL  = 0        CAPMOD = 2    XPART = 0.5
+CGDO = 2.22E-10 CGSO = 2.22E-10 CGBO = 1E-9
+CJ   = 7.306902E-4 PB  = 0.9419671 MJ  = 0.4929212
+CJSW = 2.535867E-10 PBSW = 0.8    MJSW = 0.3168453
+CJSWG = 6.4E-11  PBSWG = 0.8    MJSWG = 0.3168453
+CF   = 0        PVTH0 = 5.98016E-3 PRDSW = 14.8598424
+PK2  = 3.73981E-3 WKETA = 1.137045E-4 LKETA = -9.773144E-3 )
*
```

A.2 TSMC 180nm Technology

NMOS Model

```

.MODEL NMOS NMOS (          LEVEL = 7
+VERSION = 3.1      TNOM = 27      TOX = 4E-9
+XJ  = 1E-7        NCH = 2.3549E17  VTH0 = 0.3627858
+K1  = 0.5873035  K2  = 4.793052E-3  K3  = 1E-3
+K3B = 2.2736112  W0  = 1E-7      NLX = 1.675684E-7
+DVT0W = 0        DVT1W = 0      DVT2W = 0
+DVT0 = 1.7838401 DVT1 = 0.5354277 DVT2 = -1.243646E-3
+U0  = 263.3294995 UA  = -1.359749E-9 UB  = 2.250116E-18
+UC  = 5.204485E-11 VSAT = 1.083427E5  A0  = 2
```

```

+AGS = 0.4289385 B0 = -6.378671E-9 B1 = -1E-7
+KETA = -0.0127717 A1 = 5.347644E-4 A2 = 0.8370202
+RDSW = 150 PRWG = 0.5 PRWB = -0.2
+WR = 1 WINT = 1.798714E-9 LINT = 7.631769E-9
+XL = -2E-8 XW = -1E-8 DWG = -3.268901E-9
+DWB = 7.685893E-9 VOFF = -0.0882278 NFACTOR = 2.5
+CIT = 0 CDSC = 2.4E-4 CDSCD = 0
+CDSCB = 0 ETA0 = 2.455162E-3 ETAB = 1
+DSUB = 0.0173531 PCLM = 0.7303352 PDIBLC1 = 0.2246297
+PDIBLC2 = 2.220529E-3 PDIBLCB = -0.1 DROUT = 0.7685422
+PSCBE1 = 8.697563E9 PSCBE2 = 5E-10 PVAG = 0
+DELTA = 0.01 RSH = 6.7 MOBMOD = 1
+PRT = 0 UTE = -1.5 KT1 = -0.11
+KT1L = 0 KT2 = 0.022 UA1 = 4.31E-9
+UB1 = -7.61E-18 UC1 = -5.6E-11 AT = 3.3E4
+WL = 0 WLN = 1 WW = 0
+WWN = 1 WWL = 0 LL = 0
+LLN = 1 LW = 0 LWN = 1
+LWL = 0 CAPMOD = 2 XPART = 0.5
+CGDO = 7.16E-10 CGSO = 7.16E-10 CGBO = 1E-12
+CJ = 9.725711E-4 PB = 0.7300537 MJ = 0.365507
+CJSW = 2.604808E-10 PBSW = 0.4 MJSW = 0.1
+CJSWG = 3.3E-10 PBSWG = 0.4 MJSWG = 0.1
+CF = 0 PVTH0 = 4.289276E-4 PRDSW = -4.2003751
+PK2 = -4.920718E-4 WKETA = 6.938214E-4 LKETA = -0.0118628
+PU0 = 24.2772783 PUA = 9.138642E-11 PUB = 0
+PVSAT = 1.680804E3 PETA0 = 2.44792E-6 PKETA = 4.537962E-5 )
*

```

PMOS Model

```

.MODEL PMOS PMOS ( LEVEL = 7
+VERSION = 3.1 TNOM = 27 TOX = 4E-9
+XJ = 1E-7 NCH = 4.1589E17 VTH0 = -0.4064886
+K1 = 0.5499001 K2 = 0.0389453 K3 = 0
+K3B = 11.4951756 W0 = 1E-6 NLX = 9.143209E-8
+DVT0W = 0 DVT1W = 0 DVT2W = 0
+DVT0 = 0.5449299 DVT1 = 0.3160821 DVT2 = 0.1
+U0 = 117.9612996 UA = 1.64867E-9 UB = 1.165056E-21
+UC = -1E-10 VSAT = 2E5 A0 = 1.7833459
+AGS = 0.407511 B0 = 1.314603E-6 B1 = 5E-6
+KETA = 0.0137171 A1 = 0.4610527 A2 = 0.6597363
+RDSW = 364.9443889 PRWG = 0.5 PRWB = -0.1129203
+WR = 1 WINT = 0 LINT = 2.007556E-8
+XL = -2E-8 XW = -1E-8 DWG = -2.835566E-8
+DWB = 8.003075E-9 VOFF = -0.1064646 NFACTOR = 2
+CIT = 0 CDSC = 2.4E-4 CDSCD = 0

```

```

+CDSCB = 0          ETA0 = 0.0141703  ETAB = -0.0398356
+DSUB  = 0.4441401  PCLM  = 2.2364512  PDIBLC1 = 9.167645E-4
+PDIBLC2 = 0.0209189  PDIBLCB = -9.568266E-4  DROUT  = 9.976778E-4
+PSCBE1 = 1.731161E9  PSCBE2 = 5E-10      PVAG   = 14.337819
+DELTA = 0.01      RSH   = 7.5      MOBMOD = 1
+PRT   = 0        UTE   = -1.5     KT1    = -0.11
+KT1L  = 0        KT2   = 0.022   UA1    = 4.31E-9
+UB1   = -7.61E-18  UC1   = -5.6E-11  AT     = 3.3E4
+WL    = 0        WLN   = 1       WW     = 0
+WWN   = 1        WWL   = 0       LL     = 0
+LLN   = 1        LW    = 0       LWN    = 1
+LWL   = 0        CAPMOD = 2      XPART  = 0.5
+CGDO  = 6.79E-10  CGSO  = 6.79E-10  CGBO   = 1E-12
+CJ    = 1.176396E-3  PB    = 0.8607121  MJ     = 0.4163285
+CJSW  = 2.135953E-10  PBSW  = 0.6430918  MJSW   = 0.2654457
+CJSWG = 4.22E-10   PBSWG  = 0.6430918  MJSWG  = 0.2654457
+CF    = 0        PVTH0 = 4.364418E-3  PRDSW  = 4.4192048
+PK2   = 3.104478E-3  WKETA = 0.0270296  LKETA  = 2.038008E-3
+PU0   = -2.3639825  PUA   = -8.41675E-11  PUB    = 1E-21
+PVSAT = -50      PETA0 = 1E-4      PKETA  = -1.444802E-3 )
*
```

APPENDIX B

PROPOSED DESIGN FLOW OF AUTOMATIC PLACEMENT AND ROUTING USING MENTOR GRAPHICS EDA TOOLS

B.1 APR with Verilog Entry

Figure B.1 (a) shows the entire design flow of APR with Verilog Entry using *Mentor Graphics* EDA tools. This flow has been used for the post-layout simulations in Chapter 2 under *Windows* operation system and can be summarized as follows:

- 1) We export EDF file compiled from Verilog source using *Mentor Graphic LeonardSpectrum*. Note that SYN. file, which supports logic synthesis, should be matched to the corresponding CMOS technology file;

- 2) TPR file is exported from EDF file in *Tanner Schematic-Edit*;

- 3) Due to already-defined grammar in SYN file, we revise the I/O port name of standard cells as the form of IN [n], OUT [n], where n is the number of input or output starting from 0;

- 4) Using the layout library of standard cells and customized rules of APR, TPR file automatically generates the total layout in *Tanner Layout-Edit*;

- 5) Using EXT file and SPICE model of the corresponding technology, post-layout simulations are implemented in *PSpice*.

B.2 APR with Customized Schematic Entry

Figure B.1 (b) shows the entire design flow of APR with customized schematic entry using *Mentor Graphics* EDA tools. This flow can drive the post-layout simulation of the digital block built by customized cells, which is not standard cells based on static logic. The proposed flow has been used for the post-layout simulations in Chapter 3 and Chapter 4 under *Windows* operation system and described as follows:

- 1) Design the schematic cells of customized logic gates in *Tanner Schematic-Edit*;

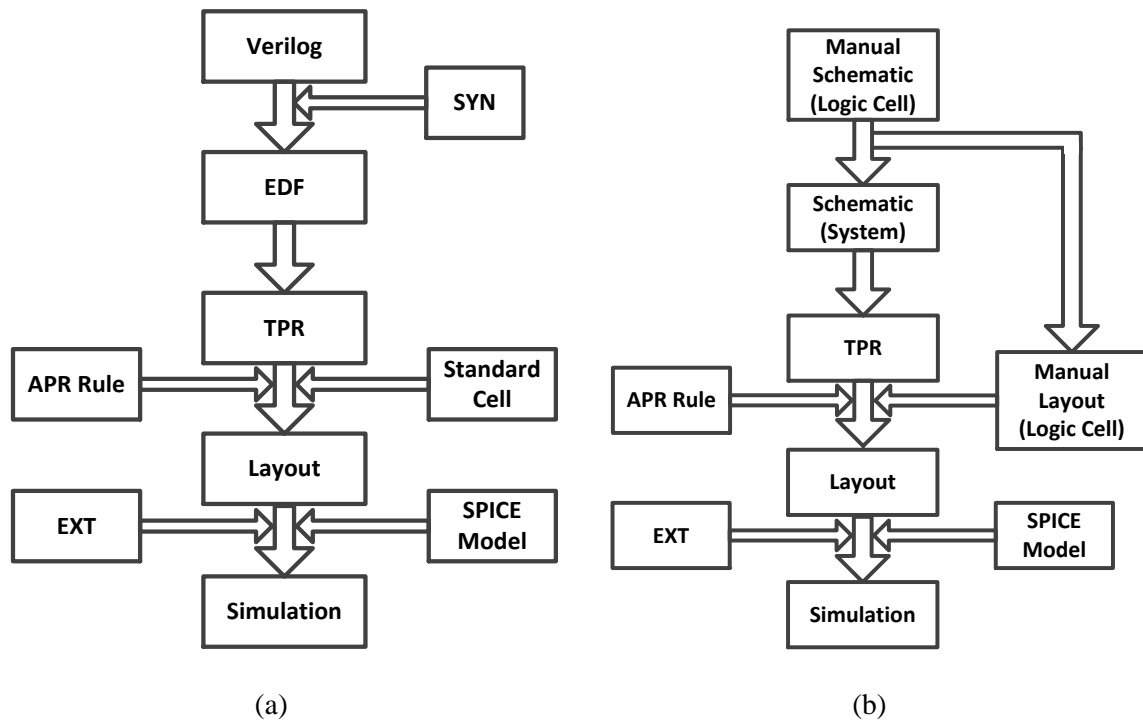


Figure B.1 Design flow of APR using *Mentor Graphics* EDA tools, (a) HDL entry, (b) customized schematic entry.

2) Design the layout cells of customized logic gates in *Tanner Layout-Edit*. The way to design these layout cells is the same as the one to design standard cells based static logic, which includes square layout, top-located power port using metal 1, bottom-located ground port using metal 1, vertical-extent data port using the connection of metal 1 and metal 2, and pre-placed via for further routing. Note that each logic gate in layout view should strictly correspond to the schematic cell in terms of connection, dimension and port name;

3) Customized schematic entry of system level is implemented in *Tanner Schematic-Edit*;

4) TPR. file can be extracted from schematic entry in *Tanner Schematic-Edit*;

5) In *Tanner Layout-Edit*, TPR file mapping to the library of customized layout cells with the customized rules of APR can generate the total layout;

6) Using EXT file and SPICE model of the corresponding technology, post-layout simulations are implemented in *PSpice*.

APPENDIX C ERROR TEST IN PSPICE SIMULATION

Figure C.1 shows the described test flow. The proposed method includes two main blocks, which are the real block simulated under SPICE model and the ideal one which is built by customized gates with delay-free, glitch-free and infinite fan-out, self-defined delay cell in *PSpice*. The combination of customized gates achieves the same function as in real block. The self-defined delay cell is set to the same latency as in real block without any PVT variation or process corner. Comparing two groups of results from two blocks using XOR gates, once logic high occurs, it means variations or corners bring the incorrect data transmission to the real block. Note that it is possible that the logic high lasts very short which should be seen as a glitch but not an error. To remove these glitches, another ideal buffer with rising slope follows the output of the XOR gate. In this buffer, the rising time, T_r , for logic switch is set to $1/(10*f)$, where f is the work frequency in the current simulation for both two blocks. Using this method, if the logic high lasts less than $1/10$ of a period, it will be removed by the buffer. Otherwise, it will be seen as an error shown after buffering. Then the output of the buffer connects to an ideal counter triggered by positive edge. Therefore, once the simulation of entire cycles is finished, the error rate can be calculated by:

$$Error\% = \frac{\sum_{i=1}^{N_o} E_i}{N_C \times N_o} \quad (C.1)$$

where N_C and N_O are the number of cycles and output nodes, respectively. E_i is the number of error corresponding to the i_{th} output node.

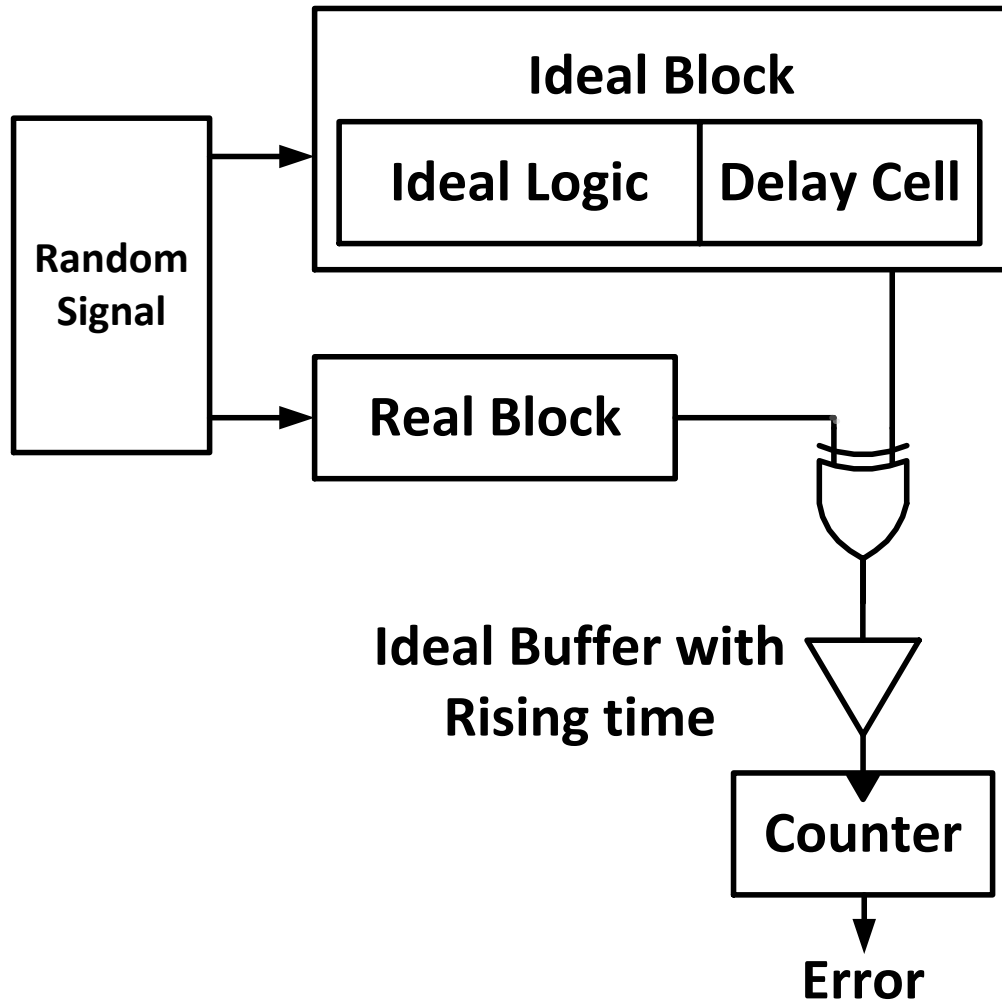


Figure C.1 Customized flow used for error testing.

VITA

Zhou Zhao was born in 1988 in Urumqi, China. He received B.S.E.E. and M.S.E.E. from the University of Electronic Science and Technology of China, Chengdu, China in July 2011 and July 2014, respectively. Since August 2014, he started his Ph.D. study under Dr. Ashok Srivastava at the Louisiana State University, Baton Rouge, Louisiana. Zhou Zhao's research interests include design of low-power mixed signal and digital circuits used for general-purpose computing, neural accelerator and digital calibration.