NORTHWESTERN UNIVERSITY

Low-Power Mixed-Signal Time-Domain Hardware Accelerator Design for Edge Computing and Machine Learning

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Computer Engineering

By

Zhengyu Chen

EVANSTON, ILLINOIS

December 2020

ABSTRACT

Low-Power Mixed-Signal Time-Domain Hardware Accelerator Design for Edge Computing and Machine Learning

Zhengyu Chen

While the entire silicon industry has been blooming under Moore's Law for decades, conventional digital implementation is approaching the "stall" of Moore's Law due to many physical design limitations. Technology innovation now is going to take a different direction. Given the increasing demand for emerging applications' computational capacity, it is urgent to find alternative computing methods that can bring efficiency beyond conventional digital approach. To improve the computing efficiency, multiple alternative solutions have been proposed, e.g. approximate computing, parallel computing, quantum computing, time-domain computing, etc. Among them, mixed-signal time-domain computing (MSTC) has drawn significant attention recently due to its high energy efficiency and low area cost. As the nature of mixed-signal design, MSTC combines the advantages of both digital and analog computing methods. On one hand, circuit-wise, MSTC utilizes digital components to encode/decode and processes information in time domain, which brings the benefit of technology scalability and compatibility of the current digital design flow. On the other hand, from the signal processing aspect, MSTC is similar to analog computing as the information can be more densely encoded in a single signal, e.g. a time pulse, leading to benefits similar to the analog-based processing. Such

benefits include the high efficiency in energy consumption and a desirable error resiliency/scalability where most-significant-bit is the least likely to show errors.

In this thesis, the innovative MSTC circuit, architecture, and algorithm design methodology are introduced to accelerate emerging applications, e.g. image processing and machine learning. To provide a concrete circuit design foundation, the variation-aware MSTC circuit design methodology is introduced. The basic arithmetic cells and other complex operation modules including time flip-flop, time-domain accumulator, and time-domain multiplier, are implemented in time domain. To demonstrate the energy and area efficiency in conducting non-linear operations, a MSTC-accelerated image processing engine is presented with over 40% area and energy improvement compared to the digital counterparts. To improve the throughput in MSTC, the first time-domain pipeline architecture for dynamic time warping (DTW) algorithm is proposed, which is enabled by the special time flip-flop design. The proposed pipeline operation leads to an order-of-magnitude improvement in throughput and a scalable processing capability for time series classification tasks.

Moreover, to demonstrate the efficiency in conducting machine learning workloads, a MSTC-based accelerator for deep neural network (DNN) applications, i.e. generative adversarial network (GAN), is developed. The proposed GAN accelerator is the first mixed-signal circuit implementation with efficient multi-bit multiplication for on-chip DNN inference and training. Different from prior time-based implementations which need successive conversion between time and digital domain when realizing MAC operation, this work computes the entire MAC operation in time domain, rendering the highest throughput of 18~5400× with similar efficiency. To explore the potential in carrying the emerging Compute-In-Memory (CIM) task, an energy efficient time-domain CIM processor is proposed. A single-phase MAC operation is realized to

remove throughput overhead of prior multi-phase operation and digital accumulation. The power bottleneck of ADC/SA is mitigated by implementing a computation-adaptive ADC skipping operation and special analog sparsity scheme, leading to additional 2~3× reduction of CIM macro power.

Finally, to deliver the missing element of design automation for time-based design and to make it compatible with the existing digital design flow, a systematic design automation flow for MSTC is presented. More specifically, a digital compatible synthesis and backend flow is developed with novel variation-aware RTL mapping and ACG-based placement algorithms to enable the automation of MSTC design. Compared to the existing analog placement methods and commercial EDA tools, the proposed design automation scheme shows significant improvement in the signal matching performance.

Acknowledgements

The Ph.D. program is not only a process that helps me gain technical knowledge and research capability, but also a journey that cultivates my soul, my personality, and my faith in achieving success in whatever situations.

Throughout my Ph.D. program, I have been supported by so many beloved people who provided me with advice, encouragement, and strength. Without these lovely people, my time at Northwestern University would not have been fulfilled by excitement and pleasure.

I would like to express deepest appreciation to my Ph.D. advisor, Professor Jie Gu, for his continuous guidance and smart ideas throughout my studies at Northwestern. He spent significant efforts and patience in guiding me to become a mature researcher. His insight on high-quality research topics greatly inspired my research passion, rendering a successful Ph.D. journey for me. I could not expect a better advisor. Thanks again Prof. Gu, for all his guidance and help during the past five years.

I want to thank other committee members, including Professor Hai Zhou and Professor Seda Ogrenci Memik. Prof. Zhou has been mentoring my research at Northwestern and providing me with a lot of valuable guidance in my research on design automation. Prof. Memik has been providing me with very helpful feedback on multiple occasions including research topics, Ph.D. prospectus and defense. Many thanks to their guidance, comments, and recommendations over the past years.

I would like to thank all the lab members who I have been working with throughout my Ph.D. period. Dr. Tianyu Jia, who joined this lab with me at the same time, is an excellent colleague and a good friend of mine. We have been extensively collaborating over the past years.

Great thanks to the Master students who have been me working with me on those tough tape-out projects, including Sihua Fu, Xi Chen, Qiankai Cao, Geng Xie and Zhenduo Zhai. They have been learning and researching with me together over the past years. I would also thank other lab members including Kofi Otseidu, Yijie Wei, Yuhao Ju and Aly Shoukry. Thank them for all the assistance and help throughout these years.

Moreover, I would like to thank my internship advisor, Mr. Christopher Clark, at Google TPU team. I would also thank other colleagues I worked with in the TPU team in Wisconsin Madison. I had a great internship experience with their help and advice and got involved into real industrial product development during the summer in 2019.

Lastly, and most importantly, I would like to show great appreciation to my family. None of the achievements in this journey would have become possible without my family. I would thank my parents for the endless love and countless support both mentally and financially. Without their support and encouragement, this exciting journey of Ph.D. could not be concluded with such a joyful ending.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Mixed-Signal Time-Domain Computing

The energy improvement of conventional digital circuits has reached a bottleneck as the dynamic energy consumption of digital logic gates is dictated by CV_{dd}^2 where both C, i.e. the capacitance of the circuits, and V_{dd}, i.e. the supply voltages, are limited by the technology [1]. Besides, the leakage power of digital design also contributes significantly to the total power consumption and the leakage power is mainly determined by the technology in use. Finding alternative computing methods is quite urgent to bring efficiency beyond the conventional digital approach. To fulfill such a demand, several non-conventional digital techniques such as approximate computing and stochastic computing have been proposed providing a good tradeoff between energy/area consumption and accuracy. Such a tradeoff is made possible based on the fact that 80% of daily application have certain degree of error tolerance leading to feasibility of approximation in computing [2, 3]. Despite of the different methodologies used in the above low power design techniques, the energy reduction is still based on conventional voltage and technology scaling leaving little room for further improvement assuming logic optimization has been well obtained from the modern design automation tools. The single-core processor performance over the recent decades is shown in Figure 1.1. As depicted in the figure, the gap between the single-core computing capacity and the prediction of Moore's Law is getting larger and larger.

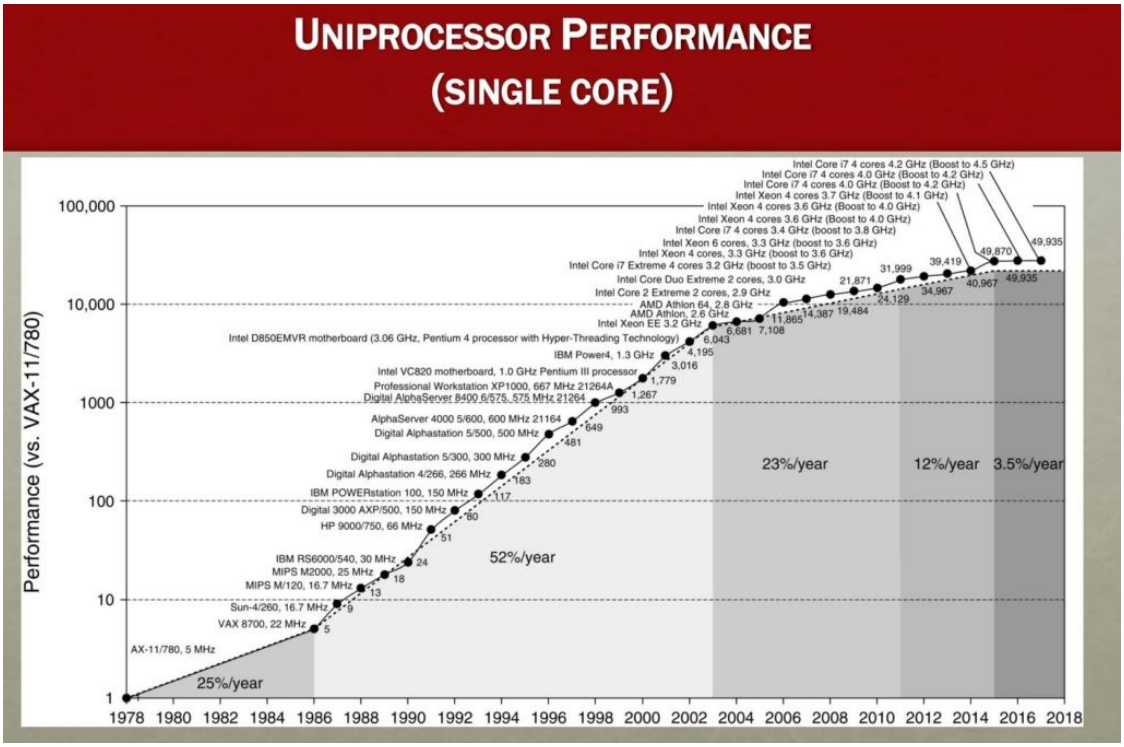


Figure 1.1 Uni-processer performance.

On the other hand, analog computing, which is potentially more energy- and area-efficient than its digital counterpart at the cost of limited accuracy, has been explored over decades. For instance, a digital-analog hybrid neural network exploits efficient analog computation and digital intra-network communication for feature extraction and classification with an equivalent digital design [4]. A switched capacitor based analog matrix multiplication design was proposed to perform multiply-accumulate (MAC) operations efficiently for machine learning tasks with similar accuracy compared to the digital counterpart [5]. In addition, memristor or RRAM-based computing explores the voltage, current and resistance relationship to achieve much higher efficiency on MAC operations for deep neural network (DNN) applications [6]. Other analog computing designs, such as analog multiplier, and mixed-signal FIR filter [7, 8, 9, 10] , offer several attractive features such as high energy efficiency in certain applications. However,

analog computing suffers from its limitation on voltage scalability due to the design of amplifiers, process variation sensitivity, the static current from amplifiers, and incompatibility to conventional digital design flow. As a result, analog computing has not been chosen as the primary design method for general purpose computing compared to its digital counterpart.

Recently, the mixed-signal time-domain computing (MSTC) emerges as an interesting alternative to the existing computing methods [11, 12, 13, 14]. MSTC combines the advantages of both digital and analog methods. After encoding information into time domain, conventional arithmetic operations can be more efficiently conducted by manipulating the signals in time domain with special time-domain operation modules. As a result, MSTC is a good candidate for realizing emerging applications like image processing and machine learning.

In this thesis, several interesting and novel research at circuit-, algorithm-, architecture- and design methodology-level are going to be presented including: (1) Circuit-level design of novel MSTC modules including the time flip-flop, time-domain accumulator, time-domain multiplier and other high-efficient time-domain logic cells; (2) Algorithm-level designs for low-power image processing, dynamic programming, and machine learning applications, e.g. convolutional neural network (CNN) and generative adversarial network (GAN); (3) Architecture-level design for scalable pipelined time-domain architecture, and multi-bit analog memory compute-in-memory (CIM) computing; (4) Design-automation methodology for MSTC including simulated annealing-based frontend (synthesis) and backend (place&route); All these techniques are verified by rigorous simulation and silicon implementation throughout fabricated chips.

1.2 Motivation

1.2.1 Circuit Variation and Nonlinear Operation in Time Domain

The basic idea of MSTC is to represent data/information in the format of delay or length of time pulses and then process the information in time domain with special mixed-signal circuits. As all the information is carried and processed in time domain, the timing control is critical to guarantee the error resiliency due to the sensitivity of delay to variations including both global variation and local mismatches. Since the least-significant-bit (LSB) resolution is pre-defined, a mismatch of timing beyond this value will lead to single-bit error. There has been a lack of discussion on robustness and variation impact to the MSTC computing, which is the crucial consideration in this type of design. Also, the efficiency of conducting non-linear operations in time domain was not well explored [11-17]. As a result, most of existing time-based works suffer from the following issues: (1) Variation impact which is critical to the time-domain computing design, is not well considered and analyzed [12, 13, 14]; (2) The existing designs utilized an inefficient and variation vulnerable multiple-gate time encoding circuit that limited the advantages of the technique [11]; (3) The strong capability of MSTC in various nonlinear operations, e.g. MAX, MIN operation, has not been well explored leaving limited improvement from the techniques [12, 13, 14]. To deal with the above issues, the following techniques are developed including (1) a variation-driven design methodology for MSTC is proposed, (2) the high-efficient time-domain nonlinear operation development are demonstrated by a MSTC image processing engine.

1.2.2 Special Purpose Accelerator Design in Time Domain

Special purpose accelerators have recently gained significant interests thanks to the bloom of machine learning applications. Several time-domain demonstrations in CNN and reinforcement-learning have been developed in recent years to improve the energy and area efficiency [15, 16, 17]. In addition, time-based Computing-In-Memory (CIM) techniques which incorporate analog computing inside memory macros have shown significant advantages in computation efficiency for deep learning applications [18, 19, 20, 21, 22, 23]. However, there are quite some limitations in the existing demonstrations: (1) There is a lack of memory in time-domain operations which significantly limits the design space of the technique; (2) Most prior works suffer from low throughput and low hardware utilization due to the non-pipelined operation; (3) There is a lack of dedicated low-power ML accelerator for particular applications in edge computing, e.g. generative adversarial network (GAN), due to the tremendous challenges on resource-limited edge devices; (4) The efficiency of the existing CIM designs is limited by the low bit-precision. To overcome these challenges, the solutions are introduced as: (a) An efficient and high-throughput time-domain pipelined architecture is presented to accelerate dynamic time warping (DTW) algorithm; (b) Through significant architecture improvement and hardware adaptation, a low-power mixed-signal GAN accelerator is developed on edge device with 8-bit resolution; (c) A high-efficient time-based CIM design is implemented, which utilizes the analog-sparsity-based low-power methods, a compute-adaptive ADC skipping, and a special weight shifting technique.

1.2.3 Design Automation in Time Domain

Despite of many existing demonstrations of highly efficient operation using MSTC [9-22], most of existing work for time-domain computing is based on analog/mixed-signal design flow, which requires significant manual design and layout effort. This is partially due to the stringent timing control requirement of the technology leading to the difficulty of adoption into a large-scale design. Hence, it is important to develop a comprehensive design methodology for the automatic synthesis and place&route for MSTC. To address such a growing demand and deliver the missing design automation element, a design automation flow for MSTC is developed.

1.3 Summary of Contributions

The rest of the thesis is organized as follows. Chapter 2 introduces the background of mixed-signal time-domain computing (MSTC) including the related prior work on multiple interesting MSTC applications. Chapter 3 presents novel time-domain circuit techniques [24], including: (1) double-encoding strategy; (2) bit-scalable design that accelerates the performance compared to previous linear coding; and (3) shared digital-to-time converter/time encoder with variation-aware design technique which significantly improves the error tolerance of MSTC. Chapter 4 presents a time-based feature-extraction processor used for real-time image recognition [25, 26]. Complex operations like median filter and winter-take-all are carried out in time domain to improve the efficiency in image classification. Chapter 5 presents a general-purpose dynamic time warping (DTW) engine for time-series classification using time-domain computing [27]. Chapter 6 presents a low-cost mixed-signal time-domain accelerator for generative adversarial network (GAN) [28]. Chapter 7 introduces a dynamic analog RAM (DARAM) based Computing-In-Memory macro and associated CNN accelerator. Chapter 8

proposes a comprehensive design automation flow for MSTC [29]. Chapter 9 concludes this thesis and introduces the future research plan.

The contributions of this thesis are summarized as below:

- In Chapter 3, a design methodology for energy efficient time domain signal processing is proposed. To understand the strength and weakness of the technique, the impacts of process variation are studied and modeled with a variation driven design strategy. Such a design achieves an optimal tradeoff between energy and robustness. Several key circuit techniques such as dual-encoding scheme and bit-scalable design are proposed to further improve the design efficiency. The test results show a 3.3× improvement in energy-delay product and a 34% area reduction compared with conventional design.
- In Chapter 4, a MSTC accelerated image process is developed based on the feature-extraction algorithm. This work proposes a series of highly efficient time-domain computing techniques including shared time generator/DTC, double-edge operation scheme, bit-split technique and high-efficient time-domain nonlinear operations. MSTC-based accelerators are used to remove the bottlenecks of the algorithm, i.e. median filter (MF) and winner-take-all (WTA), rendering significant speedup. A total of 24% to 42% area saving is observed in MF and WTA accelerators compared to ASIC implementation. A 1.7× speedup and 20% to 23% power saving are also observed using MSTC. The overall image recognition processor operates at 1.33GHz with a throughput of 72 frames per second (fps).
- In Chapter 5, a general-purpose DTW engine using time-domain computing is designed for time-series classification. The first pipeline architecture in time domain is presented. A special time-domain storage cell, namely time flip-flop, has been developed with extendable

ring-based structure and embedded accumulation functionality. The developed DTW engine also allows high-throughput pipelined data flow and unfolded operation for longer time series through a specially designed pipeline architecture. The measurement shows a throughput improvement of more than 9× compared to prior works. A post-silicon calibration scheme is also incorporated to reduce the impact from process variation leading to 3× reduction of distance measurement error.

- In Chapter 6, a low-cost mixed-signal time-domain accelerator is developed to accelerate generative adversarial network (GAN). A significant reduction in hardware cost is achieved through delicate architecture optimization for 8-bit GAN training on edge devices. As a result, the total training time of MNIST database takes only 4.5 minutes which is 82× less than a high-performance CPU (2.6GHz Intel i7 Quad-core with a power of 197W). As most of the existing AMS designs suffer from low throughput, this work achieves the highest throughput of 18~5400× [15, 16, 17] with similar efficiency.
- In Chapter 7, a 3T Dynamic-Analog-RAM-Based Computing-in-Memory Macro and CNN Accelerator is presented to accelerate CNN inference workload. A special ADC skipping scheme brings 65% saving of ADC energy with less than 0.4% accuracy degradation. Combining all the sparsity features, the macro power was reduced by 2.1× on average under VGG16 model. Compared to the closest system implementation in [18], an 8× system energy efficiency improvement at 44.7TOPS/W is achieved along with 3× area reduction in macro size. Overall, this work achieves a state-of-art macro efficiency of 217TOPS/W at 4 bits, which is 3× improvement from prior work and is only 32% lower than that reported in a recent 7nm technology. In addition, an effective bit cell size of only 75% of 6T foundry SRAM cell is achieved.

- In Chapter 8, a comprehensive digital compatible design flow including frontend synthesis and backend placement for MSTC is presented. In the synthesis stage, the proposed technique can handle the variation requirement while minimizing the estimated area of the circuit. During the backend stage, an ACG-based placement algorithm is proposed to handle the complex placement constraints for MSTC design. The comparison to the previous analog placement scheme shows 4× matching improvement from the proposed method.

Chapter 2

Background

2.1 Mixed-Signal Time-Domain Computing

The basic concept of mixed-signal time-domain computing (MSTC) is to represent data in the format of delay or length of time pulses and process the information in time domain with special mixed-signal circuits. As shown in Figure 2.1, digital information, e.g. digit 2 and 5, are converted into time domain represented by time pulses. Afterwards, computations in time domain are realized by manipulating the pulse width of time pulses utilizing time-domain logic circuits, e.g. time_domain(2+5) as depicted in Figure 2.1. In the end, the time-domain results are converted back to digital domain for further operation, e.g. convert time pulsed to digit 7 and 3.

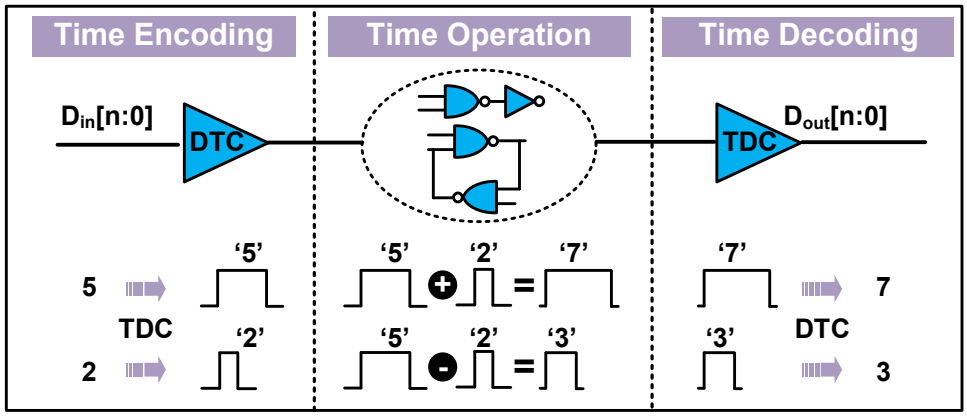


Figure 2.1 Concept of mixed-signal time-domain computing.

A digital-to-time converter (DTC) is used to convert digital information into time domain. Correspondingly, time-to-digital converter (TDC) carries the job to convert time-domain information back into digital domain. More technical details of DTC and TDC designs are presented in Chapter 3. Most of the arithmetic and Boolean operations are supported in time

domain such as, addition, multiplication, accumulation, shift, max, min, comparison, etc. More complex operations/algorithms can be implemented based on these building arithmetic operations such as, multiply-accumulate (MAC), median filter (MF), winner-take-all (WTA), etc. As a result, many emerging applications with complex algorithms can be conducted in time domain efficiently, e.g. image recognition [25, 26], machine learning applications [28, 29], etc. Memory related operations such as load and store can also be realized in time domain by utilizing special time flip flop design resulting in the first high-throughput time-domain pipeline architecture [27]. Moreover, MSTC is by nature a good candidate for near sensor computing as the overhead of analog-to-digital conversion can be mitigated by using time-domain techniques. Table 2.1 summarizes the implemented operations in time domain.

TABLE 2.1 LIST OF OPERATIONS IMPLEMENTED IN TIME DOMAIN

Operation	Type	Supported in MSTC
addition	basic arith.	Yes
subtraction	basic arith.	Yes
multiplication	basic arith.	Yes
division	basic arith.	Partially
modulus	basic arith.	Yes
increment	basic arith.	Yes
decrement	basic arith.	Yes
shift right	basic arith.	Yes
shift left	basic arith.	Yes
comparison	Boolean op.	Yes
maximum	Boolean op.	Yes
minimum	Boolean op.	Yes
store	pipeline op.	Yes
load	pipeline op.	Yes
MAC	complex op.	Yes
MF	complex op.	Yes
WTA	complex op.	Yes
CNN	ML algori.	Yes
FCN	ML algori.	Yes
GAN	ML algori.	Yes

2.2 Related Work

2.2.1 Prior Work in Analog Computing

There has been a growing interest in analog computing which utilizes non-Boolean analog voltage or physical resistance for computing. For instance, a digital-analog hybrid neural network explored the efficient analog computation and digital intra-network communication for feature extraction and classification with 7.5× energy efficiency compared to equivalent digital design [4]. A switched capacitor based analog matrix multiplication design was proposed to perform MAC operations efficiently for machine learning tasks with similar accuracy compared to digital counterpart [5]. A memristor- or RRAM-based computing explores the voltage, current and resistance relationship to achieve much higher energy efficiency on multiplier-accumulator (MAC) operations for deep neural network (DNN) applications [6]. In addition, a spike timing dependent plasticity (STDP) inspired wavefront recording scheme is implemented to capture incoming wavefronts in [30].

2.2.2 Prior Work in Time-Domain Computing

Several demonstrations have been developed in recent years using MSTC for realizing emerging applications [11, 15, 16, 17, 31, 14]. For instance, a time-domain low-density parity-check (LDPC) design was demonstrated with 2× reduction in area compared to the digital implementation [11]. A swarm robotic system incorporating a time-domain reinforcement learning accelerator was implemented with over 30% saving of energy compared to the digital counterpart [15, 14]. A time-domain CNN engine showed 12× improvement for energy efficiency compared to the other state-of-art digital implementations [16]. A high-efficient time-

based in-memory computing graph engine was realized using wave-front expansion and 2D gradient control for solving single-source shortest path problems [17].

2.2.3 Prior Work in DNN Accelerator and Compute-In-Memory Design

The technology trends of big data, social networks, and autonomous driving bring high volume of data for processing and high demands on computing devices. As a result, many new markets have grown significantly justifying the cost of special purpose accelerator chips with examples of tensor processing unit (TPU) from Google [32], AWS Inferential chip from Amazon [33], and self-driving AI chip from Tesla [34]. On one hand, the digital implementations accelerate the machine learning (ML) workloads by utilizing novel dataflow architecture, e.g. systolic array, and data-parallel etc. On the other hand, analog-based Computing-In-Memory (CIM) has drawn significant attention recently as it shows significant advantages in computation efficiency for ML applications. Several interesting designs based on SRAM bit-cell show significant area/energy improvement over CNN tasks [18, 19, 20, 21, 22, 23]. While earlier CIM macro was limited by lower bit precision, e.g. binary weight in [18], recent works have shown 4 to 8 bit-precision for the weights/inputs and up to 20bits for the output values [19, 20]. Sparsity and application features have also been exploited at system level to further improve the computation efficiency [21, 22, 23].

2.2.4 Prior Work in Design Automation for Analog Computing

Design automation for mixed-signal analog computing has been developed over decades. Although automatic placement has been proposed previously for analog/mixed-signal design [35, 36], MSTC poses special challenges, i.e. massive-stage-symmetry (MASS), and hence requires

special techniques not available from the prior work. Topological representations are widely used in solving analog placement problems (in back-end), in which the relative positions between the modules are encoded. Typical topological representations are slicing tree [37], sequence-pairs (SP) [38], O-tree [39], B*-trees [40], and TCG-S [41]. Most of these works have been applied to handle the symmetry constraint and other constraints like the centroid constraint.

Chapter 3

Energy-Efficient Mixed-Signal Time-Domain Circuit Design Methodology

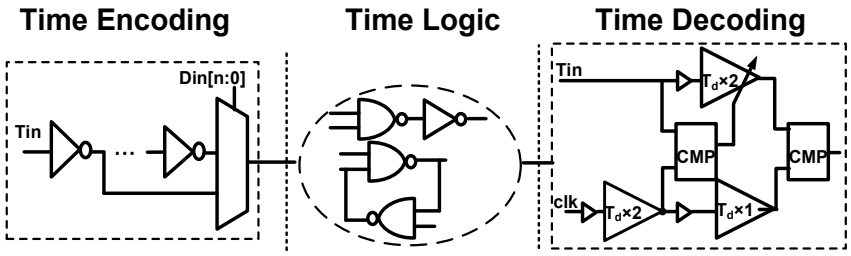
In this chapter, the operating principle and design methodology of energy efficient MSTC is systematically presented. Variation impact of MSTC is evaluated and a variation driven design methodology is proposed. Several novel circuit level-design techniques including double encoding strategy and bit-scalable schemes are proposed, which significantly improve the area and energy efficiency of MSTC.

The reset of Chapter 3 is organized as follows: the basic MSTC design methodology along with circuit implementation are introduced in Chapter 3.1. An energy-efficient design methodology including double-edge operation and bit-split scheme, is presented in Chapter 3.2. A variation-aware design methodology for MSTC is proposed in Chapter 3.3. Chapter 3.4 concludes this chapter.

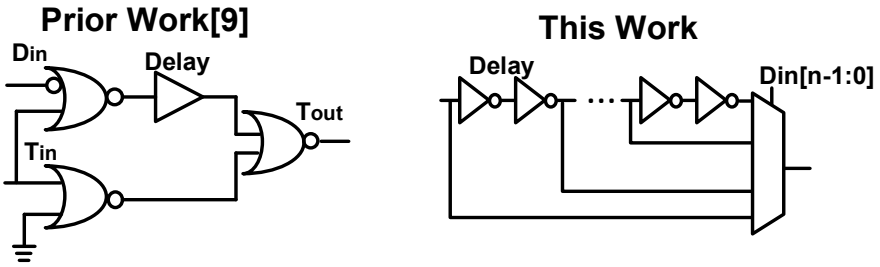
3.1 Mixed-Signal Time-Domain Computing Circuit Design Overview

3.1.1 Mixed-Signal Time-Domain Computing Building Modules

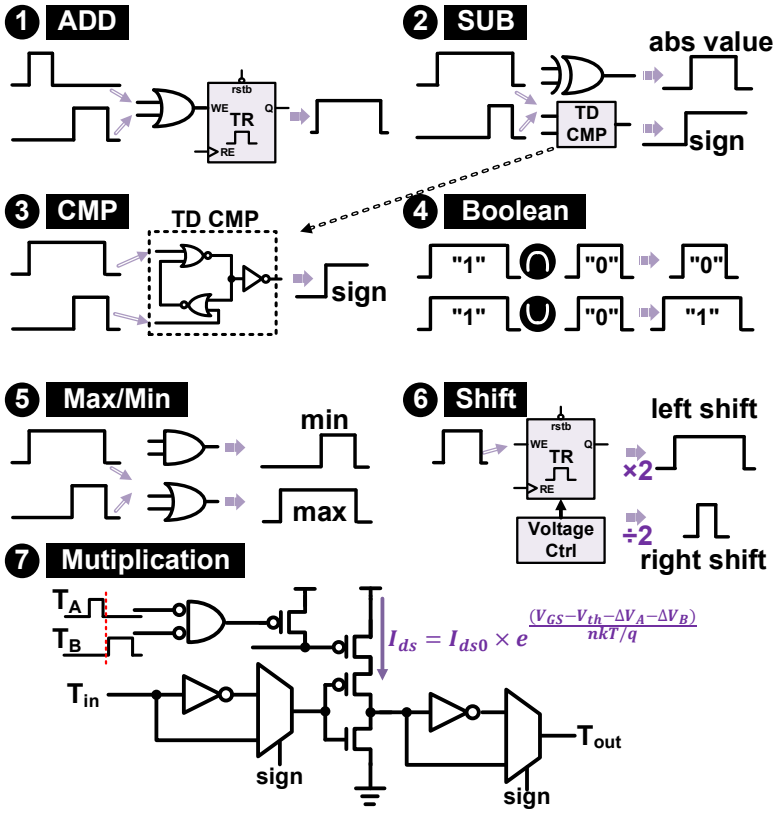
Mixed-Signal Time-Domain Computing (MSTC), or also referred as time-domain computing (TC), converts the task of signal processing into “time” or delay of digital cells which can be processed efficiently for numerous operations. The digital binary inputs are first encoded and processed in time domain and either reconverted back into digital domain through time-to-digital converter (DTC) or results are directly obtained at time domain without time decoding.


Time Encoding
Time Logic
Time Decoding
Din[n:0]
Tin
Tin
Td×2
CMP
CMP
clk
Td×2
Td×1

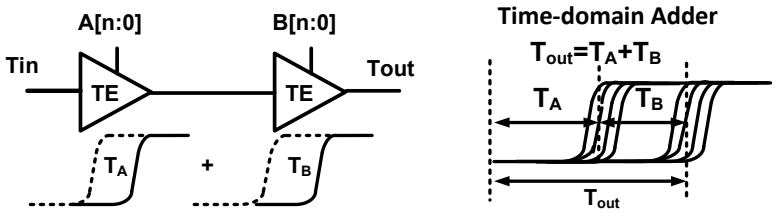
(a)


Prior Work[9]
Din
Delay
Tout
Tin
This Work
Delay
Din[n-1:0]

(b)


1 ADD
rstb
WE
TR
RE
2 SUB
abs value
TD CMP
sign
3 CMP
TD CMP
sign
4 Boolean
"1"
"0"
"0"
"1"
"0"
"1"
5 Max/Min
min
max
6 Shift
left shift
rstb
WE
TR
RE
×2
Voltage Ctrl
÷2
right shift
7 Mutiplication
TA
TB
Tin
sign
Tout
sign

(c)



(d)

Figure 3.1 (a) System overview of time-domain computing; (b) Time encoder circuit: prior work [11] (left), proposed DTC in this work (right); (c) Time-domain operation circuits; (d) An example of n-bit MSTC adder: circuit schematic (left), waveform (right).

Figure 3.1 (a) shows an overall setup of the MSTC which consists of key processing stages: (1) time encoding, (2) information processing in time domain, and (3) time decoding. Here, T_{in} is the time-domain input signal, D_{in} is the digital input signal which used to determine the time delay of generated time-domain signal from digital-to-time converter (DTC) or time encoder (TE), and T_d is referred as the time delay of a single delay cell, e.g. buffer.

3.1.2 Time Encoder (TE) or Digital-to-Time Converter (DTC)

Figure 3.1 (b) shows 1-bit DTC from prior work (left) [11] and the proposed n-bit DTC (right) in this work. Our proposed DTC has a simple inverter chain to generate the delay passed through selection multiplexer to convert from digital input to time-domain signal. Compared with the design in LDPC work [11], our design has benefits in area, energy and robustness.

3.1.3 Energy- and Area-Efficient Time-Domain Operations

Figure 3.1 (c) summarizes the schematic of basic time-domain operation/logic cells that process the signal in time-domain once the information is encoded into time domain from DTC. Compared to conventional CMOS logic design, many operations can be performed much more efficiently. For example, the MAX, MIN and Compare (CMP) operations are realized by a single

or two logic gates leading to tremendous saving from conventional CMOS operation. Note that, the symmetric output load constraint is critical to the CMP circuit. In addition, some more complex operations, e.g. shift and multiplication, are also implemented efficiently in time domain. More details of such time-domain operations will be introduced in Chapter 5, 6, and 7.

3.1.4 Time-digital-converter

Time decoder (TD), or time-digital-converter (TDC) has been extensively developed in all-digital phase-locked-loop (ADPLL) design with the state-of-art TDC achieving 1*ps* resolution [42, 43]. The right-hand side of Figure 3.1 (a) shows a 2-bit base-line TDC design based on binary-search. However, due to the high energy and area cost of such a TDC, in this work, we develop algorithms that eliminate the use of TDC leading to much improved area and energy efficiency.

3.1.5 Benefits of MSTC

The benefits of MSTC come from the following interesting facts: (1) similar as analog computing, multiple bits can be encoded into single transition leading more efficient information delivery. An example is the adder circuit as shown in Figure 3.1 (d) where the n-bit operation only consumes transitions of a few inverters rendering 3× improvement in energy efficiency; Here, A[n:0] and B[n:0] are the digital control signal of the DTCs; T_A and T_B are the corresponding time-domain signals. (2) Some logic operation can be efficiently carried (3) Owing to the analog nature of operation, the MSTC is intrinsically immune to large magnitude of error. In other word, compared to digital design, the analog-based design has much less chance to have the error happen at the most-significant bit (MSB). Since the computation output is more

affected by MSB errors in applications like facial recognition, MSTC is more error tolerant compared with digital counterpart when error occurs; (4) Although the information is processed in time-domain, the information carriers are still binary digital signals processed by conventional logic circuits which makes the entire design digital-friendly.

3.2 Energy-Efficient Design Methodology in MSTC

3.2.1 Double-edge Operation

In this work, we propose a double-edge operation, where logic operation is processed at both rising and falling transitions as shown in Figure 3.2 (a). The energy taken from source to charge the gate capacitance is as $E = V * Q = V * V * C = CV^2$. Half of the energy is dissipated during rising transition; the other half is dissipated during falling transition.

In previous design [9], only single transition is used. Thus, the other half is purely a waste. In our proposed design, we utilize both transitions, which provides us with 2× energy efficiency. Area consumption is also reduced by around 30% because the buffer stage is shared for both rising and falling transitions. Figure 3.2 (d) shows the energy and area saving come from the double-encoding scheme. Figure 3.2 (b) and (c) shows the logic encoding concepts between conventional complementary logic design where pull-up and pull-down realize complementary logic functions and the dual-encoding strategy for MSTC.

Interestingly, in MSTC design, for rising and falling operation, the design could perform two totally different logic operations as compared with conventional design where complementary operations have to be performed.

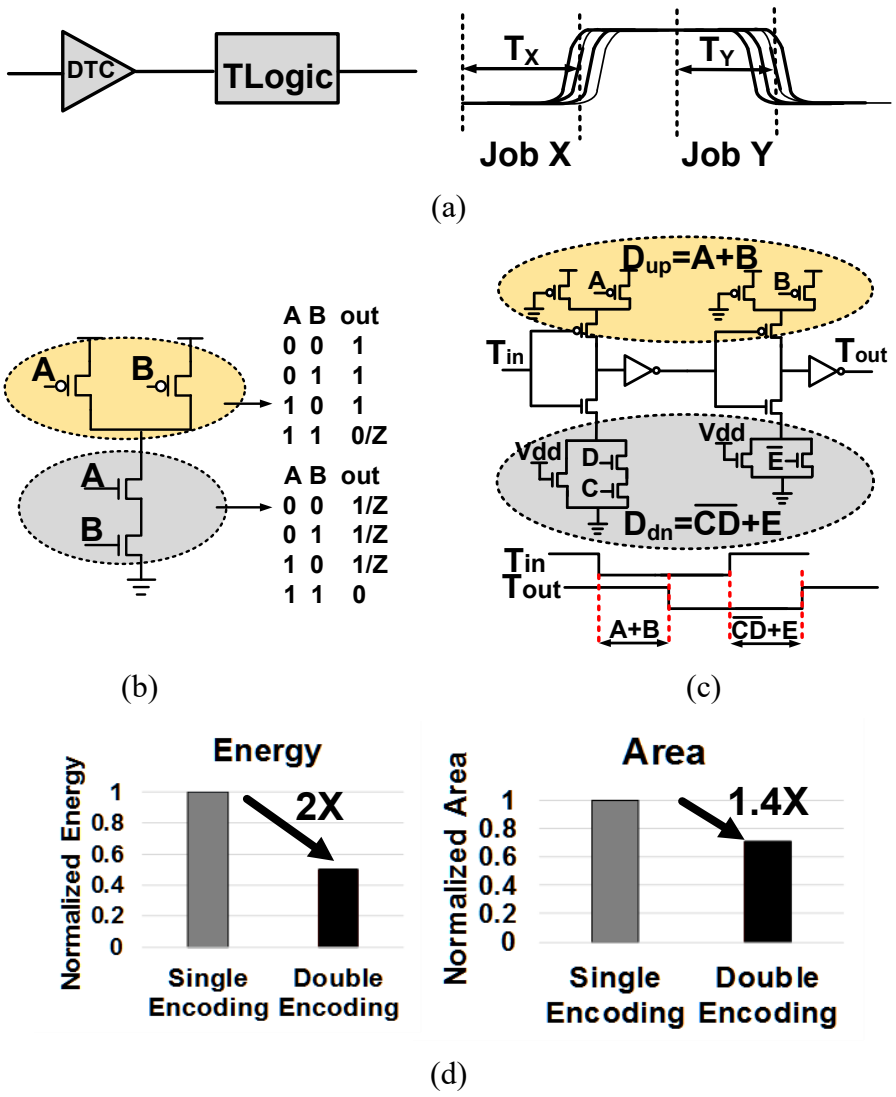


Figure 3.2 Proposed double-edge operation: (a) overview, (b) conventional digital complementary logic, (c) Two 1-bit adders using dual-encoding scheme with different logic operations for rising and falling transitions, (d) energy and area comparison between single

As shown in Figure 3.2 (c), (1) during the falling edge of time-domain input signal Tin, the pull-up part of the circuit is turned on, which processes $D_{up} = A + B$; (2) During the rising edge of Tin, the pull-down part of the circuit is turned on, which processes $D_{dn} = \overline{CD} + E$. This means that theoretically MSTC could realize more functionality with the same amount of pull-up and pull-down logic circuits as compared with conventional digital design. Simulation shows

that the pull-up and pull-down operation can be completed decoupled without delay impact to each other as long as the input slew rate is much faster than the encoded 1-bit delay, which is guaranteed by adding inter-stage buffers.

3.2.2 Bit-Split Scheme

As previous work shows only limited bit precision, e.g. 3 bits [11, 12], we propose a bit-split technique that splits an input vector into smaller bit groups leading to a scalable high-resolution encoding without exponentially increasing the delay. As shown in Figure 3.3 (a), 8-bit inputs A and B are split into 2 sub-groups, e.g. A[7:4] (referred as MSBs) and A[3:0] (referred as LSBs). In this work we encode 4-bit MSBs operation in the falling-edge and 4-bit LSBs in the rising-edge rending 16× reduction of delay and 2-4× reduction of energy on a 8-bit non-linear operation, e.g. MIN(MAX(A, B), (C+D)) (Figure 3.3 (b)). This technique also makes MSTC designs scalable with the number of bits since large number of bits can be split into several small groups. To allow the split of the bits, a digital combination logic has to be added to combine the decision from each sub-group. This incurs digital equal comparison to deal with the situation that MSBs are equal. An example of the combine algorithm is shown in Figure 3.3 (c): the comparison between A and B goes through the following steps: (1) check if A's 4-bit LSBs equals to B's, if no, go to step (2), else go to step (3); (2) Check weather A's 4-bit MSBs is larger than B's; (3) Check weather A's 4-bit LSBs is larger than B's. Then based on the value of these conditions to determine the mathematical relation between A and B.

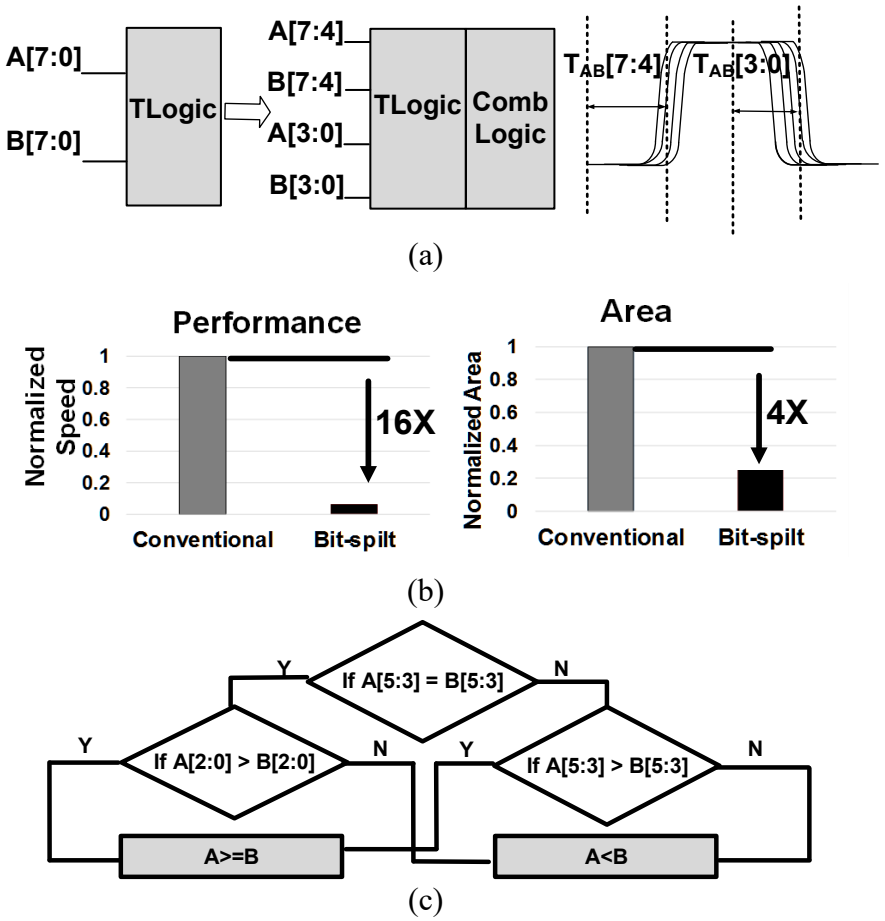


Figure 3.3 Proposed bit-split technique: (a) technique overview, (b) performance and energy saving come from the bit-split scheme, (c) combine algorithm.

3.3 Variation-Aware Design Methodology in MSTC

3.3.1 Overview of Variation-Aware Design Methodology

Figure 3.4 shows the overall flow of our proposed variation-awareness design methodology. The flow includes two steps: (1) initial circuit generation which first convert the target algorithm/application into time-domain algorithm and then map the time-domain algorithm into time-domain logic cells; (2) Resizing the time-domain logic cells and adjusting

the single-bit delay (resolution) to meet the design specification, e.g. variation and performance target. More specifically, (a) the capacitive loads of the time-domain circuits determine the delay of the circuits, e.g. time encoder; (b) The larger size of the transistor, the smaller variation it has but the capacitive load/energy consumption also increases. Thus, the size of time-domain logic cells must be carefully selected to satisfy the design specification. Meanwhile the time-domain logic cells are characterized based on Monte-Carlo simulation.

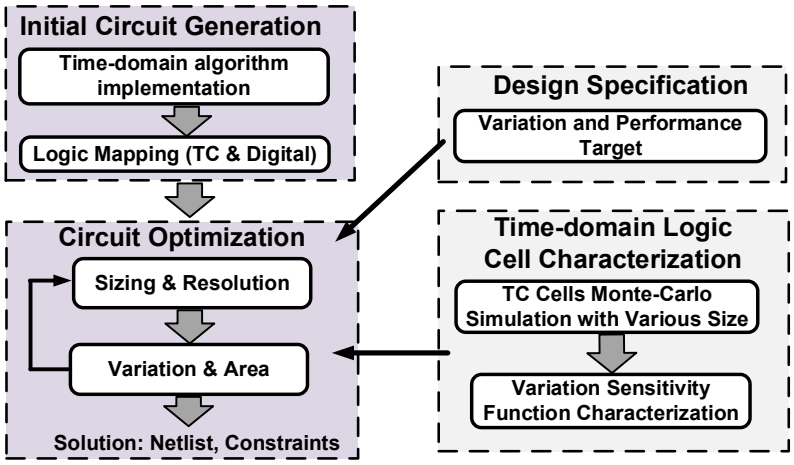
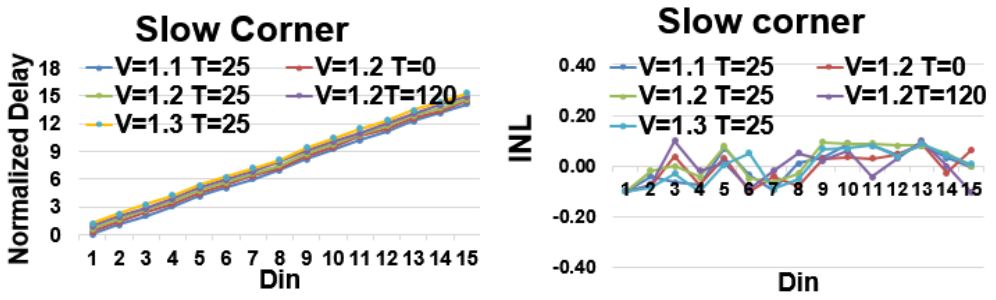


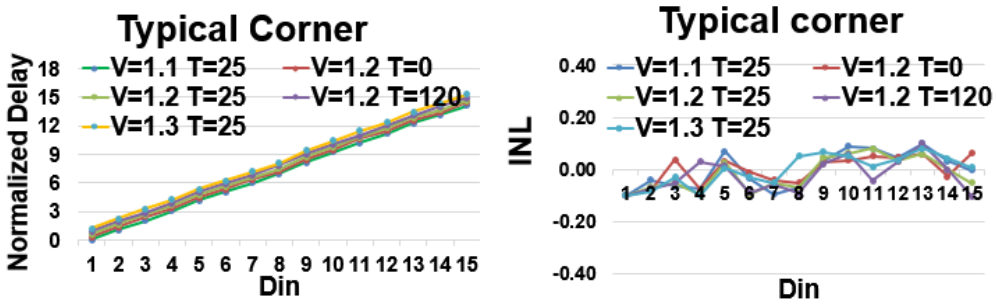
Figure 3.4 Variation-awareness design flow of MSTC.

3.3.2 Global Variation VS. Local Variation

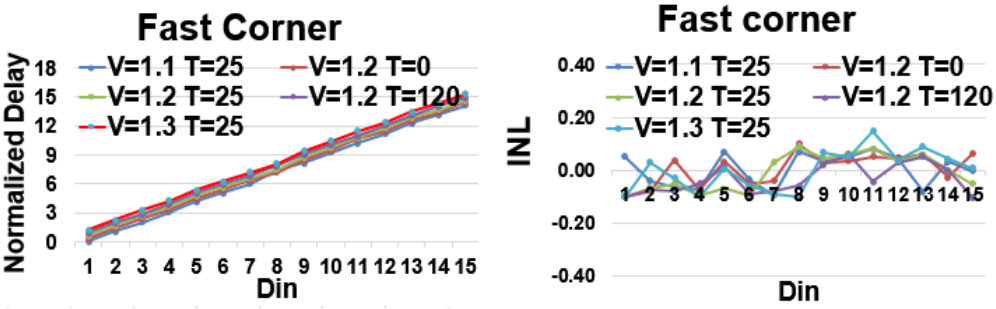
As TE/DTC holds the most stringent requirement on the timing control accuracy, we performed the global process-voltage-temperature (PVT) variation analysis to a 4-bit DTC in a 55nm technology using the base-line 1-bit DTC.



(a)



(b)



(c)

Figure 3.5 Normalized delay across PVT corners and INL of (a) slow corner, (b) typical corner, (c) fast corner.

Since relative delay among signals is most critical to keep the computation error-free or at low error rate, linearity matters the most compared with the absolute delay values. As shown in the left side of Figure 3.5 (a), (b), and (c), the linearity is well preserved across PVT corners. The right side of Figure 3.5 (a), (b), and (c) show the integral nonlinearity (INL) of the time encoder where the integrated nonlinearity is represented as a fraction of least-significant bit (LSB). The INL variation of the time encoder is well maintained within 15% of LSB across PVT corners.

Hence, through a proper budgeting of variation, the global PVT variation does not introduce significant concern to the functionality of the MSTC design.

On the other hand, local variation/mismatch poses more challenges to the MSTC design compared with global variation due to the linearity requirement of time encoder. Monte-Carlos simulation with random threshold voltage variation is performed to evaluate the impact of local mismatch.

3.3.3 Shared TDC/TE Design

The digital-to-time (DTC) converter poses the most stringent variation requirement on the timing control accuracy as the generated signal experiences multiple stages of variation impact during signal generation stage. For example, for a 4-bit DTC, the longest delay generated by such a DTC sums up the variation of 15-stage inverter chain. Assume the variation (also referred as the standard deviation) of a single stage inverter is σ_{inv}, the variation V_{ind-TE} for a n-bit individual DTC (Figure 3.6 (a)) in the worst scenario is shown in (3.1):

$$V_{indiv-DTC} = \sqrt{(2^n - 1)\sigma_{inv}{}^2 + n\sigma_{inv}{}^2} = \sigma_{inv}\sqrt{(2^n - 1) + n} \tag{3.1}$$

The term, $(2^n - 1)\sigma_{inv}{}^2$,represents the variation comes from $(2^n - 1)$-stage inverter chain when input is set to maximum value. The other term, $n\sigma_{inv}{}^2$, comes from the n-stage multiplexer at the end of the inverter chain. Relative delay among signals is most critical, we derive the mismatch between two individual TDCs in (3.2):

$$V_{indiv-DTC-CMP} = \sigma_{inv}\sqrt{2(2^n - 1) + 2n} \tag{3.2}$$

The mismatch between two individual DTCs/TEs increases dramatically when the number of bits increases. In order to relieve the variation concern comes from the time encoder, we

proposed the shared time generator (TG) scheme which uses a common inverter chain to generate timing signals relieving the variation impact.

$$V_{TG-CMP} = \sigma_{inv}\sqrt{2n} \tag{3.3}$$

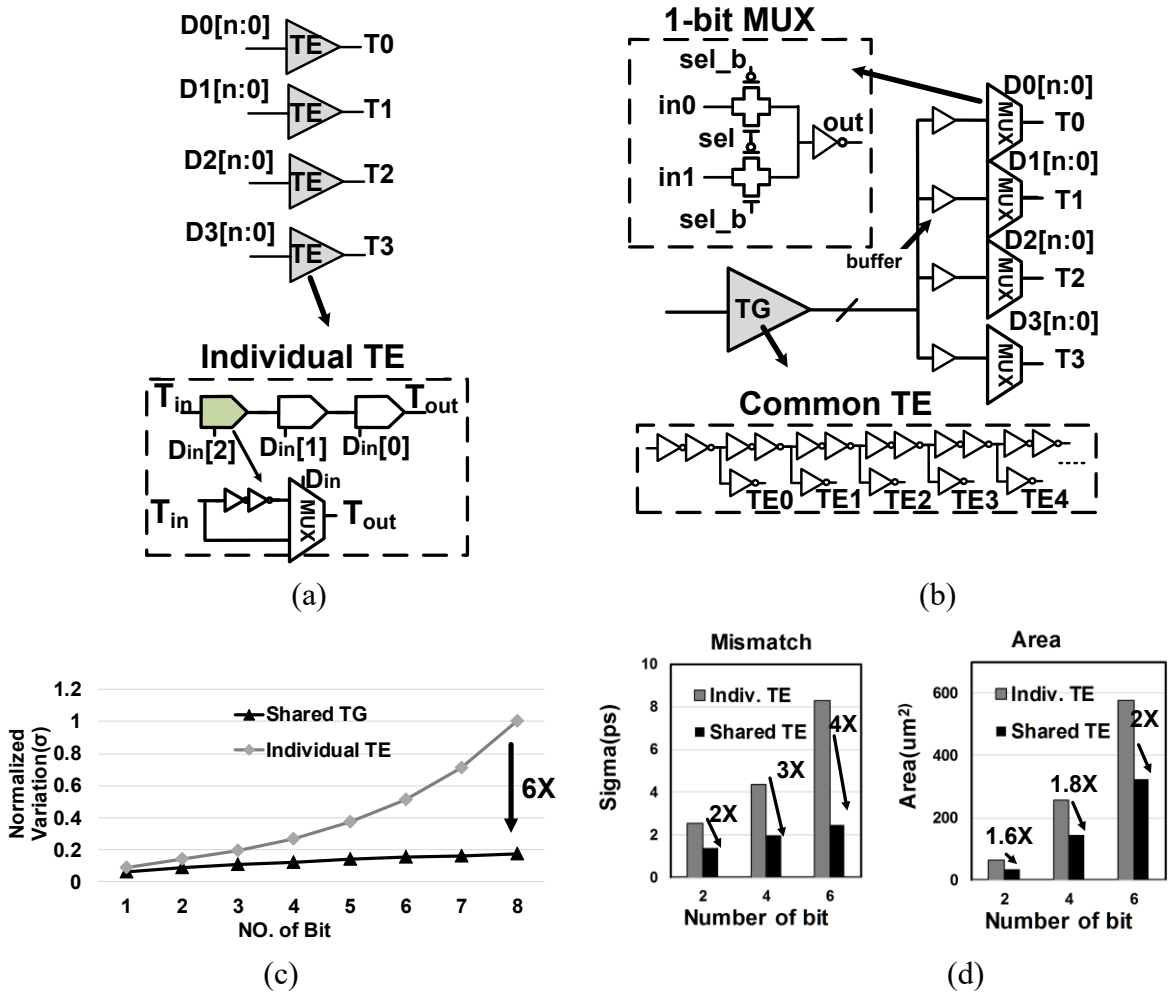


Figure 3.6 (a) Circuit schematic of individual DTC scheme [11], (b) proposed shared time encoder scheme, (c) variation comparison between shared TG and individual TE/DTC, (d) mismatch and energy comparison between induvial and shared TE/DTC (8-output).

Figure 3.6 (b) shows the circuit schematic of the shared time generator which consists of (1) a common inverter chain used to generate the all possible delay, (2) distributing multiplexers which are used to select the desired delay for each output. When we consider the mismatch of

two different output from TG, the worst scenario is when the delay difference of two output is just single-bit delay away from each other. Since all the output are generated by the common inverter, the variation comes only from the distributing multiplexers eliminating the mismatch source from inverter chains rendering much improved variation resiliency. The mismatch between two outputs from TG can be represented by (3.3):

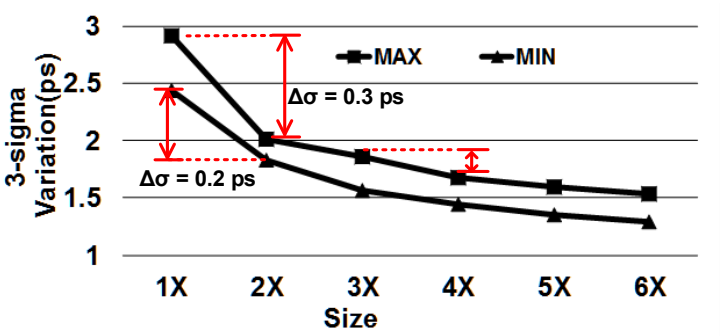
Compared with individual DTC design whose mismatch in a n-bit DTC is proportional to $\sqrt{(2^n - 1) + n}$, the shared TG has mismatch only proportional to $\sqrt{n}$ leading to 3~4× less mismatch rendering shortened single bit delay and smaller cell size. Figure 3.6 (c) shows the variation trend of shared TG and individual DTC cases. As we can see, the variation of individual DTC case grows dramatically compared with the shared TG case. Besides, the area consumption of TG is also smaller due to the sharing of common inverter chain. Figure 3.6 (d) shows the mismatch improvement and area saving come from proposed common TG design. Note that, the more paths/operations that share the same TG, the more complexity of distributing network will be needed, e.g. tree-structure buffers and distributing MUXs. In our case, it is a clear win for using shared TG. But it is possible that the shared TG becomes too expensive if too many signals are generated. In that case, the design needs to be performed towards using individual DTC.

3.3.4 Variation-Awareness Design

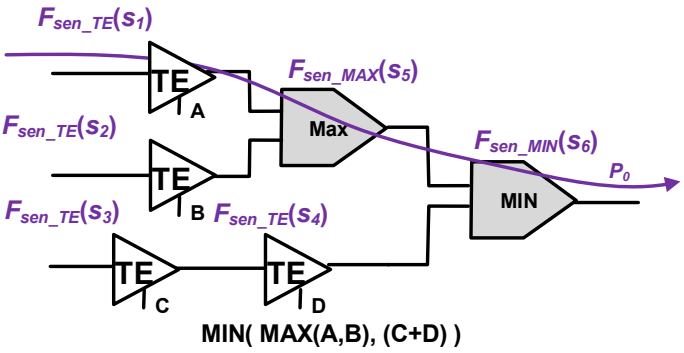
In this section, we introduce the variation-awareness design flow. We define the 3-sigma variation of MSTC modules, which is a function of the size s as $\sigma(s)$. Apparently, the $\sigma(s)$ decreased as s increases. Also, the area is a function of the size s as $A(s)$. The sensitivity of variation of a module can be defined as (3.4):

$$F_{sen}(s) = \gamma \frac{d\sigma(s)}{dA(s)} \tag{3.4}$$

Where $\frac{d\sigma(s)}{dA(s)}$ term represents the variation sensitivity comes from module itself without considering the whole placement topology. The γ term represents the significance of the module, e.g. module in a convergent path.



(a)



(b)

Figure 3.7 (a) Variation of MAX (NAND2) and MIN (NOR2) at various sizes; (b) Example schematic.

Figure 3.7 (a) shows the variation-area(size) curve of MAX (NAND) and MIN (NOR) gates. Since most time-domain cells are standard-cell like, we follow standard cell sizing convention of 1×, 2×, 3×, etc. An example of MSTC operation, MIN(MAX(A, B), (C+D)), is shown Figure 3.7 (b). Given a simple task of decreasing the variation of critical path P_0, it is obvious that we can gain more variation improvement if we give the sizing priority to the module

whose current variation sensitivity is larger than the rest. Based on this, we propose our netlist optimization in following.

Assume we have totally n modules, X_1, X_2, …X_n, the size of each module is s_1, s_2, … s_n. Besides, there are *P* paths need to be considered in the placement. The optimization problem of netlist is then formed in (3.5) and (3.6):

$$\text{Minimize } \sum_{i=1}^{n} A_i(s_i) \tag{3.5}$$

$$\forall\, paths \in P, s.t. \sqrt{\sum_{i=1}^{n} {\sigma_{pi}}^2(s_i)} \leq \sigma_T \tag{3.6}$$

where $\sigma_i(s_i)$ is the variation of Xi, and $A_i(s_i)$ is the module area of X_i. The optimization algorithm is described as follows:

Table 3.1 Variation-Awareness Optimization Algorithm

Algorithm 1 Variation-Awareness Optimization Algorithm

Input: Initial schematic/netlist of module $X_1, X_2, \ldots X_n$, with minimum sizing $s_1, s_2, \ldots s_n$.
Output: Netlist which satisfies variation budget with minimum estimated area

while $\forall\, paths \in P, \sqrt{\sum_{i=1}^{n} {\sigma_i}^2(s_i)} > \sigma_T$ **do**
  for *i* = 1 to *n* **do**
    find the module *j* = *i*,with maximum $F_{sen\_j}(s_j)$
  end
  Increase the size of module *j* by 1×, update s_j
end
Return the schematic/netlist with current sizing

Given the initial schematic/netlist with minimum size for each module, we first check if the variation of critical path meets the budget σ_T. If yes, the optimization is completed. Otherwise, the second step is performed where we traverse the netlist to find out the module in the critical path with maximum variation sensitivity with their current size. The size of module is then increased by 1×. In the following steps, we continue to check whether the variation budget is satisfied. If not, the optimization repeats by upsizing the most effective module, i.e. highest

sensitivity. The pseudo code is shown in Table 3.1 above.

3.4 Summary

This chapter proposes a design methodology for energy-efficient mixed-signal time-domain computing. To understand the strength and weakness of the technique, the impacts of process variation are studied and modeled with a variation driven design strategy proposed to achieve an optimal tradeoff between energy and robustness. Several key circuit techniques such as dual-encoding scheme and bit-scalable design are proposed to further improve the design efficiency.

Chapter 4

A Time-Domain Accelerated Image Recognition Processor Design

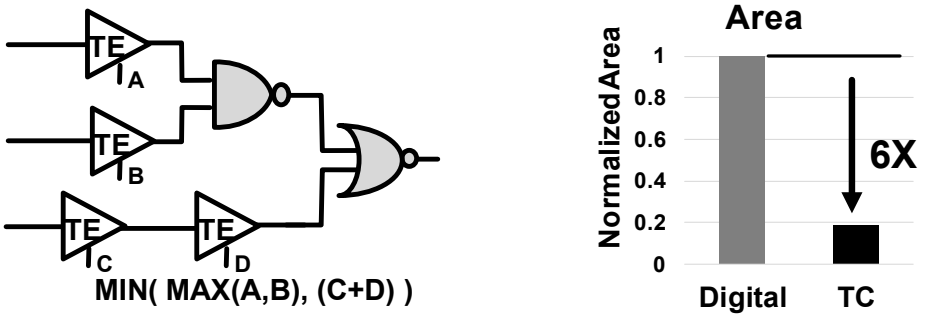
In this Chapter, a feature-extraction and vector-quantization processor accelerated by MSTC is developed to conduct real-time image recognition. A 55nm prototype chip shows 72 fps/core (@1.33 GHz) operation with up to 42% area and power saving from time-domain computing compared to the conventional digital implementation.

The reset of Chapter 4 is organized as follows: Chapter 4.1 exploits the complex MSTC non-linear operation and shows benefits in both area and energy consumption compared to the conventional digital counterparts. The MSTC-accelerated image processing engine is presented in Chapter 4.2. The silicon implementation with measurement results are presented in Chapter 4.3. The summary of this chapter is given in Chapter 4.4.

4.1 Exploiting Complex MSTC Non-Linear Operations

Many image processing applications such as pattern classification and facial recognition, require a large amount of non-linear signal processing operations, e.g. comparison, sorting, minimum, maximum, etc. [44, 45]. Among them, winner-take-all (WTA) and median filter (MF) are two of the most critical building blocks commonly used for pattern classification. In WTA and MF, a deterministic decision is made based on excessive compare and sorting operation which are highly expensive to be implemented in standard CMOS ASIC design and even more difficult for a CPU operation [46, 47]. As introduced in Chapter 3, many nonlinear operations can be efficiently implemented in MSTC. Figure 4.1 (a) shows the circuit diagram of MSTC implementation for operation MIN(MAX(A, B), (C+D)), which was introduced in previous

section. As depicted in Figure 4.1 (b), by using several NAND and NOR gates, such a complex operation can be easily implemented in MSTC rendering an energy saving of 6×. Therefore, we can improve the area efficiency by increasing the utilization of such efficient MSTC non-linear functions, e.g. MAX, CMP, in the design.



(a) (b)

Figure 4.1 (a) MSTC implementation of a non-linear operation; (b) Area comparison between digital and MSTC approaches.

4.1.1 Winner-Take-All

We derived our time-domain WTA algorithm from a binary comparison tree scheme which takes advantages of the efficient MAX/MIN and CMP operations in time domain (Figure 4.2).

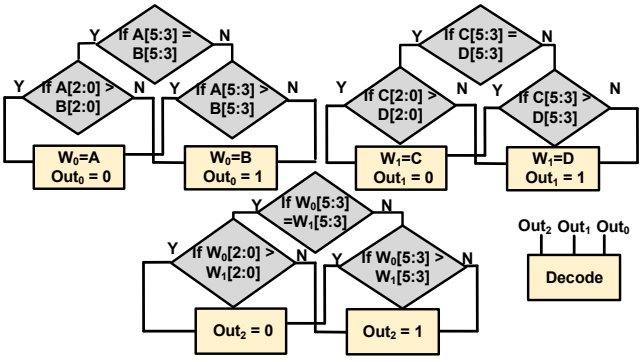


Figure 4.2 Algorithm of 6-bit 4-input WTA design with 3-bit split in MSBs and LSBs.

Figure 4.3 (a) shows the circuit implementation of the proposed 8-input 6-bit WTA accelerator. The winners or MIN value from each branch in each stage are selected in parallel

and propagated to the subsequent stage to be compared again. After converting the input digital value into time-domain, the comparison can be simply made by using time-domain CMP.

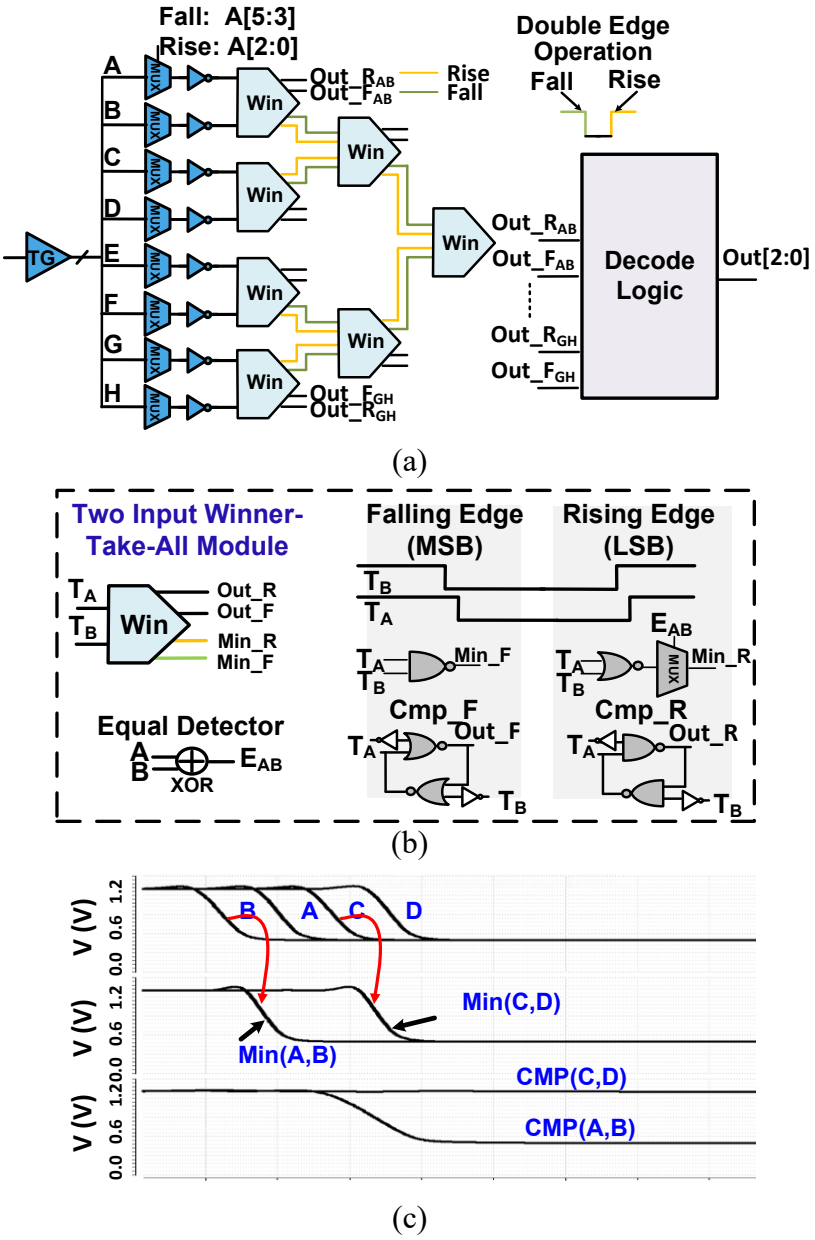


Figure 4.3 (a) Circuit design of WTA, (b) example of double-edge and bit-split technique, (c) example waveform of MIN function used in this design.

The MIN function which is built by a single NAND/NOR gate directly propagates the winner to next stage without intermediate restoration or regeneration. As a result, a massive

parallel operation with mostly NAND or NOR gates, is realized in MSTC. All comparison results are finally decoded in digital domain to find the final winner. Shared TE/DTC, double-edge operation and bit-split technique are also utilized in this design.

Figure 4.3 (b) shows the example of double-edge operation and bit-split techniques utilized in the WTA design: (1) the 6-bit input are divided into two groups as in[5:3] and in[2:0]; (2) in[5:3] (referred as MSBs) and in[2:0] (referred as LSBs) are encoded into falling and rising edge of the same clock cycle respectively; (3) During falling transition of signal, MSBs are processed while LSBs are processed during rising transition. Figure 4.3 (c) shows the example waveform of MIN function used in this design. As a result, the MSTC WTA achieves lower area consumption with faster speed. The area of proposed time-based WTA is improved by 42% compare to digital implementation as shown in Figure 4.4.

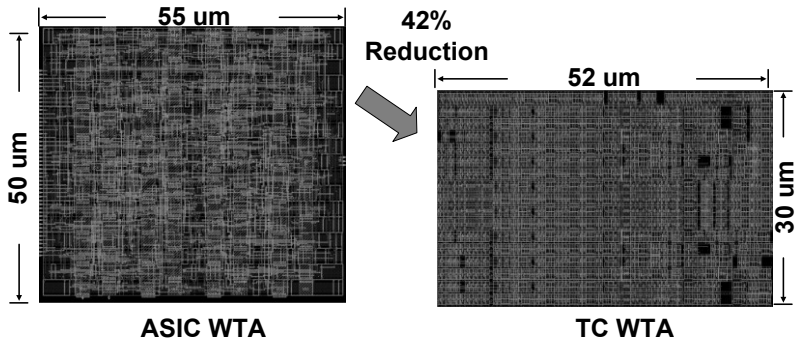


Figure 4.4 Layout comparison between (a) conventional digital WTA and (b) MSTC WTA.

4.1.2 Median Filter

As a core building block in applications such as facial recognition, median filter (MF) consumes up to 70% of total CPU cycles due to its enormous amount of CMP and swapping operation in a typical bubble sorting algorithm [48]. In order to remove the bottlenecks of the

application, several efficient MF algorithms have been proposed such as [49, 50]. The majority voting algorithm [50] improves energy efficient, but still suffers from the drawbacks of analog-based design such as cannot scale with technology which requires substantial amount of effort in tuning the circuit and designing the layout.

We propose an energy efficient time-based MF with high performance. The core idea of the algorithm is to have a massive comparison between each of the two inputs and order the inputs from high to low. The final median value is filtered/selected by the proposed decoder. Figure 4.5 shows the algorithm and detailed implementation of the proposed 12-input 8-bit time-domain median filter design as following steps: (1) each pair of the 12 inputs is compared, thus a total number of 66 comparisons are processed. The comparison result is recorded as "0" or "1", e.g. if A > B, OUTAB = 1, OUTBA = 0; (2) The related comparison results of each input are summed up, e.g. OUTA = OUTAB + OUTAC + … + OUTAL; (3) Finally, all the summation results are compared with N/2, where N is the number of inputs. The input whose summation result of the comparisons equals to N/2 is marked as the median value.

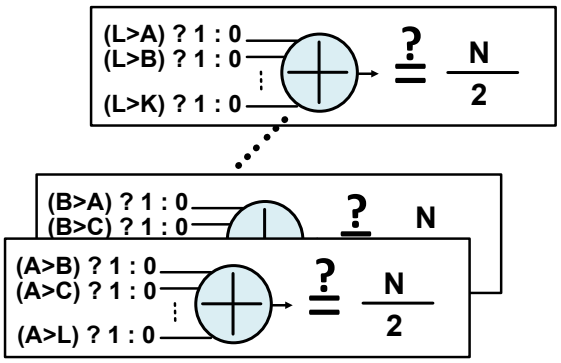


Figure 4.5 Algorithm of 12-input MF.

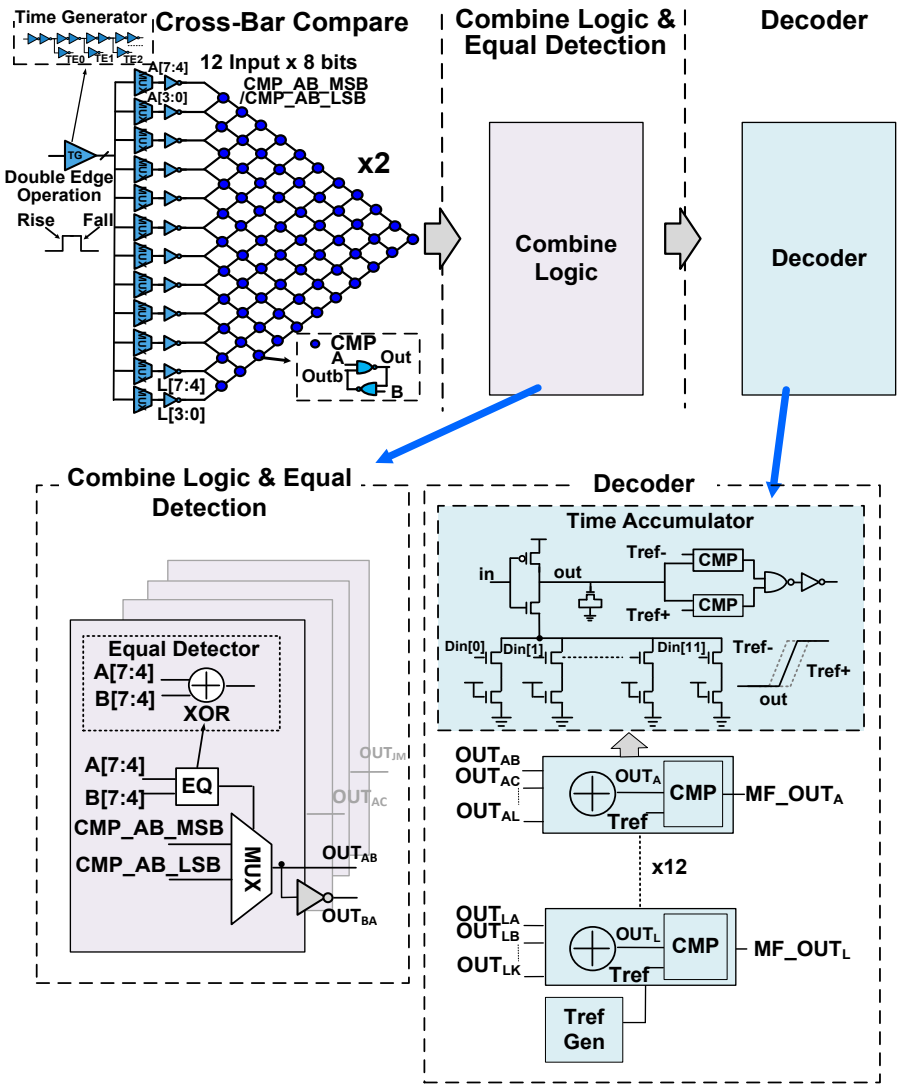


Figure 4.6 Circuit diagram of MF with detailed circuit schematic of combine logic & equal detection and the circuit detail of decoder.

Figure 4.6 shows the corresponding MSTD circuit diagram of MF. During the comparison stage, the digital inputs are first converted into time-domain by the proposed shared TE/DTCs. Each pair of input is compared in parallel in time-domain with double-edge and bit-split design, with overall 66 comparisons for both MSBs and LSBs for all 12 input vectors. The 66 comparisons are processed parallelly in the cross-bar compare module as shown in the left-hand side of Figure 4.6. During the combine & equal-detection stage, all the comparison results are

then processed by the following digital logic for purpose of equal detection and MSBs/LSBs grouping to obtain the comparison results for all input vectors. In the decoder stage, to decode the comparison results into final median value, a special time accumulator/adder design is implemented where all 11 digital comparison results from each input are summed in time-domain and compared with a reference median time-domain signal.

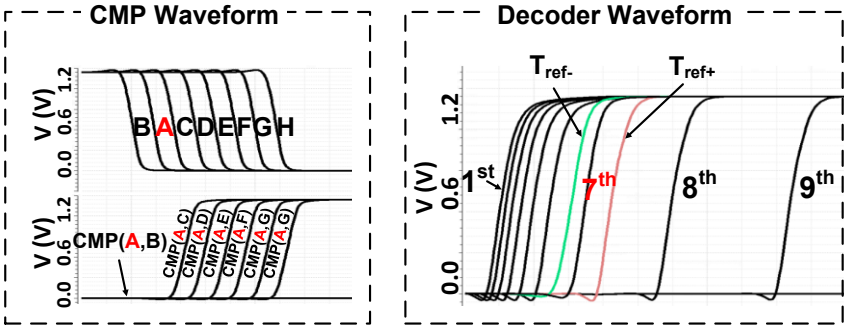


Figure 4.7 Example of waveform of MF operation.

Although the 66 comparison results can be further decoded into final result in conventional digital design, the decoding logic will incur large overhead due to the complex operations. To reduce the overhead, we proposed a time-accumulator based time-domain decoding logic. The bottom-left side of Figure 4.6 shows the detailed circuit implementation of the combine & equal detection stage. The bottom-right side of Figure 4.6 shows the detailed circuit implementation of decoder. The core idea of the time-domain decoder is to form a detection window by Tref- and T_{ref+}. As the 12 inputs are ordered and represented by the delay of the rising edge, the median value is carried by the 7th rising edge. The T_{ref-} is set to be located in middle of 6th and 7th signal while T_{ref+} is set to be located in middle of 7^{th} and 8^{th} signal. In this way, the 7^{th} signal which represents the median value can be captured by the detection window as shown in the decoder waveform in Figure 4.7. Compared to digital decoder design, the time-based decoder

dramatically reduced the area by 3×. Overall, the final area of proposed time-based MF is improved by 24% compared to conventional digital implementation as shown in Figure 4.8.

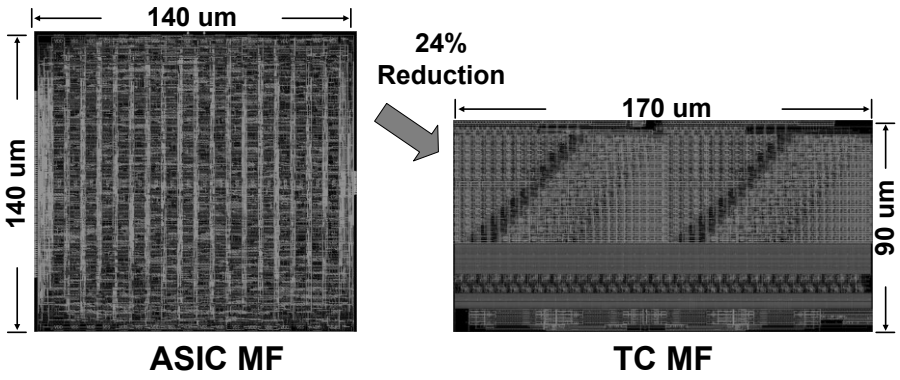


Figure 4.8 Layout comparison between (a) conventional digital MF and (b) MSTC MF.

4.2 Time-domain Computing Accelerated Image Recognition Processor

To demonstrate the proposed circuit techniques, we adopt a basic image recognition algorithm as shown in Figure 4.9 into a hybrid ASIC design with time-domain accelerators [50].

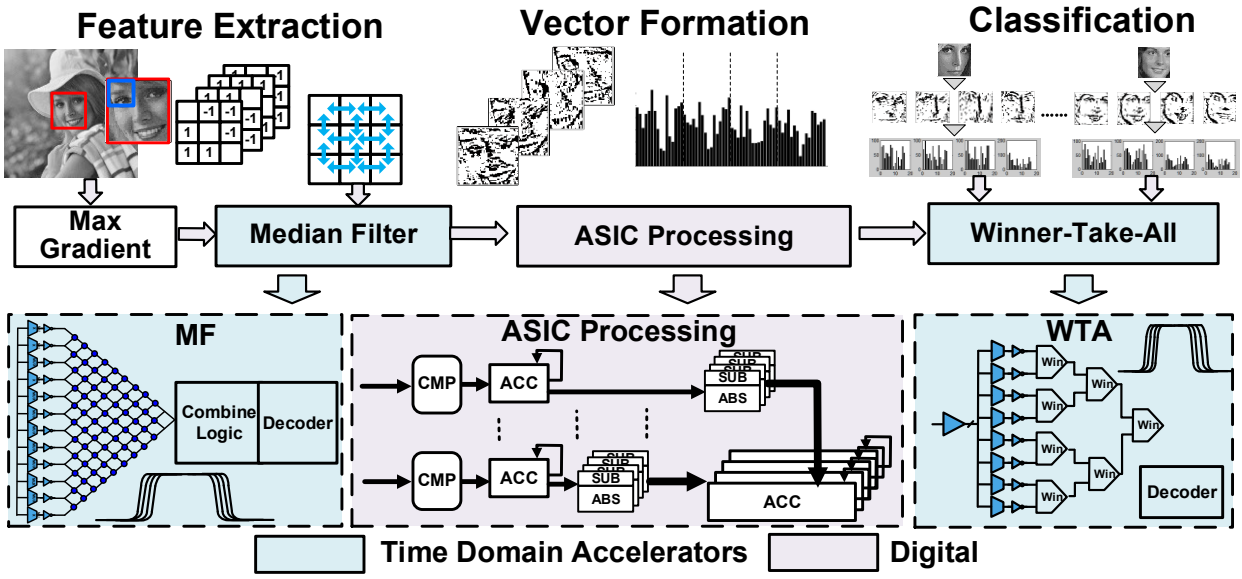


Figure 4.9 Overview of image recognition algorithm used in this work.

4.2.1 Implemented Image Processing Algorithm

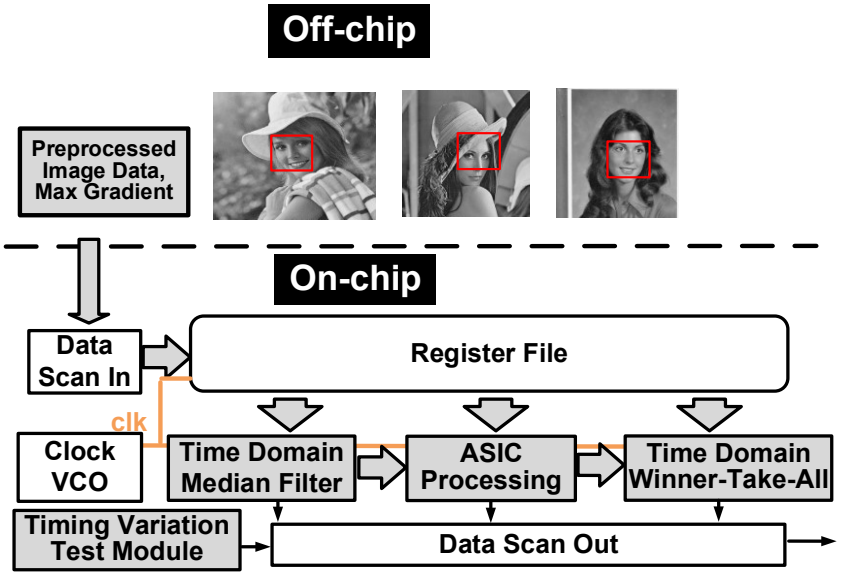
As shown in Figure 4.9, the operations of the image recognition algorithm involve three main steps: (1) feature extraction which detects edges in four directions: horizontal, vertical, +45˚ and -45˚. In order to determine the threshold value for edge detection, all the absolute-value differences between each two neighboring pixels are calculated in the 3×3 kernel and the median detection of the 12 difference-value is adopted as the threshold; (2) Vector formation where edge flags in all directions are counted and the spatial distribution of edge flags is represented by a vector of 64 elements; (3) Classification: the generated feature vector is then classified by a winner-take-all (WTA) classifier.

The subtractors and absolute value circuits shown in Figure 4.9 are used for calculating the distance between template feature vector and input feature vectors, e.g. $D_{xy} = |x - y|$. The compare (CMP) and 1st stage accumulator (ACC) compute the 64 elements. Then the subtractor and absolute value circuits calculate the distance of each element between the template feature vector and input feature vectors. At the end, the 2nd stage accumulator calculate the accumulated distance of the 64 elements between template feature vector and each input feature vectors. The heavily used nonlinear computations such as comparison (CMP), MIN/MAX function, are expensive for CPU/GPU based design or even state-of-art ASIC design. In this work, MSTC based accelerators are used to remove the bottlenecks of the algorithm, i.e. MF and WTA with significant speedup as shown in Chapter 4.4.

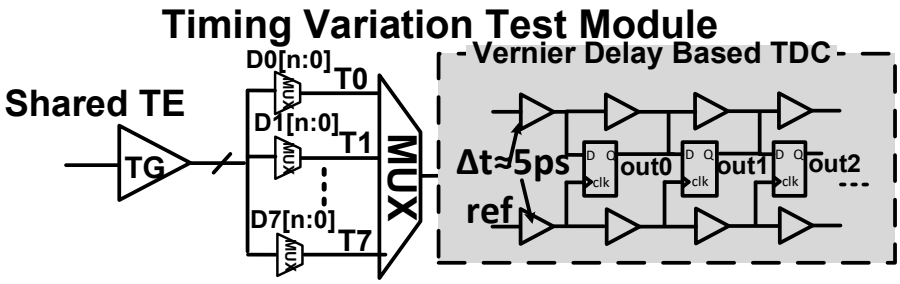
4.2.2 Test Chip Implementation

Figure 4.10 shows the test chip implementation of the proposed image recognition processor in a 55nm low power CMOS process at 1.2V. Scan chains are used to fetch image data

to on-chip register files and read out all internal register/comparator values for test verification. A special timing test module (Figure 4.10 (b)) is built to exam the linearity and robustness of the proposed shared DTC design. A Vernier-delay-chain based TDC with ~5*ps* bit-resolution is used to characterize the timing variation of DTC. The DTC used throughout this work is implemented with a ~25*ps* single-bit resolution which can be tuned from 13*ps* to 35*ps* for further evaluation.



(a)



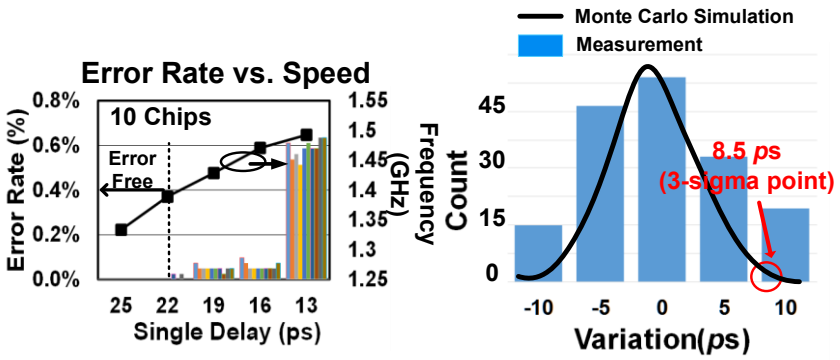
(b)

Figure 4.10 (a) Top level implementation of the proposed test chip; (b) Circuit diagram of timing variation test module.

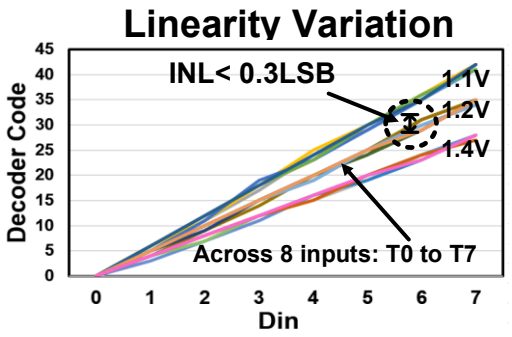
4.3 Measurement Results

In the test chip setup, there is a separate power supply for DTC modules to change the single-bit delay from 13*ps* to 35*ps*. Note that, we cannot directly measure the single-bit delay on the test chip due to the limited measurement resolution. However, based on extracted simulation from SPICE, we can estimate the single-bit delay on the test chip of the current supply voltage. Figure 4.11 shows measurement results. Robustness of the design was verified across 10 chips. As shown in Figure 4.11 (a), by default, no error was observed at the design target speed of 1.33GHz. When pushing the DTC resolution beyond 22*ps* (estimated based on simulation), small error was observed at the MF's output at LSBs while no error was observed at the final WTA output. The error rate from MF reached 0.6% when reducing the resolution to 13*ps* which led to an operating speed of 1.5GHz, a 13% boost of performance without observing error at the final output. This shows the strength of MSTC where small errors may be generated at LSBs at stringent timing condition but does not lead to significant error at final output.

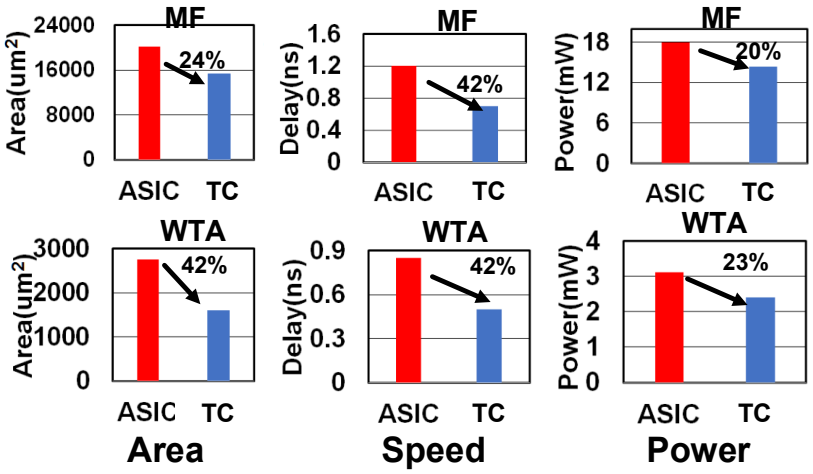
The linearity of DTC was also measured for all eight inputs across 10 chips and supply voltages from 1.1V to 1.4V as shown in Figure 4.11 (c). Only small deviation (~8*ps*) from ideal value was observed across all the measurement leading to an integral nonlinearity (INL) of less than 0.3 LSB. The measured (blue histogram) vs. the simulated variation from SPICE Monte Carlo simulation of DTC across chips are shown in Figure 4.11 (b). The results match the expectation from the simulation. Note that, the measurement results in the figure are from all chips and all paths with Din set as 7. As the matching among paths on the same chip is more critical, the measured variation on a single chip is quite small, which is within 5*ps*.



(a) (b)



(c)

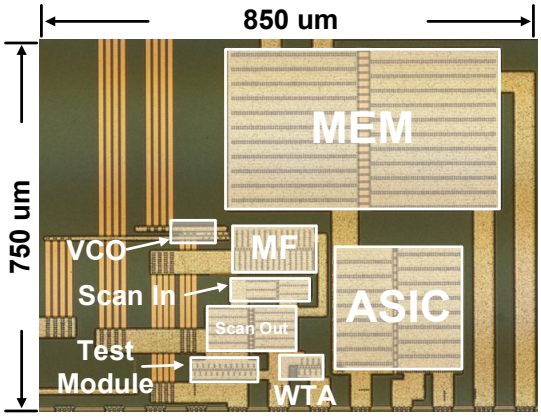


(d)

Figure 4.11 Measurement results on (a)performance, (b) measured (blue histogram) vs. simulated variation through chips, (c) linearity of the TE/DTC, (d) area, speed and power comparison.

4.4 Comparison and Discussion

The design is compared with conventional ASIC design in the same process with standard synthesis and place & route implementation. As shown in Figure 4.11 (d), 24% to 42% area saving is observed in MF and WTA accelerators compared with ASIC implementation. A 1.7× speedup and 20% to 23% power saving are also observed using MSTC. The overall image recognition processor operates at 1.33*GHz* with a throughput of 72 frames per second (fps). Figure 4.12 shows the die micrograph and the detailed design specifications. As the focus of this work is on robust and efficient techniques for time-based design, a direct comparison with prior work is difficult. We made comparison in two aspects: (1) time-based work, and (2) image recognition processors.



Process		Low Power 55 nm
Area		0.64 mm²
Frequency		1.33 GHz
Power	ASIC	59 mW @1.33 GHz
	MF	14 mW @1.33 GHz
	WTA	2 mW @1.33 GHz
Throughput		72 Frames/s

Figure 4.12 Die micrograph and specifications.

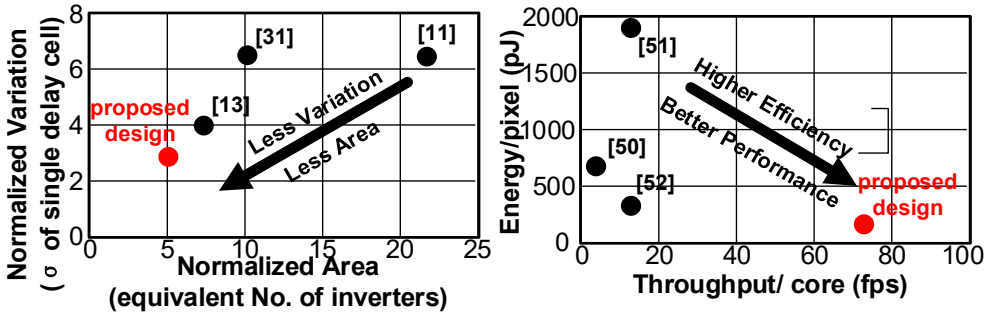
4.4.1 Time-based Work

As shown in Table 4.1 and Figure 4.13, compared with prior time-based work [11, 13, 31], (1) we achieved the fastest operation speed with a single-bit delay which is 2×~4× shorter; (2) We encoded largest number of bits by the bit-split technique; (3) We achieved lowest mismatch/variation which is 3× smaller compared with [11, 13, 31] ;(4) We had the least

encoding effort with lowest transistor count.

4.4.2 Image Recognition Processors

As shown in Table 4.1 and Figure 4.13, compared with image recognition processors with similar algorithms, e.g. feature vector based, we achieved (1) the highest throughput per core and throughput per area, (2) highest energy efficiency for single processor core with more than 9× improvement. However, it is notable that prior work involves more configurations and numbers of processing units [50, 51, 52]. Note that, (1) compared with our implementation, only [50] implements very similar design. We have significant advantages due to both time-domain design as well as technology scaling from 180nm to 65nm. (2) For designs in [51] [52], their algorithms are much more complex and support more throughput. For example, the work presented in [51] is based on vector parallel image recognition algorithm. Work presented in [52] is based on principal component analysis (PCA), and the proposed hardware utilizes the technique described above to reduce data dimensionality and uses support vector machine (SVM) as a final classifier for face recognition. Our comparison is focused on "single core/PE".



(a) (b)

Figure 4.13 Efficiency vs. variation of timing encoding circuits from prior work [11, 13, 31] and our proposed work; (b)Performance vs. energy for prior image processing designs [50, 51, 52].

4.5 Summary

In this chapter, a series of highly efficient time-domain signal processing techniques are proposed including shared time generator, double-edge operation scheme, bit-split technique and high-efficient time-domain operations. In our approach, the use of MSTC-based accelerates the pipeline operation bottleneck by 40% due to the limitation of MF and WTA operations. The strength of MSTC including error resiliency, highly efficient non-linear operations and better energy/area efficiency compared with digital counterpart is demonstrated by a test chip. The test chip on image recognition processor is fabricated in 55*nm* low power CMOS showing state-of-art energy efficiency and throughput with significant improvement from time-domain techniques compared with conventional digital implementation.

TABLE 4.1 PERFORMANCE COMPARISON

		[50] JSSC 2007	[51] JSSC 2014	[52] JSSC 2017	[11] JSSC 2014	[31] CICC 2017	[13] ASSCC 2016	**This work**
Image Recognition	Technology	180nm	180nm	40nm	65nm	65nm	55nm	55nm
	Voltage (*V*)	1.8	1.8	0.6	1.2	1.2	1.2	1.2
	Area (mm^2)	33.64	82.3	5.9	0.063	0.24	3.61	0.64
	Power (*mW*)	85	630	23	4	0.3	-	75
	Accuracy (No. of bit)	8	8	10	-	-	-	8
	Frequency (*GHz*)	0.1	0.05	0.1	-	0.1	-	1.33
	Throughput/core (fps) *	6.1	16	14.7	-	-	-	72
	Throughput/area (fps/mm^2)	11.6	12.4	80.3	-	-	-	116
	Energy/pixel (*pJ*)	756	2126	84	-	-	-	54
Time-based	Single bit delay (*ps*)	-	-	-	100	50	-	25
	Maximum No. of bit encoded	-	-	-	3	3	5	8
	Timing Mismatch**				6.5	6.5	4	2.8
	No. of equiv. inverters to generate 1-bit delay (4-bit TE/DTC)	-	-	-	22	10	7	5

Chapter 5

A Scalable Pipelined Time-Domain DTW Engine for Time-Series Classification

Time-series classification (TSC) is a challenging problem in machine learning and significant efforts have been made to improve its speed and computation efficiency. Among various approaches, dynamic time warping (DTW) algorithm is one of the most prevalent methods for TSC due to its succinctness and generality. To improve the throughput of the operation, this chapter presents a mixed-signal DTW accelerator utilizing MSTC where signals are encoded and processed using time pulses. A pipelined operation is enabled by a specially designed time flip-flop (TFF) circuit leading to dramatic improvements in performance and scalability of the operation. A 65nm CMOS test chip was implemented and measured. The results show more than 9× improvements in throughput compared to prior work on time-series classification. As most existing time-domain designs suffer from the lack of time-domain storage elements, this work utilizes sequential circuit elements in time-domain computing extending the capability of time-based circuits.

The reset of Chapter 5 is organized as follows: The basics and background of time-series classification and dynamic time warping (DTW) are introduced in Chapter 5.1. The special circuit techniques including the novel time flip-flop design is carried out in Chapter 5.2. The domain-domain pipeline architecture is presented in Chapter 5.3. The physical implementation and testing results are shown in Chapter 5.4. The comparison and discussion are presented in Chapter 5.5 with summary written in Chapter 5.6.

5.1 Time-Series Classification and Dynamic Time Warping

5.1.1 Time-Series Classification

A time series is a series of data points indexed, listed or graphed in time order [53]. Time series are encountered in many real-world applications ranging from electronic health records to human activity recognition. Typical examples of time series are stock price, voice, human motion, electrocardiogram (ECG) signal, etc. The classification of time series signals, e.g. an ECG signal, is commonly used for detection of special events or operational anomaly. However, time series classification (TSC) has been considered as a significantly challenging problem in data mining due to its variable speed, lack of alignment, random appearance of sparse events, and long time-sequence [53, 54]. Three conventional classification methods are being developed including the distance-based, model-based, and feature-based methods [53, 55]. The model-based and feature-based methods are case-specific and complex to implement. For example, the HMM algorithm as a model-based method can only be useful when dealing with voice signal classification. On the other hand, the distance-based methods, e.g. Euclidean-based, DTW-based or cosine-based, are comparatively easy to implement with good accuracy results. Specially DTW, a variant of the dynamic programming algorithm, has been widely used for time-series classification. In addition, as machine learning introduced promising results in dealing with classification and detection workloads, a few neural network (NN) based works for time-series classification were implemented showing good classification results [56]. Even though NN-based designs sometimes show better accuracy, they rely on large database for training which may not be available and requires large computation efforts. The NN-based design usually consumes more area and energy compared to the succinct distance-based methods. The strong

capability for distance measurement for variable-speed temporal sequences makes DTW a popular method for time-series classification in broad applications, such as ECG diagnosis, motion detection, voice recognition, stock prediction, etc. [53]. In addition, a similar dynamic programming based approach is also being used in DNA sequencing for comparison of similarity between DNA pairs [54]. To accelerate the operation, a DNA sequencing hardware accelerator based on dynamic programming algorithm was previously implemented resulting in 15 giga-cell-update per second (GCUPS) throughput at 70mW power consumption [54].

5.1.2 Dynamic Time Warping (DTW)

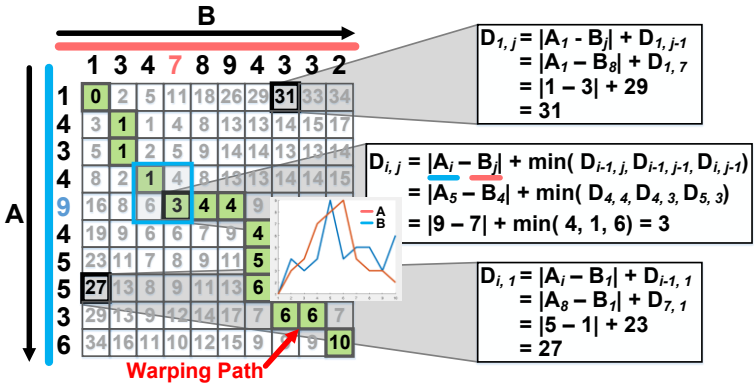


Figure 5.1 Dynamic time warping (DTW) algorithm.

Figure 5.1 shows the basic principle of DTW, which detects similarity among temporal signals with variable speed. As shown in Figure 5.1, for two time series A and B, $D_{i,j}$ can be formulated as the summation of absolute difference $|A_i - B_j|$ and the minimum value of its three ancestor nodes $\min(D_{i-1,j}, D_{i,j-1}, D_{i-1,j-1})$ where A_i and B_j denotes the ith and jth elements of A, B, and $D_{i,j}$ denotes the DTW value at node (i, j). The equation is written as:

$$D_{i,j} = |A_i - B_j| + \min(D_{i-1,j}, D_{i-1,j-1}, D_{i,j-1}) \quad (5.1)$$

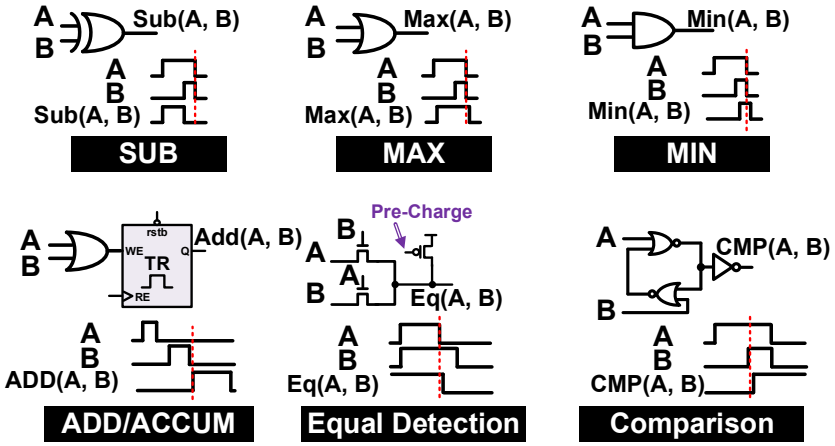
A "warping path" is produced in order to align the two signals in time, as highlighted in

Figure 5.1. The value of bottom-right node denotes the DTW distance between the two inputs. The lower distance represents more similarity between the inputs and can be directly used for classification tasks. As will be shown later, time-domain design holds significant advantages in performing simple operations such as MIN and ABS, which are repetitively used in DTW operations. As a result, in this work, we aim at utilizing time-domain computing to accelerate the DTW operations.

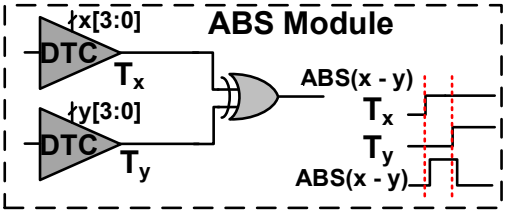
5.2 Time-Domain Acceleration Technique

5.2.1 Basic Time-Domain Computing Circuits

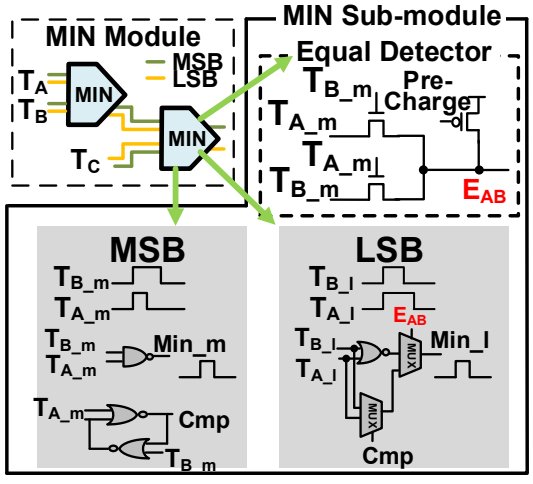
As the fundamental building blocks, basic time-domain operations, i.e. subtraction (SUB), maximum (MAX), minimum (MIN), addition/accumulation (ADD), equal detection (EQ), comparison (CMP), are specially designed with high energy and area efficiency as depicted in Figure 5.2 (a). As shown in the figure, some of the input signals are required to be overlapped while others are not. In order to guarantee the correctness of time-domain operations, we introduced the following mechanisms: (1) The overlap and non-overlap fashion of signals are pre-defined for different operations. For most operations besides ADD are working in the fashion of overlap. (2) We have special technique to make sure the rising or falling edges of two input time-domain signals are aligned in order to conduct the operation correctly. For example, by using the proposed time flip-flop to latch time-domain signals, the output time-domain pulses are aligned by falling edge. The operations such as CMP, MAX, MIN, can be easily implemented in time domain using few standard cell gates. DTW algorithm also requires some sophisticated computing modules, i.e. ABS and MIN, which are generally not easy to be implemented in digital domain. Figure 5.2 (b) (c) show the MIN and ABS modules used in this work.



(a)



(b)



(c)

Figure 5.2 Circuit details of time-domain circuits implemented in this work. (a) Basic time-domain circuits; (b) ABS module; (c) 3-input MIN module.

In the MIN module, computation is split into MSB and LSB groups. Both modules consist of only simple digital gates, e.g. NAND, rendering 6× reduction compared with equivalent

digital implementation. The 3-input MIN module consists of a 2-input MIN module and one equal detector module. The data path is divided into MSB and LSB paths. As shown in the figure, both MSB and LSB MIN modules are built by simple NAND, NOR, and MUX gates with corresponding waveform depicted.

As mentioned in Chapter 4, the existing time-domain demonstrations suffer from excessive digital and time domain conversion and the lack of internal storage. Missing the storage mechanism in time domain causes a lack of time-domain sequential logic which is required for high throughput pipelined structure or design of finite state machines in non-combinational circuits [54]. Thus, in this paper, a novel time-domain storage cell, namely time flip-flop (TFF) is introduced in the following section.

5.2.2 Time Flip-Flop Circuit

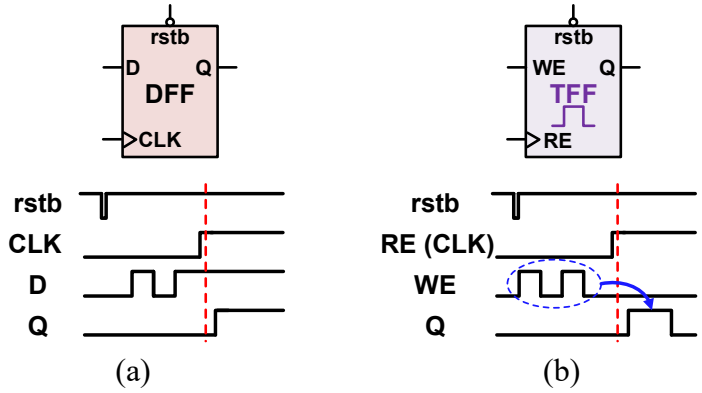


Figure 5.3 Differences between DFF and TFF. (a) DFF, (b) TFF.

As depicted in Figure 5.3, the proposed TFF takes time pulse as inputs and generates time pulse as the output triggered by the read enable signal. Compared to digital DFF, the proposed TFF operates in a similar fashion but has some advanced features: (1) TFF can store multi-bit information in time domain; (2) TFF takes multiple time pulses as input in a sequential order; (3)

Accumulation operation can be naturally realized – the output pulse width equals to the width summation of input pulses.

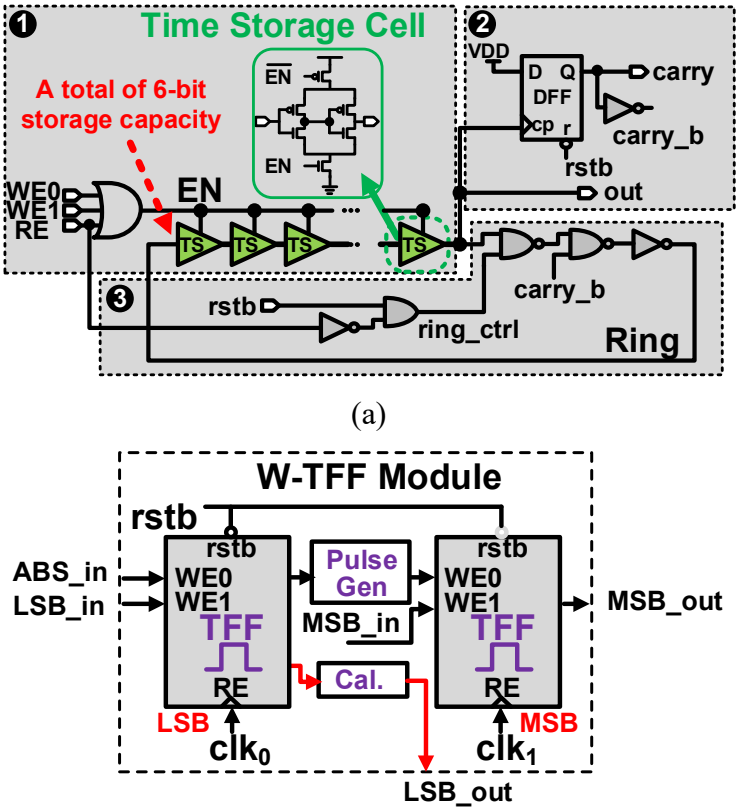


Figure 5.4 Time-domain flip-flop designs. (a) Circuit diagram of TFF; (b) Circuit diagram of the W-TFF module.

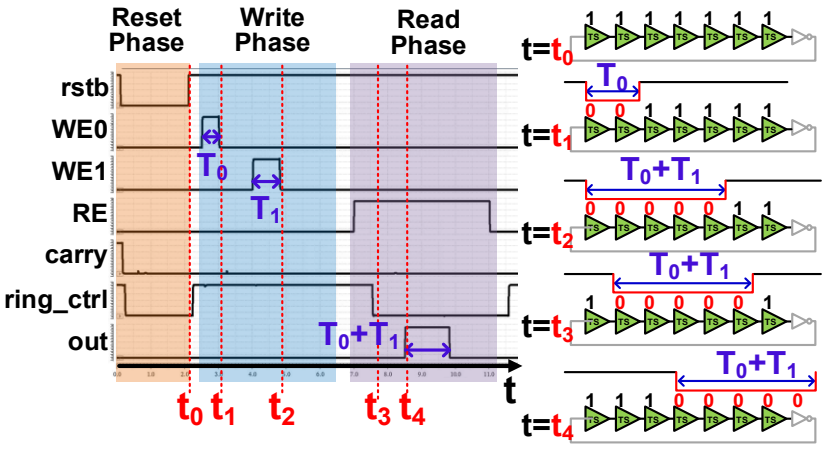
Figure 5.4 (a) shows the circuit diagram of a ring-based multi-bit TFF design which contains three parts: (1) a 33-stage tri-state inverter chain serves as the storage unit. In this design a total of 6-bit time-domain information with 40*ps* single-bit resolution (a total of 2520*ps* capacity) can be stored in such a tri-state inverter ring. (2) A carry signal detection module is used to generate a carry signal when the ring is fully filled. Due to the nature of the ring structure, the storing process can continue without the need of resetting the circuit after the ring is full. (3) A peripheral module which is used to reset the ring at the very beginning of the

computation. Besides, such a peripheral circuit is also used to flip the polarity of the output pulse when the ring is fully filled. In this design, each TFF can store a 6-bit time domain signal and two TFFs are used to construct a 10-bit time domain values separated into MSB and LSB units, leading to a wide-TFF module (WTFF) as shown in Figure 5.4 (b). In WTFF, once the LSB TFF is full, a carry signal is sent to a pulse generator to generate an extra pulse to be stored in the MSB TFF, extending the operation into 10 bits. In addition, a minimum pulse generator circuit is used to create a removable offset to keep the pulse from being too narrow (less than 100*ps*) to be propagated.

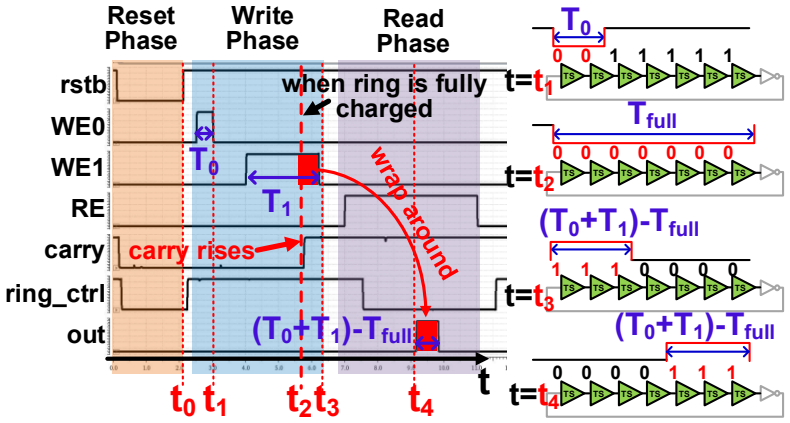
The write and read mechanism are described in Figure 5.5. In the scenario when the input pulses are not large enough to fully fill the ring (overflow), the simulated waveform is shown in Figure 5.5 (a). During reset phase ($t=t_0$), rstb signal is sent to reset voltages in the internal nodes of TFF. During the write phase ($t=t_1$, t_2), input pulses are sent to the ring, which allows propagation of "0" through the ring with a duration of input pulses. Multiple input pulses can be repeatedly sent to TFF and will be accumulated through the propagation of the ring. During readout phase ($t=t_3$, t_4), the stored pulse is sent out from the output pin of the ring with pulse width equivalent to summation of the stored values. Note that while the inputs are quantized time pulses, the information is stored as analog voltages on the internal nodes of the inverter chain, so no quantization loss occurs inside the TFF.

In another scenario when the ring is filled during write phase, the corresponding simulated waveform is shown in Figure 5.5 (b). At $t=t_2$ when the ring is filled, the operations are identical to the first scenario. At the moment of $t=t_2$, the ring is fully occupied by the input pulses while the writing process is still going on since the second pulse is not fully finished yet. A carry signal rises by the carry detection peripheral circuit and the ring will rotate back with remainder values

stored inside ($t=t_3 \sim t_4$). The "rotation" operation conveniently allows cascading TFFs into multi-bit groups rendering a scalable large numerical range of TFF.



(a)



(b)

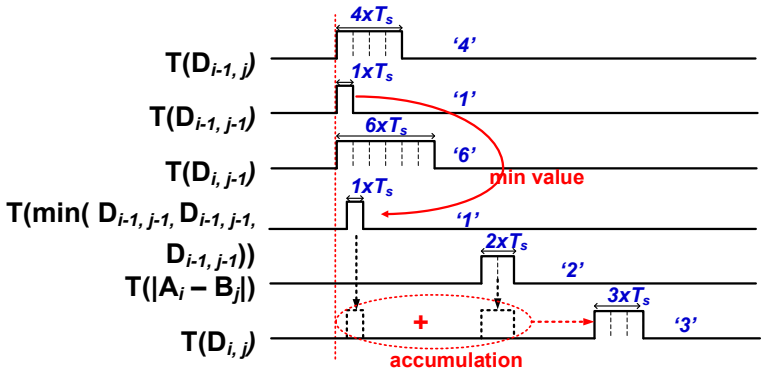
Figure 5.5 Simulated waveform of TFF when (a) ring is not fully filled, (b) ring is fully filled.

5.3 Time-Domain DTW Architecture

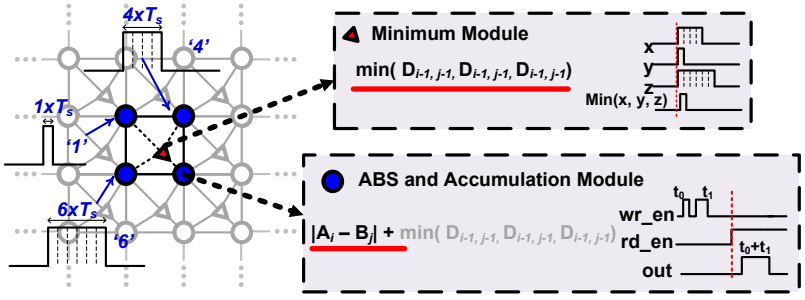
5.3.1 Time-Domain DTW Algorithm Mapping

As shown in eq. (5.1), the core computations of DTW contain two non-linear operations – the ABS and MIN. Such operations can be efficiently realized in time domain. The

corresponding time-domain waveform for node $D_{i,j}$ of eq. (5.1) is depicted in Figure 5.6 (a). The minimum value of its three ancestor nodes is carried by time-domain signal T(min($D_{i-1,\ j}$, $D_{i,\ j-1}$, $D_{i-1,j-1}$)) which is generated by the time-domain MIN module. The absolute difference is carried by time-domain signal T($|A_i - B_j|$) which is generated by the time-domain ABS module. The two time pulses are subsequently summed to generate the local DTW value of the current node. By recursively calculate the local nodes' DTW values in the matrix, the final DTW distance of the two time-series input can be obtained. The high-level circuit diagram of such a time-domain implementation is shown in Figure 5.6 (b) with succinct topology and data path.



(a)



(b)

Figure 5.6 Time-domain DTW algorithm. (a) Waveform of time-domain DTW; (b) Time-domain implementation of DTW.

5.3.2 Pipelined Time-Domain DTW algorithm

The time-domain implementation of DTW described in the above section is in the combinational logic fashion – there is no internal clock to synchronize the computation. This solution has its own benefits such as compact architecture, simple circuit requirement, and smaller latency when dealing with single time-series pair. However, it suffers from low throughput without pipelining, low utilization of hardware, and the bounded length of input time-series data limited by the dimension of hardware implementation. For such reasons, a pipelined architecture is developed to overcome the above issues.

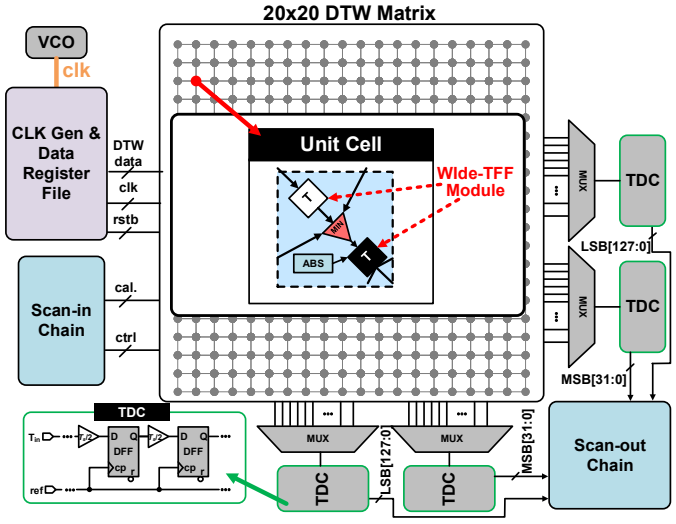


Figure 5.7 Architecture diagram of implemented pipelined time-domain DTW.

One key element to enable the time-domain pipelined design is the time-domain information storage cell, i.e. time flip-flop (TFF). By inserting TFF to every node of the DTW matrix, the pipelined architecture can be realized. Figure 5.7 shows the pipelined DTW engine with 20×20 DTW unit cells and scalable operation to construct longer time series. The DTW matrix contains a group of DTW unit cells with a diagonal pipeline structure. The unit cell, as depicted in Figure 5.7, contains 2 WTFF modules, an ABS module, and a MIN module. The

second WTFF module (marked in white) in the unit cell is used to copy the data from last pipeline stage, because the data stored in node (*i-1, j-1*) is one pipeline stage earlier than the nodes (*i-1, j*) and (*i , j-1*).

A 4-bit digital-time-converter (DTC) is implemented inside ABS to convert input digital values into time-domain pulses. The DTC consists of an inverter-based delay chain and multiplexers. The inputs of ABS modules are stored in on-chip SRAMs and sent to the 20×20 DTW array in the fashion of the systolic data streaming.

5.3.3 Pipelined Structure and Data Streaming Flow

Due to the use of the TFF, in every clock cycle, the time-domain pulses are propagated along the diagonal direction of the matrix as depicted in Figure 5.8. A total of 39 pipeline stages in the diagonal direction are synchronized by the global clock and reset signals. Note that, the TFF is the largest component and takes about 40% area of each DTW node. Hence, 40% overhead is added to enable pipeline operation. However, the throughput improvement of pipeline mode is 7× compared to the non-pipeline mode.

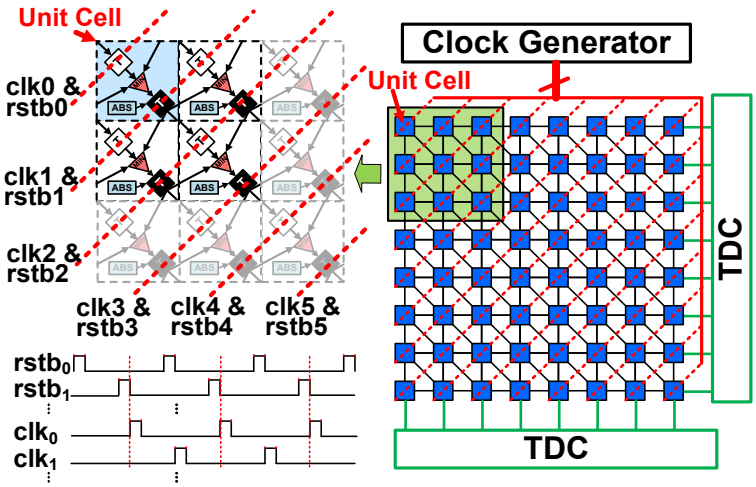
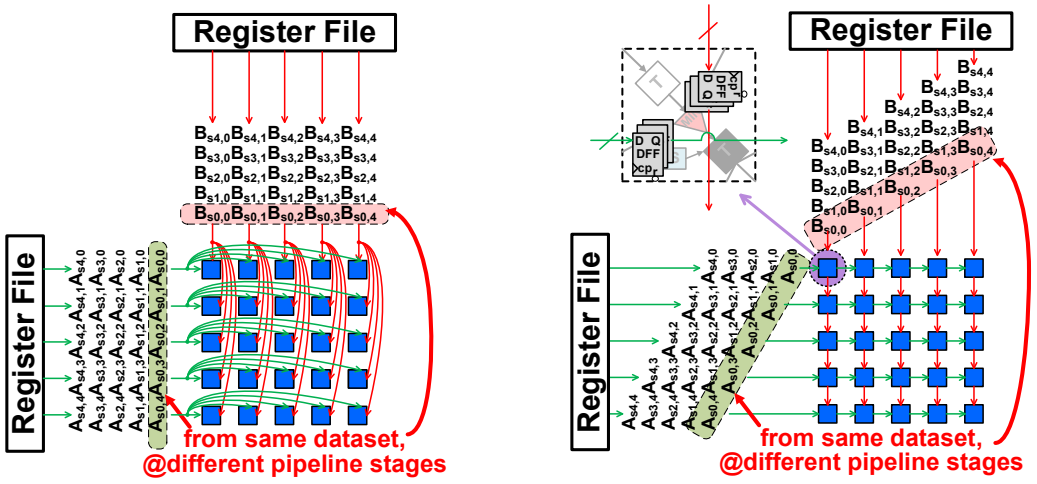


Figure 5.8 Diagonal data path and pipeline stage structure of DTW engine.

Data interaction can always be a challenge for array-based accelerator design, especially in a mixed-signal design which is very sensitive to the quality of signal routing. One straightforward solution for DTW data signal routing is shown in Figure 5.9 (a), with a massive routing broadcasting all signal connections. This would not only introduce signal crosstalk but also lead to the top-level signal routing congestions. Instead, in this work, a systolic data streaming flow is implemented where each data item is piped through the DTW matrix as inputs to ABS modules both vertically and horizontally (Figure 5.9 (b)). Such a flow is similar to a systolic dataflow in other accelerators e.g. Google's TPU design [32]. With such a solution, we reduce the signal crosstalk and eliminate massive data signal routing by more than 15×: The routing signals of ABS inputs are reduced from 2×20×20×4b into 2×20×4b at 20× reduction. However, some calibration signals still need to be explicitly routed into each DTW node which makes the total reduction into 15×.



(a) (b)

Figure 5.9 Data streaming flow comparison between (a) brute-force data streaming flow, (b) systolic data streaming flow.

5.3.4 Unfolding DTW Operation

The pipelined operation allows fixed dimensions of the DTW engine to be unfolded for longer data sequences, as shown in Figure 5.10. The total unfolded length is ultimately limited by internal register storage capacity, i.e. 10 bits in this implementation but can be easily extended further using the WTFF design. All output pulses from the bottom and right boundaries are decoded by shared time-to-digital converters (TDCs) every clock cycle and re-sent back for processing by subsequent sections.

Please note that due to the nature of analog/mixed-signal (AMS) computing, this design also has limitation on the scalability compared with digital implementation although we intend to improve this drawback by adding an unfolding operation in the special pipelined mode. In our study, most of our results are based on the final distance which require the value at the bottom right point of the matrix given that the distance measurement of two time series can be obtained at the bottom right corner of the matrix. For the goal of retrieving all intermediate data for post-processing for a larger matrix, multiple similar cores (not implemented in this work) can be stitched together on the same chip. In that case, the data from TDC can be send out to the next core for further operation with some degradation of the throughput due to data transmission. Such an operation is only supported in pipelined mode because the non-pipelined mode in this work would generate data asynchronously leading to a high cost in obtaining intermediate data.

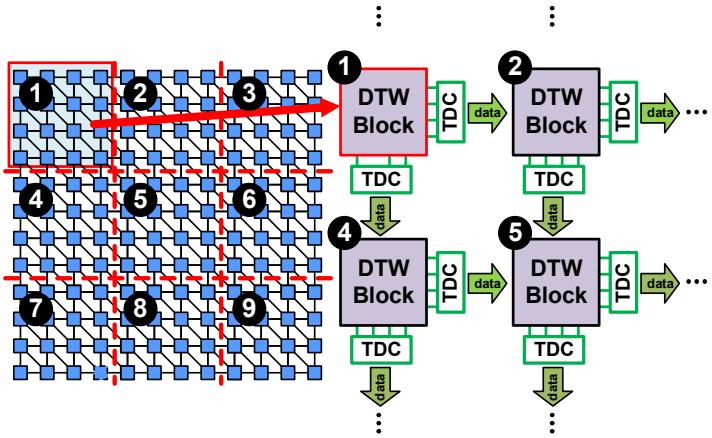


Figure 5.10 Unfolding mode of the proposed DTW engine.

5.3.5 Non-Pipelined DTW Mode

The pipelined mode is essentially designed for accelerating multi-bit time-series classification. And each pipeline period is determined by the capacity of the WTFF module, which is 10 bits in this design. As the processing time scales with the number of bits in time-domain operation, the pipelined mode is not efficient for low resolution time-series classification, e.g. DNA sequencing that only requires 1-bit operation. In such a case, the throughput is higher in non-pipelined operation than the pipelined operation due to the extremely fast operation at each node with only 1-bit input. Hence, to speed up the operation for simple data sequence case, a non-pipelined mode is implemented by bypassing the TFF modules and allowing signal edges to directly propagate through the matrix as shown in Figure 5.11. Different from pipelined case, in non-pipelined case, we encode information by the delay of rising edges instead of the pulse width of time pulses (similar to prior work [57]). Note that, the rising edge is naturally accumulated through the combinational block for “ADD” operation as depicted in Figure 5.6 (b).

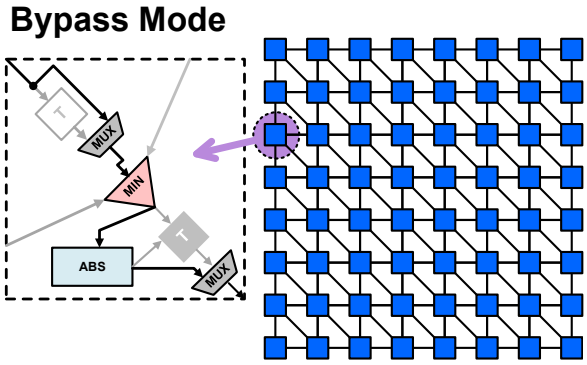
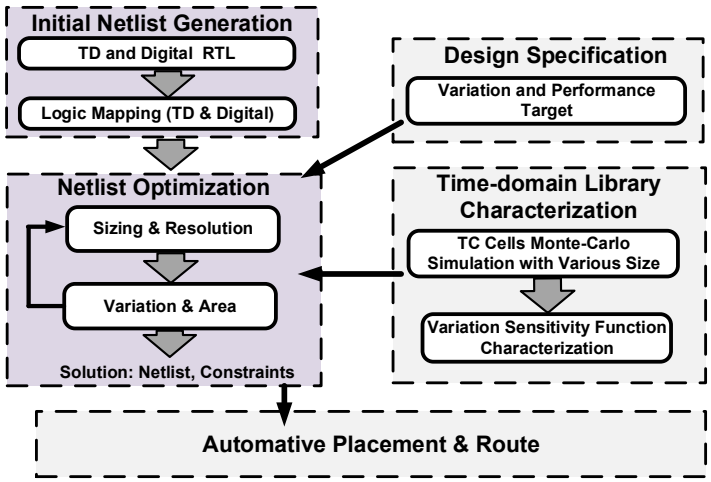


Figure 5.11 Architecture diagram of non-pipelined DTW mode.

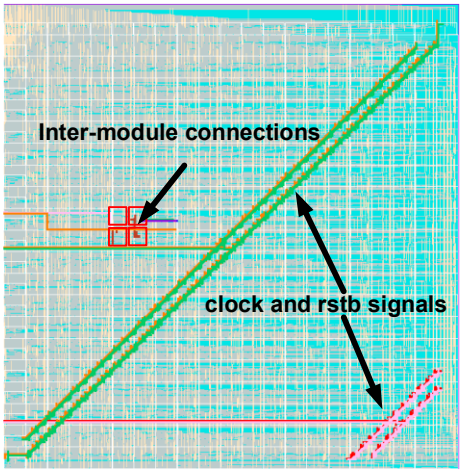
5.3.6 Design Automation for Mixed-Signal Circuit Design

Mixed-signal circuit design typically suffers from the requirement of manual layout efforts to enhance the integrity of the signals. To ease the large amount of design effort for the 2-D array, a time-domain design automation technique is utilized as shown in Figure 5.12.

In the local module level, the implemented automation technique includes both the synthesis and place & route parts. The synthesis process involves two steps: (1) the RTL with customized syntax for time-domain logics is utilized to perform a special mixed-signal time-domain (MSTC) logic synthesis process which generates an initial gate-level netlist; (2) The size of each module in the initial netlist is tuned by a special optimizer to meet the variation budget while keeping the area consumption small. The place & route process utilizes an adjacent constraint graph-based placement algorithm to realize the special signal mapping requirement in time domain [58]. As a result, majority of the modules are automated except critical local cells, e.g. ring core of TFF.



(a)



(b)

Figure 5.12 Design automation techniques used in this work. (a) Design automation flow chart; (b) Layout result of 20×20 DTW matrix.

In the higher level, we developed placement script and utilized digital tool to conduct the layout as such an example shown Figure 5.12 (b). The neighbor DTW nodes are placed abut to each other to minimize the routing length of inter-module connections. The critical global signals, i.e. clock and reset, are routed in a structured way by routing script with higher metal layer to relieve the signal crosstalk effect. As a result, the massive manual signal routing can be avoided at the higher level of the design while still maintaining routing quality/matching

performance compared to hand layout.

5.3.7 DTW Matrix Calibration Scheme

Similar to analog computing, variation is also a significant concern in time-domain computing [16]. To relieve such an issue, special calibration scheme is introduced to calibrate the 20×20 DTW matrix as shown in Figure 5.13. A 2b tunable delay cell is implemented in each unit cell to tune the output pulse width, compensating for process variations.

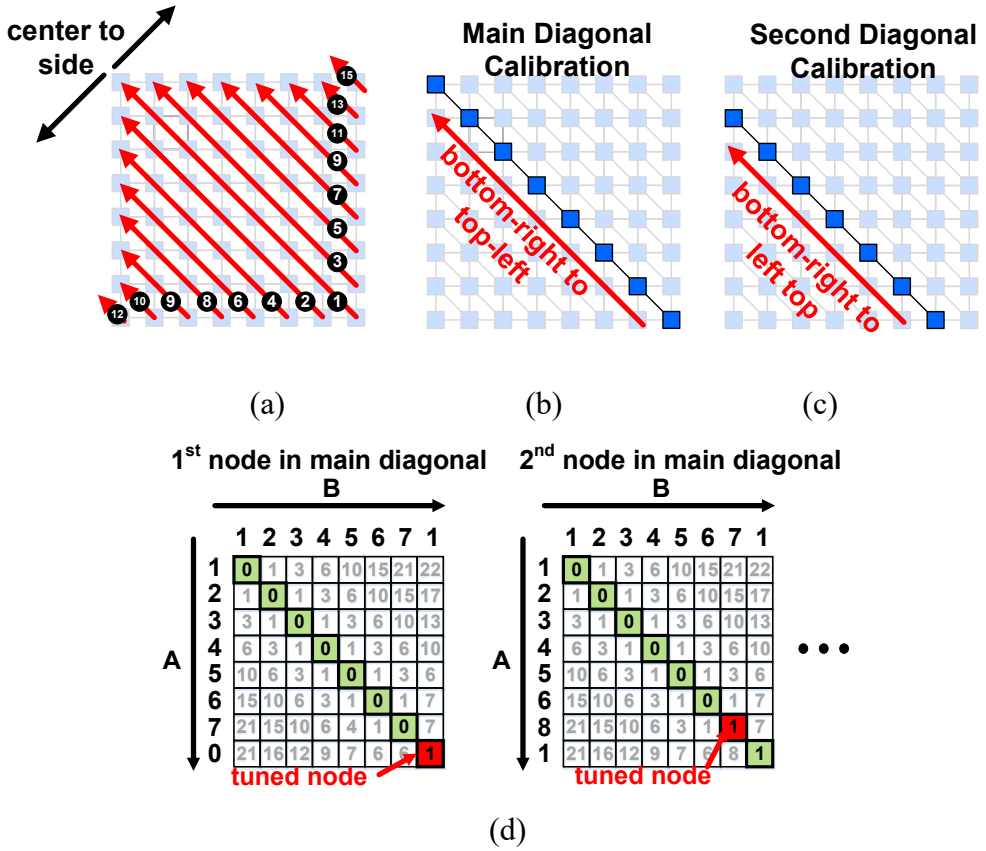


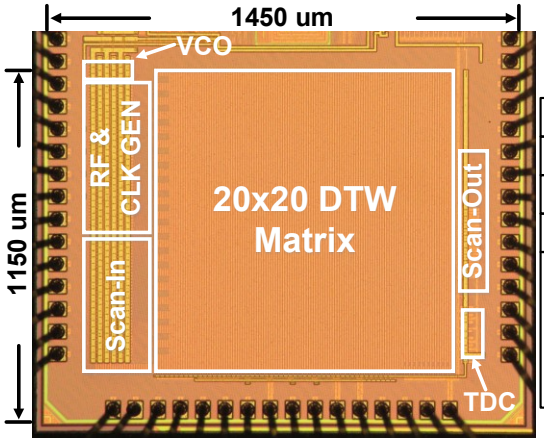
Figure 5.13 Calibration scheme of the 20×20 DTW matrix. (a) Calibration order through different diagonals. (b) Calibration order of each DTW node on the main diagonal. (c) Calibration order of each DTW node on the second diagonal. (d) Example of special input sets to enable the calibration of different node on the main diagonal.

The DTW nodes are calibrated through each diagonal path following a center-to-side order as depicted in Figure 5.13 (a). On each diagonal path, the nodes are calibrated from bottom-right

to top-left as shown in Figure 5.13 (b) (c) and the calibration is performed node by node. The basic idea is to construct special input sets which make the warping path to lie into the particular diagonal path and to be calibrated. By specially manipulating the input data patten, each node is further calibrated in that particular diagonal path one by one. Once the diagonal path is properly calibrated, the next diagonal path will be calibrated following center-to-side order until all the nodes on all diagonals are calibrated. This systematic calibration flow allows each cell to be tuned sequentially without back and forth operations and can be easily automated using the PC. The calibration results are shown in the next section.

5.4. Measurement Results

5.4.1 Test Chip Setup



Technology			65nm
Die Area			1.67 mm^2
Frequency			110 MHz
Nom Vdd			1V
Power	DTW Core	Pipelined	111 mW
		Non-pipe.	20 mW
	RF& CLK GEN	Pipelined	25 mW
		Non-pipe.	15 mW

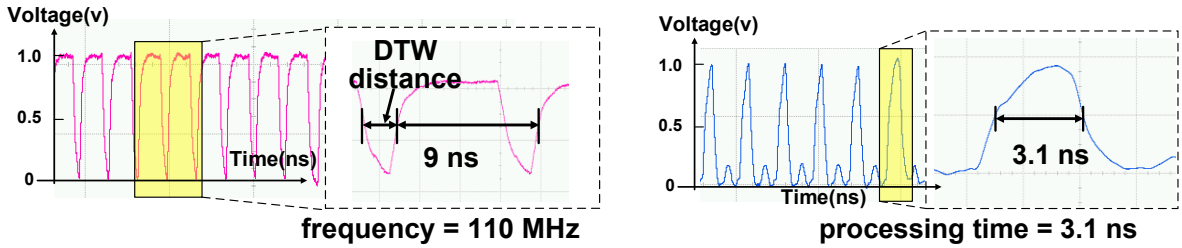
Figure 5.14 Die photo and chip specification.

A test chip of the proposed DTW accelerator engine was implemented in a 65nm CMOS process with die photo and specification table shown in Figure 5.14. The chip is running at 110 MHz with a nominal supply voltage of $1V$. Two sets of TDCs, based on Vernier delay chains, are placed at the right and bottom sides to decode time-domain signals at the boundaries.

A single-bit resolution of 40ps is used in the DTW design, while a resolution of 20*ps* is used in the TDC to reduce quantization errors at the boundary of operation. All the input and output data can be scanned in and out through a scan chain for verification.

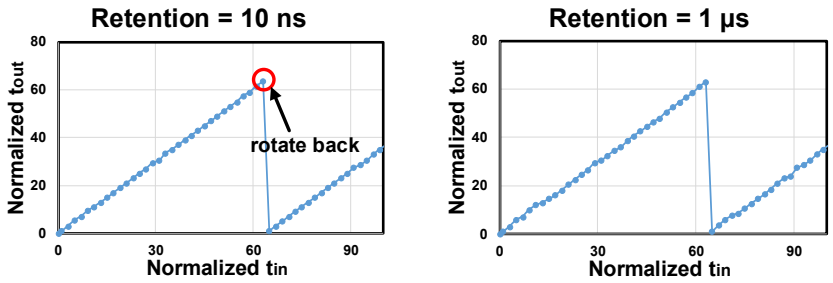
5.4.2 Measurement Results

Figure 5.15 (a) shows the measured waveform in the pipelined mode which confirms the expected output pulse at a frequency of 110*MHz*. The negative pulses depicted in the zoomed-in window carry the DTW distance information in time domain. Figure 5.15 (b) shows 3.1*ns* processing time in DNA-sequencing non-pipelined mode.



(a) (b)

Figure 5.15 Measured waveform of (a) pipelined mode, (b) non-pipelined mode.



(a) (b)

Figure 5.16 Linearity measurement of TFF at nominal 1.0*V* with (a) retention time of 10*ns*, (b) retention time of 1μs.

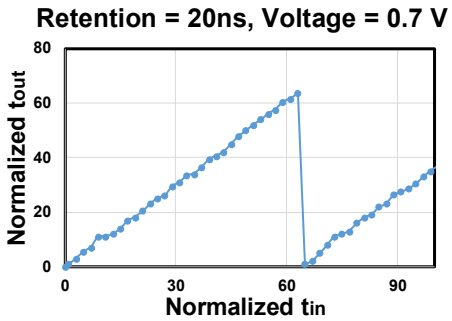
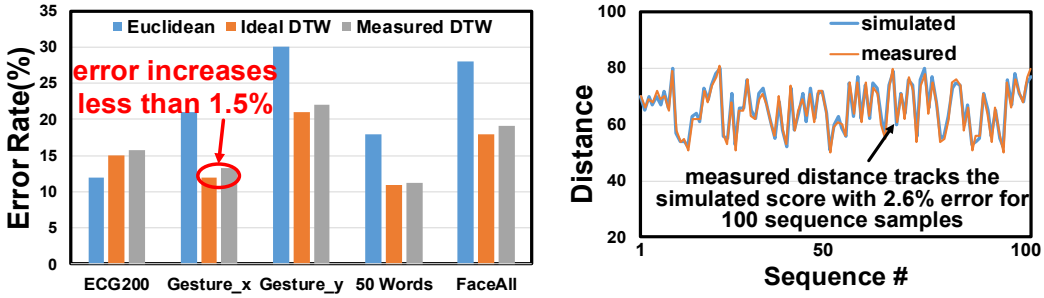


Figure 5.17 Linearity measurement of TFF in low voltage case (0.7V) with retention time is 20ns.

The linearity of the TFF is key to the accuracy of the DTW computation. Also, the retention capability of TFF for time-domain signals is important since the degradation of time-domain signal over the time due to leakage will cause information loss for the computation. The linearity of TFF is measured and verified under different retention time condition. As shown in Figure 5.16 (a) (b), the TFF is verified to retain data for over 1us at a supply voltage of 1V, with less than 0.5 LSB linearity loss due to leakage. This retention time is sufficient for the target application whose retention requirement is only 7ns. The linearity of TFF is also verified at a lower supply voltage of 0.7V. As shown in Figure 5.17, the linearity loss is 1.5 LSBs which results to classification error increase (2%) in the low voltage operation.

Figure 5.18 (a) shows measurement results on classification error using the fabricated DTW chip. UCR time-series classification databases were used with five databases from four typical applications including ECG signal classification, gesture recognition, words recognition and, face detection [59]. The measured error rate for classification by the DTW engine is only 1.5% higher than ideal DTW operation (floating point results in software). The increased error rate is mainly due to quantization loss (contributing about 0.5%) and process variation effect (contributing about 1%).



(a) (b)

Figure 5.18 Measurement results of different applications. (a) DTW classification error rate of UCR archive (pipelined Mode); (b) Simulated vs. measured DNA alignment distance (non-pipe. mode).

In order to test the performance of the non-pipeline DTW mode, a measurement of the DNA sequencing application is conducted. 100 sets of DNA sequence data from the human genome database (GDB) were tested for comparison between ideal DTW operation and measurement results. As shown in Figure 5.18 (b) the measured distance closely tracks the ideal results, having an error within 2.6%.

As shown in Figure 5.19, in order to test the robustness of the chip, the chip was verified at different supply voltages in pipelined mode down to $0.7V$, with a 2.3% increase in error rate compared with ideal DTW operation on the UCR database.

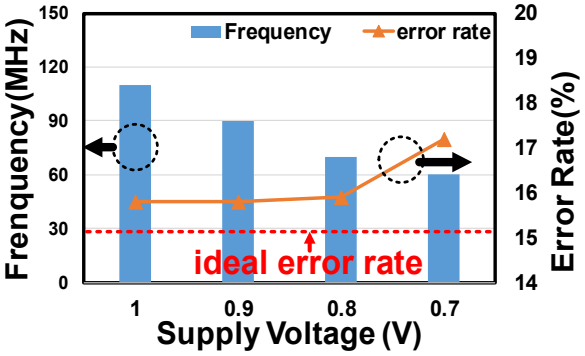


Figure 5.19 Chip operating frequency and error rate measurement under different supply voltages.

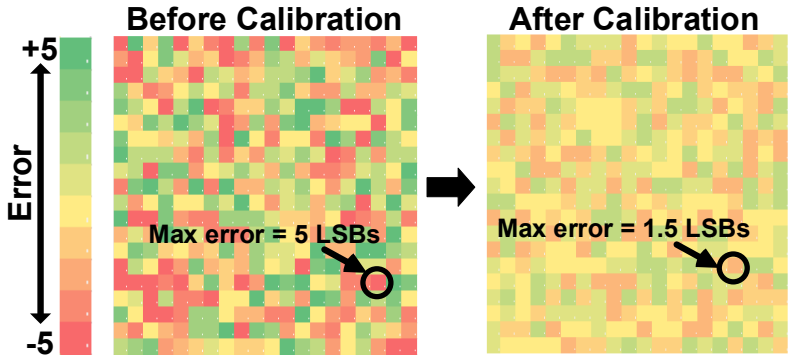


Figure 5.20 DTW node error measurement before and after calibration.

Figure 5.20 shows the chip calibration results before and after calibration operations. In this experiment, a 20×20 time-series classification task was conducted with 4-bit inputs. The scale for the figure is the measurement distance error in the unit of LSB. The final absolute computation different is 1 LSB. After calibrating the 20×20 DTW matrix, the maximum DTW distance computation error drops from 5 LSBs to 1.5 LSBs. Table 5.1 shows the comparison with prior work. A throughput of 140 giga-cell-updates-per-second (GCUPS) for DNA sequencing is achieved with 9× improvement over previous work [54]. The number of bits in this work are 4 bits as input and 10 bits in internal operation as compared with low resolution in most prior work, e.g. 1 bit [54]. More than 20× higher throughput per area (GCUPS/mm^2) is observed compared with prior CPU, GPU and ASIC implementations. This is mainly due to the area efficiency of time-domain circuit technique in special operations, e.g. compare, maximum and minimum. Overall, 1.5×~50× improvement of energy per GCUP is realized in this work compare to prior chip implementations. Over 20× and 18× improvements on inference per second per mm^2 and inference per second per watt are achieved respectively.

5.5 Comparison and Discussion

In order to form an apple-to-apple comparison, the technology scaling effect is also taken

into consideration. Compared to [54], whose throughput is limited by their time resolution which is 2*ns*. We assume the bit resolution scales with technology (which is not typically true in analog/mixed-signal design), our technology advances about 3 generation with scaling of about 0.7^3 leading to about 3× improvement in throughput. On the other hand, our design has shown 9× improvement of throughput, so we observe 3× improvement if taking into account of the technology impact. Compared to [60], we further scale down the process impact by 0.7 (from 90*nm* to 65*nm*) and the bit precision impact (from 32 bit to 4 bit), this leads to an throughput improvement of about 11× for the ASIC implementation of [60].

In addition, the use of time flip-flops enables the first pipelined architecture for time-domain design which not only improves the throughput but also increases the hardware utilization. Compared to non-pipelined operation, the pipelined design shows 7× improvement in throughput for general DTW applications. The hardware utilization has been improved from 11% to 93% due to the pipeline architecture.

In addition of the fabricated prior test chips, Kin Fun Li et al. proposed a DTW single element processing unit to investigate the suitability of using it as a building block for more complex architecture for embedded applications [63]. V.K Sundaresan et al. introduced parallel DTW algorithm [64]. Xiaowei Xu et al. proposed a memristor-based DTW accelerator design [56]. Compared to the digital implementations in [60] [63], our design improved the throughput by over 4×. Compared to the analog mixed-signal design in [64], we realized a throughput improvement over 200×.

TABLE 5.1 DTW ACCELERATOR DESIGN AND COMPARISON TABLE

	[61]	[62]	[60]	[54]	This work
Architecture	CPU	GPU	ASIC/ CPU	Time-domain ASIC	Time-domain ASIC
Process (*nm*)	65	28	90	180	65
Area (mm^2)	143	300	6.4	4	1.67
Number of bits	floating point	floating point	32	1	4 (input) 10 (internal)
Power (*mW*)	9.5×10^4	2×10^5	2732	70	136 (pipeline) 35 (non-pipe.)
Clock period (*GHz*)	2	1	0.6	0.01	0.11 (pipeline)
Throughput for DNA sequencing (GCUPS)	3	119	9 **	15 **	140 *
Throughput per Area (GCUPS/mm^2)	0.02	0.4	1.4	3.75	84
Throughput for general DTW App. (GCUPS)	-	-	-	-	71 (pipeline) 10 (non-pipe)
Energy per GCUP (*pJ*/CUP)	3.2×10^4	1×10^3	304	4.7	0.25
Inferences/Second (Giga)	0.006	0.276	0.021	0.036	0.32
Inferences/Second/mm^2 (Mega/ mm^2)	0.041	0.92	3.3	9	191
Inferences/Second/*W* (Mega/*W*)	0.0006	0.0014	0.0076	0.51	9.2
Error rate	-	-	-	2.9%	1.5~2.6%

* *In DNA application, single bit non-pipeline mode with input length of 20 is utilized for fair comparison with prior work.

** Technology scaling is considered and is further discussed in the above paragraph.

5.6 Summary

In this chapter, a general-purpose DTW engine using time-domain computing is designed for time-series classification. A special time-domain storage cell, namely time flip-flop, is developed with extendable ring-based structure and embedded accumulation functionality. The developed DTW engine also allows high-throughput pipelined data flow and unfolded operation for longer time series through a specially designed pipeline architecture utilizing the time flip-flop circuits. A 65*nm* CMOS test chip is fabricated and tested. The measurement shows a throughput improvement of more than 9× compared to prior works. In addition, a design automation methodology was applied to ease the mixed-signal design effort. A post-silicon calibration scheme was also incorporated to reduce the impact from process variation leading to 3× reduction of distance measurement error.

Chapter 6

A Mixed-signal Time-Domain Generative Adversarial Network Accelerator

In this chapter, a low-cost mixed-signal time-domain accelerator for generative adversarial network (GAN) is presented. A significant reduction in hardware cost is achieved through delicate architecture optimization for 8-bit GAN training on edge devices. An area efficient subthreshold time-domain multiplier is designed to eliminate excessive data conversion for mixed-signal computing, enabling high throughput mixed-signal online training that demonstrated in a 65*nm* CMOS test chip.

Chapter 6 is organized as follows: Chapter 6.1 introduces the background and design challenges in building a GAN accelerator for edge computing. Chapter 6.2 introduces the architecture innovation for GAN accelerator. Chapter 6.3 presents the circuit innovation including a high-efficient time-domain multiplier design. Chapter 6.4 shows the measurement results and Chapter 6.5 compare the proposed design with other time-based ML accelerator designs. Chapter 6.6 concludes the design.

6.1 Design Challenge in Generative Adversarial Network (GAN)

GAN is rendered as one of the most interesting and challenging applications in deep learning space. As shown in Figure 6.1, GAN contains two deep neural networks (DNN), i.e. a generator and a discriminator, contesting and evolving with each other [65]. Despite its broad real-time applications in gaming, authentication, VR, there is a lack of dedicated low power

GAN accelerator due to the tremendous challenges on resource-limited edge devices. From the algorithm aspect, GAN is extremely difficult to train due to model collapses from unbalanced models and high sensitivity to hyper-parameters. From the hardware aspect, GAN involves two DNNs with complex training sequences, e.g. 41 different training stages as in this work. Moreover, the typical floating-point training and complex calculation, e.g. batch normalization and optimizers, are very expensive for a resource-limited edge device [65].

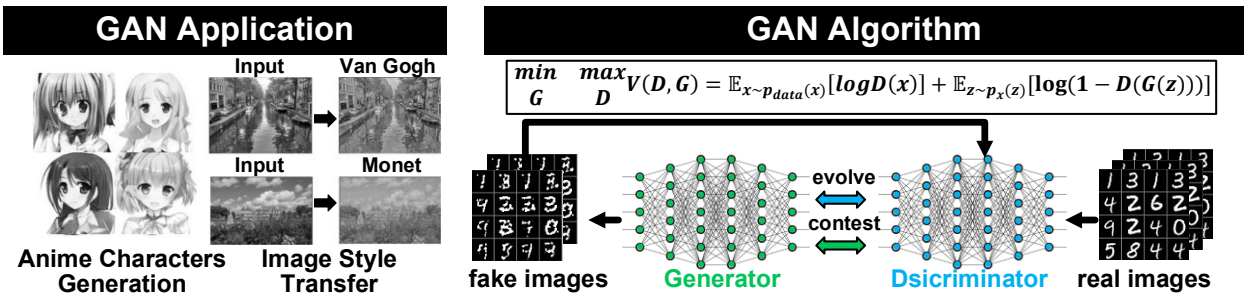


Figure 6.1 GAN applications and algorithm.

This work, through significant architecture improvement and hardware adaptation, presents a mixed-signal GAN accelerator with 8-bit resolution for cost-effective implementation on edge device. The contributions include: (1) for the first time, a complete GAN training core was implemented on an 8-bit low-power ASIC chip consuming only $39mW$; (2) An efficient subthreshold time-domain (TD) multiplier was designed with significant area saving compared to digital design; (3) On-chip training was performed in mixed-signal TD for the first time. The design eliminated 94% overhead from domain conversion, leading to the state-of-art throughput for a mixed-signal based accelerator which normally suffers from slow operation speed.

6.2 Time-Domain GAN Accelerator Architecture Design

Figure 6.2 shows the implemented GAN architecture with model compression that used in this work. For fitting with a small chip budget on edge device, we targeted a low-budget architecture implementation of DCGAN [65] using greyscale image with a size of 28×28 pixels.

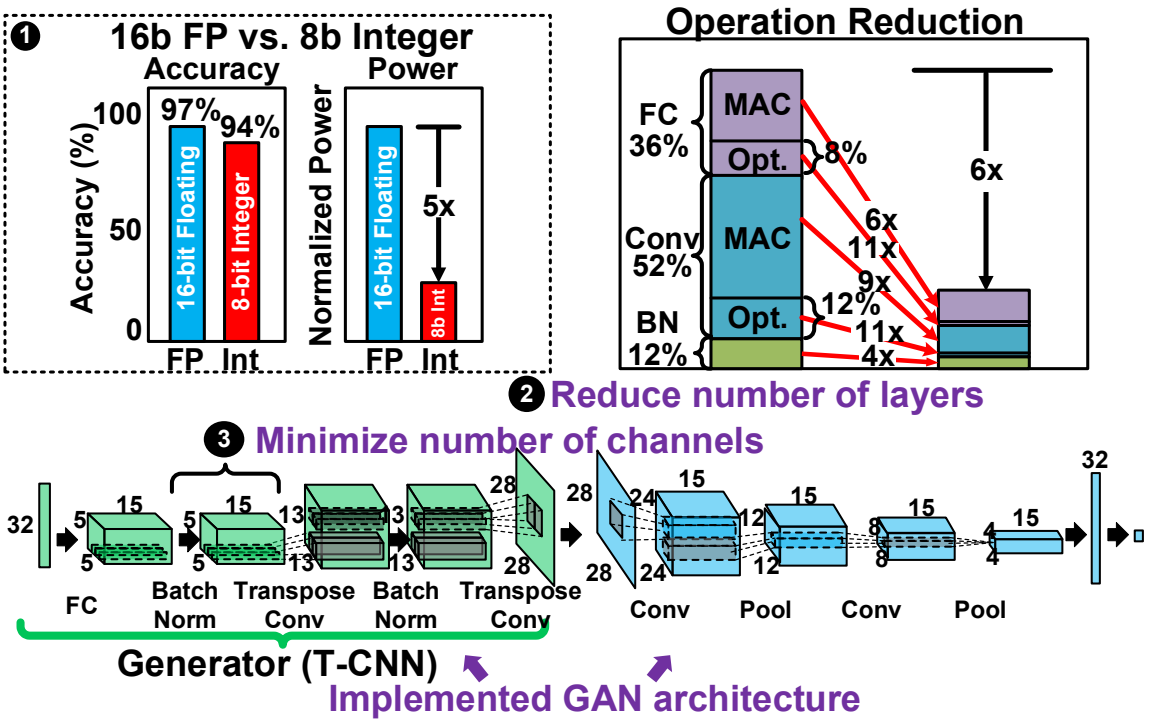


Figure 6.2 Model compression techniques utilized in this work.

As depicted in Figure 6.3 and Figure 6.4, the following techniques were specially developed: (1) model balancing and adaptive training were utilized to enable 8-bit training versus conventional floating-point training, leading to a 5× reduction in hardware cost; (2) The challenging and memory consuming operations of batch normalization were simplified by disabling low-impact runtime operations, rendering a 77% removal of the associated operations; (3) The expensive ADAM optimizer was replaced by a succinct momentum stochastic gradient

descent optimizer suitable for integer implementation with an 11× reduction of the optimizer's computation; (4) The number of layers and channels were further minimized to reduce the computation load by 6× to 9×. Overall, a 6× reduction of training complexity, a 6.5× hardware cost reduction, and an 11× reduction of on-chip memory were achieved through the algorithm simplification with about a 3% loss of accuracy.

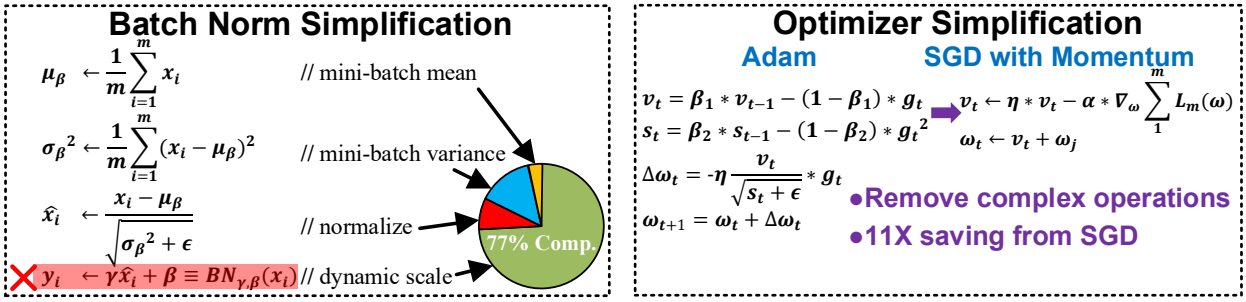


Figure 6.3 Hardware adaptation techniques utilized in this work.

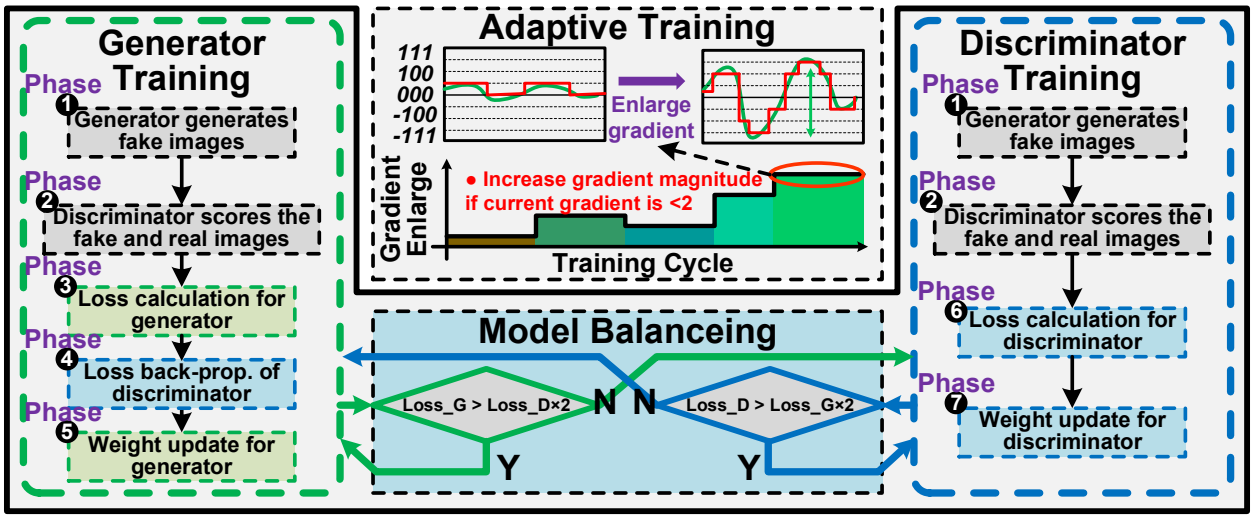


Figure 6.4 Adaptive training techniques in GAN accelerator design.

Figure 6.5 shows the training sequence. Each training iteration consists of 7 unique phases (e.g. forward prop., loss cal.) with 5 phases for the generator and 4 phases for the discriminator. Each phase also contains 4 to 6 sub-tasks (e.g. Conv, FC, pooling, etc.). To avoid model

collapsing, an adaptive training and model strength control scheme was implemented which ceases the training of discriminator if its strength is too high and adaptively increases the magnitude of the gradients during backpropagation (presented in Figure 6.4). The training sequence is managed by an ASIC training management unit (TMU) shown in Figure 6.6. A total of 41 training stages were implemented in the TMU as a finite state machine. Special operations such as pooling, sigmoid, data transpose etc. were handled by the dedicated hardware modules inside the TMU. Register files were used to store temporary weights and feature map outputs, bridging the throughput mismatch between SRAM and MAC arrays.

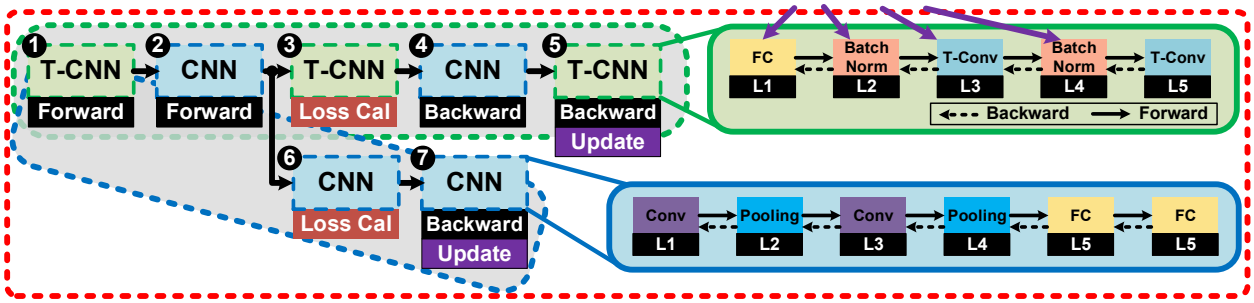


Figure 6.5 Training sequence of GAN.

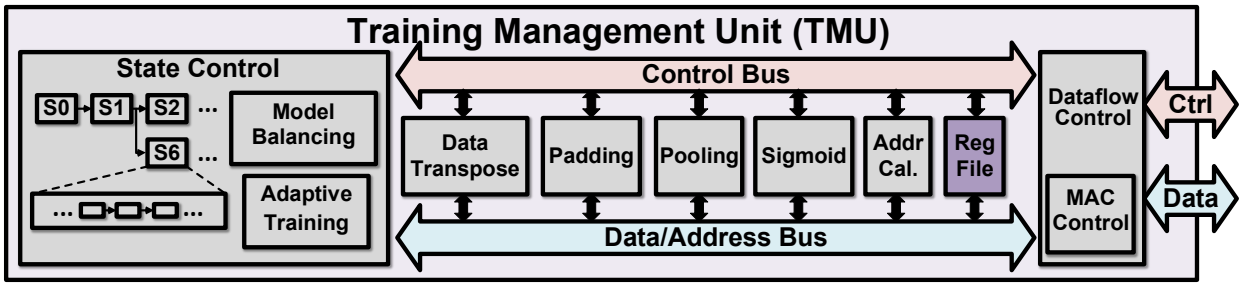


Figure 6.6 Block diagram of ASIC training management unit (TMU).

Figure 6.7 shows the test chip architecture diagram including the TMU, a 10×10 time-domain (TD) MAC matrix, SRAM modules and supporting blocks. All the MAC operations of CNN and Transpose-CNN are performed by a TD MAC matrix to improve area and energy efficiency. The circuit diagram of TD MAC matrix/array is depicted in Figure 6.8. The time

pulses generated from digital-to-time converters (DTC) are processed by the subsequent multiplication, accumulation and activation all in time domain and are finally converted back into digital domain using time-to-digital converters (TDC). A special 16b time-pulse based time-domain accumulator (TD-ACC) is designed using four 4-b ring-based time accumulators [27] with carry propagation to realize accumulation efficiently.

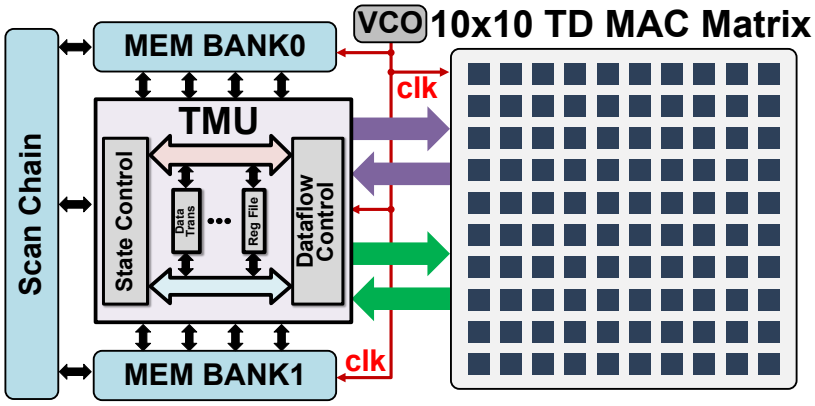


Figure 6.7 Top-level architecture diagram of proposed GAN accelerator.

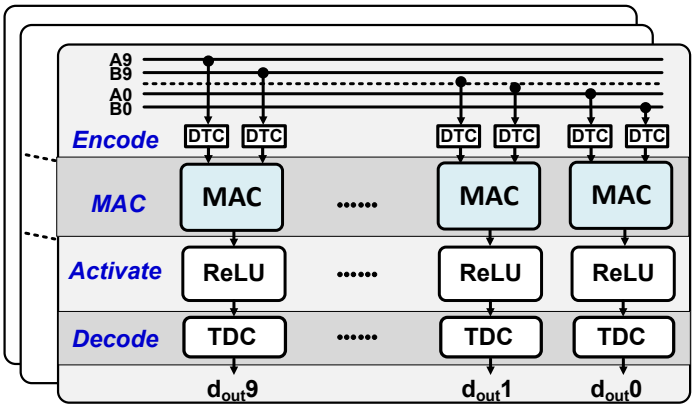


Figure 6.8 Circuit diagram of time-domain MAC array.

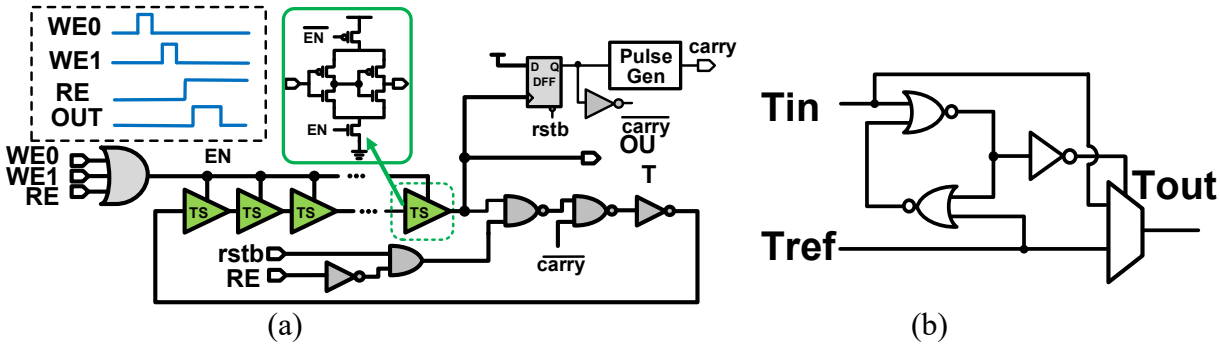


Figure 6.9 Circuit details of (a) 4b time-domain accumulator, (b) time-domain ReLU function.

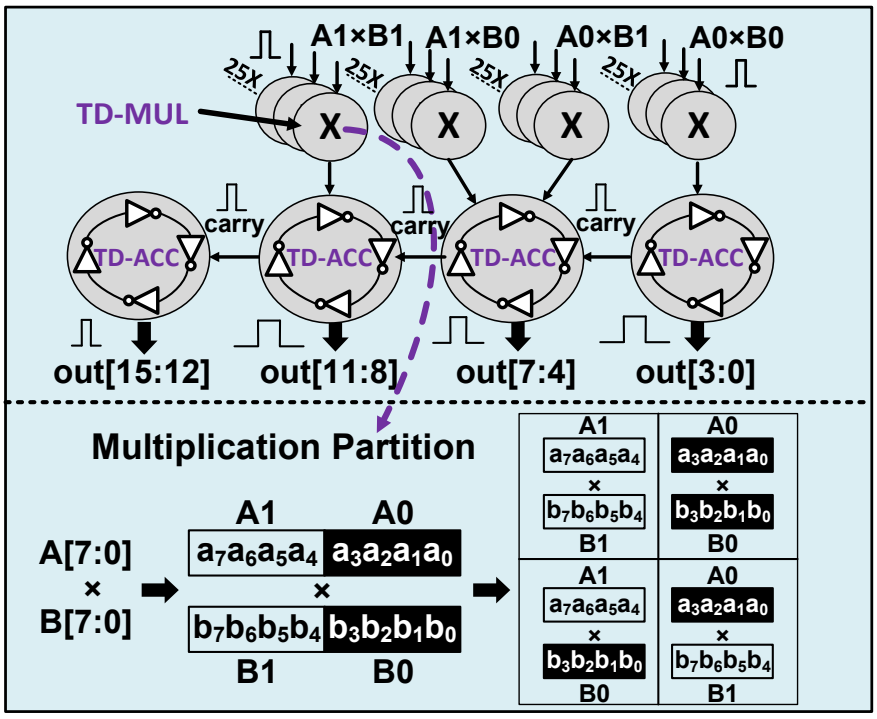


Figure 6.10 Circuit diagram of time-domain MAC unit.

As presented in Figure 6.10, with the special TD-ACC, the TDC is only activated once every 25 MAC operations, removing 94% of the time and power overhead from the expensive TDC operations. Pushing all operations in time domain significantly reduces the cross-domain data conversion, rendering a 160× speed-up in MAC operation compared with previous counter-based TD designs [15]. The 8-b TD multiplication is partitioned into four 4-bit multiplications to

improve the computation accuracy and speed. The detailed circuit implementation of 4-bit time-domain accumulator is shown in Figure 6.9 (a) with operating waveform of time-domain MAC operation shown in Figure 6.11

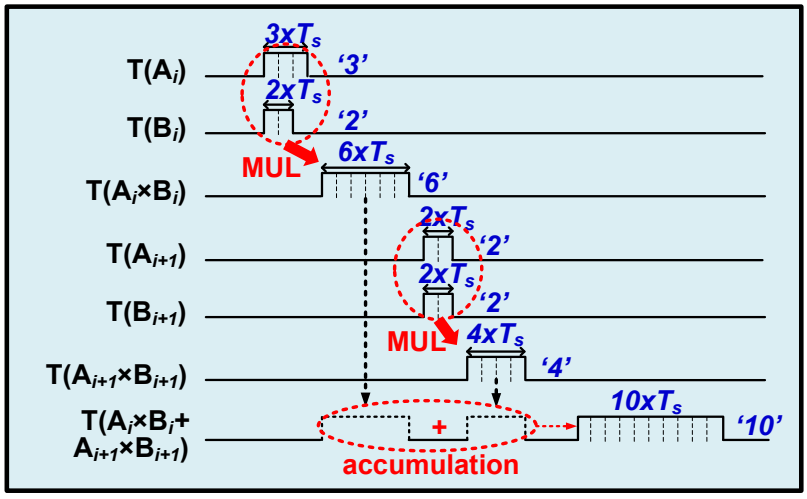
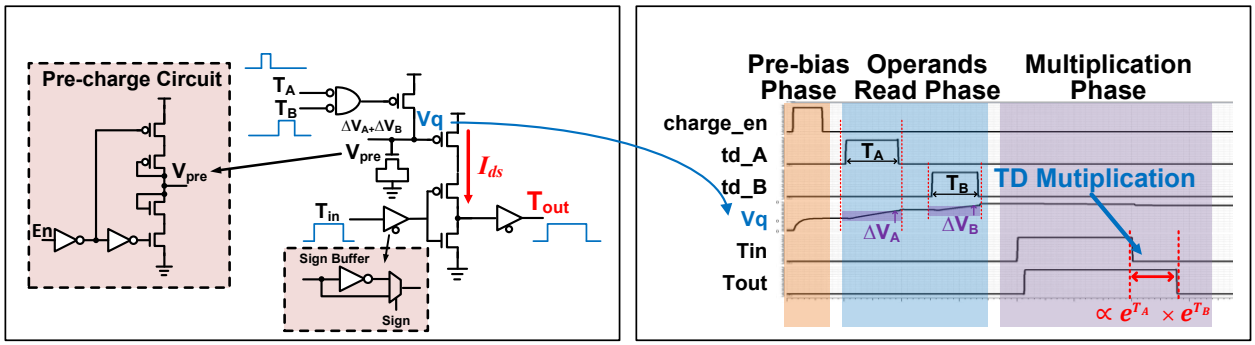


Figure 6.11 Time-domain MAC operation waveforms.

6.3 Time-domain GAN Accelerator Circuits Design

Figure 6.12 (a) shows the detailed circuit design featuring a subthreshold (sub-vth) TD multiplier (TD-MUL) and a DTC- based linearization technique. The TD-MUL takes input time pulses and generates output pulses of the multiplication results.

As in Figure 6.12 (a), the current starving PMOS transistor is pre-biased at subthreshold region and generates a delay equals to the multiplication results through charge accumulation at the gate with logarithmic addition, i.e. a multiplication is addition in log domain.



(a) (b)

Figure 6.12 Time-domain multiplication, (a) circuit details, (b) simulation waveform.

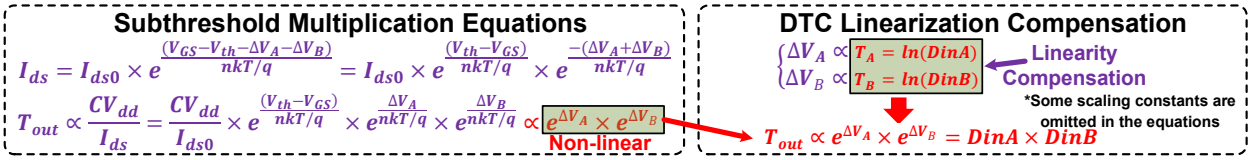


Figure 6.13 Nonlinearity compensation in time-domain multiplier.

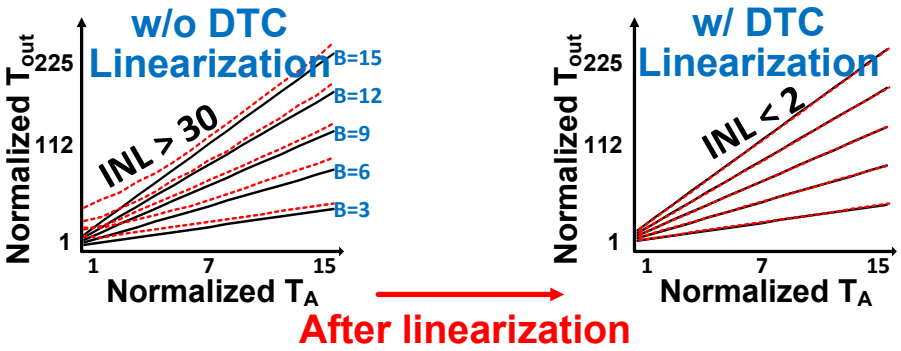


Figure 6.14 Nonlinearity compensation simulation results.

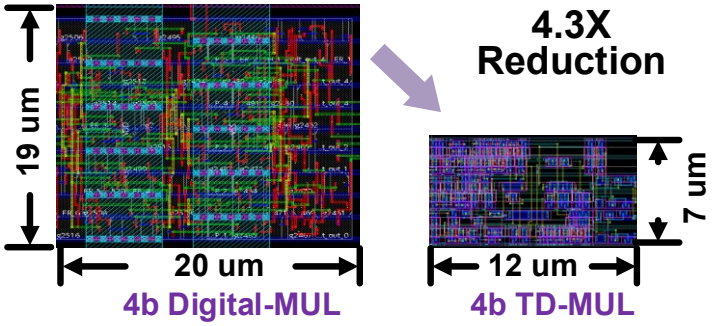


Figure 6.15 Layout comparison between 4b digital multiplier and 4b timed-domain multiplier.

The layout comparison between Compared to the digital implementation is shown in Figure 6.15. Overall, the implemented sub-vth multiplier renders a 4.3× reduction of area. However, as shown in simulation, significant nonlinearity is observed in sub-vth multiplication. The nonlinearity is compensated by a logarithmic encoding of DTC. As shown in both equation and the simulated waveforms in Figure 6.13 and Figure 6.14, the linearization technique elegantly removes nonlinearity with negligible overhead. After the multiplication, the resulting time pulses are sent into TD-ACC for accumulation of 25 cycles avoiding time-consuming digitalization as [15, 16, 66]. Simple TD ReLU function (depicted in Figure 6.9 (b)) is also implemented at each CNN layer except the final layer which uses digital sigmoid function.

6.4 Measurement Results

Figure 6.16 shows the measured linearity from both the TD-MUL and TD-ACC. For the multiplier, although up to 4% error is seen in the result, most of the error is just a small scaling factor shift. Less than 1b error is observed in the TD-ACC design.

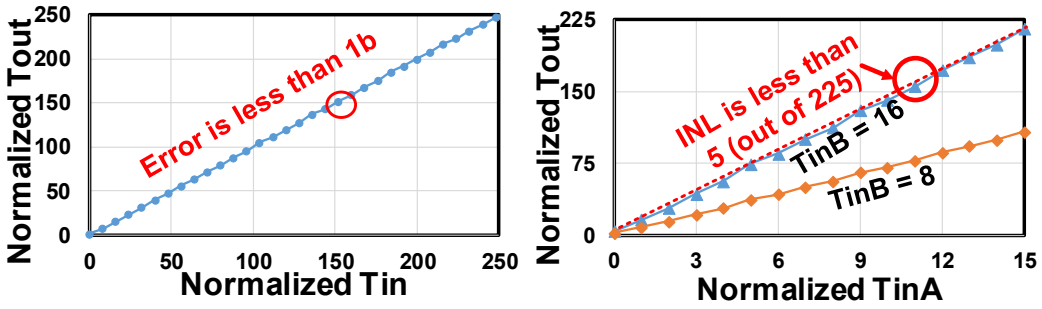


Figure 6.16 Linearity measurement of (a) time-domain accumulator and (b) time-domain multiplier.

We trained the GAN with 3 databases, i.e. a digit-MNIST, a fashion, and an emoji database [67, 68]. As depicted in Figure 6.17, accuracy of the generated images with conditional GAN

from 3 databases shows less than 1% error compared to the ideal integer 8-bit training on CPU and 3% compared with ideal floating-point training (1.6% comes from quantization loss). The final training results of the 3 databases are shown in Figure 6.18.

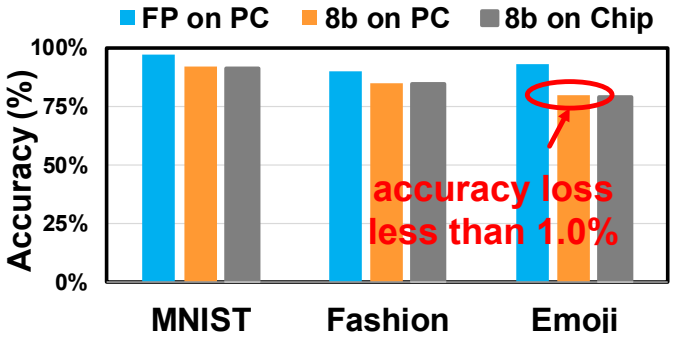
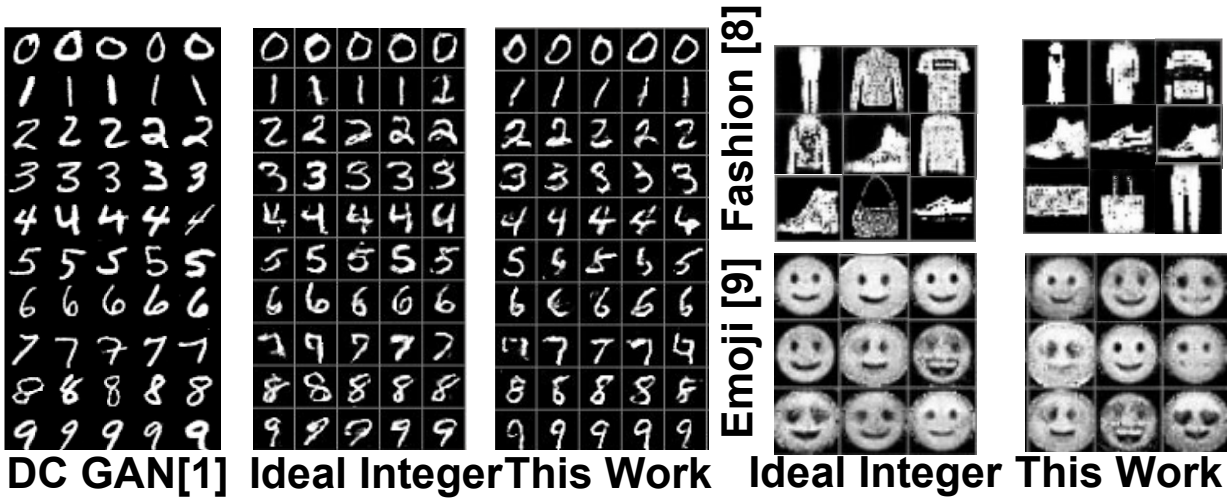


Figure 6.17 Measurement results of classification errors on different databases.

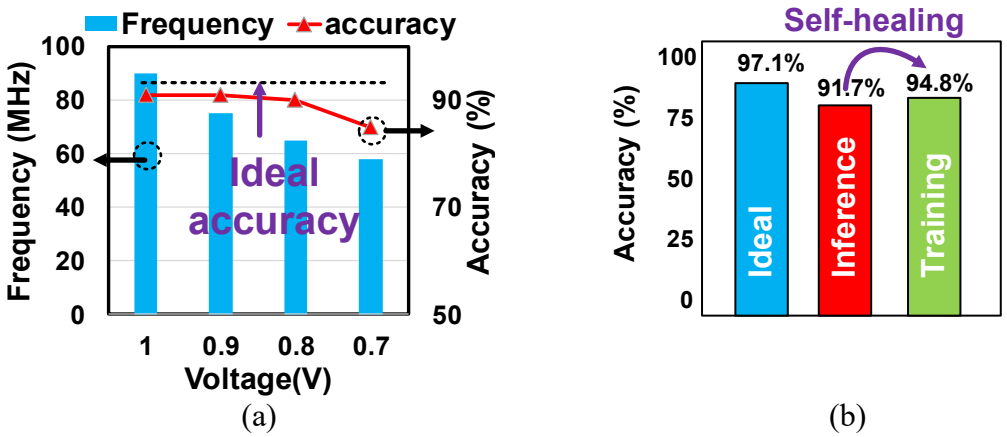


(a) (b)

Figure 6.18 Training results of GAN on (a) MNIST digit database, (b) Emoji and Fashion MNIST databases.

As results shown in Figure 6.19 (a), the chip is verified with supply voltages down to $0.7V$ with up to 5% degradation of accuracy compared with ideal GAN operation. Interestingly, a "self-healing" feature of GAN is observed as depicted in Figure 6.19 (b), recovering most of the error loss from on-chip variations compared with no on-chip training. This intrinsic resiliency

presents a merit for training empowered design using mixed-signal circuits. The chip consumes 39mW power with TD-MAC at 90MHz. The total training time of MNIST database takes 4.5 minutes which is 82× less than a high-performance CPU (2.6GHz Intel i7 Quad-core with a power of 197W). The die photo and comparison table with prior analog mixed-signal (AMS) designs are shown in Figure 6.20.



(a) (b)

Figure 6.19 (a) Measurement result of voltage scaling, (b) measurement result of 'self-healing'.

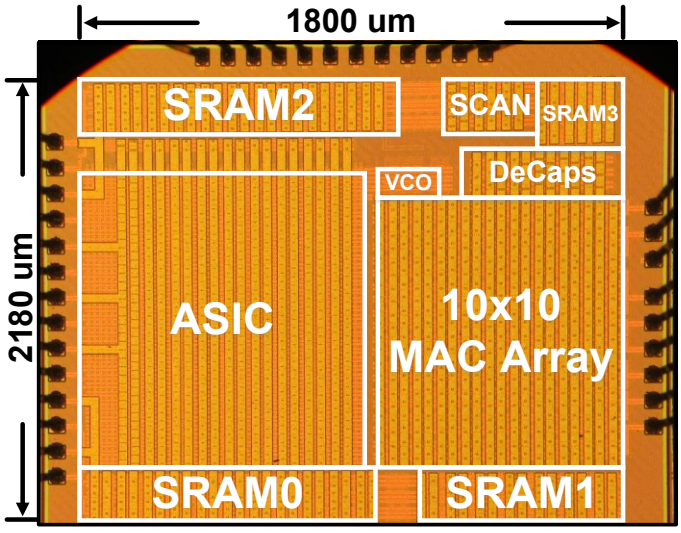


Figure 6.20 Die photo.

6.5 Comparison and Discussion

The comparison table is in As most of existing AMS designs suffer from low throughput, this work achieves the highest throughput of 18~5400× [31, 15, 16, 66, 69] with similar efficiency. In addition, a low-cost 8-bit on-chip training was realized for AMS design on the very challenging GAN operation.

TABLE 6.1 COMPARISON TABLE OF TIME-DOMAIN GAN ACCELERATOR

	[66] ISSCC 2016	[69] VLSI 2018	[31] CICC 2017	[15] ISSCC 2019	[16] ISSCC 2019	This work
Architecture	Switch Capacitor	TD ASIC	TD ASIC	TD ASIC	TD ASIC	TD ASIC
Application	Gradient Descent	DNN Inference	Image Recog.	Reinforcement Learning	CNN Inference	GAN
Process (*nm*)	40	28	65	65	40	65
Area (mm^2)	1.44	0.02	0.24	2.0	0.12	3.94
Input/Weight Resolution (bit)	6/3	8/8	1/3	8/8	8/1	8/8
Learning	Offline	Offline	Offline	Online	Offline	Online
Freq. (*MHz*)	2500	780	99	1.5	25	90
Power (*mW*)	0.65	0.15	77	0.003	0.03	8 (MAC) 31 (ASIC)
Throughput (GOPS)	1	0.8	0.75	0.0033	0.365	18*
MAC Efficiency (TOPS/W*Bit)	16	112	0.004	1.1	12	18*
On-die SRAM	-	-	0.1 KB	16 KB	No	52 KB

6.6 Summary

In this Chapter, the first mixed-signal design for GAN accelerator is presented with efficient subthreshold time-domain 8b multiplier. A few novel circuit designs including time-domain multiplier and time-domain accumulator are proposed. The time-domain multiplier allows a 2.6× area improvement compared to digital counterpart. Model compression is utilized to improve the hardware efficiency for edge computing. To further stabilize the training process of the fragile GAN algorithm, the adaptive training technique is introduced. Compared with prior mixed-signal design, the highest GOPS for mixed-signal neural network computing is reported.

Chapter 7

A 3T Dynamic Analog RAM-Based Computing-in-Memory Macro and CNN Accelerator Design

In this chapter, a Dynamic-Analog-RAM (DARAM) based Computing-In-Memory (CIM) macro and associated CNN accelerator is demonstrated in a 65*nm* CMOS test chip. With special analog sparsity techniques and retention enhancement, the design achieves state-of-art energy efficiency of 217TOPS/*W* at CIM macro level and 44 TOPS/*W* at system level for 4 bits weight/input operation. An effective bit cell size of only 75% of 6T foundry SRAM cell is achieved.

Chapter 7 is organized as follows: the design basic of CIM is introduced in Chapter 7.1. The special circuit techniques including the special 3T dynamic analog RAM are presented in Chapter 7.2. Architecture innovation of the proposed CIM-based CNN accelerator is presented in Chapter 7.3. Special energy saving techniques are introduced in 7.4. Measurement results and comparison with prior work are introduced in Chapter 7.5 and 7.6. Chapter 7.7. summarizes the proposed CIM design.

7.1 Computing-In-Memory Design and Challenges

Computing-In-Memory (CIM) techniques which incorporate analog computing inside memory macros have shown significant advantages in computing efficiency for deep learning applications. While earlier CIM macro was limited by lower bit precision, e.g. binary weight in [18], recent works have shown 4 to 8bit precision for the weights/inputs and up to 20bits for the

output values . Sparsity and application features have also been exploited at system level to further improve the computation efficiency [21, 22, 23]. To enable higher precision, bit-wise operations were commonly utilized [21, 22, 23]. However, there is limitation on existing solutions using the bit-wise operations with SRAM cells.

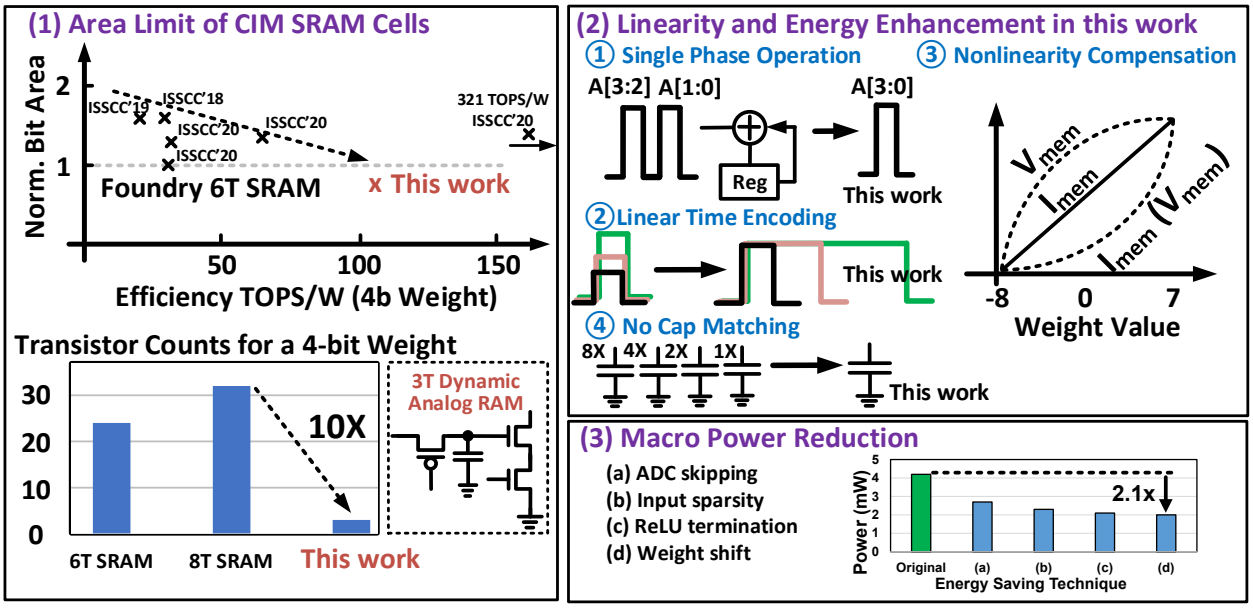


Figure 7.1 Challenges in CIM design and our proposed solution..

Figure 7.1 shows the summary of challenges and solutions in this work. First, all existing solutions utilize 6T/8T/10T SRAM as a CIM cell which fundamentally limits the size of the CIM array. In this work, we replace the commonly used SRAM cell with a 3-transistor (3T) analog memory cell, referred as dynamic-analog-RAM (DARAM) which represents a 4-bit weight value in an analog voltage. This leads to ~10× reduction of transistor counts and achieves an effective CIM single-bit area smaller than the foundry supplied 6T SRAM cell for the first time. Secondly, as no bit-wise calculation is needed in this work, only single-phase MAC operation is performed, removing throughput degradation due to previous multi-phase operation and associated digital

accumulation in [20, 21]. Furthermore, analog linearity issue is mitigated by highly linear time-based activation, removal of matching critical multi-bit caps [21, 23], and a special read current compensation technique. Thirdly, to mitigate power bottleneck of ADC or SA, this work applies analog sparsity based low power methods which includes a compute-adaptive ADC skipping operation when analog MAC value is small (or "sparse") and a special weight shifting technique, leading to additional ~2× reduction of CIM-macro power. We demonstrated the proposed techniques using a 65nm CIM-based CNN accelerator showing a state-of-art energy efficiency.

7.2 Dynamic Analog RAM-Based CIM Circuit Design

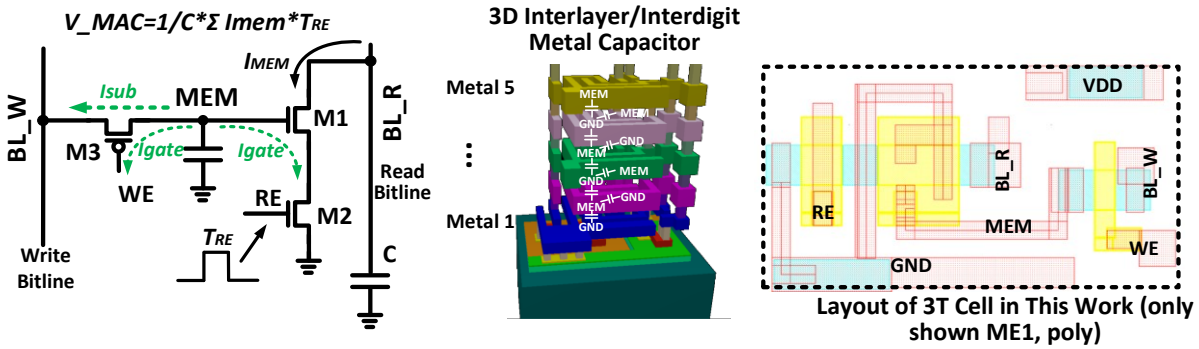


Figure 7.2 Proposed 3T DARAM design, (a) circuit schematic, (b) 3D diagram of metal capacitor, (3) layout.

Figure 7.2 shows the design details 3T dynamic-analog-RAM (DARAM). Similar as a conventional CIM bit cell, the charge drawn to BL_R is proportional to the multiplication of read current I_{mem} from the read access transistor M1 and time pulse duration of RE through switch M2. A 4-bit weight is stored as an analog voltage on the internal "MEM" node generating a read current proportional to the weight value. Due to the 4-bit lumped analog weight, a 4-bit MAC operation is realized by a single read of the DARAM, much simpler than the previous bit-wise operation. Designed with regular logic transistors, the critical read-access transistor M1 is sized

with larger W and L to reduce device variation.

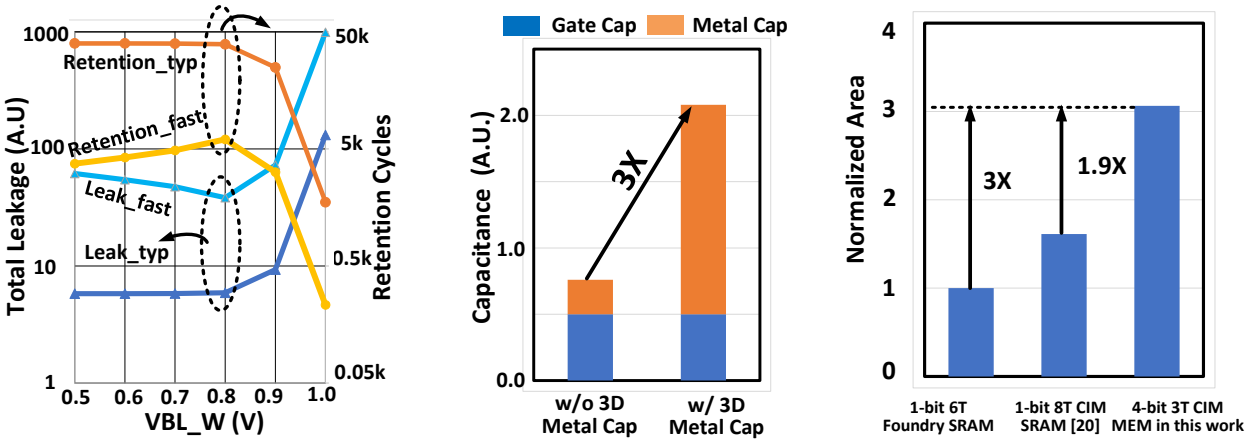


Figure 7.3 Simulation of proposed DARAM, (a) leakage simulations over different design corners, (b) capacitance improvement, (c) area comparison between DARAM and prior design.

Simulation results of the proposed DARAM are depicted in Figure 7.3. The DARAM cell has an area of 1.9× of previous 8T CIM cell and 3× of foundry 6T SRAM cell leading to an effective bit area of 47% of the 8T CIM cell and 75% of foundry 6T SRAM cell [20]. During write, write-access transistor M3 is used to write the analog voltage from BL_W into the "MEM" node from a column-wise DAC with an adjustable voltage range from 0.45V to 1V. Each write can be finished within one clock cycle with totally 64 clock cycles for writing into the entire CIM macro.

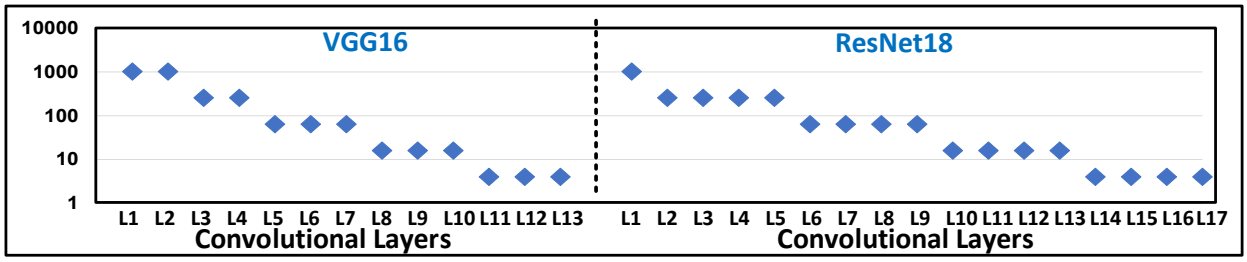


Figure 7.4 Stationary cycles of weights on CNN models.

Subthreshold and gate leakage are minimized to maintain a constant analog voltage during the life cycles of stationary weights for the CNN operation. As shown in Figure 7.4, the weight stationary cycles of CNN models, e.g. VGG16, ResNet18 vary from a few tens of cycles to thousands of cycles for a single image and increase proportionally with the batch size, driving the retention requirement of the analog voltage. A special 3D inter-layer and inter-digit metal capacitor using M1 to M5 interleaving MEM and GND nodes vertically and horizontally are added inside each DARAM cell to enhance the storage capacitance by 3×. As shown in Figure 7.3 (a), during CNN inference, a separate biasing of BL_W at $0.8V$ leads to about 20× reduction of subthreshold leakage current. This allows a retention time of ~41k cycles (for a voltage drift less than half of a single bit) at typical corner and more than 5k cycles at fast corner. As a result, a batch size of 5~40 images can be processed without a rewrite (refresh) operation with negligible accuracy loss. For a larger batch size, a 64-cycle DARAM refresh operation is needed at every 5.5k-41k cycles, leading to a throughput overhead of less than 1.2% or CIM macro energy overhead of less than 0.4%. Note for smaller batch size or CNN layers with less stationary weight, refresh is not needed.

7.3 Dynamic Analog RAM-Based CIM Architecture Design

Figure 7.5shows the architecture of the CNN accelerator with 4 CIM macros. Each CIM macro contains a 64×32 DARAM array. A row-wise digital-time-converter (DTC) is used to convert a 4-bit activation into a time pulse with 50ps resolution. A 5-bit SAR ADC and a 4-bit current DAC are implemented at each column to provide MAC read-out and analog write-in. The design natively supports 4bit input/weight operation and can also support 8b/8b by combining

two DARAM cells and operating in successive two cycles. Similar to prior schemes, global SRAMs are used to store weight and input/output activation data before being fetched into CIM macro. As depicted in Figure 7.6, an ASIC core is used to manage data sequencing and pre/post-processing including (a) offsetting of data values due to the non-2's complementary format of weight in comparison with the support of both non-2's and 2's complement formats in prior works [20, 21]. The offset calculation has negligible overhead as it is commonly shared by all the columns; (b) 4bit to 8bit conversion if needed; (c) Accumulation at inter-macro loop similar as in [21].

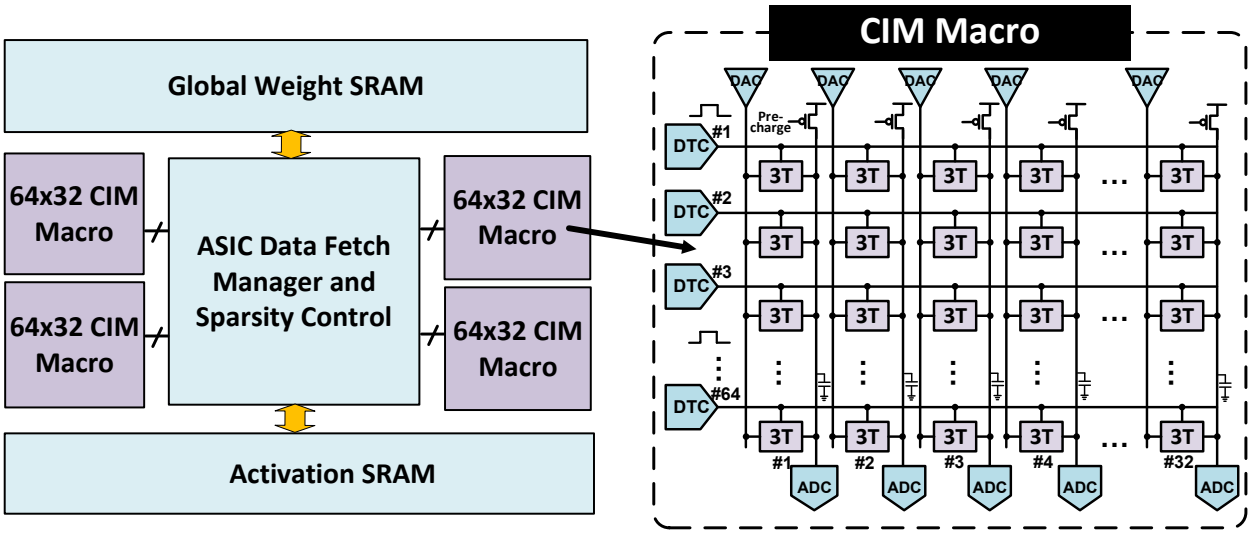


Figure 7.5 Top-level architecture diagram of proposed CIM-based CNN accelerator.

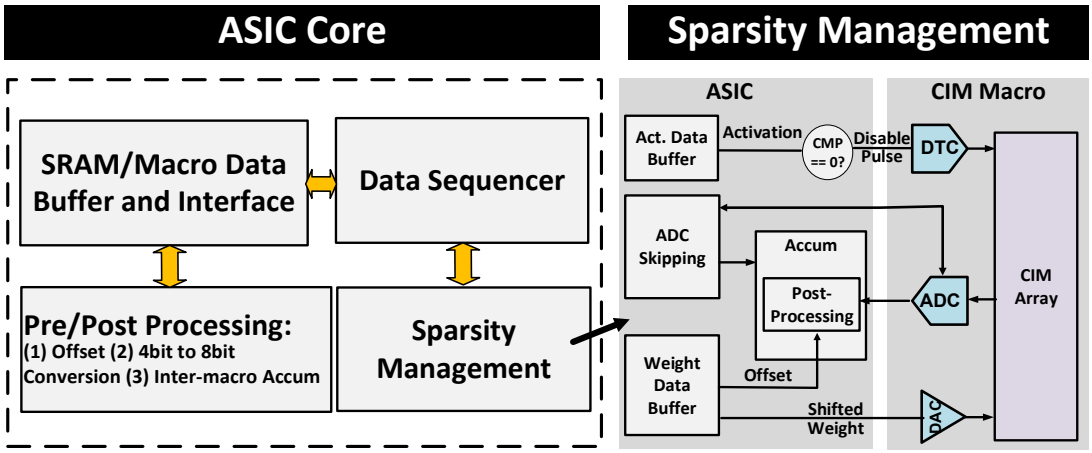


Figure 7.6 Sparsity management module in ASIC core.

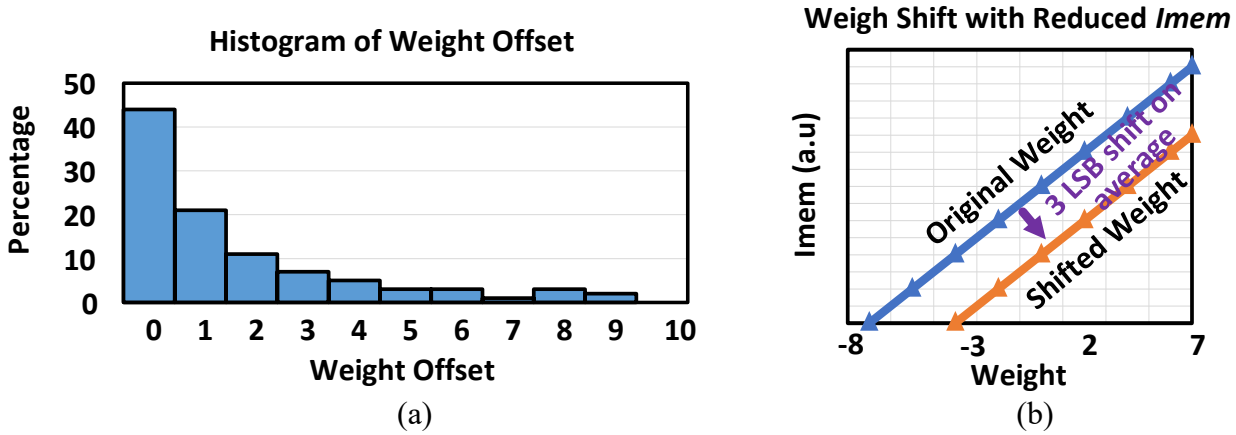


Figure 7.7 (a) Histogram of weight offset, (b) weight-shift-based I_{mem} reduction based.

Additional three features are introduced in this work. (1) An input-stationary operation mode is supported, which is more efficient for later layers in VGG/Resnet. (2) Because the MAC energy consumption using analog weight favors lower weight value compared with zero weight in digital SRAM, a special analog weight shifting technique is introduced where the weights are shifted down whenever the weight range in a column is not fully utilized. The shifted weights are pre-determined off-chip according to the weight being used and associated MAC offsets are added back in ASIC to restore the values. As shown in Figure 7.7, an average of 3-bit weight

shifting is achieved providing a 1.3× energy reduction on the MAC operation of the DARAM cells. (3) Input sparsity is also leveraged by detecting zero inputs from ASIC and disabling row-wise DTC and the associated MAC operation in the CIM macro.

7.4 Dynamic Analog RAM-Based CIM Energy Saving Techniques

Figure 7.8 presents the ADC skipping technique exploiting "analog sparsity" in MAC operation to save the dominant ADC power in CIM macro.

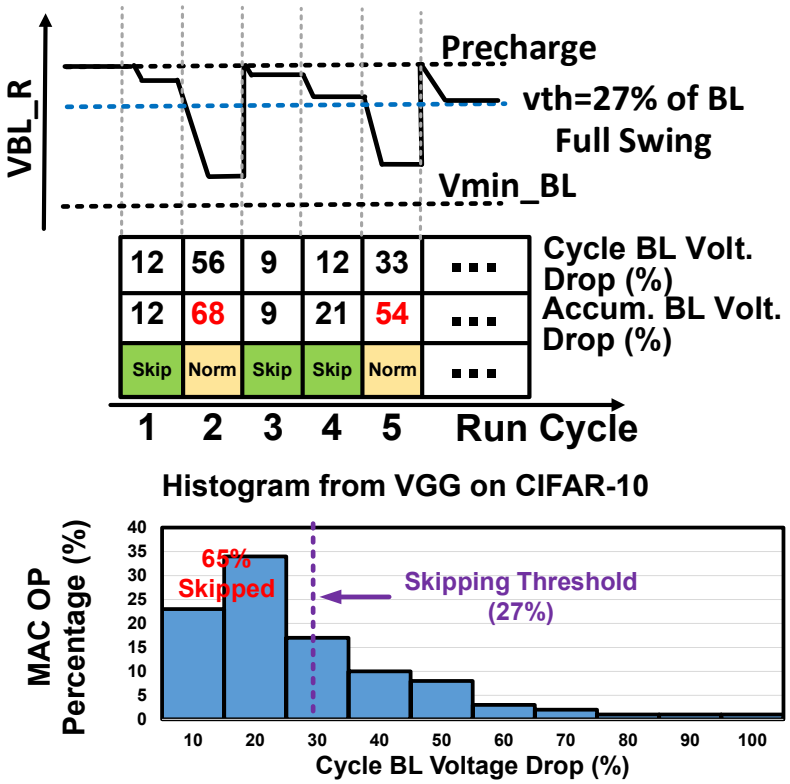


Figure 7.8 MAC-based ADC skipping scheme.

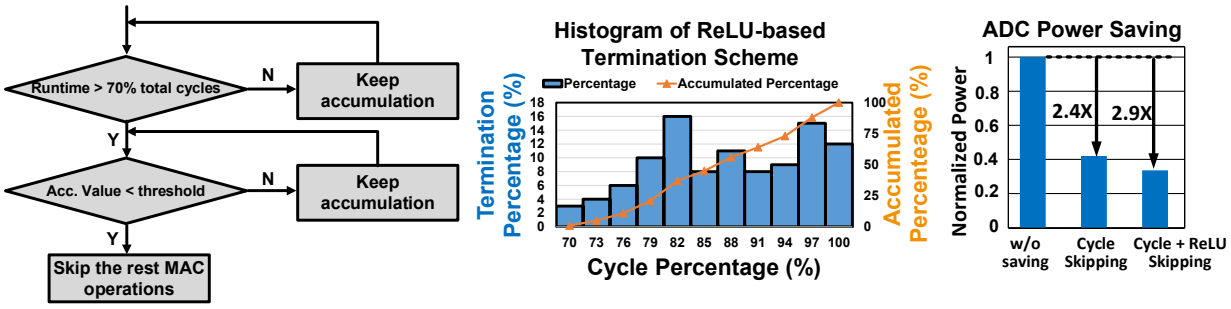


Figure 7.9 ReLU-based ADC skipping scheme.

As shown in the histogram of the bitline voltage drop, i.e. analog MAC value, based on the VGG model, over 60% of the cases have bitline voltage drop less than 27% of full swing leading to the possibility of merging two or more MAC accumulation without activating ADC and bitline precharge with small accuracy impact of 0.1~0.4% from occasional overflow. Different from [19] which only reduces ADC conversion steps at low MAC value, this work skips entire ADC operations leading to higher energy saving, i.e. an average ADC power reduction of 2.4×. In addition, as depicted in Figure 7.9, we apply an early termination of MAC operation based on the ReLU function, i.e. the accumulation has become negative enough that the sign of accumulation results cannot be flipped by remaining MAC operations. The detection is performed in ASIC with a preset negative threshold. Combining both approaches, an average of about 2.9× saving can be achieved on ADC energy consumption.

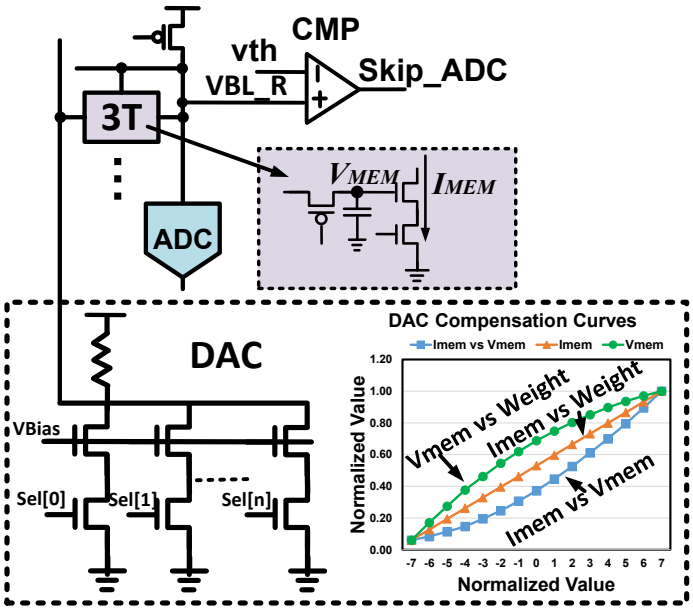


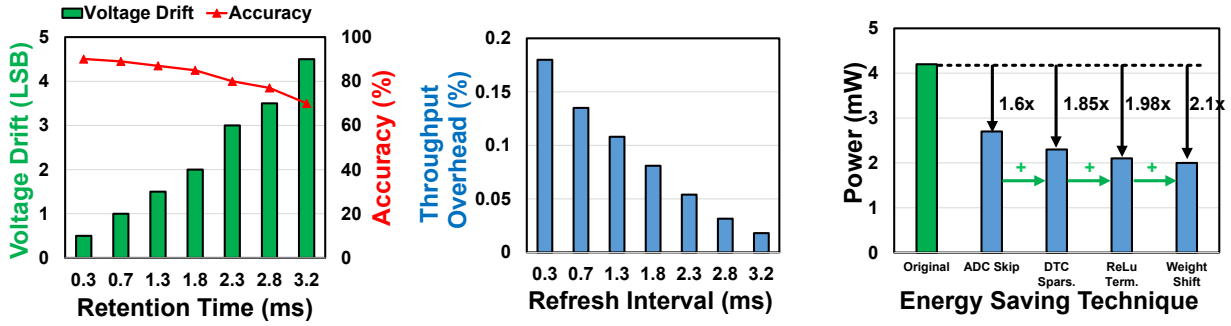
Figure 7.10 Weight nonlinearity compensation technique for DARAM.

Figure 7.10 shows a nonlinearity compensation scheme where the nonlinear relationship between the bitline current and MEM voltage from the read transistor M2 is compensated by a non-linear analog voltage generated from the DAC. As a result, a highly linear I_{mem} versus the weight is achieved.

7.5. Measurement Results

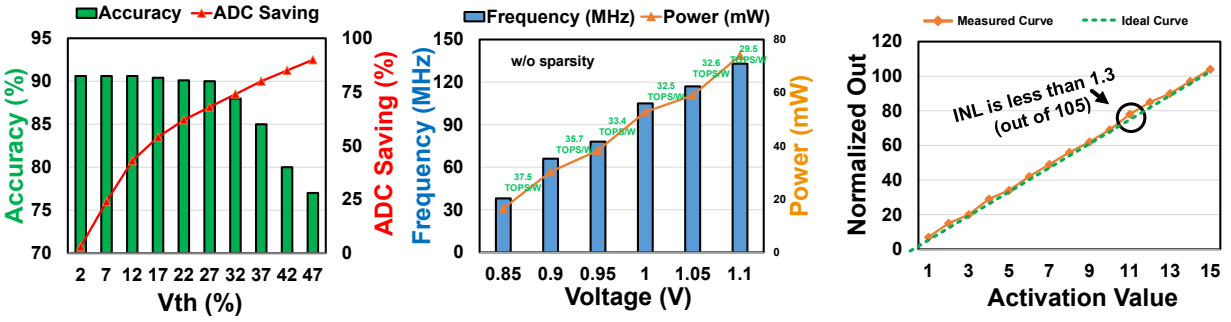
A 65nm CMOS test chip was fabricated to demonstrate the DARAM in a CNN accelerator running at 105*MHz* at 1*V*. Calibration was performed to remove variation impacts, e.g. ADC, DAC offset, etc. by adding small offsets in ASIC. As shown in measurement results in Figure 7.11, a retention time of up to 0.36*ms* (38k cycles) without refresh was observed with negligible accuracy degradation supporting a batch size of 37 images in VGG16. With larger batch size, the refresh operations incurred only up to 0.17% throughput overhead. The ADC skipping scheme

brings 65% saving of ADC energy with less than 0.4% accuracy impact using a 27% MAC value as the skipping threshold. Combining all the sparsity features, the macro power was reduced by 2.1× on average under VGG16 model. As shown in Figure 7.12, the CNN accelerator was measured from 1.1V down to 0.85V showing a system efficiency from 29TOPS/W to 37TOPS/W without sparsity enhancement.



(a) (b) (c)

Figure 7.11 Measurement results: (a) DARAM cell retention time, (b) weight refresh overhead, (c) CIM macro power improvement.

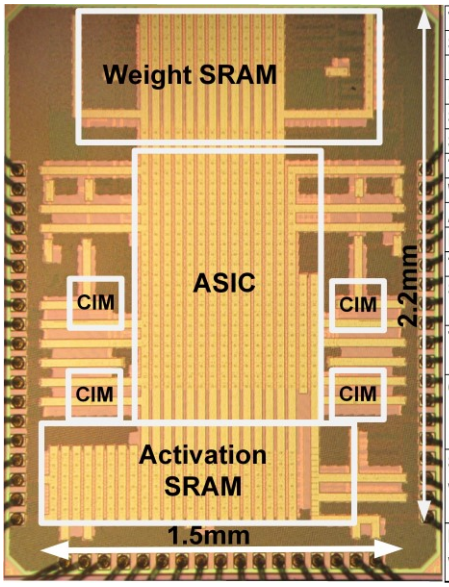


(a) (b) (c)

Figure 7.12 Measurement results: (a) ADC saving vs skipping Vth of bitline cap, (b) voltage-frequency scaling, (c) MAC linearity.

7.6 Comparison and Discussion

Figure 7.13 shows the die photo and additional information. Comparison with prior work was made in Table 7.1. Compared to the closest system implementation in [21], at 4-bit weight/input operation, an 8× system energy efficiency improvement at 44.7TOPS/*W* is achieved along with 3× area reduction in macro size. Overall, this work achieves a state-of-art macro efficiency of 217TOPS/*W* at 4 bits, which is more than 3× improved from those reported in closer technologies and is only 32% lower than that reported in a recent 7*nm* technology. In addition, an effective bit cell area smaller than foundry supplied 6T SRAM is achieved.



Technology	65nm LP
Supply voltage	0.85V-1.0V
Nominal Freqency	105MHz
Processor area	1.5mm x 2.2mm
Single CIM macro area	0.23mm x 0.22mm
Single CIM macro array size	64 x 32
Total effective CIM bits	33kb (=4x64x32x4)
Weight bit resolution	4 or 8 bit
Activation bit resolution	2 or 4 or 8 bit
Number of channels	64
Total digital SRAM	172KB
Single macro power	4.2mW (w/o sparsity) 2.03mW (w/ sparsity)
Total system power	52.8mW (w/o sparsity) 38.4mW (w/ sparsity)
CIM macro energy efficiency at 1V w/o and w/ sparsity (TOPS/W)	w=4b, in=4b: 102.2 / 216.8 w=8b, in=4b: 53.1 / 112.2 w=8b, in=8b: 26.1 / 54.8
System energy effciency at 1V w/o and w/ sparsity (TOPS/W)	w=4b, in=4b: 32.5 / 44.7 w=8b, in=4b: 16.1 / 22.3 w=8b, in=8b: 8.3 / 10.9
Measured accuracy (CIFAR-10) with 4b weight and input	90.6% (VGG16) 91.2% (ResNet18)

Figure 7.13 Die photo and chip specifications.

7.7 Summary

In this Chapter, a dynamic analog RAM based Computing-In-Memory macro and associated CNN accelerator is demonstrated in a 65nm CMOS test chip. With special analog

sparsity techniques and retention enhancement, the design achieves state-of-art energy efficiency of 217TOPS/W at CIM macro level and 44 TOPS/W at system level for 4 bits weight/input operation. An effective bit cell size of only 75% of 6T foundry SRAM cell is also achieved.

TABLE 7.1 COMPARISON TABLE OF PROPOSED 3T DARAM CIM CNN ACCELERATOR.

	[23]	[19]	[20]	[21]	This work
Memory Bit	8T SRAM	6T SRAM	Twin-8T SRAM	8T SRAM	3T Analog RAM
Tech. (*nm*)	7	28	55	65	65
Frequency (*MHz*)	222	240	-	100	105
System Area (*mm2*)	-	-	-	9	3.3
Size of Macro (bit)	64×64	512×64	64×60	64×64	4×64×32
Area of Macro (*mm2*)	0.0032	-	-	0.148	0.05
Activation Precision (bit)	4	4/8	1/2/4	2/4/6/8	4/8
Weight Precision (bit)	4	4/8	2/5	4/8	4/8
ADC Precision (bit)	4	5	5	5	5
Digital Storage	-	-	-	164KB	172KB
Sparsity Support	-	-	-	Activation + weight	Activation + Weight + ADC
Power of CIM Macro (*mW*)	-	-	-	3.8	4.2 (raw)
Energy Efficiency of CIM Macro at 4bit Weight/Input (TOPS/*W*)	321	68.44	22.96	25.83	102.2 * (raw)

Chapter 8

Digital Compatible Synthesis, Placement and Implementation of MSTC

In this chapter, a comprehensive design flow for MSTC is presented. In the frontend, a variation-aware digital compatible synthesis flow is proposed. In the backend, a placement technique using graph-based optimization engine is proposed to deal with the especially stringent matching requirement in MSTC. Simulation results show significant improvement over the prior analog placement methods. A 55*nm* test chip is used to demonstrate that the proposed design flow can meet the stringent timing matching target for MSTC with significant performance boost over conventional digital design.

Chapter 8 is organized as follows: the challenges and background of design automation in time domain is carried out in Chapter 8.1. The synthesis methodology is proposed in Chapter 8.2, and the backend design automation methodology in presented in Chapter 8.3. Chapter 8.4 shows the experimental results with summary written in Chapter 8.5.

8.1 Design Automation in Mixed-Signal Time-Domain Computing

8.1.1 Challenges of Time-domain Computing Design Automation

As MSTC relies on the precise timing control for information processing, variation and mismatch of signal timing could lead to computation errors. As the least-significant-bit (LSB) resolution is pre-defined, e.g. 25*ps* used in this work, a variation of timing beyond this value will lead to single-bit error. Specially, local variation or mismatch creates the largest threat to the

operation similar to analog computing. Comparing to digital design, a much more stringent backend layout is needed in consideration of matching, variation, crosstalk and signal slew rate. In addition, as MSTC usually performs more complex algorithms [11, 14], the signal paths and matching components in MSTC are much more complicated than a typical analog design leading to more challenges in the front-end or back-end design for MSTC.

8.1.2 Proposed Digital Compatible Design Methodology

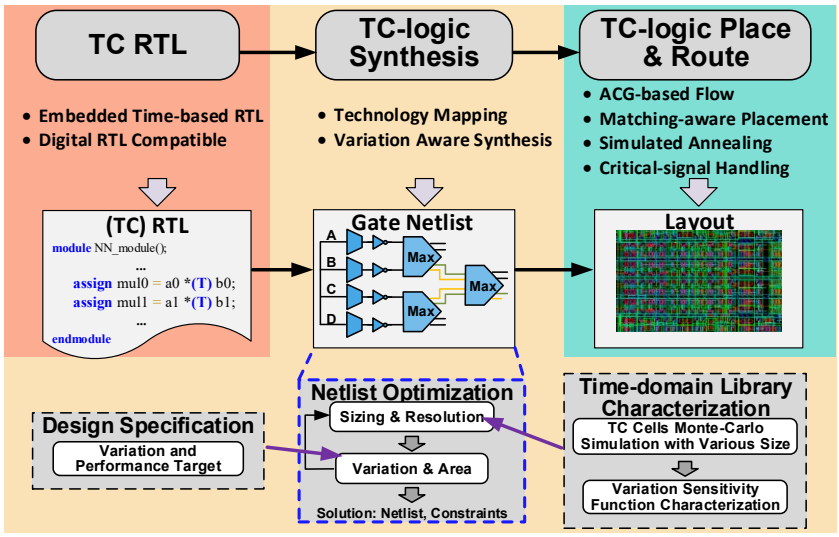


Figure 8.1 Flowchart of proposed MSTC automation flow.

Figure 8.1 shows the overview of the proposed digital compatible design automation flow. Particularly, a specially developed time-domain RTL code is attached to conventional Verilog language to denote the special design of the time-domain logic operation. Based on the hybrid RTL codes, the synthesis tool provides logic synthesis and technology mapping to create a gate-level netlist using both standard cells and digital friendly time-domain modules. Variation awareness is implemented into the synthesis process. At the back-end, an ACG-based placement technique is developed to handle the stringent signal matching requirement of MSTC design.

8.2 Synthesis of Time-Domain Logic

To create a digital-compatible design flow for MSTC design, synthesis needs to create gate-level netlist similar to the conventional digital design. The proposed technique is realized by embedding a special plug-in script into existing RTL/synthesis flow. It handles not only the generation of time-domain cells but also special requirements in MSTC, such as variation.

8.2.1 Overview of Proposed MSTC Synthesis Technique

The bottom of Figure 8.1 shows the flow of the proposed synthesis technique: (1) the RTL with customized syntax for time-domain logics is utilized to perform a special MSTC logic synthesis process. As a result, both conventional digital and time-domain logics are synthesized into an initial gate-level netlist. The size of each cell is set to the smallest size at this step. (2) The initial netlist is then sent to a netlist optimizer to exercise the sizing options of each module to meet the variation budget while minimizing area consumption.

8.2.2 Implementation of MSTC Synthesis

The proposed logic synthesis script can recognize special syntax used for the MSTC RTL. In the MSTC RTL, a special syntax is developed to denote the MSTC operation, e.g. add and multiplication. The special keyword “(T)” after the operation symbol “+” or “×” is used to denote the MSTC operation as shown in Table 1. The synthesis script works as a plug-in script on top of conventional synthesis tool. Special mapping functions are called for generating time-domain circuits similar to the conventional technology mapping. For instance, the “?” operation symbol in time-domain RTL, is mapped into a time-domain.

The variation sensitivity function is introduced for netlist optimization. We define the 3-sigma variation of MSTC modules, which is a function of the size s as $\sigma(s)$. Apparently, the $\sigma(s)$ decreases as s increases. The area of MSTC modules is a function of the size s as $A(s)$. The variation sensitivity function is shown as:

$$F_{sen}(s) = \gamma \frac{d\sigma(s)}{dA(s)} \tag{8.1}$$

where $\frac{d\sigma(s)}{dA(s)}$ term represents the variation sensitivity comes from the module, and γ term represents the significance of the module, e.g. module in a convergent path. As most MSTC cells are standard-cell like, we follow the standard cell sizing convention, e.g. 1×, 2×, etc.

Assume that we have totally n modules, X_1, X_2, … X_n, the size of each module is s_1, s_2, … s_n. Besides, there are p critical paths need to be considered in the placement. The optimization problem of netlist is then formed in (8.2) and (8.3):

$$\text{Minimize } \sum_{i=1}^{n} A(s_i) \tag{8.2}$$

$$\forall\, paths \in P, s.t. \sqrt{\sum_{i=1}^{n} {\sigma_p}^2(s_i)} \leq \sigma_T \tag{8.3}$$

where $\sigma_p(s_i)$ is the variation comes from Xi, and $A(s_i)$ is the area of Xi. The pseudo code of the optimization is shown as follows.

TABLE 8.1 NETLIST OPTIMIZATION ALGORITHM

Algorithm 1 Netlist Optimization Algorithm

Input: Initial netlist of module X_1, X_2, ...X_n, with minimum sizing $s_1, s_2, \ldots s_n$.
Output: Netlist which satisfies variation budget with minimum area
1: for all critical paths p in the netlist **do**
2: **while** $\sqrt{\sum_{i=1}^{n} \sigma_i^2(s_i)} > \sigma_T$ **do**
3: **for** i = 1 to n **do**
4: find the module $j = i$, with maximum $F_{sen}(s_j)$
5: **end**
6: Increase the size of module j by 1×, update s_j
7: **end**
8: end
9: Return the netlist with current sizing

TABLE 8.2 EXAMPLE RTL IMPLEMENTATION OF MSTC-NEURAL NODE.

```
1   module NN_module (a0, a1, a2, a3, b0, b1, b2, b3, out);
2    input [1:0] a0, a1, a2, a3;
…   …
6    assign mul0 = a0 *(T) b0;
…   …
11   assign mac1 = mul2 +(T) mul3;
12   assign out = (mac0 >= mac1) ?(T) 0 : 1;
13  endmodule
```

TABLE 8.3 EXAMPLE NETLIST OF MSTC-NEURAL NODE FROM SYNTHESIS.

```
1   module NN_module (a0, a1, a2, a3, b0, b1, b2, b3, in, out);
2    input [1:0] a0, a1, a2, a3;
…   …
6    TC_TE_X3 I0 (.IN(in), .DIN(a0), .OUT(te0));
…   …
13   TC_MUX_X4 I7 (.A(mul2), .B(te3), .S(b3), .OUT(mac1));
14   TC_CMP_X2 I8 (.a(mac0), .b(mac1), .out(out));
15  endmodule
```

Given the initial netlist generated by MSTC logic mapping from MSTC RTL with minimum sizing, we first check if the variation of all critical paths meets the budget σ_T. If yes, the optimization is completed. Otherwise, the following step is performed in which we traverse

the netlist to find out the most effective module in the critical path, i.e. highest variation sensitivity. The size s of this module is then increased by 1×. We keep repeating the previous steps until the variation targets of all critical paths are met.

8.3 Proposed Mixed-Signal Placement

Due to the lack of prior techniques on automatic placement for MSTC circuits [11, 14, 13, 31], in this section, we propose a practical and efficient placement technique for MSTC circuit utilizing adjacent constraint graph (ACG) based optimization engine to deal with the stringent matching requirements. It is worth to mention that although automatic placement has been proposed previously for analog/mixed-signal design [35, 36], MSTC poses special challenges, i.e. massive-stage-symmetry (MASS), as referred in this paper, and hence requires special techniques not available from the prior work. The special matching requirement of MASS for time domain circuits are highlighted as follows:

1) Module symmetry and stage symmetry constraint: modules within certain groups must be placed symmetrically with respect to a horizontal or a vertical axis to maintain the matching of critical MSTC signal. Moreover, modules on symmetry paths need to be place symmetrically in each stage.

2) Clustering constraint: certain MSTC modules must be placed near to each other in order to isolate the critical MSTC modules from other digital modules.

3) Shortest critical signal path constraint: the wire length of critical paths must be minimized in order to relieve the variation impact of MSTC circuit and improve slew rate of the signals.

Similar constraints are observed in the existing analog placement/floorplan design, but MSTC design has more challenges due to its larger numbers of components as described in the follows.

8.3.1 Preliminaries

Topological representations are widely used in solving analog placement problems, in which, the relative positions between the modules are encoded. Typical topological representations are slicing tree [37], sequence-pairs (SP) [38], O-tree [39], B*-trees [40], and TCG-S [41]. Most of these works have been applied to handle the symmetry constraint and other constraints like the centroid constraint. However, these representations are not suitable for solving the MASS placement problem of MSTC design as explained as follows.

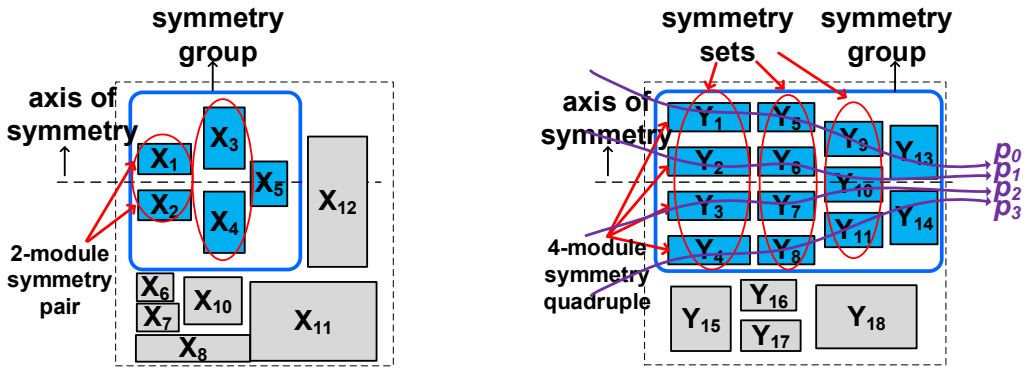


Figure 8.2 Symmetry group in (a) conventional analog design, (b) time-domain computing design.

1) A complete representation is preferred in order to efficiently handle the special constraints like symmetry and critical path constraints. For example, tree-based representation doesn't provide complete topological information, which makes it harder to check the relations, e.g. horizontal relation, between modules.

2) When dealing with symmetry constraint, we form a symmetry group with multiple symmetry pairs. However, in most of analog placement problem, each symmetry pair in the symmetry group only contains few modules as shown in Figure 8.2 (a). On the other hand, in the MSTC design, large numbers of modules, defined by the algorithm, e.g. LDPC [11], need to be allocated symmetrically through hierarchies as shown in Figure 8.2 (b).

3) For MSTC design, we not only need to place the modules symmetrically within a set, but also need to guarantee the matching across different hierarchy on the long signal paths. As shown in Figure 3.3 (b), the modules on path p0 must be symmetric with the modules on paths p1 – p3 leading to stringent multi-path matching problems for sequence of modules. This not only requires a massive symmetry placement within a symmetry group but also requires carefully match at each stage. Thus, the MASS becomes a special challenge in the MSTC placement.

Adjacent Constraint Graph (ACG) [70]representation is chosen in this work due to the following advantages: compared with existing placement techniques, ACG has the advantage of efficiency and succinctness when dealing with the symmetry and other constraints. Without the redundant edges, the number of edges in ACG is O(nlog(n)), much smaller than the O(n2) number of edges in TCG-S or SP. ACG is also more flexible than other representations in performing packing.

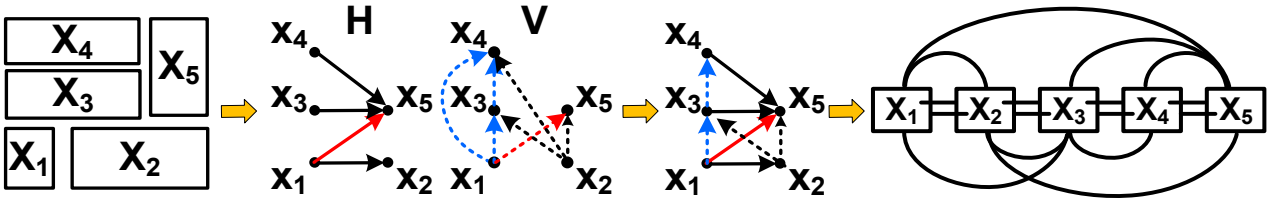
Assume we are given a set of n modules with areas Ai where i = 1. . . n, together with a set of j nets N1, N2 . . . Nj. Our objective is to obtain a placement F of the circuit satisfying all the placement constraints mentioned previously while minimizing a cost function:

$$\mathrm{C}(F) = \mathrm{A}(F) + \alpha \times \mathrm{W}(F) + \beta \times \mathrm{W\_penalty}(F) \tag{8.4}$$

where A(F) is the total area of F, W(F) is the total wire length of F, W_penalty(F) is the total wire length of wires between the modules which violated the constraint after the packing stage. α and β are empirical coefficients used for regulating the weights of wire length and wiring violation.

8.3.2 Adjacent Constraint Graph (ACG) Representation

The basic idea of the ACG representation, briefly described below, is to encode any rectangle packing as an ordered module sequences with edges which indicates the spatial relations [70].



(a) (b) (c) (d)

Figure 8.3 (a) A floorplan, (b) constraint graphs in horizontal (solid edges) and vertical (dotted edges) directions, (c) ACG Graph, (d) ACG data structure.

As an illustration, for a floorplan given in Figure 8.3 (a), its constraint graph in both horizontal and vertical directions are shown in Figure 8.3 (b). As the essential idea of constraint graph is used for avoiding module overlap, any two modules must have at least one relation ("left" or "below to"). Thus, over-specification has no benefit in terms of representation. Since those redundant edges are unnecessary for placement, we can remove those edges and the result is an ACG representation (Figure 8.3 (c)). The corresponding ACG data structure is shown in Figure 8.3 (d). The vertices will be doubly linked in a linear order. Edges are all directed from left to right. The edges above the vertex line represent horizontal (H) relations and those below

represent vertical (V) relations.

8.3.3 Proposed MSTC Placement Approach

Simulated annealing is employed as the basic searching engine in our approach with ACG as the representation. Our proposed placement algorithm works as follows. It first generates an initial ACG representation following the default cells order, which also satisfies all the constraints proposed by the designer. After the initial solution is generated, the simulated annealing process is applied. In each iteration the following steps are performed: (1) three categories of perturbations/moves are introduced. All these perturbations are complete in terms of the searching space; (2) After the perturbation, a new ACG is generated and the corresponding packing is produced based on the longest path algorithm; (3) Area and interconnect cost with extra penalties are computed based on the new packing. (4) Check whether the annealing process should continue based on the current temperature and cost. The flowchart is shown in Figure 8.4.

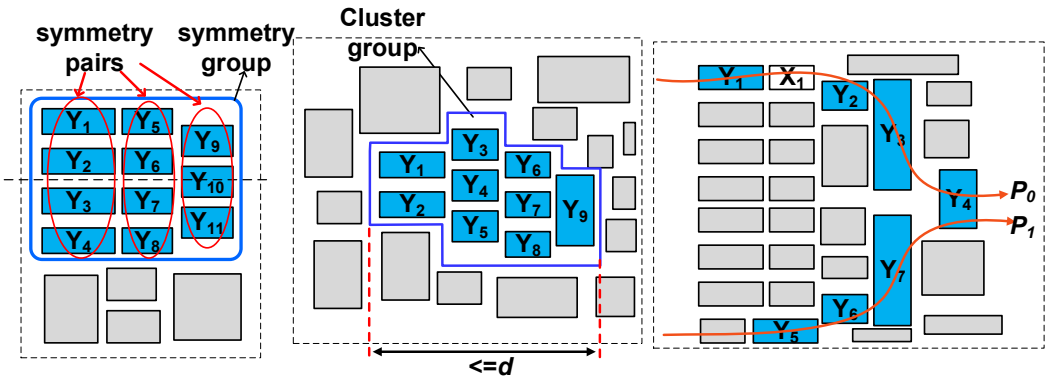
8.3.4 Handling of Placement Constraints in MSTC

In MSTC circuit, symmetry constraint (marked in blue in Figure 8.5 (a)) can be handled as follows (we assume the symmetric modules are symmetric with respect to a horizontal axis):

1) If modules Y_1, Y_2, Y_3, and Y_4 are required to be symmetric, all of them must be in vertical relations. In the other word, every two of them must be connected by horizontal edges in the ACG.

2) The x coordinates of modules Y_1, Y_2, Y_3, and Y_4 must be same which can be regulated during the packing stage.

3) The distances between adjacent modules must be same.



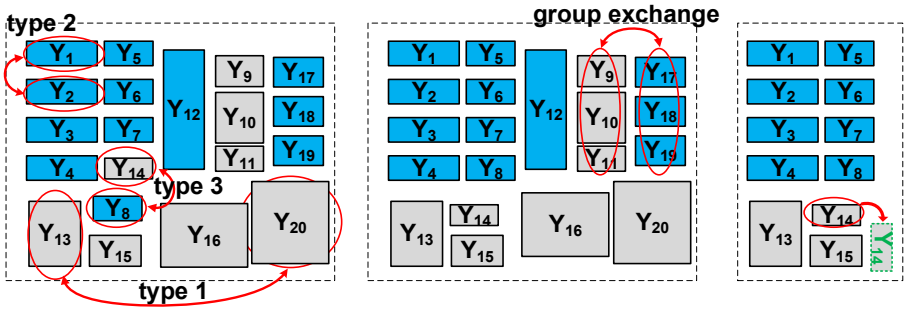
(a) (b) (c)

Figure 8.4: Example of (a) symmetric constraint, (b) clustering constraint, (c) critical signal path constraint.

Clustering constraint can be handled by forcing the modules in the same clustering group to abut each other in ACG representation. Besides, we introduce the penalty term in the cost function to force the placement to obey the constraint. An example of clustering constraint among modules Y_1-Y_9 is shown in Figure 8.5 (b).

To handle this constraint, the total wire lengths of these paths need be as short as possible (P_1 and P_0 in Figure 8.5). The constraint can be handled by (1) guaranteeing horizontal relations for the modules in same critical path in ACG, e.g. Y_1, Y_2, Y_3 and Y_4; (2) increasing the weight of nets which are on the critical paths when calculating the cost of total wire length. As a result, the placement engine tends to move the modules which are not on critical signal path, e.g. X_1, away from the critical path P_0.

8.3.5 Set of Perturbations/Moves



(a) (b) (c)

Figure 8.5 Example of moves in (symmetry group are marked in blue): (a) category 1, (b) category 2, (c) category 3.

We employ the following set of moves to perturb a current candidate ACG. The moves/perturbations can be divided into three categories: (a) exchange of two random modules, (b) group exchange of the symmetric sets, and (c) editing edges in the current ACG representation. The details of moves are given as follows:

1) In the first category (Figure 8.5 (a)), there are three different types of exchanges: (1) Exchange two random modules which are not in any of the symmetry groups. (2) Exchange two random modules within a symmetric set. (3) Exchange one module which is inside of one symmetry group and another module which is outside of that symmetry group. This movement cannot be guaranteed to not violate the symmetry constraint. Thus, a special checker is implemented to check the feasibility of the new generated ACG. If such a move violates the constraints, penalty will be added to the cost function shown in eq. (8.4).

2) Figure 8.5 (b) shows one example of second category. This group exchange also needs special checker to check the feasibility of the new ACG after such a move. It provides the chance of moving away the modules which are located inside of a symmetry group.

3) The third category involves the modification of ACG edges including (1) changing current edge type from horizontal to vertical or vice versa; (2) Adding or removing the existing current edges while following the ACG requirement. We only allow modifying the edges of the modules which are outside of symmetry group. In this way, all the constraint within the symmetry group cannot be violated. An example of modify the edge between Y14 and Y15 from vertical to horizontal is shown in Figure 8.5 (c).

8.3.6 Packing and Routing

A new packing algorithm is derived from conventional packing scheme based on the longest path algorithm. Different from previous work, the proposed packing algorithm allows us to pack the selected modules in respect to the symmetry axis instead of only to the lower bottom corner of plane [38, 40]. The packing example of conventional and our proposed ways are shown in Figure 8.6 with symmetric modules marked in blue.

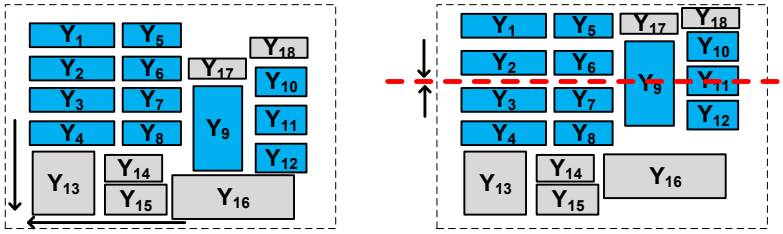


Figure 8.6: Example of packing (a) to lower-bottom corner, and (b) respect to the symmetry axis.

We utilize the Innovus tool to handle the routing job. Since the MSTC cells follow the digital cell's implementation and are well organized after the proposed placement, e.g. the cells on the same critical path are placed abut to each other, the Innovus tool can handle the routing job appropriately.

8.4 Experimental Results

8.4.1 Time-domain WTA Operation Implementation

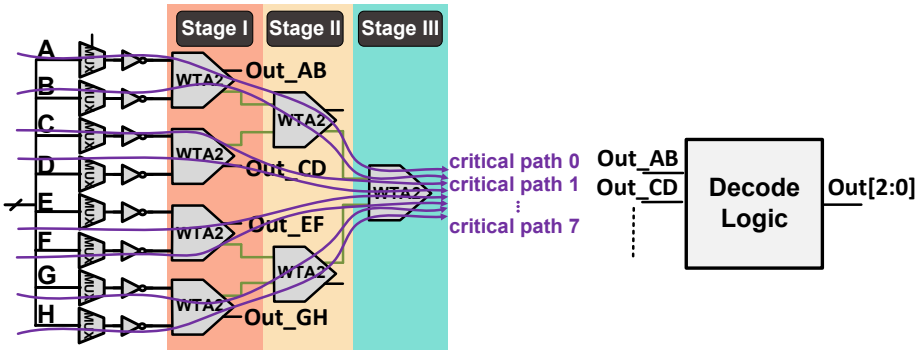
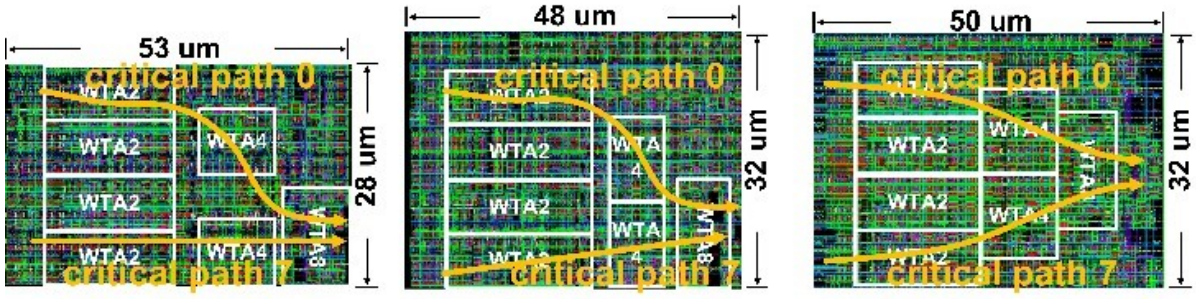


Figure 8.7 Topology and implementation of WTA in MSTC.

We compare our proposed ACG-based placement flow to other existing work [38, 40] on a winner-take-all (WTA) circuit, which is a commonly used digital module in machine learning based classifiers. Figure 8.7 shows the design of the 8-input 6-bit WTA. The algorithm of WTA is based on binary comparison tree. The critical signals are propagated through 3 stages and the matching of 8 critical paths is the key concern of the design. The total number of critical digital modules for matching are 84 which is much larger than a typical matching problem observed in an analog design.

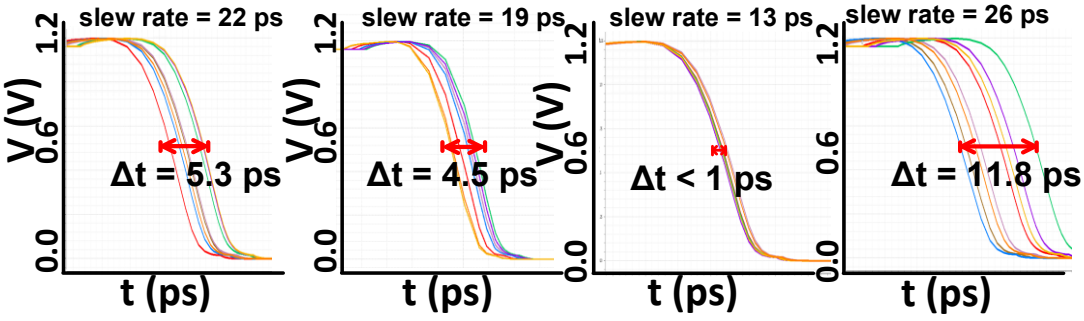
We experiment the placement of WTA by different approaches: (a) use B* tree based placement method from [40], (b) use sequence pair (SP) based placement method from [38], (c) use the proposed placement method. The layout results of approaches (a), (b) and (c) are shown in the Figure 8.8. All the methods maintain a good symmetry property in the 1st stage (WTA2). However, both B* tree based and SP based placement methods have troubles in placing the modules properly in the stages 2 and 3 as (1) the modules in 2nd and 3rd stages are not placed in the central region with respect to the 1st stage leading to large signal routing mismatch between

critical signals; (2) The critical MSTC modules are not separated with other non-critical modules causing the slew rate degradation of the critical signals. These failures are mainly due to the following reasons: (1) both previous placement approaches pack the modules from lower bottom corner leading to difficulty in placing the selected modules in respect to the symmetry axis; (2) Both previous placement methods are short of the ability to deal with the clustering and critical-path constraints.



(a) (b) (c)

Figure 8.8 Layout of placement methods: (a) B* tree based [40], (b) sequence pair based [38], (c) proposed design in this work.



(a) (b) (c) (d)

Figure 8.9 Simulation result of mismatch for (a) B* tree based placement [40], (b) sequence pair based placement [38], (c) our proposed technique, (d) conventional digital design.

As a result, they failed to place the critical time-domain modules to be close to each other avoiding non-critical modules to block the critical paths. On the other hand, due to the efficiency and succinctness of ACG-based representation, it's much easier to handle the cluster and critical

path constraints. As a result, the above issues can be properly resolved by the proposed ACG-based placement with good matching through stages of critical paths (Figure 8.8 (c)).

After the layout is generated from Innovus, we import the layout back into Cadence Virtuoso to perform spice simulation with parasitic extraction. The simulation result of matching for the 8 critical paths is shown in Figure 8.9 in comparison among B* tree method, SP method, proposed method and conventional digital design using EDA tools. As we can see, the mismatch from using B* tree based and SP based placement method are better than that from the conventional digital flow. However, the mismatch from these two methods are still significantly larger than our proposed ACG-based placement method whose mismatch is less than 1*ps*. Thus, the proposed placement methodology provides both the efficiency and accuracy in dealing with MSTC design. Table 3 summarizes the performance of different methods. The algorithms are implemented in C++ and run on a Windows machine with 2.6GHz i7 Quad-core and 8GB RAM. Note that ACG-based placement method also achieves the lowest runtime mainly due to the efficient and succinct representation when deal with complex matching constraints. For example, the number of edges in ACG is O(nlog(n)), while it's O(n2) in SP. Even though the edge number is only O(n) in B* tree, it lacks a complete topology information used for dealing with MSTC constraints which makes the number of searching iteration larger.

TABLE 8.4 PERFORMANCE COMPARISON FOR PLACEMENT METHODS.

Methods	B* tree [40]	SP [38]	This work
Mismatch (*ps*)	5.3	4.5	1
Slew rate (*ps*)	22	19	13
Run time (*s*)	23	85	18
Area (um^2)	1484	1536	1600

8.4.2 Time-domain Image Processing Implementation

For demonstration, we adopt a basic facial recognition algorithm into a hybrid ASIC design with time-domain accelerators. The operations of the image recognition algorithm involve three steps: (1) feature extraction which performs median filtering and detects edges in four directions. (2) Vector formation; (3) Classification where the generated feature vector is classified by a winner-take-all (WTA) classifier. In our design, the median filter for feature extraction and WTA for final classification were designed in time-domain to remove the bottlenecks of the algorithm [26]. In particular, the proposed synthesis and placement techniques were applied on the WTA design leading to the layout for the fabricated chips.

8.5.3 Measurement Results

The 55nm test chip was fabricated and measured across 10 chips. No error was observed at internal time-domain results or final classification at the design target speed of 1.33GHz.

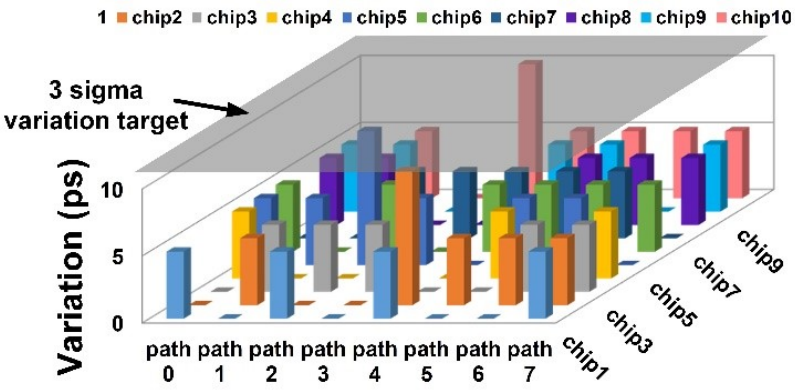
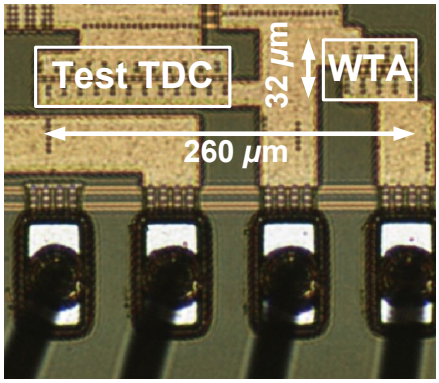


Figure 8.10 Mismatch measurement results; y axis denotes the absolute variation from the nominal delay.

Figure 8.10 shows the measured on-chip mismatch of 8 critical paths from 10 chips in WTA circuits. The mismatches were measured by using an on-chip time-digital-converter (TDC)

with 5*ps* resolution. As shown, the measured mismatch is within 0.5 LSB, which verifies the feasibility of handling variation (synthesis) and layout mismatch (placement) of the proposed methodology. No systematic mismatch was observable from the measurement proving the good matching performance of the placement algorithm. The mismatch was dominated by the random process variation which has been properly budgeted (within half of LSB, i.e. 12*ps* as 3-sigma variation target) from the proposed synthesis flow. The die micrograph and the specification of WTA is shown in Figure 8.11. The design is compared with conventional ASIC with standard synthesis and place and route implementation. A 42% area saving, a 1.7× speedup and a 23% power saving, is observed in the time-domain WTA accelerator compared to ASIC implementation. The overall image recognition processor operates at 1.33*GHz* with a state-of-art throughput of 72 frames per second.



Technology	55 nm	
Frequency (GHz)	1.33	
Total Chip Area (mm^2)	0.64	
	ASIC	TC
WTA Area (μm^2)	2800	1600
WTA Power (mW)	3.1	2.4
WTA Frequency (GHz)	1.2	2

Figure 8.11 Die photo and specifications of the WTA design.

8.5 Summary

This chapter proposes a comprehensive digital compatible design flow including frontend synthesis and backend placement for MSTC. In the synthesis stage, our proposed technique can handle the variation requirement while minimizing the estimated area of the circuit. In the

backend stage, an ACG-based placement algorithm is developed to handle the complex placement constraints for MSTC design. The comparison with prior analog placement schemes shows much improved matching performance from the proposed method. The proposed synthesis and placement flow are demonstrated by a 55nm test chip showing on-target mismatch results and significant performance enhancement from MSTC compared with digital implementation.

Chapter 9

Conclusion and Future Work

Special purpose accelerators have recently gained significant interests thanks to the bloom of machine learning applications. It is predicted that the special purpose artificial intelligence (AI) chips with built-in machine learning accelerators will grow from $6 billion in 2018 to $90 billion in 2025 specially contributed by the edge devices [71]. Conventional digital design started to encounter difficulties in fulfilling the rapid-growing computation demand for these computation-intensive applications. Alternative solutions like mixed-signal time-domain computing have drawn significant attention recently due to their computation efficiency in conducting both arithmetic and non-linear operations.

In this thesis, a comprehensive design realm in MSTC is presented including circuit-, algorithm-, architecture- and design methodology-level innovations.

At the circuit level, high-precision low-cost time-domain operation modules are introduced to efficiently conduct computation workloads in time domain. Time encoding and decoding circuits, i.e. digital-to-time converter and time-to-digital converter, are implemented to convert signals between time and digital domains. Basic arithmetic and Boolean operation modules are realized by succinct digital cells to conduct basic computation in time domain. Most of these time-domain operations improve the energy/area efficiency by over 10× compared to digital counterparts. Borrowing the ideas from analog computing, a sub-threshold time-domain multiplier is proposed in time domain, which renders an improvement of 4× compared to digital multiplier. More complex operations like median filter and multiply-accumulate (MAC) are

implemented based on those element modules. In addition, a special time-domain flip-flop circuit is developed to enable the very first pipeline architecture in time domain. Finally, a multi-bit dynamic RAM memory cell is introduced to conduct Compute-In-Memory (CIM) tasks in time domain, showing 8× of improvement in energy and 3× in area compared to the state-of-art CIM design.

At the algorithm level, many interesting adaptations of complex algorithms like median filter and winner-take-all are developed to show the computation efficiency in time domain. Moreover, some hardware-friendly algorithms are introduced to make the complicated application applicable to time-domain computing. For instance, to overcome the resource limitation of realizing AI applications on edge devices, an adaptive training and a model balancing algorithm for time-domain GAN accelerator design are introduced. Finally, to improve the hardware utilization in conducting AI tasks, some sparsity-aware algorithms are developed. For example, in time-domain CIM accelerator design, the MAC-based and ReLU-based ADC skipping method is introduced, rendering an 50% energy saving for CIM macro.

At the architecture level, the very first pipeline architecture in time domain is introduced, resulting in a 9-20× throughput improvement over prior work in both time and digital domains. A domain-conversion-free architecture that allows MAC operation to be completed computed in time domain, is proposed to remove the energy bottleneck coming from signal conversions.

At the design methodology level, a comprehensive digital-compatible design flow including frontend synthesis and backend placement for MSTC, is proposed to mitigate the original manual design effort. The proposed synthesis and placement flow are demonstrated by a 55*nm* test chip showing on-target mismatch results and significant performance enhancement from MSTC compared to digital implementation.

As for the future work, we are planning to conduct research on a general-purpose mixed-signal time-domain microcontroller used for near-sensor computing. Recently, near-sensor computing draws significant interests for low-power Internet of Things (IoT) devices as it relieves the overhead of data communication by processing the sensor data locally. However, the existing works suffer from the high cost of data conversion, e.g. the use of ADC. In this future work, we are going to propose a near-sensor microcontroller with embedded time-domain computing which eliminates the conversion of signal between digital and analog mixed-signal domains. Time-domain arithmetic logic units and pipelines will be seamlessly implemented into a RISC-V ISA to directedly process the signals generated from sensors. In addition, we will develop a dynamic time scaling technique to bridge the physical world with the digital world. Finally, we are planning to implement this work into a real silicon chip to verify the performance and robustness of the design.

References

[1] R. Hameed and et al., "Understanding sources of inefficiency in general-purpose chips," in *ACM International Symposium on Computer Architecture (ISCA)*, Saint-Malo, 2010.

[2] S. Hashemi, H. Tann, F. Buttafuoco and S. Reda, "Approximate Computing for Biometric Security Systems: A Case Study on Iris Scanning," in *IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Dresden, 2018.

[3] N. R. Shanbhag, R. A. Abdallah, R. Kumar and D. L. Jones, "Stochastic computation," in *IEEE Design Automation Conference (DAC)*, Anaheim, 2010.

[4] F. N. Buhler, P. Brown, T. C. J. Li, Z. Zhang and M. P. Flynn, "A 3.43TOPS/W 48.9pJ/pixel 50.1nJ/classification 512 analog neuron sparse coding neural network with on-chip learning and classification in 40nm CMOS," in *IEEE Symposium on VLSI Circuits*, Kyot, 2017.

[5] D. Bankman and B. Murmann, "An 8-bit, 16 input, 3.2 pJ/op switched-capacitor dot product circuit in 28-nm FDSOI CMOS," in *IEEE Asian Solid-State Circuits Conference (A-SSCC)*, Toyama, 2016.

[6] S. Yu and et al., "Binary neural network with 16 Mb RRAM macro chip for classification and online training," in *IEEE International Electron Devices Meeting (IEDM)*, San Francisco, CA, 2016.

[7] P. Godoy and J. L. Dawson, "Chopper Stabilization of Analog Multipliers, Variable Gain Amplifiers, and Mixers," *IEEE Journal of Solid-State Circuits,* vol. 43, no. 10, pp. 2311-2321, Oct. 2008.

[8] S. T. Kim, J. Choi, S. Beck, T. Song, K. Lim and J. Laskar, "Subthreshold current mode matrix determinant computation for analog signal processing," in *IEEE International Symposium on Circuits and Systems*, Paris, 2010.

[9] K. Abdelhalim, L. Kokarovtseva, J. L. P. Velazquez and R. Genov, "915-MHz FSK/OOK Wireless Neural Recording SoC With 64 Mixed-Signal FIR Filters," *IEEE Journal of Solid-State Circuits,* vol. 48, no. 10, pp. 2478-2493, Oct. 2013.

[10] M. Gu and S. Chakrabartty, "A 100 pJ/bit, (32,8) CMOS Analog Low-Density Parity-Check Decoder Based on Margin Propagation," *IEEE Journal of Solid-State Circuits,* vol. 46, no. 6, pp. 1433-1442, June 2011.

[11] D. Miyashita and et al., "An LDPC Decoder With Time-Domain Analog and Digital Mixed-Signal Processing," *IEEE Journal of Solid-State Circuits,* vol. 49, no. 1, pp. 73-83, 2014.

[12] M. Liu, L. R. Everson and C. H. Kim, "A scalable time-based integrate-and-fire neuromorphic core with brain-inspired leak and local lateral inhibition capabilities," in *IEEE Custom Integrated Circuits Conference (CICC)*, Austin, TX, 2017.

[13] D. Miyashita, S. Kousai, T. Suzuki and J. Deguchi, "Time-domain neural network: A 48.5 TSOp/s/W neuromorphic chip optimized for deep learning and CMOS technology," in *IEEE Asian Solid-State Circuits Conference (A-SSCC)*, Toyama, 2016.

[14] 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," in *IEEE International Solid - State Circuits Conference (ISSCC)*, San Francisco, CA, 2018.

[15] N. Cao, M. Chang and A. Raychowdhury, "A 65nm 1.1-to-9.1TOPS/W Hybrid-Digital-Mixed-Signal Computing Platform for Accelerating Model-Based and Model-Free Swarm Robotics," in *IEEE International Solid- State Circuits Conference - (ISSCC)*, San Francis, CA, 2019.

[16] A. Sayal, S. Fathima, S. S. T. Nibhanupudi and J. P. Kulkarni, "All-Digital Time-Domain CNN Engine Using Bidirectional Memory Delay Lines for Energy-Efficient Edge Computing," in *IEEE International Solid- State Circuits Conference - (ISSCC)*, San Francisco, CA, 2019.

[17] L. Everson and et al., "A 40X40 Four-Neighbor Time-Based In-Memory Computing Graph ASIC Chip Featuring Wavefront Expansion and 2D Gradient Control," in *IEEE ISSCC*, San Fransicso, CA, 2019.

[18] W. Khwa and et al., "A 65nm 4Kb algorithm-dependent computing-in-memory SRAM unit-macro with 2.3ns and 55.8TOPS/W fully parallel product-sum operation for binary DNN edge processors," in *IEEE International Solid - State Circuits Conference - (ISSCC)*, San Francisco, CA, 2018.

[19] X. Si and et al., "A 28nm 64Kb 6T SRAM Computing-in-Memory Macro with 8b MAC Operation for AI Edge Chips," in *2020 IEEE International Solid- State Circuits Conference - (ISSCC)*, San Francisco, CA, 2020.

[20] X. Si and et al., "A Twin-8T SRAM Computation-In-Memory Macro for Multiple-Bit CNN-Based Machine Learning," in *IEEE International Solid- State Circuits Conference - (ISSCC)*, San Francisco, CA, 2019.

[21] J. Yue and et al., "A 65nm Computing-in-Memory-Based CNN Processor with 2.9-to-35.8TOPS/W System Energy Efficiency Using Dynamic-Sparsity Performance-Scaling Architecture and Energy-Efficient Inter/Intra-Macro Data Reuse," in *IEEE International Solid- State Circuits Conference - (ISSCC)*, San Francisco, CA, 2020.

[22] R. Guo and et al., "A 5.1pJ/Neuron 127.3us/Inference RNN-based Speech Recognition Processor using 16 Computing-in-Memory SRAM Macros in 65nm CMOS," in *Symposium on VLSI Circuits*, Kyoto, Japan, 2019.

[23] Q. Dong and e. al., "A 351TOPS/W and 372.4GOPS Compute-in-Memory SRAM Macro in 7nm FinFET CMOS for Machine-Learning Applications," in *IEEE International Solid-State Circuits Conference - (ISSCC)*, San Francisco, 2020.

[24] Z. Chen and J. Gu, "Analysis and Design of Energy Efficient Time Domain Signal Processing," in *IEEE/ACM International Symposium on Low Power Electroni cs and Design*, San Francisco, 2016.

[25] Z. Chen and J. Gu, "An Image Recognition Processor with Time Domain Accelerators Using Efficient," in *IEEE Asian Solid state Circuits Conference (A-SSCC)*, Taiwan, 2018.

[26] Z. Chen and J. Gu, "A Time-Domain Computing Accelerated Image Recognition Processor With Efficient Time Encoding and Non-Linear Logic Operation," *IEEE Journal of Solid-State Circuits (JSSC),* vol. 54, no. 10, pp. 3226-3237, Nov. 2019.

[27] Z. Chen and J. Gu, "A Scalable Pipelined Time Domain DTW Engine for Time Series Classification Using Multibit," in *IEEE International Solid- State Circuits Conference - (ISSCC)*, San Francisco, CA, 2019.

[28] Z. Chen, S. Fu, Q. Cao and J. Gu, "A Mixed-Signal Time-Domain Generative Adversarial Network Accelerator with Efficient Subthreshold Time Multiplier and Mixed-Signal On-Chip Training for Low Power Edge Devices," in *IEEE Symposium on VLSI Circuits*, Honolulu, HI, 2020.

[29] Z. Chen and J. Gu, "A 65nm 3T Dynamic Analog RAM-Based Computing-in-Memory Macro and CNN Accelerator with Retention Enhancement, Adaptive Analog Sparsity and 44TOPS/W System Energy Efficiency," in *IEEE International Solid-state Circuit Conference*, San Francisco, CA, 2021.

[30] A. Madhavan and et al., "Storing And Retrieving Wavefronts with Resistive Temporal Memory," in *arXiv:2003.09355v1 [cs.ET]* , March 2020.

[31] M. Liu, L. R. Everson and C. Kim, "A scalable time-based integrate-and-fire neuromorphic core with brain-inspired leak and local lateral inhibition capabilities," in *IEEE Custom Integrated Circuits Conference (CICC)*, 2017.

[32] N. P. Jouppi and et al., "In-Datacenter Performance Analysis of a Tensor Processing Unit," in *ACM ISCA*, 2017.

[33] E. Beck and et al., "In-Datacenter Performance Analysis of a Tensor Processing Unit," in *arXiv*, 2019.

[34] V. James, "Tesla's new AI chip isn't a silver bullet for self-driving cars," 24 April 2019. [Online]. Available: https://www.theverge.com/2019/4/24/18514308/tesla-full-self-driving-computer-chip-autonomy-day-specs.

[35] H. P. Lin and et al., "Analog Placement Based on Hierarchical Module Clustering," in *IEEE/ACM Design Automation Conference (DAC)*, 2008.

[36] B. Xu, S. Li, N. Sun and D. Z. Pan, "A scaling compatible, synthesis friendly VCO-based delta-sigma ADC design and synthesis methodology," in *ACM/EDAC/IEEE Design Automation Conference (DAC)*, Austin, TX, 2017.

[37] C.-T. Lin, D.-S. Chen and Y.-W. Wang, "An efficient genetic algorithm for slicing floorplan area optimization," in *IEEE International Symposium on Circuits and Systems. Proceedings*, Phoenix-Scottsdale, AZ, 2002.

[38] Q. Ma, L. Xiao, Y. Tam and E. F. Y. Young, "Simultaneous Handling of Symmetry, Common Centroid, and General Placement Constraints," *in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 30, no. 1, pp. 85-98, 2011.

[39] P.-N. Guo, C.-K. Cheng and T. Yoshimura, "An O-tree representation of non-slicing floorplan and its applications," in *IEEE/ACM Design Automation Conference*, New Orleans, LA, 1999 .

[40] P.-Y. Chou and et al., "Heterogeneous B*-trees for analog placement with symmetry and regularity considerations," in *Proceedings of the International Conference on Computer-Aided Design (ICCAD)*, Nov. 2011.

[41] J.-M. Lin and Y.-W. Chang, "TCG-S: orthogonal coupling of P*-admissible representations

for general floorplans," in *IEEE/ACM Design Automation Conference (DAC)*, Oct. 2002 .

[42] Y. Seo, J. Kim, H. Park and J. Sim, "A 1.25ps Resolution 8b Cyclic TDC in 0.13μm CMOS," *IEEE Journal of Solid-State Circuits,* vol. 47, no. 3, pp. 736-743, 2012.

[43] S. Henzler and et al., "A Local Passive Time Interpolation Concept for Variation-Tolerant High-Resolution Time-to-Digital Conversion," *IEEE Journal of Solid-State Circuits,* vol. 43, no. 7, pp. 1666-1676, July 2008.

[44] E. Chicca, F. Stefanini, C. Bartolozzi and G. Indiveri, "Neuromorphic Electronic Circuits for Building Autonomous Cognitive Systems," *Proceedings of the IEEE,* vol. 102, no. 9, pp. 1367-1388, 2014.

[45] Y. Hung and B. Liu, "High-reliability programmable CMOS WTA/LTA circuit of O(N) complexity using a single comparator," *IEE Proceedings - Circuits, Devices and System,* vol. 151, no. 6, pp. 579-586, Dec 2014.

[46] S.-C. Liu and M. Oster, "Feature competition in a spike-based winner-take-all VLSI network," in *IEEE International Symposium on Circuits and Systems*, Island of Kos, 2006 .

[47] Y. Fang, M. A. Cohen and T. G. Kincaid, "Dynamic Analysis of a General Class of Winner-Take-All Competitive Neural Networks," *IEEE Transactions on Neural Networks,* vol. 21, no. 5, pp. 771-783, May 2010.

[48] G. M. Blair, "Low cost sorting circuit for VLSI," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Application,* vol. 32, no. 6, pp. 515-516, June 1996.

[49] S. Chen, X. Zhang, H. Sun and N. Zheng, "sWMF: Separable weighted median filter for efficient large-disparity stereo matching," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, Baltimore, MD, 2017.

[50] H. Yamasaki and T. Shibata, "A real-time image-feature-extraction and vector-generation VLSI employing arrayed-shift-register architecture," in *European Solid-State Circuits Conference*, Grenoble, 2005.

[51] C. Shi and et al., "A 1000 fps Vision Chip Based on a Dynamically Reconfigurable Hybrid Architecture Comprising a PE Array Processor and Self-Organizing Map Neural Network," *IEEE Journal of Solid-State Circuits (JSSC),* vol. 49, no. 9, pp. 2067-2082, 2014.

[52] D. Jeon and et al., "A 23-mW Face Recognition Processor with Mostly-Read 5T Memory in 40-nm CMOS," *IEEE Journal of Solid-State Circuits (JSSC)),* vol. 526, pp. 1628-1642, 2017.

[53] H. Ding and et al., "Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures," in *Proceedings of VLDB*, Auckland, New Zealand, 2008.

[54] A. Madhavan, T. Sherwood and D. Strukov, "A 4-mm2 180-nm-CMOS 15-Giga-cell-updates-per-second DNA sequence alignment engine based on asynchronous race conditions," in *IEEE Custom Integrated Circuits Conference (CICC)*, Austin, TX, 2017.

[55] H. I. Fawaz and et al., "Deep Learning for Time Series Classification: A Review," in *arXiv*, 2019.

[56] X. Xu and et al., "Accelerating Dynamic Time Warping With Memristor-Based Customized Fabrics," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,*

vol. 37, no. 4, pp. 729-741, April 2018.

[57] A. Madhavan, T. Sherwood and D. Strukov, "Race Logic: A hardware acceleration for dynamic programming algorithms," in *ACM/IEEE 41st International Symposium on Computer Architecture (ISCA)*, Minneapolis, MN, 2014 .

[58] Z. Chen, H. Zhou and J. Gu, "Digital Compatible Synthesis, Placement and Implementation of Mixed-Signal Time-Domain Computing," in *ACM/IEEE Design Automation Conference (DAC)*, Las Vegas, NV, 2019.

[59] "UCR Archive," [Online]. Available: http://www.cs.ucr.edu/~eamonn/time_series_dat.

[60] N. Neves, N. Sebastião, D. Matos, P. F. P. Tomás and N. Roma, "Multicore SIMD ASIP for Next-Generation Sequencing and Alignment Biochip Platforms," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 23, no. 7, pp. 1287-1300, July 2015.

[61] M. Farrar, Striped Smith–Waterman Speeds Database Searches Six Times Over Other SIMD Implementations, Bioinformatic, 2007.

[62] Y. Liu and et al., "Cudasw++ 3.0: Accelerating Smith-Waterman Protein Database Search by Coupling CPU and GPU SIMD Instructions," in *BMC bioinormatics*, 2013.

[63] K. F. Li and et al., "Dynamic Time Warping in Hardware," in *iiWAS*, Dec. 2012.

[64] V. K. Sundaresan, N. R. S. Nichani and R. Sankar, "A VLSI hardware accelerator for dynamic time warping," in *Proceedings., 11th IAPR International Conference on Pattern Recognition*, 1992 .

[65] A. Radford, L. Metz and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," in *arXiv:1511.06434*, Jan 2016.

[66] E. H. Lee and S. S. Wong, "A 2.5GHz 7.7TOPS/W switched-capacitor matrix multiplier with co-designed local memory in 40nm," in *IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, 2016.

[67] Z. Research, "Fashion MNIST," 2017. [Online]. Available: https://www.kaggle.com/zalando-research/fashionmnist.

[68] "Emoji database," [Online]. Available: https://getemoji.com/.

[69] K. Yoshioka and et al., "PhaseMAC: A 14 TOPS/W 8bit GRO Based Phase Domain MAC Circuit for in-Sensor-Computed Deep Learning Accelerators," in *IEEE Symposium on VLSI Circuits*, Honolulu, HI, 2018.

[70] H. Zhou and J. Wang, "ACG–Adjacent Constraint Graph for General Floorplans," in *IEEE ICCD*, 2004.

[71] D. Stewart, "Edge AI chips come into their own," [Online]. Available: https://www2.deloitte.com/us/en/insights/industry/technology/technology-media-and-telecom-predictions/2020/ai-chips.html..

[72] Z. Chen and J. Gu, "High-Throughput Dynamic Time Warping Accelerator for Time-Series Classification With Pipelined Mixed-Signal Time-Domain Computing," *IEEE Journal of Solid-State Circuits,* 2020.

[73] R. Cutler and L. S. Davis, "Robust real-time periodic motion detection, analysis, and application," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 781-796, Aug. 2000.